

26  
28



**Universidad Nacional Autónoma de México**

**FACULTAD DE INGENIERIA**

**DISEÑO Y APROVECHAMIENTO DE  
APLICACIONES PARA  
COMPUTADORAS**

**TESIS**

Que para obtener el título de:

**INGENIERO EN COMPUTACION**

Presentan :

**ALBERTO TEMPLOS CARBAJAL  
EDUARDO HERNANDEZ OLIVA  
JAIME GARCÉS CORELLA**

**Director Ing. Carlos Strassburger F.**

**México, D. F.**

**1985**



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.

# I N D I C E

CAPITULO	PAG.
I.- Introducción.	2
II.- Evolución del soporte lógico de sistemas.	12
II.1.- Los inicios.	12
II.2.- Los primeros compiladores y monitores de proceso en lote.	13
II.3.- Sistemas de tiempo compartido.	14
II.4.- Sistemas en línea.	17
II.5.- Soporte lógico de sistemas de tercera generación.	18
II.6.- El soporte lógico de sistemas, durante los primeros años de la década de los setentas.	21
II.6.1.- Sistemas operativos.	21
II.6.2.- Ayudas a la programación.	25
II.6.3.- Otras ayudas.	28
II.6.4.- Sistemas manejadores de bases de datos.	28
II.6.5.- Programas de control de comunicaciones.	31
II.6.6.- Otro soporte lógico de sistemas.	32
II.7.- Lenguajes de cuarta generación.	34
II.7.1.- Limitaciones.	38
III.- Problemas y soluciones con programas de aplicación.	40
III.1.- Desarrollo de soporte lógico dentro de la empresa.	42
III.1.1.- Metodología.	43
III.1.1.1.- Fase de planeación.	44
III.1.1.2.- Fase de desarrollo.	48
III.1.1.3.- Fase de mantenimiento.	50
III.2.- Adquisición de soporte lógico comercial.	52
III.2.1.- Clasificación.	54
III.2.2.- Situación de mercado.	55
III.2.3.- Fuentes de paquetes de soporte lógico.	55
III.2.4.- Compra de un paquete de soporte lógico comercial.	56
III.2.4.1.- Evaluación de paquetes de soporte lógico.	58
III.2.4.2.- Desarrollo de un sistema de puntuación.	59
III.2.4.3.- Ventajas y desventajas de los paquetes de soporte lógico.	64
III.2.5.- Perspectivas.	65
III.2.6.- Un par de inquietudes.	66
III.2.7.- Adaptación de soporte lógico comercial.	68
III.3.- Contratación del desarrollo del soporte lógico.	68

IV.- Aspectos legales del soporte lógico.	71
IV.1.- Contratación de licencias de uso.	71
IV.2.- Trámites ante el Registro Nacional de Transferencia de Tecnología, para la inscripción de contratos de soporte lógico.	73
IV.3.- Protección legal del soporte lógico en México.	74
V.- Aplicación a un proyecto real.	80
APENDICE A	
Evolución de los manejadores de bases de datos.	173
APENDICE B	
Clasificación de los paquetes de soporte lógico de 1982 a 1984.	186
APENDICE C	
Directorio de los principales distribuidores de soporte lógico.	195
APENDICE D	
Catálogos y directorios de productos de soporte lógico.	201
APENDICE E	
Referencias bibliográficas de metodologías para el desarrollo de sistemas de soporte lógico.	203
APENDICE F	
Formas de trámites legales.	205
BIBLIOGRAFIA	248

**CAPITULO I**

**INTRODUCCION**

## INTRODUCCION

Dentro del tema del seminario, se ha pretendido dar la mayor cobertura a este vasto tema que de hecho es imposible detallar y ejemplificar; por ésto en cuanto a diseño se presenta la evolución de las aplicaciones para computadoras desde sus orígenes hasta los lenguajes de 4a. generación (L4G) haciendo, descripciones y clasificaciones de la aplicaciones sin entrar a detalles de qué tecnología de programación (HIPO, programación estructurada, etc) se utilizó ni cómo se aplica; no es una historia del diseño de aplicaciones, pero seguramente se pueden asociar las características y alcances de cada aplicación con las técnicas de diseño implicadas en su tiempo, teniendo presente el ámbito dentro del cual se desarrollieron.

Anteriormente se consideraba un arte al diseño, programación y aprovechamiento de computadoras, pero al transcurrir del tiempo se ha convertido en toda una profesión como la nuestra, dentro de éste el aprovechamiento de aplicaciones para computadora amerita dar soluciones a los usuarios de equipo de cómputo desde la adquisición de aplicaciones, conociendo las formas de evaluar hasta proceder a un diseño total utilizando las técnicas de análisis y programación de sistemas con toda la teoría y tecnología actuales, con la gran responsabilidad de realizarlo dentro del marco económico y tiempo de implantación adecuados.

El objetivo de esta tesis es precisamente orientar a los lectores de cómo proceder a adquirir, adaptar y desarrollar externa o internamente aplicaciones para computadora asegurando su operación posterior, así como el poderlas aprovechar.

A lo largo del trabajo se encontrará el término "no procedural" o "lenguaje no procedural" que de hecho no tiene una traducción literal y se ha dejado en el idioma original; pero con esto se quiere decir que la aplicación para computadora debe ser más simple y más eficiente en su uso. Por ejemplo si se necesita adquirir un BASIC no solo interesarán las características de que sea compilador e interactivo, porque hay muchos BASICs, y de éste se deberá seleccionar el que permita lograr la programación con el menor esfuerzo, o en otras palabras con el menor número de instrucciones; es decir, si el BASIC X no tiene las funciones intrínsecas MATINV, MATMUL, etc. y el BASIC Y sí tiene las funciones de matrices se dice que BASIC Y es menos procedural que el BASIC X al menos en operaciones con matrices. Igualmente una aplicación para computadora no procedural tiene ventajas sobre otra que necesita hacer definiciones detalladas para lograr los mismos objetivos; estas ventajas son al menos en tiempo de programación o especificación del problema a resolver, pero se debe tener presente que "no procedural" no implica ser más eficiente, aunque pudiera serlo.

Los lectores de esta tesis con experiencia podrán recordar los pasajes que posiblemente vivieron y a la vez actualizarse; y aquellos sin experiencia podrán documentarse y adquirir experiencia para lograr el mejor aprovechamiento de aplicaciones para computadora, como lenguajes de programación, sistemas operativos, bases de datos, comunicación de datos y paqueterías en general que puedan ser de uso específico o incluir las anteriores.

Para el caso de adquisiciones de soporte lógico se presentaron métodos para evaluar, seleccionar e implicaciones legales, como trámites de permisos de importación, licencias, contratos, registros ante el gobierno, conocimiento de normas y leyes correspondientes.

Para la adaptación de aplicaciones de computadora no se hace mucho énfasis porque se considera que una vez tomada la decisión de adaptar, los pasos a seguir son los de desarrollo.

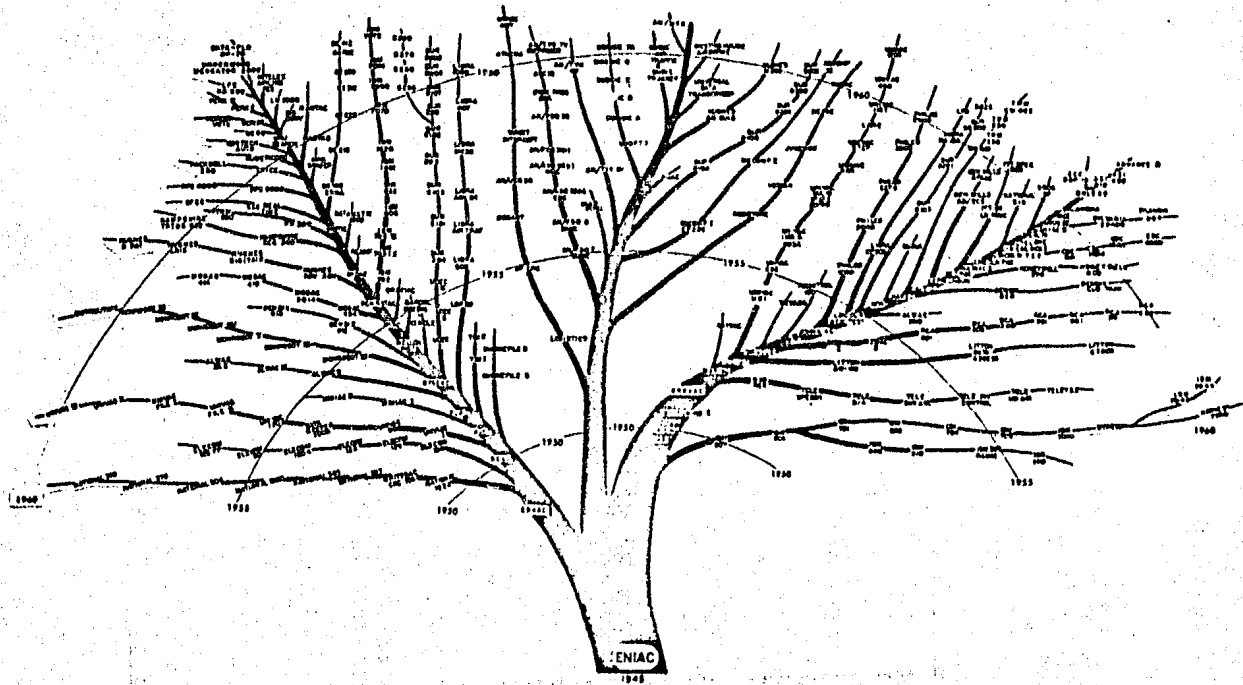
En el caso de desarrollo de aplicaciones se presentaron los pasos a seguir, distinguiendo si es desarrollo por terceros o por los interesados.

Finalmente se presenta un ejemplo de desarrollo de un sistema utilizando la tecnología mas moderna para productividad de aplicaciones de cómputo de tipo administrativo, mostrando e ilustrando los pasos seguidos y el desarrollo en un lenguaje "no procedural" de cuarta generación.

Uno de los problemas mas grandes en las empresas es la evolución en automatización tanto en ofrecimiento de nuevos servicios, como en llevar a un mayor grado de complejidad a los sistemas existentes por no utilizar métodos de aprovechamiento de aplicaciones para computadora.

Esperamos que la tesis presente sirva para evitar que el personal de programación permanezca en un status de corrección de sistemas y haga de esto, deliberada o inconcientemente un modus vivendi; y además resuelva o solucione los problemas anteriormente planteados.

En las páginas siguientes se muestra una clasificación ilustrativa por generaciones, tanto del soporte físico como del soporte lógico.



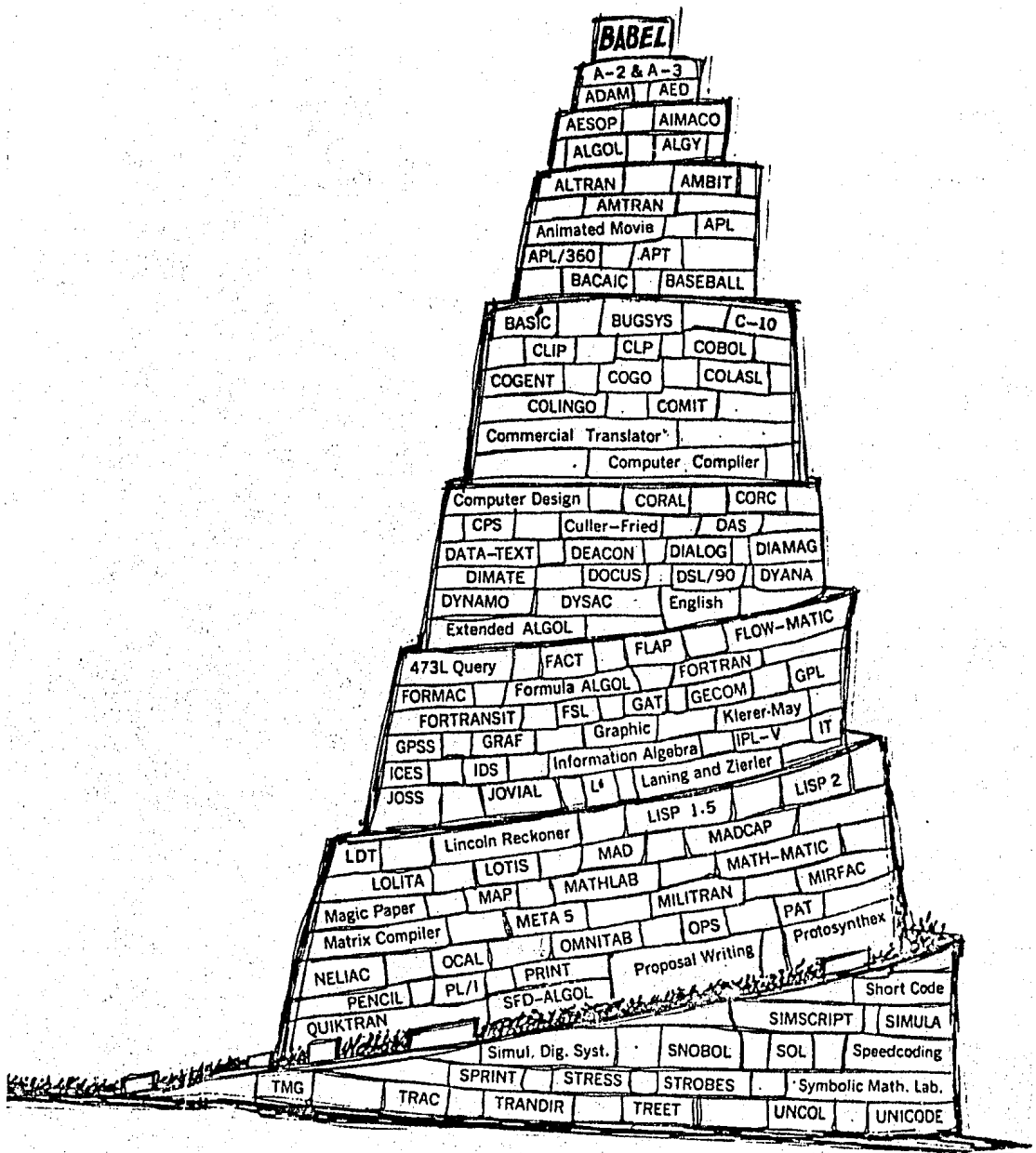


-----		
I		I
I		I
I	Computadoras de la primera generación	I
I	-----	I
I		I
I	Entrada al mercado:	1950 aproximadamente.
I		I
I	Aplicación principal:	Instrumentos de cálculo.
I		I
I	Tecnología utilizada:	Tubos de vacío.
I		Memoria de cilindro magnético.
I		I
I	Unidades periféricas:	Lectoras y perforadoras de
I		tarjetas y cinta de papel, equipo
I		unitario, etcétera.
I		I
I	Sistema operativo:	No existía.
I		I
I	Lenguajes de	
I	programación:	Lenguajes de máquina,
I		ensambladores primitivos.
I		I
I	Alfabeto:	Numérico.
I		I
I	Administración:	Trivial, no se requería.
I		I
I	Aspectos cuantitativos:	M. Central 1000 a 8000 palabras.
I		Proceso $10^4$ ops/seg.
I		Precio 100000 a 2.5 millones
I		EE.UU.
I		I
I	Modelos típicos:	IBM-650, Bendix-G15, Univac SS90.
I		Bull-pt, IBM-709.
I		I
I		I
I		I
-----		

Computadoras de la segunda generación	
Entrada al mercado:	1960 aproximadamente.
Aplicaciones principales:	Proceso de datos. Instrumento de cálculo.
Tecnología utilizada:	Transistores y ferritas.
Unidades periféricas:	Lectoras y perforadoras de tarjetas, impresoras y cintas magnéticas.
Sistema operativo:	Rudimentario, controla periféricos, inicia y termina tareas.
Lenguajes de programación:	Ensambladores y primeros compiladores (FORTRAN, ALGOL).
Alfabeto:	Números y letras, algunos caracteres especiales.
Facilidades adicionales:	Existencia de bibliotecas.
Administración:	Primitiva, planeación de producción con procesos masivos.
Aspectos cuantitativos:	MC 8000 a 32000 palabras. Procesadores $10^5$ ops/seg. Precios $10^5$ a $10^7$ EE.UU.
Modelos típicos:	CDC-160, IBM-7090, IBM-1401, Burroughs 5500, RCA-305, Bendix G20, CDC-3600, CEC-6600.

Computadoras de la tercera generación	
Entrada al mercado:	Aproximadamente entre 1968 y 1970.
Aplicaciones principales:	Sistemas de información.
Tecnología utilizada:	Circuitos integrados (LSI) y memoria de partículas magnéticas.
Unidades periféricas:	Cintas y discos magnéticos, terminales de video y teletipos.
Arquitectura:	Multiprogramación, multiproceso, sistemas de interrupción.
Lenguajes y facilidades de programación:	Lenguajes de alto nivel, COBOL, PL/I, base de datos (DMS).
Alfabeto:	Números, letras y caracteres especiales.
Sistema operativo:	Manejo de discos, multiproceso, memoria dinámica, memoria virtual, etcétera.
Facilidades adicionales:	Edición y prueba interactiva de programas.
Administración:	Compleja y especializada.
Aspectos cuantitativos:	MC 64 a 256 K <sub>6</sub> palabras. Procesador 10 <sup>6</sup> ops/seg. Memoria secundaria 10 <sup>4</sup> caracteres. Rango de precios 5x10 <sup>4</sup> a 10 <sup>8</sup> EE.UU.
Modelos típicos:	IBM-360, Burroughs 6700, PDP 10, PDP 11, Univac 1106, CYBER 170.

Computadoras de la cuarta generación	
Entrada al mercado:	Entre 1977 y 1981.
Aplicaciones principales:	Sistemas de comunicación, sistemas de información para negocios pequeños, uso personal.
Tecnologías utilizadas:	Microelectrónica (VLSI), memorias MOS (Metal Oxide Sylicates).
Unidades periféricas:	Terminales inteligentes, discos y cintas magnéticas, equipo de graficación, lectores ópticos y digitalizadores.
Arquitectura:	Proceso distribuido, uso de microprocesadores.
Lenguajes y facilidades de programación:	Bases de datos distribuidas, lenguajes interactivos, descriptivos y gráficos.
Alfabeto:	Irrestringido, mayúsculas y minúsculas, símbolos matemáticos, alfabeto japonés, etcétera.
Sistema operativo:	Proceso sin interrupción, comunicación entre máquinas, rutinas de recuperación, etcétera.
Facilidades adicionales:	Metaprocesadores, correo electrónico, manejadores de texto.
Administración:	Muy simple para equipos personales. Muy complejo para redes de proceso distribuido.
Aspectos cuantitativos:	Memoria central 64K a $10^7$ caracteres. Procesador $10^7$ ops/seg. Memoria secundaria $10^{10}$ caracteres. Rango de precios $10^3$ a $10^8$ EE.UU.
Modelos típicos grandes:	IBM-4330, Univac 1100, Burroughs B6900, 7900.
Medianos:	Prime 550, MP 3100 VAX.
Pequeños:	Apple, TR80, IBM-PC.



GENERACIONES DE LENGUAJES				
GENERACION	ENTRO EN USO	LENGUAJES		
1	1940's 1950's	LENGUAJE DE MAQUINA		
2	1950's	LENGUAJE ENSAMBLADOR		
3	1960's	LENGUAJES DE ALTO NIVEL. ALGOL, FORTRAN, COBOL, PL/1, ETC.		
4	1970's 1980's	NOMAD, FOCUS, LINC, MAPPER, MANTIS, APL, COGEN, ETC.		

CAPITULO II

EVOLUCION DEL SOPORTE LOGICO DE SISTEMAS

## II.- EVOLUCION DEL SOPORTE LOGICO DE SISTEMAS.

### II.1.- LOS INICIOS.

A partir de la aparición de las primeras máquinas orientadas a la realización de cálculos -a las cuales se programaba mediante el alambrado de tableros -la comunicación entre el hombre y la computadora ha estado en un proceso de evolución permanente, haciendo posible el desarrollo de aplicaciones cada vez más complejas.

Las primeras computadoras (antes de 1955) frecuentemente eran vendidas sin soporte lógico de sistemas. Los programadores escribían sus programas en lenguaje de máquina (código binario) y les era necesario recordar las direcciones absolutas de las instrucciones y las modificaciones que hacían en su programación. Es obvio, que escribir un programa en esta forma era un proceso que requería mucho tiempo y en el cual el programador estaba propenso a cometer muchos errores. Rápidamente se reconoció la necesidad de crear una mejor herramienta para la programación, como respuesta a ello se idearon los lenguajes ensambladores. Estos ensambladores permitían a los usuarios escribir programas usando direcciones simbólicas para los valores de los datos y para las instrucciones de control de secuencia (branch), así como también el uso de mnemónicos para los códigos de operación.

Los programadores pronto se dieron cuenta que con mucha frecuencia recodificaban los mismos procedimientos, lo cual consumía mucho tiempo y resultaba sumamente caro. Como consecuencia se concibió la noción de macro, y rápidamente entraron en uso los macroensambladores, los cuales permitían a los usuarios desarrollar programas más fácilmente. Los macroensambladores fueron las principales herramientas de programación de la década de los sesentas y aún tienen un uso muy extendido.

Los dispositivos primarios de Entrada y Salida (E/S) de las primeras computadoras fueron las lectoras de tarjetas, las impresoras; también tuvieron un uso común las cintas de papel perforado. El almacenamiento auxiliar consistió principalmente de manejadores de cintas magnéticas. Las computadoras fueron esencialmente de naturaleza serial; es decir, cuando se emitía una instrucción de lectura, ya sea de una tarjeta o de un registro en cinta, no se podían ejecutar otras instrucciones hasta que esta no estuviese terminada. Los usuarios pronto se percataron de que una forma de mejorar el funcionamiento de un sistema de cómputo era formar bloques con los registros de una cinta. Por ejemplo, si bloqueaban cuatro registros lógicos en un solo registro físico en la cinta, podían reducir el período de tiempo que tenían que esperar para el reinicio del sistema a través de cinta (que frecuentemente era el factor dominante) en tres cuartas partes. Sin embargo, fue necesario desarrollar soporte lógico altamente complejo para manejar las actividades



de bloqueo y desbloqueo, así como para el manejo de las condiciones de error, fin de archivo, y fin de cinta. Algunas de las primeras computadoras también tenían una capacidad limitada de translapo (overlap) de cintas, por medio del cual un programa podía emitir un requerimiento de lectura o escritura y continuar procesando. Era posible escribir programas que pudieran expedir un requerimiento de lectura, realizar algunas otras operaciones de cómputo, y entrar en un ciclo de espera hasta que la lectura terminara. Sin embargo, para el programador promedio fue sumamente difícil el escribir un código razonablemente complejo, y en forma simultánea bloquear y desbloquear los registros, haciendo un uso, lo más extenso posible, de la capacidad de lectura, escritura y cómputo simultáneos. Los fabricantes de computadoras, en conjunción con usuarios "complejos", pronto implementaron paquetes de sistemas de control de entrada y salida (IOCS-Input Output Control Systems) para el control de esas y otras funciones. Los paquetes IOCS pudieron ser utilizados por el programador que hacía uso de un lenguaje ensamblador al incluir macros de entrada y salida. A finales de los años cincuenta, muchos fabricantes de computadoras ya proveían tanto macroensambladores como sistemas de control de E/S.

Durante la década de los cincuenta, era usual que las computadoras no pudieran procesar más de un programa en forma simultánea. Los operadores tenían que sentarse en la consola y cargar las cintas del programa; sólo hasta que el programa finalizaba, podían "limpiar" las lectoras y perforadoras de tarjetas, la impresora, desmontar las cintas y (si fuera necesario), hacer un vaciado de la memoria principal. Después de esto, podía comenzar el proceso de inicio del siguiente trabajo. En muchas instalaciones se reconoció con prontitud la ineficiencia de esta forma de operación. Se desperdiciaba una gran cantidad de tiempo de computadora durante la preparación, inicio y retiro de un trabajo, del sistema. Esto condujo al desarrollo de monitores simples (sistemas operativos simples) de proceso en lote a finales de la década de 1950.

## II.2.- LOS PRIMEROS COMPILADORES Y MONITORES DE PROCESO EN LOTE.

Desde el principio se tenía la esperanza de que las computadoras pudieran ser diseñadas para que interpretaran y ejecutaran expresiones algebraicas en forma directa. A mediados de los años cincuenta, un sistema de translación algebraica, FORTRAN (Formula Translation), comenzó a ser aceptado y usado por un número reducido de personas. Sin embargo, los primeros compiladores fueron difíciles de usar. El compilador tenía que ser cargado al sistema por el operador. Una vez instalado el compilador, este leía el programa de las tarjetas perforadas y producía una versión del programa, en lenguaje ensamblador, en otro conjunto de tarjetas perforadas.

A continuación el operador retiraba las tarjetas de la perforadora y las cargaba nuevamente en la lectora de tarjetas, antes de que la ejecución del programa objeto pudiera comenzar. Se desperdiciaba una gran cantidad de tiempo cuando el operador o el programador llevaba y traía las tarjetas de la perforadora a la lectora y cargaba el compilador y el ensamblador, de las tarjetas.

No pasó mucho tiempo para que los usuarios desarrollaran monitores de proceso en lote (batch), los cuales aceleraron la ejecución de los trabajos. Estos monitores leían las tarjetas de control preparadas por los usuarios, cargaban de un sistema de cintas el compilador o el ensamblador necesario, realizaban la compilación o el proceso de ensamblado, y cargaban el programa objeto para su ejecución. Continuamente los sistemas IOCS estándar permanecían en memoria principal; al finalizar la ejecución del programa, el control era transferido al monitor, el cual leía las tarjetas de control del siguiente programa. Estos monitores simples de proceso en lote, incrementaron drásticamente la eficiencia de muchas computadoras. Sin embargo, la gran resistencia que presentaron muchos usuarios hacia estos sistemas fue debida a varias razones; principalmente a que los usuarios sentían una pérdida de control sobre la computadora.

### II.3.- SISTEMAS DE TIEMPO COMPARTIDO.

Mientras que el uso de las primeras computadoras se realizó en modo lote, también hubo alguna interacción en tiempo real con los usuarios. Por ejemplo, era usual que una rutina simple, tal como el programa que calcula una raíz cuadrada, emitiera resultados intermedios conforme fuera realizando los cálculos, de manera que el usuario podía determinar cuando el algoritmo progresaba hacia la solución. De hecho, el usuario del sistema tenía la capacidad para interactuar con la computadora para monitorear el estado de su proceso. Los monitores simples de proceso en lote, introducidos para máquinas tales como: la IBM 704, 709 y 7090 eliminaron la posibilidad de tal interacción con la computadora. Entonces fue imposible determinar exactamente cuando un trabajo específico iba a ser ejecutado; el monitor tenía todo el control sobre la ejecución y no podía ser interrumpido convenientemente. Aún con ésta pérdida de control, no podrían haberse tenido consideraciones de mejoras económicas sin el uso de monitores de proceso en lote.

Mucha gente se dió cuenta de que sí tenía la capacidad de interactuar con una computadora de gran tamaño, como lo había estado haciendo, tenía que compartir la computadora con otros usuarios; de aquí nació la noción de "Tiempo Compartido". La idea esencial fue permitir al programador sentarse en una consola e interactuar con su programa conforme este se ejecutaba. Algunos de los primeros sistemas de tiempo

compartido fueron desarrollados en las universidades debido a la gran población de programadores independientes que allí existía. Uno de los primeros sistemas fue el llamado "Compatible Time-Sharing System (CTSS)", desarrollado en el Instituto Tecnológico de Massachusetts (MIT) para la IBM 7094.

En el sistema CTSS un programa ejecutivo residía en memoria e interactuaba con varias terminales. Cada vez que una terminal mandaba un mensaje se generaba una interrupción, el mensaje era procesado al tener cargado el programa requerido por el usuario en memoria principal, posteriormente era ejecutado. Cuando terminaba la ejecución, un mensaje apropiado era mandado a la terminal y el programa era transferido a memoria secundaria. Como la mayoría de las tareas que el usuario deseaba realizar podían ser procesadas en unos cuantos segundos, este sistema podía manejar varios usuarios sin que ninguno de ellos se "diera cuenta" de que los demás estaban presentes en el sistema.

La actividad primaria de los programadores es escribir programas, y como los sistemas de tiempo compartido fueron desarrollados por ellos, el uso principal de estos sistemas fue en el proceso de desarrollo de programas. Se adicionaron facilidades de edición de textos, de manera que los programadores podían preparar un programa desde el teclado (usualmente en un lenguaje de alto nivel, tal como FORTRAN), modificarlo y cambiarlo usando un programa de edición de textos que se invocaba desde el teclado para corregir los errores sintácticos reportados por el compilador, cuando un programa era compilado sin errores, la ejecución de este era iniciada.

Aunque la idea de tiempo compartido fue conceptualmente simple, las personas que desarrollaron los primeros sistemas, pronto se dieron cuenta que tenían que enfrentarse a muchas dificultades técnicas y de interfase con el usuario. Por ejemplo, se dieron cuenta de que una cantidad sorprendente de tiempo de computadora se requería para transferir los programas, dentro y fuera del sistema, entre las interacciones con los usuarios individuales. También muchos de los programas eran muy pequeños, especialmente en las universidades donde muchos estudiantes trabajaban en línea; y solamente una pequeña porción de la memoria principal era utilizada en un determinado momento. Además, pronto se descubrió la necesidad de elaborar técnicas de seguridad para prevenir la destrucción inadvertida (y algunas veces maliciosa) de los programas y datos. Lo más molesto de todo fue que si un usuario desarrollaba un programa, que requería mucho tiempo de proceso y lo ejecutaba desde su terminal, el tiempo de respuesta de los otros usuarios se volvía inaceptablemente largo. Los diseñadores de sistemas desarrollaron métodos para controlar la autorización de acceso por medio de la "programación de tiempos" (scheduler) de las tareas, para prevenir que los trabajos simples en línea monopolizaran el sistema, y también exploraron nuevas maneras de compartir la memoria principal entre varios usuarios.

Los autores de los sistemas de tiempo compartido, pronto aprendieron que los ensambladores y compiladores desarrollados para las aplicaciones en lote, no eran apropiadas para su uso en tiempo compartido.

Aquellos usuarios que deseaban compilar un programa grande, monopolizaban el sistema y hacían que el tiempo de respuesta de los demás usuarios se incrementara considerablemente. También durante el proceso de "tecleo" del código en modo línea, los errores no eran indicados o corregidos inmediatamente; esto dió origen a un desperdicio de ejecuciones y a depuraciones complejas e innecesarias. Como resultado se desarrollaron nuevos sistemas, a los que se llamó sistemas de tiempo compartido de segunda generación, en base a lenguajes interpretativos especiales, siendo diseñados de manera que cuando una línea de código era introducida en el sistema, se trasladaba a una forma intermedia, la cual posteriormente era interpretada en el tiempo de ejecución. La primera fase de la compilación se realizaba conforme se introducía la línea de código, y la segunda fase era realizada hasta que el programa completo era terminado. Un sistema pionero de esta naturaleza fue el sistema BASIC, el cual se diseñó en Dartmouth College.

Además, los sistemas de tiempo compartido tuvieron otros objetivos para hacer más conveniente a los programadores el uso de una computadora. Se hizo evidente que si se iban a desarrollar grandes sistemas, muchos usuarios necesitaban compartir una base de datos de programas, común. Por lo tanto, uno de los objetivos de los primeros sistemas de tiempo compartido, tales como el CTSS (desarrollado en MIT) fue el proporcionar a los programadores la facilidad de compartir código escrito por otros y este objetivo rápidamente fue expandido hasta proporcionar la facilidad de compartir datos. La mayoría de los primeros sistemas de tiempo compartido trataron con estructuras de archivos de tipo simple. Las dos organizaciones normalmente disponibles a los usuarios fueron: la organización secuencial (archivos en cinta) o, en algunos sistemas más avanzados, archivos de acceso directo para los cuales, el usuario tenía que desarrollar el método de acceso. Sin embargo, pronto se desarrollaron muchos sistemas complejos de manejo de directorios para los sistemas de acceso directo. Estos sistemas fueron indispensables debido a la necesidad de que el usuario que terminaba su sesión posteriormente necesitaba volver a entrar en una nueva sesión y localizar los programas o datos previamente usados. Estos sistemas de directorio también trataron de solucionar los problemas que tenían los usuarios al usar el mismo nombre para sus archivos y los problemas de autorización, así como las prioridades de los programas.

Los sistemas de tiempo compartido siempre han tenido que tratar con problemas de requerimientos de usuario que son básicamente insolubles. El "programador de tiempos" debe tratar con requerimientos dinámicos de recursos. En cualquier instante, el usuario puede requerir nuevos recursos tales como:

un archivo, un compilador, o cualquier dispositivo tal como una cinta magnética, etc. El sistema operativo debe ser capaz de distribuir óptimamente estos recursos entre los usuarios, especialmente cuando son insuficientes para satisfacer la demanda. Además, los usuarios normalmente no están cooperando entre sí. Sino que por el contrario, los usuarios compiten por los recursos y no están muy interesados en la integridad del espacio de trabajo de los demás. Los problemas de aislar y contener a los usuarios, fueron difíciles de resolver en los primeros sistemas de tiempo compartido. Aún no está claro si estos problemas están completamente resueltos, particularmente con la adición progresiva de los problemas que implican los ambientes multimodo.

#### II.4.- SISTEMAS EN LINEA.

Al mismo tiempo que algunos usuarios de computadoras consideraban a los sistemas de tiempo compartido, otros usuarios (notablemente las líneas aéreas, la industria financiera, y la industria militar) concibieron los sistemas en línea. Se esperaba que en estos sistemas los operadores (oficinistas típicamente) interactuaran con la computadora por medio de una terminal, mandando y recibiendo mensajes. Por ejemplo, se esperaba que en los primeros sistemas de reservación aérea se pudiera manejar la información sobre los lugares disponibles en los vuelos, programados con varios meses de anticipación, para responder a las consultas de disposición y reservación, que se realizaba desde las agencias de ventas.

La mayoría de los primeros sistemas en línea, fueron diseñados para el mismo tipo de máquinas para los cuales fueron diseñados los primeros sistemas de tiempo compartido. Inicialmente, se conceptualizó a los sistemas en línea en forma similar a los sistemas de tiempo compartido, debido a que en ambos casos varios usuarios trataban de usar en forma simultánea a la computadora.

Los programas de sistemas, necesarios para soportar a los sistemas en línea, son muy diferentes a los que se requieren para soportar a los sistemas de tiempo compartido, debido a la naturaleza del trabajo que cada uno implica. En un sistema en línea típico, existe un número muy limitado de tipos de transacciones que pueden ser iniciadas por los usuarios desde sus terminales. En un sistema simple de reservación aérea, pueden pedir información sobre la disponibilidad de los lugares en un vuelo específico o pueden hacer la reservación de los mismos en un vuelo determinado. Básicamente estos son los únicos dos programas de proceso de transacciones, y la cantidad de tiempo requerido para ejecutar cada uno está bien establecido. Los archivos son de dimensiones modestas, de estructura simple y rigurosamente controlada, adicionalmente puede esperarse que los usuarios cooperen con procedimientos de operación estándar. Los requerimientos de recursos son

conocidos de antemano- por ejemplo, solo habrá un cierto número de terminales en un momento dado, originando una cantidad predecible de transacciones, de manera que el espacio en memoria principal puede ser planeado, el rendimiento puede modelarse, etc.

En contraste, en un sistema de tiempo compartido, con frecuencia es difícil estimar en que momento la gente estará haciendo uso del sistema y aún es más difícil de estimar que es lo que estarán haciendo. El desarrollo de las técnicas de "programación en el tiempo" de las tareas, distribución de recursos, etc., para los sistemas en línea ha sido simple en comparación con las técnicas necesarias en los sistemas de tiempo compartido. Por otra parte, han surgido problemas técnicos difíciles de resolver. Con frecuencia los sistemas en línea tienen que manejar grandes cargas de transacciones, y los empleados de las oficinas de reservación necesitan que el tiempo de respuesta de la computadora sea de pocos segundos. El usuario de sistemas de tiempo compartido, es algo más tolerante a una respuesta lenta y frecuentemente puede esperar 30 segundos o más sin molestarse. En suma, los sistemas en línea se caracterizan por un acceso muy frecuente a un número limitado de archivos, mientras que los sistemas de tiempo compartido se caracterizan por densas ediciones de texto y grandes tiempos de ejecución, una vez que los compiladores o los programas desarrollados por el usuario son invocados y obtenidos de los archivos. Esta diferencia ha significado que los programas de sistemas necesarios para soportar a los sistemas en línea, se hayan orientado principalmente al manejo de archivos y a proporcionar un tiempo de respuesta mínimo. En contraste con aquellos necesarios para soportar a los primeros sistemas de tiempo compartido (recientemente ha ocurrido una convergencia, conforme las transacciones en línea se han vuelto más complejas y diversas, y conforme los sistemas de tiempo compartido se han orientado más a bases de datos). Otro camino de evolución hacia los sistemas multimodo.

## II.5.- SOPORTE LOGICO DE SISTEMAS DE TERCERA GERERACION.

A principios de la década de los setentas, ya se habían desarrollado sistemas prácticos de monitores en lote, "primeros en entrar primeros en servirse", sistemas de tiempo compartido y sistemas en línea. Los lenguajes de alto nivel, tales como FORTRAN y COBOL, fueron ganando una aceptación general, y cada vez más programas comenzaron a ser escritos en estos lenguajes. También creció el interés sobre la posibilidad de desarrollar programas para el manejo de archivos de propósito general. Se conceptuó que los programas de propósito general pudieran manejar estructuras complicadas de archivos y que permitieran al usuario obtener información de una base de datos usando un lenguaje de consulta simple y fácil de aprender. Se desarrollaron dos prototipos de tales sistemas, principalmente para aplicaciones militares. El

primero se orientó a usuarios sin conocimiento de programación y su propósito principal fue el permitir la interacción con la base de datos y la recuperación en línea de la información en un modo interactivo. El segundo prototipo estuvo relacionado con la facilidad de generación de reportes, el cual permitía a el usuario (sin experiencia en programación) describir el tipo de reporte que necesitaba obtener de la base de datos, generar el programa, acceder los archivos y producir el reporte. Sin embargo, los sistemas prototipo fueron ineficientes, difíciles de usar, y no cumplieron con las expectativas que de ellos se tenían.

A principios de los sesentas, se reconoció que muchas de las técnicas que estaban siendo usadas en los sistemas de tiempo compartido ,y en los sistemas en línea para la "programación de tiempos" de los trabajos de los usuarios y para la ejecución simultánea del proceso de cómputo de un usuario con el proceso de E/S de otro, podían ser usados en un medio ambiente de procesos en lote para mejorar el tiempo de respuesta (turnaround time) de los usuarios y la utilización de los recursos del sistema.

A mediados de los sesentas, las firmas más grandes de computadoras introducen una nueva generación de equipos de cómputo. Se incluyen muchas características en estos equipos con el fin de facilitar la operación del soporte lógico de sistemas. Por ejemplo, se usaron técnicas especiales de direccionamiento de memoria para facilitar el uso concurrente de éste recurso, por varios programas; se introdujeron mecanismos de protección para prevenir la sobrescritura inadvertida entre el sistema operativo y los usuarios; se crearon instrucciones especiales para asistir al desarrollo de actividades concurrentes que cooperan entre sí. Además se introdujeron relojes para facilitar las interrupciones necesarias en los sistemas de tiempo compartido.

El soporte lógico de sistemas introducido en estas computadoras fue extensivo- por primera vez, el soporte lógico de sistemas era considerado como una ventaja de ventas. Los fabricantes hicieron un marcado énfasis en advertir sobre las grandes facilidades ofrecidas por el soporte lógico de sistemas incluido en sus máquinas.

Este soporte lógico de sistemas de tercera generación representó un gran avance conceptual. Al principio, en forma general se adoptó la noción de multiprogramación por medio de la cual varios trabajos podían correr concurrentemente en modo lote. Se introdujeron complicados algoritmos de "programación en el tiempo" de trabajos, los cuales permitían a los usuarios asignar prioridades a los trabajadores, de manera que el monitor de procesos en lote podía iniciarlos en una modalidad diferente a la de "primero en llegar primero en servirse".

Los sistemas operativos fueron diseñados para soportar los tres tipos de ambientes de cómputo- modo lote, modo línea, y

tiempo compartido- algunas veces concurrentemente. La mayoría de los sistemas operativos de tercera generación, notablemente GCOS de Honeywell (originalmente desarrollado por General Electric), fueron diseñados de manera que al mismo tiempo, los usuarios podían estar procesando en lote, en línea e interactivamente en una computadora. También se introdujeron nuevos y avanzados procesadores de lenguajes y otras utilerías en los equipos de tercera generación. Aparecieron compiladores más complejos para COBOL y FORTRAN. IBM introduce un lenguaje procedural mas avanzado, PL/1.

Se introdujeron compiladores para FORTRAN, cuyos objetivos primarios fueron el generar mejores diagnósticos y asistir al programador en el desarrollo de sus programas, estos compiladores optimizaban el código objeto para maximizar la eficiencia en el tiempo de ejecución. También se introdujeron utilerías medianamente complejas para el manejo de datos en los sistemas de tercera generación.

Mientras que en el pasado las rutinas de entrada y salida estaban relacionadas principalmente con la lectura y escritura de cintas y equipo de registro unitario, el soporte lógico de tercera generación estuvo muy relacionado con las técnicas de lectura y escritura de varios tipos de organizaciones de datos en disco. El método de acceso secuencial indexado, fue introducido en varias versiones por todos los grandes fabricantes de computadoras.

Hasta mediados de la década de 1960, los autores de sistemas de tiempo compartido y de los sistemas en línea eran responsables de escribir sus propios programas de control de comunicaciones. En los sistemas de tercera generación, estos programas fueron adicionados al sistema operativo y eran proporcionados por el fabricante de computadoras.

En los sistemas de tercera generación, también se introdujeron utilerías más complejas para el mantenimiento y proceso de archivos secuenciales. Sin embargo, durante mucho tiempo esas rutinas no fueron completamente diferentes a las de los sistemas de segunda generación- las rutinas de ordenamiento (sorts), por ejemplo, fueron escritas en una forma más eficiente, aunque utilizaban los mismos algoritmos.

El nuevo soporte lógico de sistemas incremento dramáticamente la funcionalidad. Sin embargo, los usuarios tenían que conocer las características del soporte lógico, así como conocieron las características del soporte físico, dado que su interfase con el sistema de cómputo era a través de éste. Al mismo tiempo, la magnitud del soporte lógico de sistemas sufrió un notable incremento.



## II.6.- EL SOPORTE LOGICO DE SISTEMAS DURANTE LOS PRIMEROS AÑOS DE LA DECADA DE LOS SETENTAS.

Desde la década de los setentas, cuando el soporte lógico de sistemas de tercera generación fue introducido, ha habido muy pocas innovaciones conceptuales de gran importancia que afecten al usuario promedio. En lugar de ello, la atención se ha enfocado hacia el refinamiento de las técnicas básicas y a la implantación de los programas de sistemas. Entre estos refinamientos se incluyen los siguientes:

- 1) Una mejor apreciación de los conceptos básicos, lo cual resultó en un diseño mas lógico y claro de los sistemas.
- 2) Una mejora en la ingeniería humana, lo cual trajo como resultado la creación de sistemas fáciles de entender y usar.
- 3) Diseños más eficientes, los cuales se reflejaron en un menor uso de memoria y de tiempo de la UCP.
- 4) Se encontró que la adición de facilidades era deseable.

Estos refinamientos se han incrementado paulatinamente entre los conjuntos de programas de sistemas ofrecidos por los fabricantes de computadoras, los cuales están en continua competencia. Cada conjunto contiene lo siguiente:

- a) El sistema operativo que controla los recursos del sistema y despacha (scheduler) los trabajos.
- b) El sistema de programación, el cual consiste de procesadores de lenguajes, usualmente compiladores e intérpretes para varios lenguajes, y los programas ligadores y cargadores asociados.
- c) Uno o más sistemas manejadores de datos.
- d) Un programa de control de red de comunicación.
- e) Rutinas de utilería para llevar a cabo las funciones "caseras" de manejo de datos.

En las subsecciones que siguen, se hará una somera descripción de las características implicadas en cada uno de los elementos anteriores; en conjunción, estos elementos conforman el medio ambiente actual del soporte lógico de sistemas de los usuarios.

### II.6.1.- SISTEMAS OPERATIVOS.

Desde principios de los sesentas, el programa más importante y complejo del soporte lógico de sistemas ha sido el sistema operativo. Los sistemas operativos fueron diseñados para

fomentar el uso de las computadoras por los usuarios, que deseaban usar los sistemas de cómputo en varios modos de operación. Algunos preferían usar los sistemas en modo lote fuera de línea, mientras que otros preferían usarlos en modo línea. Otros mas estaban interesados en el desarrollo interactivo de programas y compartir los datos y programas de diferentes usuarios. Frecuentemente un usuario esperaba hacer uso de todas estas modalidades. Al mismo tiempo, muchas instalaciones deseaban correr programas que habían sido preparados para antiguos modelos de computadoras. Podría ser suficiente para un usuario, emular una computadora antigua un cierto número de veces durante el día, pero más frecuentemente, los usuarios deseaban llevar a cabo la "emulación integrada", en la cual los trabajos en la mezcla podían contener programas, que requerían ambos modos de operación, el nativo y el emulado. En algunos casos fue necesario soportar simultáneamente un sistema operativo nuevo y un sistema operativo antiguo para acomodar los programas que corrían en ambos modos.

Además de utilizar las computadoras en diferentes formas, los usuarios deseaban que sus sistemas fueran fáciles de usar y de respuesta inmediata. En el caso de los sistemas en modo lote, los usuarios esperaban que sus resultados aparecieran rápidamente; en el caso de los sistemas en línea o de tiempo compartido, los usuarios esperaban un tiempo de respuesta corto en sus terminales. Por razones económicas, también es necesario mantener una amplia utilización de todos los componentes del sistema. Esto resultó inicialmente en sistemas operativos cuyo propósito primario era soportar uno u otro modo de operación y proporcionar una respuesta rápida mediante una utilización eficiente del equipo destinado para este modo de operación. Para otros modos de operación, los sistemas operativos usualmente tenían como características una mala utilización de los recursos o un tiempo de respuesta deficiente.

Por ejemplo, GE/Honeywell desarrollaron MULTICS, un sistema que fue diseñado principalmente para facilitar el desarrollo en línea de programas de computadora y el permitir a varios usuarios compartir programas desarrollados en forma independiente, así como también compartir bases de datos. El objetivo principal fue la facilidad de uso para el desarrollo en línea de sistemas. Como resultado, el sistema fue algo deficiente cuando corría trabajos en modo lote.

Sin embargo, otros sistemas operativos enfatizaron el proceso eficiente de trabajos en modo lote. Por ejemplo, el sistema SCOPE de Control Data, fue diseñado para procesar fácil y efectivamente trabajos en modo lote. Otros sistemas fueron diseñados principalmente para aplicaciones en línea o en tiempo real. Un ejemplo de este tipo de sistemas operativos es el RSX, desarrollado por Digital Equipment Corporation para su serie PDP-11. De hecho, Digital ofreció cinco sistemas operativos, cada uno diseñado para soportar un modo de

operación en particular.

No obstante, muchos fabricantes no han estado dispuestos a desarrollar un sistema operativo para un modo de uso específico, debido a que desean que sus productos sean atractivos al mercado mas amplio posible. En lugar de ello, se han desarrollado sistemas operativos de propósito general, los cuales , aunque hacen un mayor énfasis en un modo de operación particular, pueden soportar otros modos de operación en un grado aceptable.

Por ejemplo, el sistema UNIVAC EXEC-1100 y el GCOS de Honeywell fueron usados principalmente para sistemas de procesamiento en modo lote, sin embargo, éstos sistemas soportan muy eficientemente aplicaciones en línea cuando es necesario. Los fabricantes aún no han resuelto el conflicto entre una respuesta rápida contra una alta utilización.

IBM también ha reconocido la dificultad de desarrollar un sistema operativo que sea efectivo, en un rango amplio de tamaños y capacidades de máquinas, el cual al mismo tiempo pueda soportar en forma efectiva varios modos de operación. En la línea System/370, se ofrecieron tres sistemas operativos (DOS, OS/MFT, y OS/MVT). En los modelos más avanzados de la línea System/370, IBM también ofreció varios sistemas operativos.

Aún cuando se han desarrollado muchos sistemas operativos diferentes, todos ellos son muy semejantes conceptualmente, debido a que deben cumplir con los mismos requisitos, a saber:

a) Deberán "programar en el tiempo" los trabajos a través del sistema.

b) Deberán extender las capacidades lógicas de la máquina para hacerla fácil de usar- por ejemplo, proporcionando soporte lógico de punto flotante en una computadora que no tenga soporte físico de punto flotante.

c) Deben administrar y distribuir los cinco recursos más importantes del sistema.

Los cinco recursos más importantes que deben ser distribuidos son los siguientes:

1) El procesador aritmético central. En varios sistemas se encuentran disponibles más de un procesador para la ejecución de los trabajos. El sistema operativo decide que trabajo debe ser asignado a alguna UCP en cualquier momento dado, y cual deberá ser el límite de tiempo, en el cual el trabajo haga uso del procesador.

2) Memoria principal. El sistema operativo debe incorporar rutinas de manejo de memoria, las cuales controlen la

asignación de éste recurso a los trabajos.

3) Canales. Estas son las trayectorias hacia los dispositivos de entrada y salida. En muchos sistemas, los canales son compartidos por varios dispositivos de E/S y frecuentemente están asignados a más de un usuario. El sistema operativo debe determinar cual requerimiento de uso de los canales debe ser atendido.

4) Dispositivos de Entrada y Salida. los dispositivos de E/S pueden estar dedicados a un solo trabajo, o pueden ser compartidos por varios trabajos que corren concurrentemente. Muchos sistemas permiten que los requerimientos de dispositivos sean manejados por clase de dispositivo, en lugar de ser hechos por unidad específica; el sistema operativo decide que unidad de las que están disponibles debe ser asignada al usuario. También debe establecer prioridades entre los usuarios que compiten por un dispositivo compartido.

5) Información. Los recursos de información de un sistema incluyen los archivos de datos de los usuarios, los programas del usuario, y algunos programas de sistemas; los cuales deben distribuirse entre los usuarios. Algunos sistemas permiten que los archivos de datos y los programas sean compartidos por varios usuarios, mientras que otros sistemas requieren que los archivos de datos y los programas esten dedicados a un solo usuario a la vez. El sistema operativo debe controlar cualquier uso concurrente tanto de los archivos de datos como de los programas.

Las técnicas de despacho o "programación en el tiempo" de cualquier sistema operativo, están estrechamente relacionadas con sus técnicas de asignación de recursos. En muchos sistemas, la asignación de recursos está íntimamente relacionada a unidades llamadas "trabajos", "pasos de trabajo" (en el caso de sistemas en línea y sistemas de tiempo compartido), o "sesiones de usuario". En cada sistema, algunos recursos son asignados a nuevas unidades de trabajo cuando comienza un paso de trabajo o una sesión. Se dice que tales recursos están "programados en el tiempo a alto nivel". Un ejemplo de un recurso que normalmente tiene una "programación en el tiempo de alto nivel" es un dispositivo no compartido, tal como un manejador de cinta. Otros tipos de recursos, tienen que ser asignados sólo por intervalos de tiempo muy pequeños; estos tienen que ser transferidos en cuestión de milisegundos entre los procesos en competencia y son administrados por una "programación en el tiempo de bajo nivel". El principal ejemplo de un recurso que tiene que ser "programado en el tiempo a bajo nivel" es la unidad central de procesamiento en un sistema de multiprogramación; en resumen las unidades rápidas reciben menos prioridad que las lentas para reducir el tiempo de respuesta.

En cualquier caso, aún cuando se han desarrollado muchos sistemas operativos con diferentes enfoques en la asignación de recursos, y a que una amplia variedad de términos es usada para

describir los diferentes productos, la arquitectura básica y los objetivos de la mayoría de los sistemas operativos son similares.

## II.6.2.- AYUDAS A LA PROGRAMACION.

Uno de los mayores problemas a los que se han enfrentado los usuarios de computadoras, ha sido el poder describir a la computadora el proceso que desea realizar. Se han diseñado muchos lenguajes para ayudar a los usuarios a describir sus procesos en una forma más natural. Obviamente, deben proporcionarse procesadores de lenguajes, necesarios para trasladar al lenguaje de la computadora un proceso descrito en un lenguaje específico (un programa). Se han diseñado muchos lenguajes y varios tipos de procesadores, así como también otras ayudas para facilitar el proceso de especificación y traslado de los programas a instrucciones de máquina. Cada uno ofrece un conjunto particular de virtudes y fallas en el proceso de satisfacer el complejo conjunto de requerimientos del usuario.

El tipo más "crudo" de lenguaje, disponible a la mayoría de los usuarios de computadoras, es el lenguaje ensamblador. Cuando se usa lenguaje ensamblador, una instrucción específica debe ser escrita antes de que cada instrucción de máquina pueda ser ejecutada; por lo tanto, el uso de un lenguaje ensamblador consume mucho tiempo en la programación. Los mnemónicos son substituidos por códigos de operación de instrucciones y los identificadores simbólicos por localidades de memoria, donde se tienen almacenados datos u otras instrucciones. La sintaxis y la semántica de un lenguaje ensamblador están estrechamente ligadas a la máquina a la cual está asociado. Mientras que para el programador inexperto los lenguajes ensambladores parecen ser muy difíciles, al programador experto le son sorprendentemente similares y fáciles de aprender. Teniendo un control completo sobre las instrucciones ejecutadas, el programador experto puede escribir programas muy eficientes en lenguaje ensamblador; los requerimientos del usuario para una operación económica son alcanzados satisfactoriamente.

Debido al grado de dificultad y al alto consumo de tiempo requerido en la programación al hacer uso de un lenguaje ensamblador, se crearon los macroprocesadores o "macros", los cuales están estrechamente relacionados con el lenguaje ensamblador. Un macroprocesador permite al usuario definir una secuencia específica de operaciones bajo cierto nombre, y escribir una instrucción en lenguaje ensamblador que ejecute la secuencia nombrada como si fuera una instrucción de la máquina. Por ejemplo, muchos macroensambladores tienen códigos de macrooperación para aritmética de punto flotante, para su uso, cuando el soporte físico es incapaz de realizar este tipo de aritmética. Cuando un programador desea ejecutar una operación aritmética de punto flotante, simplemente usa la

pseudoinstrucción asociada a la macro; la secuencia adecuada de instrucciones de aritmética de punto flotante es insertada en el tiempo de ejecución.

Aun con el uso de macroensambladores, la programación continuó siendo un proceso prolongado, lo que originó que se desarrollaran los compiladores o lenguajes. En la actualidad los principales lenguajes en uso son: COBOL, FORTRAN, BASIC, ALGOL, PASCAL, PL/1 y APL.

FORTRAN y BASIC permiten al programador describir un proceso de una manera algebraica. El programador puede establecer algoritmos como una secuencia de fórmulas escritas en forma similar a como se escriben en el lenguaje algebraico común. Se pueden manipular y practicar operaciones sobre escalares, vectores y matrices haciendo uso de estos lenguajes. Sin embargo, las operaciones sobre cuerdas de caracteres son mucho más difíciles de realizar. Usando COBOL es fácil describir algoritmos que requieren el movimiento y formateo de textos, así como la realización de operaciones simples de tipo contable sobre números. COBOL ofrece facilidad de uso cuando los programadores tratan estrictamente con problemas de tipo administrativo básico.

En muchas versiones de COBOL no existen facilidades para el manejo de expresiones algebraicas. En PL/1, se realizó un intento para combinar las características de manipulación algebraica de FORTRAN, con las capacidades de manipulación y edición de textos de COBOL. Mientras que este lenguaje posee características que le permiten describir procesos en un amplio rango de áreas, su uso es muy embarazoso; en el intento de universalidad tiene comprometida su facilidad de uso.

Además de estos lenguajes, se diseñaron otros más, cuyos objetivos principales fueron aumentar la facilidad para escribir ciertas clases de algoritmos. SNOBOL, por ejemplo, se diseñó para asistir a los usuarios en la escritura de algoritmos que requerían la manipulación de cuerdas de caracteres.

Muchas organizaciones han implantado variantes de los lenguajes orientados para aumentar su potencia, aprovechando características del equipo físico. Al hacer esto y en un intento por aumentar la facilidad de uso de los lenguajes, han desarrollado dialectos o versiones para cada uno de los principales lenguajes. En muchos casos, estos dialectos (especialmente FORTRAN), han mejorado el lenguaje desde el punto de vista de la escritura de programas. Sin embargo, la proliferación de dialectos ha causado otros problemas serios. Por ejemplo, un programador que escribe en un dialecto frecuentemente tiene problemas al iniciarse en el uso de otra versión del lenguaje. Además, un programa escrito en un dialecto es posible que no pueda ser trasladado a otro dialecto por medio de un filtro. Como resultado, se iniciaron esfuerzos extensivos de estandarización para satisfacer las necesidades del usuario, en cuanto a la portabilidad de los

programas y de la experiencia del mismo. No obstante, los estándares resultan en lenguajes que no son óptimos en cuanto a su facilidad de uso para varias subclases de usuarios.

La mayoría de los compiladores tienen la facilidad de uso de subrutinas. Esta capacidad permite al usuario escribir una porción de código en el lenguaje y, en varios lugares del programa principal hacer llamadas a la subrutina. Esto proporciona dos ventajas importantes -primero, permite reusar programas que previamente han sido probados; segundo, reduce drásticamente el tamaño de los programas trasladados, debido a que las subrutinas no necesitan estar incluidas. Por lo tanto, la facilidad de usar subrutinas ayuda a satisfacer los requerimientos del usuario en cuanto a facilidad de programación y confiabilidad. Los macroprocesadores fueron desarrollados para asistir a los usuarios en el desarrollo de programas con compiladores.

Aún con los compiladores, subrutinas, macroprocesadores y otros tipos de ayudas, es muy difícil para el usuario no programador, quien no está particularmente interesado en el proceso de datos, describir un proceso para su ejecución en una computadora. Como consecuencia, se desarrollaron lenguajes orientados a problemas específicos. Los lenguajes orientados a problemas, permiten al usuario este tipo de lenguajes en varias áreas, especialmente en la ingeniería civil. Por ejemplo, COGO(Coordinate Geometry), fue diseñado para la descripción y solución de problemas de la ingeniería civil. El lenguaje es fácil de usar; con pocas horas de entrenamiento, cualquier ingeniero civil puede describir y solucionar problemas complejos. También se han ideado lenguajes para describir problemas de ingeniería estructural.

Mientras que los lenguajes orientados a problemas de ingeniería civil, han sido ampliamente aceptados y son extensivamente usados, en otras disciplinas se han desarrollado muy pocos de estos lenguajes. La razón de ello parece estar principalmente relacionada a la naturaleza altamente estructurada de la terminología del campo de la ingeniería civil.

En tales circunstancias, cuando los procedimientos de descripción para una clase amplia de problemas y sus soluciones son bien entendidas y concordantes, o cuando un lenguaje de discurso sobre los problemas dentro de una disciplina son congruentes y generalmente entendidas, es posible desarrollar lenguajes orientados a problemas. En otras situaciones, tales como en contabilidad, donde se usan varios términos diferentes para describir el mismo concepto y los procedimientos para realizar tareas ampliamente usadas no están estandarizadas; es difícil desarrollar lenguajes orientados a problemas específicos.

En el futuro, se espera que haya más lenguajes orientados a problemas. Sin embargo, parece que el crecimiento de estos lenguajes será lento. No obstante, nuevos lenguajes intrínsecamente superiores harán frente a la inercia de la familiaridad de los usuarios con los lenguajes actuales.

### II.6.3.- OTRAS AYUDAS.

En los primeros días de la computación, los compiladores trasladaban la versión de un proceso en lenguaje procedural, a una versión en lenguaje ensamblador. En la actualidad existen técnicas muy complejas, mediante las cuales los lenguajes pueden ser traducidos directamente a un lenguaje de máquina. De hecho, existen algunos sistemas que pueden aceptar la definición de un lenguaje y producir en forma automática un compilador (el sistema XTL es un ejemplo). Tales procesadores son llamados compiladores de compiladores (compiler-compilers), y aumentan la facilidad de uso al soportar una amplia variedad de lenguajes orientados a problemas.

Otros tipos de procesadores de lenguajes que se desarrollaron fueron los intérpretes y generadores. En lugar de traducir un proceso escrito en un lenguaje de programación procedural, a lenguaje de máquina y ejecutar el programa objeto, un intérprete lee los datos sobre los que va a operar el programa y posteriormente interpreta y ejecuta, instrucción por instrucción el lenguaje procedural original. Los procesadores de APL y BASIC trabajan de este modo, operan en tiempos de respuesta cortos, pero consumen muchos recursos.

Frecuentemente los usuarios desean utilizar programas que han sido escritos con anterioridad, en combinación con programas que han sido escritos recientemente. A fin de satisfacer esta necesidad, los vendedores diseñaron los programas ligadores. Estos permiten tomar varios programas traducidos separadamente y combinarlos en uno solo. En sistemas de memoria real, los ligadores hacen posible especificar en que parte de la memoria el programa residirá y como deberá superponerse asimismo durante la ejecución.

### II.6.4.- SISTEMAS MANEJADORES DE BASES DE DATOS.

Los principales servicios de la mayoría de los sistemas de información siempre han consistido en crear archivos, su mantenimiento mediante adiciones, bajas, y cambios en los registros; la preparación de reportes sobre las transacciones procesadas y el estado actual de los archivos. Por ejemplo, en muchos sistemas de procesamiento en lote se preparan varios reportes y documentos en base a las transacciones procesadas, las cuales son usadas para actualizar los archivos básicos. Con cierta periodicidad se preparan reportes de sumariación que describen el estado de los archivos. Aplicación tras



aplicación, los programadores tenían que escribir programas para cargar la base de datos, implantar procedimientos para tener acceso a registros específicos de la base de datos, y preparar reportes de un subconjunto de ésta. Se reconoció que su labor podría reducirse al hacer uso de un paquete de soporte lógico de sistemas, el cual debería consistir de:

1) Un lenguaje de definición de datos, el cual se usará para definir la base de datos.

2) Un lenguaje de consulta, de modo que puedan obtenerse los registros individuales, un subconjunto de la base de datos y reportes escritos.

3) Un sistema anfitrión para uno o más compiladores, desde desde los cuales se manejará la base de datos.

4) Programas de utilería para realizar funciones "caseras".

Tales paquetes de soporte lógico han sido llamados "Sistemas Manejadores de Bases de Datos". Además de la capacidad para la definición de archivos y el mantenimiento de los datos en los mismos, un sistema manejador de bases de datos proporciona el mecanismo para ligar los datos a los programas en cualquiera de las siguientes dos formas: 1) los datos pueden ser elegidos y formateados escribiéndolos en archivos en cinta, o 2) pueden ser obtenidos directamente desde un programa de aplicación, el cual corre dentro del medio ambiente de un sistema manejador de bases de datos. Los sistemas manejadores de bases de datos algunas veces también son capaces de facilitar la base de datos a ciertas personas mediante un subsistema de control de terminales y un lenguaje de consulta asociado, el cual permite a los usuarios elegir y procesar un subconjunto de la base de datos en modo línea, independientemente de cualquier lenguaje de programación normal. Tales lenguajes de consulta generalmente son diseñados para que sean muy fáciles de usar.

El concepto de sistema manejador de base de datos generalizado es muy atractivo y la percepción de las necesidades del usuario siempre ha sido muy variada, lo que ha traído como consecuencia que alrededor de un centenar de sistemas manejadores de bases de datos hayan sido producidos. Sin embargo, muchos de estos sistemas no se encuentran en uso activo. Muchos de los sistemas manejadores de bases de datos no han satisfecho las especulaciones de los usuarios, ni han reconocido la gran diversidad de sus requerimientos, los cuales deben ser cubiertos por un sistema manejador de bases de datos generalizado. Aún cuando un enfoque generalizado a la solución de los problemas de manejo de datos conduce a una pérdida de eficiencia en situaciones específicas, la tendencia multimodal actual en el procesamiento de datos ha estimulado el desarrollo de sistemas manejadores de datos generalizados. La incorporación de muchas características dentro de los sistemas, las cuales reducen los requerimientos de programación por parte del usuario, parecen compensar el costo de una operación lenta para muchos usuarios.

Los sistemas manejadores de bases de datos son mejor entendidos en términos de tres características principales:

1) La complejidad de la base de datos que puede ser manejada por el sistema.

2) La complejidad de las consultas que pueden ser realizadas sobre la base de datos.

3) Las características de funcionamiento del sistema para la consulta y actualización de registros, y los procesos de adición y eliminación de los mismos.

Algunos sistemas manejadores de bases de datos son capaces de manejar estructuras de archivos de tipo red extremadamente complejas. Este tipo de sistemas está caracterizado por el sistema IDS de Honeywell y el IDMS Standard de CODASYL. Otros sistemas solo manejan estructuras simples de archivos, tales como los archivos orientados a estructuras jerárquicas u orientados a estructuras de cadenas. Paquetes tales como TOTAL e IMS de IBM están en esta categoría.

Algunos de los sistemas solo permiten a los usuarios interactuar con los datos a través de programas de aplicación escritos en lenguajes procedurales, usualmente COBOL y PL/1. Por ejemplo, TOTAL de CINCOM e IMS de IBM tienen estas características. Otros sistemas manejadores de bases de datos han sido diseñados para permitir al usuario sin conocimientos de procesamiento de datos, hacer consultas a los archivos en una forma relativamente simple. Frecuentemente, no es necesario que el usuario entienda la forma en que el archivo está orientado lógicamente al hacer sus consultas. Los sistemas de archivos invertidos (tales como algunos sistemas relacionales de bases de datos), permiten al usuario no programador hacer consultas a la base de datos.

En resumen, las diversas percepciones de las necesidades del usuario, han propiciado la construcción de un gran número de sistemas manejadores de bases de datos. Aunque los fabricantes de computadoras han diseñado e implantado muchos de estos sistemas, un número considerable de ellos han sido diseñados, implantados, y comercializados por compañías de soporte lógico independientes. Los sistemas varían ampliamente en sus objetivos: algunos enfatizan la facilidad para representar y estructurar relaciones complejas entre elementos de datos, sin darle mucha importancia al tiempo de respuesta, o la facilidad de que el programador obtenga elementos de datos específicos, algunos otros sistemas están poco interesados en la complejidad de las relaciones entre los elementos de datos y hacen mayor énfasis en proporcionar lenguajes de consulta naturales al humano, para el acceso a la base de datos, de manera que se pueden hacer accesos espontáneos y no formateados con una respuesta rápida.

Algunos sistemas no se preocupan por la velocidad de actualización, adición o eliminación de los registros de un archivo, sino solamente hacen énfasis en la velocidad de la consulta. En otros sistemas se ha reconocido la necesidad de que muchas aplicaciones requieren la adición o eliminación continua de los registros de un archivo y enfatizan la habilidad para realizar estas operaciones en forma rápida y eficiente. Ver Apéndice A.

#### II.6.5.- PROGRAMAS DE CONTROL DE COMUNICACIONES.

Cuando los usuarios de computadoras comenzaron a usar sistemas en línea y en tiempo real (desde mediados de la década de 1950), eran responsables de desarrollar todos los programas de control de comunicaciones, y debían entender como coordinar varias tareas asíncronas y diferentes actividades de electrónica que normalmente no tenían porque entender. Pronto se reconoció que muchas de las técnicas usadas en control de comunicaciones podían generalizarse y ser similares a las técnicas usadas por los métodos de acceso de datos. Como resultado, los fabricantes de computadoras introdujeron programas de control de comunicaciones como parte de su conjunto de soporte lógico de sistemas.

La función más complicada de un programa de control de comunicaciones es controlar la red. El programa debe manejar la entrada y salida al sistema de las terminales. Además, como muchos sistemas utilizan más de una terminal en una sola línea, el programa debe identificar que terminales estan mandando o recibiendo mensajes. Además de mandar y recibir mensajes, el programa debe mandar y recibir instrucciones de control de los dispositivos -por ejemplo, controlar el carro de la impresora, limpiar la pantalla de un CRT (tubo de rayos catódicos), etc; por lo tanto el programa debe conocer la naturaleza y el repertorio de instrucciones de cada terminal. También se debe reconocer todo tipo de tráfico de mensajes y órdenes en la red, tanto del mismo controlador de teleproceso como de otros sistemas de enlace. Se desarrollaron procedimientos llamados "protocolos de comunicación" para la inserción de "bits" de chequeo en los mensajes y la diferenciación entre "bits" de control, reconocimiento, y texto. Los protocolos más ampliamente usados son: el protocolo de arranque-parada (start-stop), y el protocolo bisíncrono de IBM, así como el SDLC. El soporte lógico de control también debe llevar a cabo las actividades de ensamblado de mensajes y control de "buffer" (área de almacenamiento temporal). En la mayoría de las redes de comunicación hay varias líneas y varias terminales transmitiendo y recibiendo señales de la computadora, carácter por carácter. El programa de control recibe los mensajes de los programas de aplicación en una forma totalmente ensamblada. El programa de control de mensajes debe primeramente subdividir estos mensajes en bloques de protocolo y mandar el mensaje carácter por carácter a la terminal correcta.

El programa de control de comunicaciones también debe realizar otras tres actividades: "encolar" los mensajes, la activación del soporte lógico de aplicaciones, y la edición de mensajes. Frecuentemente, cuando se recibe un mensaje desde una terminal, el programa de aplicación que debe procesarlo no está disponible, ya sea porque no se encuentra activo, o porque está procesando un mensaje anterior de otra terminal. En este caso el mensaje debe ser puesto en una cola de espera, ya sea en memoria o en disco. En forma similar, frecuentemente la computadora genera mensajes a una velocidad mayor a la que pueden manejar, ya sea la línea asignada o las terminales. En esta situación, los mensajes generados también deben ser encolados para su manejo posterior. Asociado a este proceso de encolamiento está el problema de intercomunicación entre procesos, para notificar a un programa de aplicación que hay un mensaje esperando ser procesado o la desactivación de un programa después de que el mensaje ha sido mandado.

En algunas situaciones el programa de control de comunicaciones debe realizar las actividades de edición de mensajes. Usualmente tiene más sentido para el programador de aplicaciones poner el mensaje destinado a una terminal en un "buffer" de salida en un formato familiar, en lugar de tener que preparar el mensaje para su transmisión. El programa de control de comunicaciones debe adicionar los encabezados y los caracteres de control necesarios, así como cambiar la representación de los caracteres cuando es necesario. Similarmente, puede requerirse la edición de los mensajes de entrada para la traslación de caracteres, la eliminación de controles especiales, y el chequeo de caracteres.

En la actualidad, todos los fabricantes de computadoras proporcionan programas muy elaborados de control de comunicaciones, lo cual permite al usuario desarrollador de aplicaciones considerar a una terminal en una red de telecomunicaciones en forma similar a un dispositivo de E/S, conectado en forma lógica. La mayoría de los programas permiten al usuario que desarrolla aplicaciones escribir un mensaje a una terminal, identificarla por un nombre lógico, en un formato que le es familiar. El programa de control tiene ingerencia sobre todas las funciones de edición de mensajes, encolamiento, y manejo de terminales.

#### II.6.6.- OTRO SOPORTE LOGICO DE SISTEMAS.

En adición a los procesadores de lenguajes, sistemas operativos, sistemas manejadores de bases de datos, y programas de control de comunicaciones, muchos vendedores de computadoras también proporcionan soporte lógico de sistemas para propósitos de mantenimiento, medición, contabilidad y utilerfías.

Los fabricantes de computadoras deben desarrollar programas para el mantenimiento del sistema, de modo que puedan probarse las máquinas y diagnosticar posibles errores. Usualmente, los

programas de ejercicio y diagnóstico para los componentes del sistema son proporcionados para probar cada uno de éstos componentes; normalmente hay un ejercitador para el sistema de memoria principal, uno para el procesador central, y uno para cada uno de los periféricos. Estos son usados principalmente por el personal de mantenimiento, pero algunos de éstos programas de prueba también son útiles al usuario para aislar las fallas de sus programas. Estos programas también son necesarios para realizar funciones tales como: la inicialización de "paquetes" de discos -esto es, checar que cada pista del paquete sea funcional y colocar la tabla de contenido de volumen en alguna localidad específica. En algunas situaciones las actividades de inicialización son realizadas por el personal del vendedor; en otras, se espera que sea el usuario quien realice estas actividades. La tendencia es hacia una dependencia mayor en el usuario para correr los programas de inicialización y diagnóstico de todo tipo, una tendencia que aumenta la presión para que los desarrolladores de estos programas los hagan infalibles y fáciles de usar.

Muchos sistemas operativos tienen integradas facilidades para reprocesar transmisiones defectuosas -lecturas de cinta, discos, tarjetas, y mensajes mandados de o hacia una terminal. La información acerca de cada una de las actividades de reproceso son registradas en un archivo de bitácora del sistema. Están disponibles utilerías para el análisis de la bitácora del sistema, de manera que el personal de mantenimiento, o los usuarios puedan identificar los recursos que están teniendo un número alto de reprocesos.

Muchos sistemas operativos y otros programas de sistemas registran información sobre la actividad del sistema. Se registra para cada trabajo procesado información tal como: el tiempo de procesador central, el número de procesos de control de canales, la cantidad de datos transferidos sobre un canal, etc. Normalmente esta información es registrada en una forma muy condensada en cinta o disco, y se proporciona soporte lógico de sistemas que puede ser usado para extraer y sumarizar la información para propósitos de contabilidad y "afinación" del sistema. Hasta ahora, estos programas de medición no han recibido mucha atención por parte de los fabricantes de computadoras. La información proporcionada para la contabilidad y "puesta a punto" del sistema frecuentemente es inadecuada o incompleta. Aún en aquellos casos donde se registra adecuadamente la información detallada, se proporcionan pocos (o ninguno) programas de propósito general para generar los reportes necesarios. Hasta la fecha, los programas de medición más exitosos han sido desarrollados por firmas independientes de soporte lógico.

Normalmente una instalación de cómputo necesita un gran número de programas de utilería. Estos incluyen rutinas de conversión de medio a medio, rutinas de administración de

memoria secundaria, y programas que lleven a cabo funciones repetitivas y simples de procesamiento de datos. Muchos vendedores proporcionan como parte de su biblioteca de soporte lógico de sistemas, rutinas de conversión de medios (las cuales pueden ser activadas con relativamente pocas tarjetas de control), para realizar actividades tales como el copiado de disco a cinta, de cinta a impresora, de tarjetas a impresora, etc. Usualmente se proporcionan otras rutinas para la administración de la memoria secundaria. Estas incluyen programas para manejar y registrar las bibliotecas de los usuarios de disco y cintas. Además, normalmente se proporcionan utilerías para determinar que conjunto de datos se encuentran en cual disco o paquete de discos, para determinar la cantidad de espacio disponible en el "paquete" de discos, y para reordenar los conjuntos de datos en los discos, para liberar mayores bloques de espacio de almacenamiento, así como para mejorar en su totalidad el funcionamiento del sistema.

El soporte lógico de sistemas para realizar funciones simples de proceso de datos, tales como: los "sorts" (métodos de ordenamiento), normalmente también son parte del conjunto de soporte lógico de sistemas. Además se proporcionan programas para "sortear" archivos de cinta y disco; también es frecuente que se proporcionen utilerías para la mezcla de archivos y/o la selección de ciertos registros de un archivo para su impresión.

Una parte de los programas de sistemas de soporte fue proporcionada con las primeras computadoras (los primeros programas de diagnóstico y los primeros "sorts"), pero en su mayor parte fueron desarrollados para acompañar a los sistemas operativos multifuncionales anunciados desde 1964.

## II.7.- LENGUAJES DE CUARTA GENERACION.

Los lenguajes de cuarta generación (L4G) aparecieron a fines de la década de los setentas y actualmente hay más de un ciento de éstos lenguajes. Los L4G se clasifican en:

- 1) Simples generadores de programas de COBOL.
- 2) Lenguajes generadores de programas de COBOL, con apoyo de sistemas administradores de bases de datos.
- 3) Lenguajes de cuarta generación "verdaderos" que además de las características anteriores, utilizan en gran parte "soporte lógico" estándar del proveedor de equipos de cómputo, como: sistemas de comunicación de datos, respaldo para protección y varias aplicaciones de utilería.

Las herramientas de cuarta generación siguen la tendencia de las primeras generaciones de soporte lógico, de reducir y simplificar el lenguaje necesario a los humanos para instruir a una computadora.

En la siguiente tabla se muestran algunos lenguajes de cuarta generación.

EJEMPLOS DE LENGUAJES DE CUARTA GENERACION

I	I DISPONIBLES PARA USUARIOS	I
I	I FINALES	I VENDEDOR
I	I	I
I LENGUAJES DE	I QUERY-BY-EXAMPLE	I IBM
I CONSULTA A	I INTELLECT	I ARTIFICIAL
I BASES DE DATOS	I	I INTELLIGENCE
I	I QUICK QUERY	I CACI
I	I EASYTRIEVE	I PANSOPHIC
I	I ASI/INQUIRY	I ASI
I	I SQL	I IBM
I	I	I
I SISTEMAS DE	I STAIRS	I IBM
I RECUPERACION DE	I CAFS	I ICL
I INFORMACION	I	I
I	I	I
I GENERADORES DE	I NOMAD	I NCSS
I REPORTES	I QUICK QUERY	I
I	I	I
I GENERADORES DE	I MAPPER	I UNIVAC
I APLICACIONES	I RAMES II	I MATHEMATICA
I	I	I INC
I	I FOCUS	I INFORMATION
I	I	I BUILDERS INC.
I	I LINC	I BURROUGHS
I	I COGEN	I JACKSONVILLE
I	I	I
I LENGUAJES DE	I APL/DI (Funciones	I IBM
I PROGRAMACION DE	I simples)	I
I MUY ALTO NIVEL	I NOMAD (Funciones simples)	I NCSS
I	I FOCUS	I INFORMATION
I	I	I BUILDERS INC.
I	I IDEAL	I ADR
I	I AME	I IBM

Continuación

I	I DISPONIBLES PARA		I
I	I PROFESIONALES	VENDEDOR	I
I	I		I
I LENGUAJES DE	I QUICK QUERY	CACI	I
I CONSULTA A	I EASTRIEVE	PANSOPHIC	I
I BASE DE DATOS	I GIS	IBM	I
I	I MARK IV	INFORMATICS	I
I	I DATATRIEVE	DEC	I
I	I SQL	IBM	I
I	I		I
I GENERADORES DE	I NOMAD	NCSS	I
I REPORTES	I QUICK QUERY		I
I	I GIS	IBM	I
I	I IBM SYSTEM UTILITIES	IBM	I
I	I RPG II	VARIOS	I
I	I RPG III	IBM	I
I	I ADRS	IBM	I
I	I MARK IV/REPORTER	INFORMATICS	I
I	I		I
I GENERADORES DE	I ADF	IBM	I
I APLICACIONES	I RAMES II	MATHEMATICA I.	I
I	I DMS	IBM	I
I	I ADMINS II	ADMINS	I
I	I USER 11	NORTHCOUNTY	I
I	I LINC/BURROUGHS	COMP. INC.	I
I	I ADS	CULLINANE	I
I	I APPLICATION BULDER	DMW	I
I	I		I
I LENGUAJES DE	I APL	IBM	I
I PROGRAMACION DE	I ADRS	IBM	I
I MUY ALTO NIVEL	I NOMAD	NCSS	I
I	I FOCUS	INFORMATION	I
I	I MANTIS	CINCOM	I
I	I NATURAL	SOFTWARE A.G.	I
I	I IDEAL	ADR	I

En la misma forma que el desarrollo de macros para compiladores de ensambladores dieron origen a los lenguajes de programación, los desarrollos específicos en superlenguajes aunados a las experiencias y prácticas de cómputo para la administración ha dado lugar a los L4G.

Los L4G se caracterizan principalmente por la reducción del tiempo requerido para desarrollar proyectos de soporte lógico y por su facilidad de uso, permitiendo que usuarios de los sistemas de cómputo o personal sin experiencia en programación



participe activamente en el desarrollo y mantenimiento de aplicaciones programadas para computadora.

Los L4G poseen además las siguientes características:

- 1) Estan diseñados para operar en línea.
- 2) Producen código estructurado.
- 3) El código generado por otras personas es fácil de entender y mantener.
- 4) Se aprenden en sesiones cortas.
- 5) Estan diseñados para facilitar la depuración del código.
- 6) El tiempo requerido para obtener resultados es menor (el 90% cuando menos) en un orden de magnitud, que el tiempo necesario en los ambientes de COBOL y PL/1 estándar actuales.

Un lenguaje de cuarta generación utiliza o incluye tecnologías para aplicaciones de computadora como:

- Compiladores,
- Administración de bases de datos,
- Sistemas para reportes,
- Sistemas para consulta,
- Comunicación de datos,
- Sistemas de procesamiento de transacciones,
- Diseño interactivo en terminales,
- Sistemas de seguridad,
- Etc.

Esta nueva tecnología esta disponible para toda persona que enfrenta un problema de procesamiento de datos y son sorprendentes los tiempos en que se aprenden (3 semanas) y se obtienen resultados entre 4 y 8 semanas, con experiencia previa en procesamiento electrónico de datos o sin ella; llegando a observarse que es más fácil aprender un lenguaje de cuarta generación para el personal sin experiencia porque no tiene que armonizar las lecciones con los conocimientos previos.

Nótese que se menciona que es más fácil solo aprender a usarlo, ya que las técnicas como: organización de sistemas; control de producción; análisis de sistemas y documentación es

necesario hacerlo todavía con personal capacitado en éstas actividades.

#### II.7.1.- LIMITACIONES.

Parece haber dos límites que restringen el uso de los L4G. Estas limitaciones se refieren a la capacidad del usuario para crear y administrar aplicaciones muy complejas y la disponibilidad de equipo de cómputo para apoyar dicha capacidad.

Con la oportunidad de crear aplicaciones más complejas en menos tiempo, el usuario formulará requerimientos mucho más elaborados y de mayor rango. Al no contar con la experiencia suficiente en diseño y mantenimiento de sistemas, este aumento súbito de complejidad limitará al usuario en su capacidad para tener un rápido acceso a aplicaciones nuevas o modificadas, a menos que se les auxilie con personal experto.

Los L4G derivan su productividad, en parte, al uso de una mayor potencia de cómputo para automatizar una programación rutinaria y detallada. Por lo tanto estos lenguajes requieren una computadora más poderosa.

Existe otra objeción debido a que se cuestiona la compatibilidad general de los L4G. La mayoría de estos mecanismos se aplican a problemas específicos y emplean sistemas de bases de datos o archivos no estándar. Las interfases con los sistemas de procesamiento de datos existentes no están implantados o son muy complejas para aplicaciones en gran escala en tiempo real. Idealmente un L4G debe utilizar el soporte lógico estándar del sistema que se emplea en un medio ambiente, especialmente para servicios de comunicaciones de datos o base de datos, o bien se deben reemplazar completamente los sistemas previos. Sin embargo, en la práctica, la mayoría de los L4G utilizan sistemas de archivos propios especializados.

### **CAPITULO III**

#### **PROBLEMAS Y SOLUCIONES CON PROGRAMAS DE APLICACION**

### III.- PROBLEMAS Y SOLUCIONES CON PROGRAMAS DE APLICACION.

Existen dos tipos generales de soporte lógico: el soporte lógico de sistemas y el soporte lógico de aplicaciones.

El soporte lógico de sistemas ayuda a la gente a usar una computadora, el de aplicaciones auxilia a la gente para llevar a cabo sus negocios o resolver otros problemas sin conocimiento profundo de campos de cómputo. El soporte lógico de sistemas, incluye a los sistemas operativos, compiladores, generadores de reportes y consulta, sistemas manejadores de bases de datos, y toda clase de programas que permitan el uso eficiente de una computadora; esto también se conoce como soporte lógico estándar.

El soporte lógico de aplicación incluye: nóminas, contabilidad, control de inventarios, manejo de calendario (PERT, Rutas Críticas etc.) y otros sistemas comunes en los negocios o empresas. Este tipo de soporte lógico se desarrolla con el fin de satisfacer las necesidades particulares de los usuarios, cuando realizan aplicaciones determinadas. El soporte lógico de aplicación, también abarca a los programas específicos de manufacturas, hospitalización, bancos y otros sistemas relacionados con la industria.

El tiempo que se requiere para el desarrollo e implantación del soporte lógico, sigue siendo el factor limitante en la mayoría de las aplicaciones para computadora en las empresas. Durante las últimas décadas los componentes de los equipos para computadora aumentaron en orden de magnitud su velocidad, a la vez que redujeron su precio. Por otra parte, el tiempo necesario para desarrollar los programas de aplicación necesarios sólo se redujo modestamente, a la vez que aumentó el costo de desarrollo del soporte lógico (en gran medida por la meno de obra). De esta manera, una parte considerable de la capacidad actual de informática permanece ociosa. Téngase en cuenta que con el sueldo mensual de un analista y de un programador se puede comprar una microcomputadora, o con el sueldo anual de ambos se puede comprar una minicomputadora.

GUIDE, la organización de los usuarios de IBM, clasifica a la productividad en el desarrollo de aplicaciones como su preocupación estratégica número uno. La mayoría de los usuarios de sistemas comerciales comparten este punto de vista.

La crisis del soporte lógico surgió a pesar de los esfuerzos realizados para crear herramientas, formular técnicas y metodologías para incrementar la productividad en el desarrollo del soporte lógico.

Una de las mejoras más dramáticas a la productividad se ha logrado con la introducción del desarrollo de soporte lógico en línea.

Sin embargo, la productividad que se ganó fue muy modesta, comparada con los avances obtenidos en el desarrollo del equipo para computadora.

Las técnicas de programación estructurada y los métodos de

diseño estructurado también han mejorado la productividad, pero es muy difícil cuantificar el aumento. Al aplicar su experiencia, los usuarios comenzaron a exigir sistemas de aplicación más complejos. Lamentablemente estas técnicas, junto con las nuevas herramientas en sistemas de bases de datos y manejo de pantallas en terminales, aumentaron el volumen de subsistemas individuales e interfaces que el programador debe coordinar. De esta forma se absorbieron las modestas ganancias en productividad.

Una de las innovaciones actuales la constituyen los L4G. Estos lenguajes permiten el manejo en forma automática de los subsistemas, que el programador debía coordinar permitiendo una ganancia considerable en productividad.

Puede considerarse, para lograr un aumento en productividad, un equilibrio adecuado entre la compra de paquetes de aplicación comerciales y el desarrollo interno de programas de aplicación. Más adelante se hablará sobre la consideración de tres enfoques para lograr la implantación de un sistema de soporte lógico de aplicación, a saber: la compra de un paquete comercial, que satisfaga por lo menos nuestros requerimientos mínimos; la contratación del desarrollo de soporte lógico de aplicación de acuerdo a nuestras necesidades, y el desarrollo dentro de la empresa de los programas de aplicación.

Cada uno de estos enfoques presentan ventajas y desventajas, dependiendo de la situación que prevalezca en el momento en que se decida o surja la necesidad de automatizar algún proceso, y por ende la necesidad de obtener los programas de aplicación necesarios.

Para elegir una de éstas opciones se requiere de un análisis de costo beneficio que cada una de ellas implica.

Para hacer más clara la explicación de estas opciones, haremos una analogía con el proceso de obtención de una casa. Supongamos que debemos satisfacer nuestra necesidad de vivienda, tenemos tres opciones: podemos diseñarla y construirla nosotros mismos; podemos comprar una casa que ya ha sido diseñada y construída; o podemos contratar a un arquitecto y a un constructor para que hagan nuestra casa. El construir nosotros mismos la casa toma tiempo y posiblemente cueste más que mandarla a hacer. La compra de una casa que ya existe, posiblemente nos proporcione más de lo que requerimos de una casa, pero raramente obtendremos lo que deseamos. Contratar para que nos construyan la casa que hemos soñado es otra posibilidad. Esta opción tomará más tiempo que comprar una casa existente, pero menos del que tomaría si la construyéramos nosotros, considerando que no somos diseñadores ni constructores. La construcción de la casa de acuerdo a nuestras necesidades y gustos puede costar más que una casa existente o que si nosotros la construimos. Si tenemos tiempo para esperar, pero ese tiempo lo tenemos dedicado a otras actividades, entonces la contratación puede ser la mejor manera de obtener la casa que hemos elegido.

Lo anterior también es verdad para el soporte lógico.

El sistema de soporte lógico de aplicación que requerimos podemos obtenerlo de tres maneras:

1) Desarrollando en la empresa el soporte lógico de aplicación que requerimos.

2) Adquiriendo soporte lógico comercial estándar, también llamado "paquetes de aplicación" o simplemente "paquetes".

3) Contratando el desarrollo de soporte lógico.

En los párrafos siguientes se hará una descripción de cada una de estas alternativas haciendo énfasis en sus aspectos más relevantes.

### III.1.- DESARROLLO DEL SOPORTE LOGICO DENTRO DE LA EMPRESA.

Una opción para obtener el soporte lógico de aplicación que requerimos consiste en desarrollarlo dentro de la empresa. Para inclinarnos por esta opción debemos tener en cuenta factores como:

- Costos,
- Tiempo disponible,
- Experiencia del personal,
- Recursos de cómputo disponibles,
- Beneficios derivados del proyecto,
- Etc.

El estudio de estos factores conforma el análisis de factibilidad del proyecto (desarrollo de programas de aplicación), el cual está integrado dentro de una metodología de desarrollo. Una metodología define como deben realizarse las tareas que nos conducen a la obtención de un sistema terminado, así como también el orden en el cual estas tareas deben ser realizadas. Es necesario para poder implantar proyectos de soporte lógico, el contar con una metodología.

Una metodología que divida en fases nuestro trabajo, rompe el proceso de desarrollo en unidades de trabajo manejables. Las fases proporcionan puntos de chequeo significativos y especifican productos intermedios, lo cual nos permite tener un mejor control del proceso.

En la literatura ( ver Apéndice E ) existe un número considerable de metodologías propuestas, estas metodologías varían dependiendo de la magnitud del esfuerzo de desarrollo (alcance del sistema, dimensión, tipo de aplicación, etc.)

En los párrafos siguientes se presenta un esquema general de una metodología para el desarrollo del soporte lógico de aplicación que necesitamos.

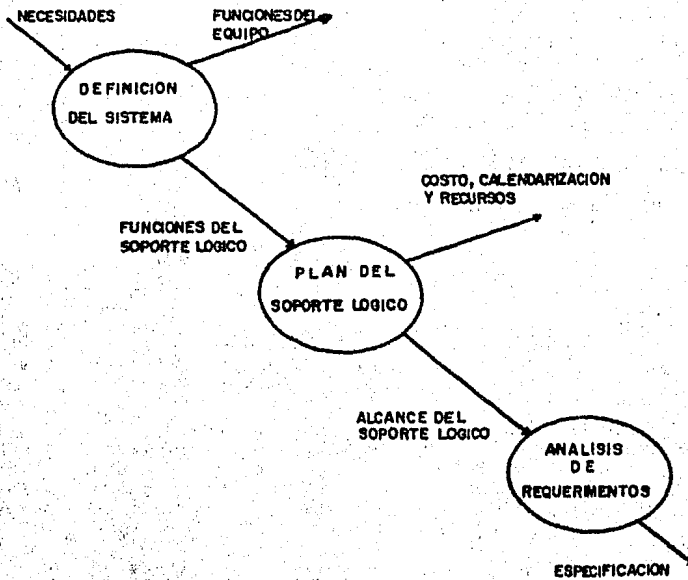
Se discutirá brevemente cada etapa de la metodología, los documentos que se producen en cada una de estas, y las revisiones que ocurren a lo largo del proceso. Cada fase descrita corresponde a una fase del ciclo de vida del soporte lógico.

El ciclo de vida es una visión a largo plazo del soporte lógico, una visión que abarca las actividades que ocurren antes de que el proceso de desarrollo comience y después de que los programas entran en uso activo.

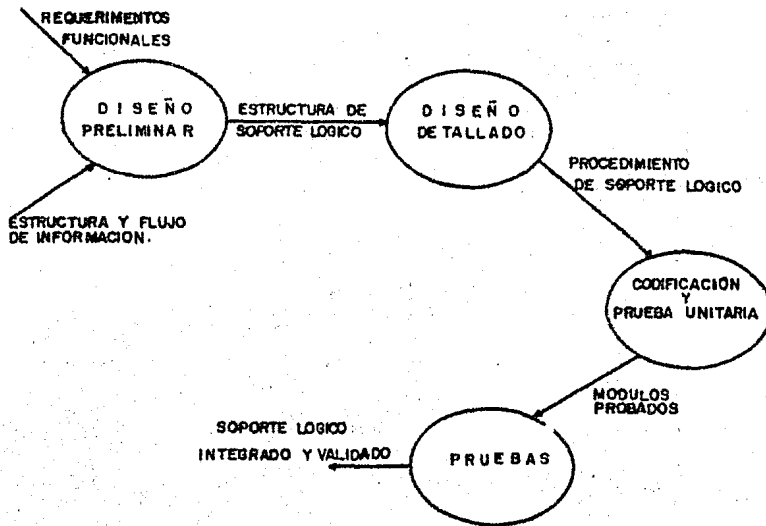
### III.1.1.- METODOLOGIA.

La metodología tiene el siguiente esquema general.

Fase de planeación



## FASE DE DESARROLLO



### III.1.1.1.- FASE DE PLANEACION.

El soporte lógico siempre es parte de un sistema mayor, basado en el uso de una computadora. Por lo tanto, la definición y el análisis del sistema deben hacerse antes de (o conjuntamente con) la planeación del soporte lógico. Las funciones del sistema deben ser asignadas al soporte lógico.

La fase de planeación del ciclo de vida del soporte lógico es un proceso de definición, análisis, especificación, estimación y revisión. El flujo de la fase de planeación se ilustra en la figura 1.

La definición del sistema es la primera etapa de la fase de planeación. En este punto de la fase de planeación la atención es enfocada en el sistema como un todo. Las funciones son asignadas al equipo, al soporte lógico, y a otros elementos del



# FASE DE PLANEACION

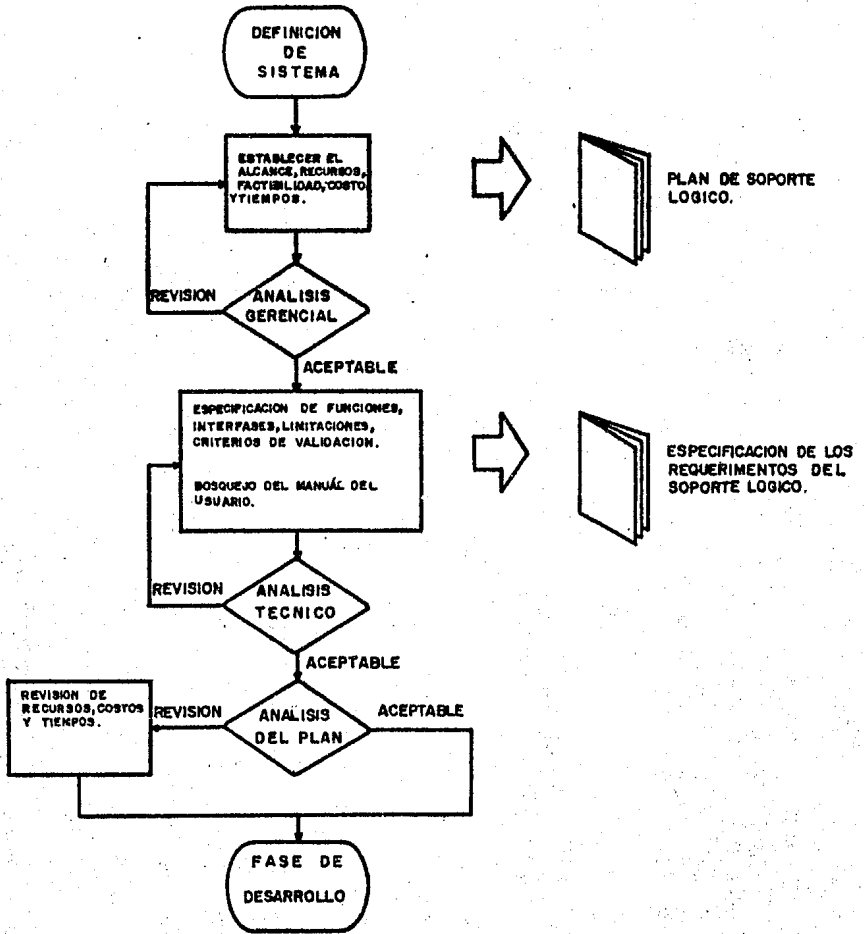


Figura 1.

sistema basados en un entendimiento preliminar de los requerimientos.

Los objetivos principales de la etapa de definición del sistema son:

1) Evaluar el concepto del sistema en los aspectos de factibilidad, costo, beneficio, necesidades, etc.

2) Describir las interfases del sistema, sus funciones y su desempeño.

3) Realizar un análisis y diseño preliminar del sistema.

4) Asignar las funciones al equipo, al soporte lógico, y a los elementos suplementarios del sistema.

5) Establecer las limitaciones de costo y tiempo.

La definición del sistema forma el fundamento para todo el trabajo subsecuente.

#### Planeación del soporte lógico.

Durante esta etapa se elabora una descripción del alcance del soporte lógico, se predicen los recursos requeridos para el desarrollo de los programas, y se establecen estimaciones en costo y tiempo. El propósito de la etapa de planeación del soporte lógico, es proporcionar una indicación preliminar de la viabilidad del proyecto en relación a las limitaciones de costo y tiempo que ya han sido establecidas. En esta etapa el administrador del proyecto elabora y revisa un documento llamado "Plan del soporte lógico".

#### Análisis y definición de los requerimientos de soporte lógico.

La siguiente etapa de la fase de planeación es el análisis y definición de los requerimientos del soporte lógico. Durante esta etapa se definen en detalle las funciones del sistema asignadas al soporte lógico. La estructura y el flujo de la información proporcionan la clave para la definición de las interfases entre los elementos del sistema y las características funcionales del soporte lógico. Los requerimientos de funcionamiento o las limitaciones en los recursos son trasladados a características de diseño del soporte lógico.

El análisis global de los elementos del soporte lógico definen criterios de validación que serán usados para demostrar que se han satisfecho los requerimientos.

La definición y el análisis de los requerimientos del soporte lógico es un esfuerzo conjunto, realizado por el analista de sistemas y la persona u organización (cliente) que solicita el sistema. Durante ésta etapa, también se elabora un documento, el cual es llamado "Especificación de requerimientos de soporte lógico".

La fase de planeación culmina con una revisión técnica de la especificación de requerimientos de soporte lógico. Esta revisión es realizada por el analista y el cliente. Una vez que ha sido establecida una definición de requerimientos aceptable, el alcance, los recursos, los costos y el tiempo identificados en el Plan de Soporte Lógico son reevaluados para su corrección. La información descubierta durante la segunda etapa puede causar impacto en las estimaciones hechas durante la primera etapa. Téngase muy presente el papel que juegan y la identificación clara entre diseñadores del sistema, usuarios y beneficiarios. Los documentos desarrollados durante la fase de planeación sirven como base para la segunda fase del proceso -la fase de desarrollo.

### III.1.1.2.- FASE DE DESARROLLO.

En la fase de desarrollo (Figura 2), se traduce un conjunto de requerimientos en un elemento operacional del sistema, al cual se le llama soporte lógico. Es muy importante realizar un diseño detallado antes de codificar cualquier porción de código.

El primer paso normalmente consiste en desarrollar una estructura modular, se definen las interfases, y se establecen las estructuras de datos. Para la evaluación de la calidad del diseño se utilizan heurísticas de diseño (pautas a seguir). Esta etapa de diseño preliminar es revisada para asegurar el cumplimiento de los requerimientos. En esta etapa se genera un bosquejo del "Documento de Diseño", el cual forma parte de la configuración del soporte lógico.

Durante la siguiente etapa de desarrollo son considerados los aspectos procedurales de cada uno de los elementos modulares del soporte lógico. Las herramientas de diseño son aplicadas para proporcionar una descripción del diseño detallado de los elementos de soporte lógico. Cada descripción procedural detallada es agregada al "Documento de Diseño" después de una revisión.

Finalmente, después de dos etapas de desarrollo (diseño), comienza la codificación, esto es, la generación de un programa, mediante el uso de un lenguaje de programación apropiado. Esta metodología ve a la codificación como una consecuencia de un buen diseño. Se revisa la claridad y el estilo de la codificación, y en el caso de encontrar alguna falla se recurre a la descripción del diseño detallado. En la etapa de codificación se genera un listado en lenguaje fuente para cada uno de los elementos modulares del soporte lógico.

Las tres etapas finales de desarrollo están asociadas con el proceso de pruebas que se le harán al soporte lógico (pruebas unitarias, pruebas de integración y pruebas de validación).

Mediante las pruebas unitarias, se intenta validar el desempeño funcional de los elementos modulares del soporte lógico.

Las pruebas de integración proporcionan un medio para ensamblar la estructura modular del soporte lógico, mientras se prueban las funciones y las interfases.

Las pruebas de validación, verifican que todos los requerimientos del soporte lógico han sido cumplidos.

Para cada una de las etapas se debe desarrollar un Plan y un Procedimiento de Pruebas. Siempre debe realizarse una revisión de la documentación de las pruebas, de los casos de prueba y de los resultados obtenidos.

## FASE DE DESARROLLO

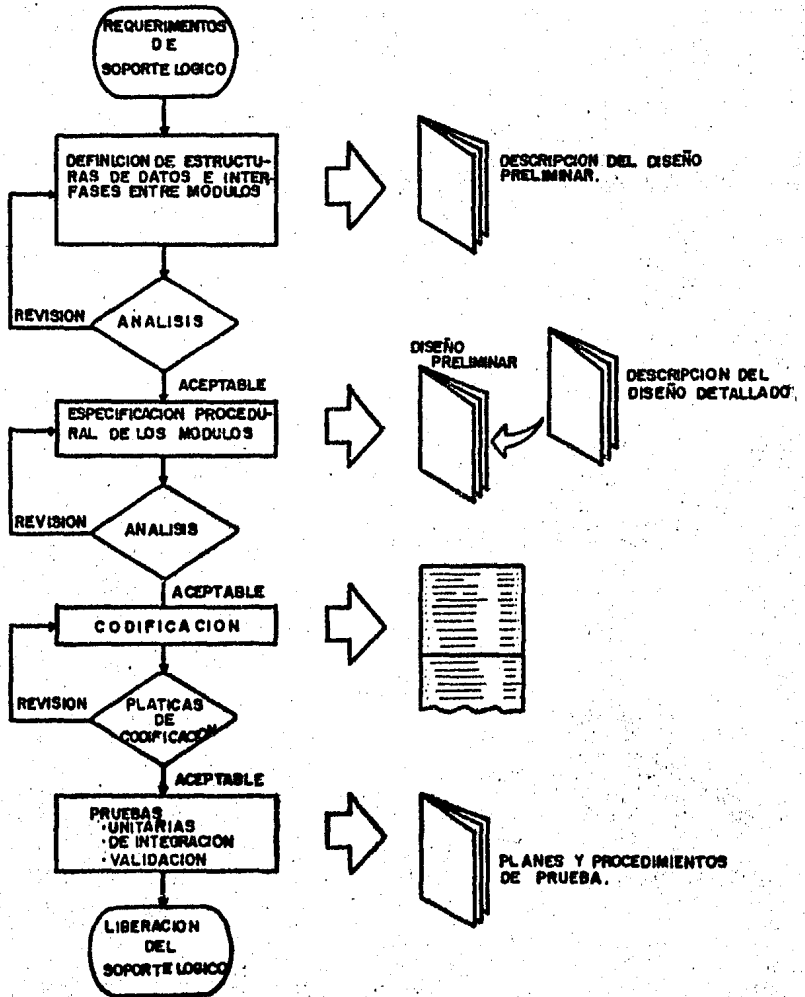


Figura 2.

### III.1.1.3.- FASE DE MANTENIMIENTO.

El soporte lógico debe ser mantenido. El reconocimiento de este hecho es el primer paso para disminuir el impacto de una tarea que consume de 40% a 70% del presupuesto de muchas organizaciones de soporte lógico.

La fase de mantenimiento (figura 3) comienza antes de la liberación del soporte lógico. Se realiza una revisión de la configuración (todos los documentos generados) del soporte lógico para asegurar que la documentación sea adecuada y este disponible para la tarea de mantenimiento que sigue. Se establece la responsabilidad del mantenimiento y se define un esquema de reportes para los errores y modificaciones del sistema.

Las tareas asociadas con el mantenimiento del soporte lógico, dependen del tipo de mantenimiento a realizar. En todos los casos, la modificación del soporte lógico incluye la configuración en su totalidad (todos los documentos desarrollados en la fase de planeación y desarrollo), no solo el soporte lógico.

A continuación se describen las cuatro actividades asociadas al mantenimiento de un programa, una vez que éste ha sido liberado para su uso.

La primera actividad de mantenimiento ocurre debido a que no es razonable asumir que las pruebas realizadas al programa descubrirán todos los errores latentes en un sistema grande, pueden ocurrir errores y ser reportados al autor del programa. El proceso que incluye el diagnóstico y la corrección de uno o más errores es llamado mantenimiento correctivo.

La segunda actividad ocurre debido al rápido cambio que es encontrado en todos los aspectos de cómputo. Con cierta periodicidad aparecen nuevas generaciones de equipo, nuevos sistemas operativos o nuevas versiones de sistemas operativos ya en uso; el equipo periférico y otros elementos del sistema son frecuentemente perfeccionados o modificados. Por otro lado la vida útil de los programas de aplicación fácilmente sobrepasa los 10 años, trascendiendo el medio ambiente para el cual fue desarrollado originalmente. Por lo tanto, el mantenimiento adaptativo, una actividad en la cual se modifica el soporte lógico para que este interactue apropiadamente con un medio ambiente cambiante, es necesario y frecuente.

La tercera actividad de mantenimiento ocurre cuando un programa o paquete es exitoso. Conforme los usuarios usan el paquete, hacen recomendaciones para adicionar nuevas rutinas, para modificar funciones ya existentes y para mejorar en forma general el paquete o programa.

La cuarta actividad de mantenimiento ocurre cuando el soporte lógico se modifica para mejorar su mantenimiento o confiabilidad o para proporcionar mejores bases para enriquecimientos futuros. Esta actividad es llamada mantenimiento preventivo, y raramente es llevada a cabo.

# FASE DE MANTENIMIENTO

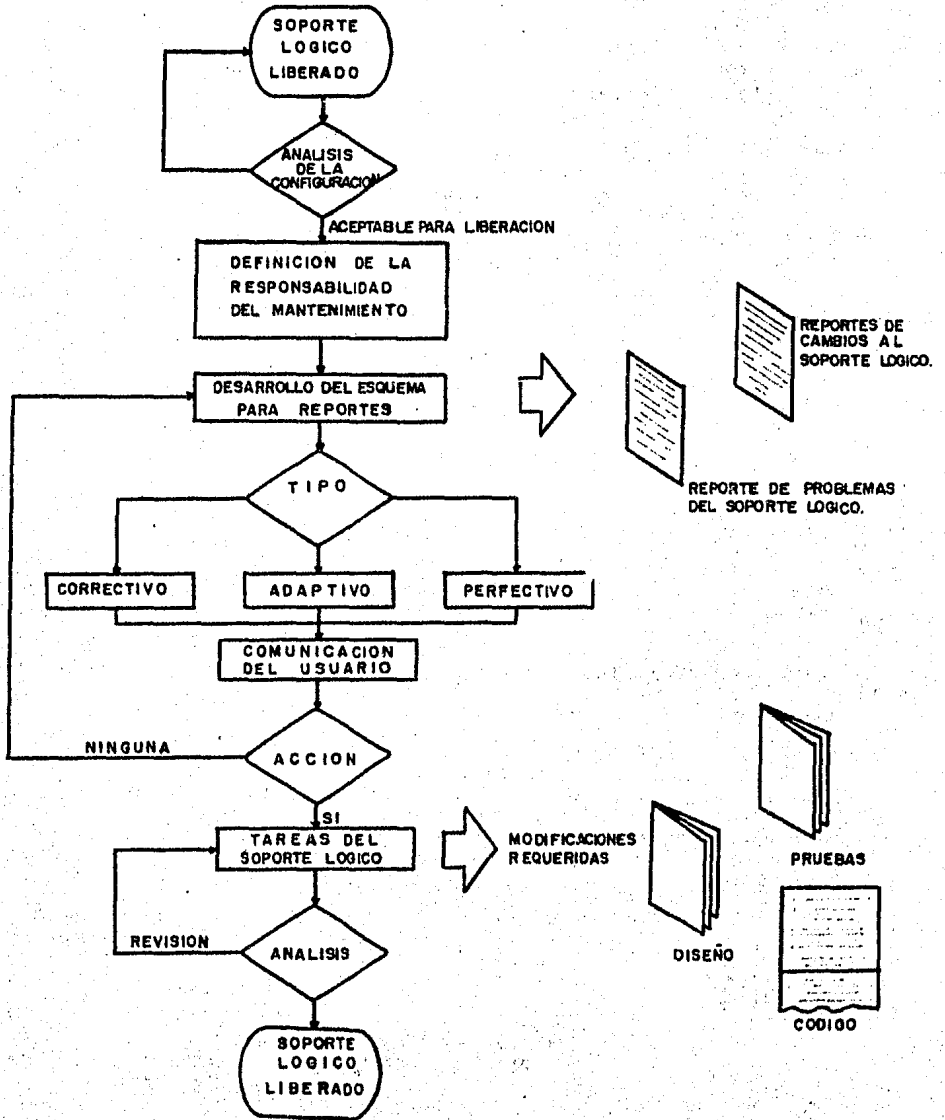


Figura 3.

### III.2.- ADQUISICION DE SOPORTE LOGICO COMERCIAL.

Cada año, el número de paquetes de soporte lógico que aparecen en el mercado se incrementa, así como también los gastos de adquisición de los mismos. Ver Apéndice B.

Existe una gran variedad de paquetes de soporte lógico. El soporte lógico de sistemas cubre cada faceta del soporte, desde utilerías hasta sistemas operativos. El soporte lógico de aplicación abarca todo el espectro de aplicaciones financieras, científicas y académicas. Mientras que algunos de estos paquetes pueden ser adquiridos en unos cuantos miles de pesos, en el otro extremo del espectro hay sistemas de bases de datos/comunicaciones de datos complejos y especializados, cuyo costo puede ser de cientos de miles de pesos.

El soporte lógico para minicomputadoras esta generando mayor interés que nunca. Esta clase de soporte lógico no refleja necesariamente precios bajos, que pueden esperarse debido al soporte físico pequeño en el cual corre. En muchos casos, el soporte lógico para una minicomputadora o para una microcomputadora es más caro que el soporte físico en sí. En el mercado de minicomputadoras, el énfasis actual es en el soporte lógico de aplicaciones, principalmente en productos de aplicación de tipo contable, diseñados para una operación efectiva en ambientes de almacenamiento limitado.

En forma clara, un número creciente de ejecutivos de procesamiento de datos se estan dando cuenta de los ahorros potenciales y los beneficios en precio / funcionamiento que pueden obtenerse mediante la adquisición de soporte lógico "previamente probado y libre de errores". También están buscando y encontrando, programas de soporte de sistemas que mejoren el funcionamiento de sus instalaciones.

Es evidente que la industria del soporte lógico, es una industria cuyos productos estan encontrando una aceptación creciente por parte de los usuarios.

#### PAQUETES DE SOPORTE LOGICO

En el pasado, siempre que se necesitó de algun sistema de información a través de una computadora, fue con el objeto de solucionar el problema de alguna aplicación o la automatización de algún proceso manual que así lo requería.

Las grandes computadoras que entonces existieron se rodearon de áreas tales como: captura, control, operaciones, y sistemas; siendo ésta última la responsable de analizar, diseñar y



desarrollar cualquier aplicación, porque estas siempre fueron realizadas con la idea de que deberían estar "hechas a la medida".

Con el tiempo este concepto fue perdiendo adeptos, debido a que era ya más costoso para las empresas desarrollar soporte lógico específico que la adquisición de los equipos de cómputo.

Así surgió la necesidad de que se desarrollaran sistemas de uso general, a los que se ha llamado genéricamente "Paquetes de Soporte Lógico". A partir de este momento comenzaron a aparecer despachos y empresas que se dedicaron a esta actividad, tratando de que los usuarios tuvieran a la mano una amplia gama de aplicaciones y que fueran ellos los que decidieran cual era la que más satisfacía sus necesidades.

Los paquetes de soporte lógico han estado presentes desde hace más de dos décadas. Sin embargo, en los primeros años, pocos de estos paquetes fueron exitosos. Debido a la gran diversidad de sus necesidades, la mayoría de los usuarios se negaban a ser forzados dentro de la rígida y limitada funcionalidad que caracterizaba a los paquetes de esta época. Aún así hubo algunos éxitos, principalmente en el área de finanzas donde las funciones de proceso estaban más estandarizadas.

En la actualidad los fabricantes de paquetes han adquirido una gran experiencia. Se ha incorporado un amplio rango de opciones y alternativas en los paquetes, con el fin de que puedan ser adaptados a una amplia variedad de necesidades.

Se han adicionado generadores de reportes y se ha aumentado su flexibilidad para que puedan satisfacer las necesidades de los usuarios, en el dinámico mundo de los negocios (fórmulas de depreciación con cambios constantes, adición de nuevos campos a los registros, preparación de nuevos reportes, etc.) se ha logrado que los paquetes funcionen en un amplio rango de ambientes de operación: sistemas operativos diferentes, diferentes familias de computadoras, etc. Los paquetes ahora son "robustos", diseñados para soportar ambientes de operación con errores y fallas en los sistemas.

No obstante que la gran diversidad en las necesidades de los usuarios siempre ha excedido la versatilidad de los paquetes, se han alcanzado progresos suficientes como para impulsar la expansión en el uso de los mismos.

El uso de paquetes de soporte lógico se está convirtiendo en un hecho común en algunas industrias. Los bancos, por ejemplo, siempre han sido relativamente receptivos a los paquetes. Los usuarios de todo tipo y tamaño están considerando cada vez más la idea de usar paquetes de soporte lógico. Una de las principales razones para ello es el incremento del proceso distribuido y la proliferación de minicomputadoras que le

acompaña. Súbitamente, se requiere de un gran número de aplicaciones basadas en minicomputadoras, y los equipos centrales de programación con frecuencia carecen de experiencia en minis. Un factor que complica la situación es la facilidad de uso requerida en los sistemas distribuidos basados en minis, los cuales deben servir a usuarios con relativamente poca experiencia en el área de procesamiento de datos; no es fácil escribir soporte lógico que desde su implementación proporcione facilidad de uso.

Al mismo tiempo, las aplicaciones de uso constante de muchos usuarios se han vuelto obsoletas. Algunos programas de contabilidad y nóminas tienen más de 20 años y se les han practicado múltiples "parches". Debido a ello, la reprogramación se vuelve esencial. Sin embargo, no existen recursos humanos suficientes en el área de procesamiento de datos. ¿Porqué no mejor usar un paquete? Las aplicaciones de personal, nóminas y contabilidad han probado ser las cartas más exitosas de los paquetes de aplicación.

Ya sea que los usuarios deseen desarrollar nuevas aplicaciones, reprogramar las antiguas, o ambas cosas, tienen problemas debido a la falta de personal de procesamiento de datos. Además las rápidas rotaciones de personal y los crecientes salarios sirven de freno a este proceso. Muchos directores en el área de procesamiento de datos, desesperados por progresar rápidamente, están considerando el uso de paquetes como una alternativa a la programación en casa. Irónicamente, esto agrava su problema, conforme el mercado de paquetes crece, los fabricantes de paquetes requieren de más programadores, y el trabajo ofrecido por ellos, frecuentemente es más atractivo que el ofrecido por el usuario.

De hecho, muchas organizaciones pequeñas que adquieren sus primeras computadoras carecen de programadores profesionales. Sin los paquetes (o sistemas que no requieren capacitación profesional para su uso), estas organizaciones no podrían considerar el uso de una computadora.

### III.2.1.- CLASIFICACION.

Existen dos tipos generales de soporte lógico (también existe un tipo de paquete de soporte lógico asociado a cada uno de ellos): el soporte lógico de sistemas y el soporte lógico de aplicaciones.

Tradicionalmente el soporte lógico de sistemas fue proporcionado por el vendedor de soporte físico. Los usuarios han demostrado estar cada vez más dispuestos a aceptar paquetes de soporte lógico de sistemas. Las compañías independientes dentro del negocio de los paquetes, son los principales productores del soporte lógico de sistemas (manejadores de

bases de datos, diccionarios de datos, ayudas para el desarrollo de programas, etc.)

Muy pocos usuarios consideran la posibilidad de desarrollar sus propios programas de sistemas, y los fabricantes de computadoras satisfacen solo una mayoría (en el mejor de los casos) de las diversas demandas de sus clientes.

La adquisición del soporte lógico de sistemas de vendedores independientes, en lugar del que el vendedor de soporte físico comercializa, generalmente está basada en una mayor capacidad, eficiencia mejorada, y menor costo del mismo. Cerca de la mitad del soporte lógico de vendedores independientes que se encuentra en uso, es soporte lógico de sistemas.

### III.2.2.- SITUACION DE MERCADO.

Los fabricantes de computadoras, acosados por presiones de mercado, han realizado varias acciones que mejoran la posición de los vendedores independientes de soporte lógico, entre estos hechos se encuentran:

1) Actualmente los fabricantes de computadoras cotizan por separado la mayoría de sus paquetes, cargo suficiente para permitir a los vendedores independientes operar provechosamente. Hasta hace poco tiempo, la mayoría de los fabricantes proveían su soporte lógico sin cargo extra.

2) Los fabricantes de computadoras han fijado los conjuntos de instrucciones y los ambientes de los programas de sistemas de sus líneas de producción. Han hecho esto para asegurar la dependencia de los usuarios a su soporte lógico. Sin embargo, los vendedores independientes también se han beneficiado. Sus paquetes ahora tienen una vida útil mayor, y son compatibles con un gran número de instalaciones.

3) Los fabricantes de computadoras han impulsado el crecimiento de los productores independientes de soporte lógico, con frecuencia comercializan activamente los productos de éstos. Muchos vendedores de sistemas pequeños recomiendan a sus clientes potenciales casas de soporte lógico o vendedores de paquetes que consideran confiables. Algunos fabricantes de computadoras pagan a las casas de soporte lógico para que reimplementen los paquetes más polupares en sus equipos.

### III.2.3.- FUENTES DE PAQUETES DE SOPORTE LOGICO.

Las principales fuentes de soporte lógico, ya sea de sistemas o de aplicaciones, son las siguientes:

1) El vendedor de soporte físico. Es la fuente tradicional del soporte lógico de sistemas. Muchos fabricantes cuentan con una gran variedad de paquetes de aplicación.

2) Casas de soporte lógico. Son empresas especializadas en el desarrollo de soporte lógico.

3) Vendedores de soporte lógico. Son compañías que comercializan el soporte lógico desarrollado por otros. La calidad de los paquetes tiende a variar considerablemente en esta fuente, debido a que el soporte lógico tiene una gran variedad de ambientes de desarrollo.

4) Grupos de usuarios. Tradicionalmente, los grupos de usuarios realizan intercambios de soporte lógico. Sin embargo, no existe un soporte adecuado para los paquetes que son intercambiados, y la documentación por lo general es inapropiada.

### III.2.4.- COMPRA DE UN PAQUETE DE SOPORTE LOGICO.

La adquisición de un paquete de soporte lógico de aplicación es una alternativa al desarrollo, por parte del usuario, de programas de aplicación. Con ellos la aplicación puede ser implantada en menor tiempo y frecuentemente a menor costo. Sin embargo, el paquete puede no cubrir exactamente las necesidades de los usuarios. Debido a su bajo costo, riesgo reducido, y al poco tiempo necesario para su implantación, los paquetes de soporte lógico deben ser considerados como una alternativa al desarrollo en casa.

Mientras que la decisión de comprar un paquete es análoga a muchas decisiones de "hacer o comprar", la adquisición de este no implica la eliminación de todos los sistemas hechos en casa, ni de todos los esfuerzos de programación, sino más bien, la sustitución de algunas tareas en el desarrollo estándar de los sistemas y en el ciclo de programación, por algún otro conjunto de tareas que en algunos aspectos son más fáciles, consumen menos tiempo y son menos caras.

Usualmente, es seguro asumir que un paquete bien probado es altamente confiable y que funcionará de acuerdo a la documentación proporcionada por el vendedor.

Una de las opciones para el usuario en búsqueda de paquetes potenciales, es consultar al vendedor de soporte físico. La búsqueda debe ser realizada usando fuentes de referencias de soporte lógico. Una fuente de referencia importante es el "ICP Software Directory", una lista detallada de los paquetes y de las compañías que los comercializan. En el Apéndice C se encuentra un directorio de las principales compañías que producen (en Estados Unidos) paquetes de soporte lógico. Otras fuentes de referencia útiles son los servicios analíticos tales

como "Datapro" y "Auerbach Software Reports" y las revistas de cómputo especializadas, las cuales publican noticias y advertencias sobre los paquetes. En el caso de los paquetes existentes en México, las referencias no son muy abundantes pero se puede considerar: Computerworld/México.

Los compradores de paquetes de soporte lógico se encuentran en el mismo predicamento que los compradores de automóviles, los paquetes vienen en diferentes formas, colores, tamaños, y precios. La selección de un paquete apropiado requiere de un curso de acción bien definido. Este procedimiento usualmente incluye:

1) El establecimiento de un proceso de adquisición de información.

2) La sumarización de la información.

3) La evaluación de paquetes alternativos contra criterios preestablecidos, y

4) La selección, recomendación eventual y aprobación.

Mientras se siguen estos pasos, el comprador de paquetes de soporte lógico puede encontrar varios problemas, tales como:

a) Las especificaciones generales de la compañía sobre el paquete pueden ser poco claras e indefinidas. Por ejemplo: ¿Debe el paquete de nóminas que está siendo evaluado incluir el almacenamiento y proceso de la información relacionada con las actividades del personal?

b) La documentación sobre las necesidades específicas de los departamentos del usuario frecuentemente no están disponibles. Los sistemas actuales son por lo regular el resultado de un diseño de sistemas por "parche". Como tales, sus capacidades son incompletas y no expresan las necesidades totales del usuario.

c) No se encuentra preestablecida una metodología para la evaluación y selección de paquetes. Esto da como resultado una revisión al azar o una recomendación basada en un proceso de evaluación defectuoso y en un criterio de selección inapropiado

d) Los enfoques de ventas de los vendedores tienden a evitar hechos difíciles, necesarios para una selección inteligente.

Muchos paquetes de soporte lógico son comprados con una evaluación y revisión mínimas. Es usual que una organización adquiera un paquete, basada en la presentación de ventas y en un chequeo rápido de las referencias sugeridas. Mientras que este proceso puede dar como resultado la elección de un paquete funcional, una revisión más sistemática puede aumentar las

posibilidades se elegir un paquete más apropiado a las necesidades de la organización.

### III.2.4.1.- EVALUACION DE PAQUETES DE SOPORTE LOGICO.

Una precondition necesaria para la evaluación de paquetes de soporte lógico es la determinación de los requerimientos. El problema a resolver debe ser definido, y las funciones básicas de proceso (entradas, proceso, salidas, etc.) deben ser especificadas, también deben identificarse los requerimientos para las interfases con el sistema operativo y con otras aplicaciones. Los requerimientos deben ser usados para pedir a los vendedores que presenten sus paquetes para someterlos a un proceso de evaluación. En base a los requerimientos, muchos de los paquetes potenciales pueden ser eliminados, dejando solo un número reducido para su evaluación detallada.

En la evaluación se puede usar un método formal, éste debe hacer uso de una serie de análisis que dejen en claro las diferencias entre los paquetes. A continuación, se da una serie de pasos útiles para llevar a cabo la evaluación. Las fuentes de información son los vendedores de paquetes, la documentación de los paquetes, usuarios actuales de los paquetes, y servicios analíticos y de clasificación. Ver Apéndice D.

1) Comparar las capacidades funcionales del paquete contra los requerimientos que han sido desarrollados.

2) Evaluar cualquier restricción de soporte físico al hacer uso del paquete. Por ejemplo, los requerimientos de memoria y equipo.

3) Evaluar los problemas de soporte lógico en que se puede incurrir al hacer uso del paquete. ¿Son adecuadas las interfases entre el paquete y el sistema operativo, compiladores, etc?

4) Evaluar las características de operación del paquete, tales como: el tiempo de respuesta, facilidad de instalación, facilidad de operación. ¿Qué tan buenas son las instrucciones de operación?

5) Evaluar las características de uso, tales como los requerimientos de entrada, formatos de reportes, controles, auditorias, detección y manejo de errores, y formas especiales. ¿Qué tan buenas son las instrucciones de usuario?

6) Evaluar el mantenimiento del paquete. ¿Qué tan buena es la documentación en cuanto a su mantenimiento? ¿Está escrito en una forma estructurada que facilita su mantenimiento? ¿Está codificado en un lenguaje estándar? ¿Quién dará mantenimiento al paquete?

7) Evaluar la flexibilidad del paquete para cumplir con necesidades cambiantes y con un ritmo de crecimiento adecuado.

Los usuarios de paquetes reportan diferencias considerables en la calidad, especialmente reflejada en la documentación para los usuarios, operadores, y programadores de mantenimiento. Por lo tanto, la investigación de las características de los paquetes es esencial para una adquisición y uso exitosos.

Los paquetes que parezcan ser viables en la investigación precedente, deberán ser demostrados en alguna forma. Quizá el mejor enfoque sea usar pruebas de referencia en nuestras instalaciones. Algunos paquetes no son lo suficientemente grandes para justificar los gastos de éste método y podrían ser demostrados en otra instalación. o podría usarse como método final de evaluación un período de prueba de los paquetes más viables.

Las negociaciones de contratación de los paquetes son muy importantes. Estos deben incluir puntos tales como: el período de garantía, privilegios de devolución del paquete si éste no es satisfactorio, penalidades por retraso en su liberación, soporte del vendedor en la instalación, acuerdos sobre modificaciones, garantías en caso de quiebra de la empresa, y mejoras en los estándares que el paquete debe cumplir.

#### III.2.4.2.- DESARROLLO DE UN SISTEMA DE PUNTUACION.

A continuación se propone un método para la evaluación de diferentes alternativas en paquetes de soporte lógico. Este método esta basado en los puntos analizados anteriormente y se resumen en la Tabla I.

TABLA			
TAREA		SUBTAREAS	
I1) Establecimiento de un equipo de trabajo para la selección de paquetes	I Es necesario tanto para el grupo de usuarios, como para el personal de procesamiento estar totalmente comprometidos con el proyecto	I*Elegir a los miembros del equipo de trabajo I*Definir las responsabilidades de cada miembro del equipo I*Desarrollar un plan de trabajo detallado	I I I I I I I
I2) Desarrollar un documento de los requerimientos de las aplicaciones (DRA)	I El DRA proporciona los requerimientos y estándares para la evaluación de los paquetes	I*Listar los objetivos del DRA I*El formato del DRA incluye: I Entradas I Procesamiento I Salidas I Requerimientos técnicos del sistema de procesamiento de datos I Interfases con otros sistemas	I I I I I I I I I I I I I
I3) Obtener información de los paquetes a evaluar	I Explorar todos los paquetes alternativos	I*Solicitar al vendedor sus propuestas I*Preparar una carta de características de usuario del paquete I*Preparar una carta de características técnicas de paquete	I I I I I I I I

Las cartas de sumarización propuestas en el punto 3 de la Tabla I se describen a continuación.



CARTA 1					
SUMARIO DE LAS FACILIDADES AL USUARIO DEL PAQUETE					
P A Q U E T E					
CARACTERISTICA	"A"	"B"	"C"		
Implicación del usuario en la adaptación del sistema a las necesidades de la organización	Máxima	Adecuada	Mínima		
Formas de entrada	Flexibilidad completa	Flexibilidad adecuada	Poca flexibilidad		
Documentación de usuario	Responsabilidad del cliente	Extensiva con orientación técnica	Orientación técnica		
Cambio dinámico en las prioridades de las transacciones	Parcial	No	Si		
Seguridad	Posible	Muy improbable	Muy posible		
Generador de Reportes	Orientado al usuario	Orientación técnica	Muy orientado al usuario		
Almacenamiento de datos del personal	Limitaciones mínimas	Potencialmente limitado	Ilimitado		
Número de usuarios	65	200	300		
Otras aplicaciones	Financieras y de personal	De personal	Financieras		
Localización del soporte	Guadalajara	D.F.	D.F.		
Precio base	\$ 27,000	\$ 37,000	\$ 29,500		

Continuación

ICuota de Imantenimiento	I \$ 2,700 I	I \$ 3,000 I	I \$ 2,300 I	I I
ICapacitación en Ila instalación	I 11 días I	I 15 días I	I 13 días I	I I
ITipo de Icapacitación I I	I Desde I requerimientos I hasta I documentación	I Capacitación a I usuarios I I	I Desde I planeación I hasta imple- I mentación	I I I I
IControles para Iproceso en lote	I Si I	I No I	I Si I	I I
IProcesamiento de Ierrores	I Identificación I de campos	I Identificación I de campos	I Localizacio- I nes fijas	I I
IControl de Ientrada	I Mínimo I	I Entrada reci- I clable	I Ninguna I	I I
IFormatos de Isalida	I Flexible I	I Rígido I	I Rígido I	I I
IReportes de Isalida I I	I No estándar I Flexibilidad I completa I	I Estándar, más I seis estándares I definidos por I los usuarios	I Estándar I I I	I I I I
IAdministración Ide proyectos Iproporcionada Ipor el vendedor	I Parcial I I I	I Parcial I I I	I Mínimo I I I	I I I I

CARTA 2				
SUMARIO DE LAS CARACTERISTICAS TECNICAS DEL PAQUETE				
P A Q U E T E				
U S U A R I O	"A"	"B"	"C"	
Memoria (max)	60-90 K	120 K	86 K	
Tamaño del registro	Número y longitud variables	Número variable de longitud fija	Número variable de longitud variable	
Número de archivos de reporte	2	5	3	
Lenguaje	COBOL y Ensamblador	COBOL	COBOL	
Control de sistema	Manejo de parámetros	Manejo de parámetros y tablas	Manejo de tablas	
Método de acceso	Secuencial o Directo	Secuencial o Directo	Secuencial	
Documentación de sistema	Extensiva	Extensiva	No proporcionada en la opción original	
Intervención del operador	Ninguna	Mínima	Mínima	
Diseño modular	Si	No	Si	
Requerimientos de conversión	Si	Si	No	
Capacidad de edición	Extensiva	Limitada	Limitada	

#### CARTA 2

La preparación de éstas cartas de sumarización es una forma de clasificar toda la información obtenida, y eliminar posibles confusiones en cuanto a las capacidades y restricciones de cada paquete. Una vez que la información ha sido recolectada y clasificada, el proceso de evaluación puede

comenzar. Este proceso consiste en el cómputo de puntos ponderados relativos, registrados para cada paquete. El cómputo de puntos ponderados proporciona un método objetivo para la comparación de paquetes, basado en las necesidades únicas de la organización. La puntuación final por paquete no necesita ser el único factor usado para la selección. Dos paquetes con puntuación relativamente iguales (dentro de pocos puntos porcentuales el uno del otro) pueden necesitar un escrutinio adicional antes de la selección final.

Este enfoque define una metodología de evaluación objetiva, eliminando los sesgos subjetivos creados por la persona que vende el paquete, y guía el proceso de evaluación de acuerdo a las prioridades y objetivos de la organización.

En resumen, los paquetes de soporte lógico proporcionan una opción al desarrollo y programación de sistemas hechos en casa. Sin embargo, se requiere cuidado en la revisión y selección del paquete apropiado. Utilizando un proceso de evaluación y revisión formal, se minimizará la posibilidad de elegir un paquete inapropiado para las necesidades de la organización.

#### III.2.4.3.- VENTAJAS Y DESVENTAJAS DE LOS PAQUETES DE SOPORTE LOGICO.

En términos generales, un paquete presenta las ventajas y desventajas enumeradas a continuación:

##### VENTAJAS

- 1) Para qué reinventar lo que ya existe ?
- 2) Costos menores a los que se obtendrían si el sistema es desarrollado en casa.
- 3) Disminución del personal de procesamiento de datos.
- 4) Uso "inmediato" del paquete.
- 5) Mejoramiento continuo.
- 6) Ventajas de oportunidad.
- 7) Liberación del personal para otros proyectos.
- 8) Altamente confiable y funciona de acuerdo a lo establecido en su documentación.
- 9) Se asegura que el sistema este documentado.
- 10) Minimización de los riesgos y esfuerzos de programación asociados con los sistemas a gran escala.

- 11) Amplio rango de capacidades.
- 12) Grupo de usuarios.

#### DESVENTAJAS

- 1) Puede o no cumplir con todos los requerimientos del usuario.
- 2) Las modificaciones extensivas al paquete, por parte del usuario, usualmente se traducen en la pérdida del soporte por parte del vendedor.
- 3) Se puede requerir capacidades adicionales de soporte físico.
- 4) Puede requerir adaptaciones.
- 5) Su mantenimiento tendrá un costo.
- 6) Siempre se dependerá del proveedor.
- 7) No dejan experiencia en "casa".

#### III.2.5.- PERSPECTIVAS.

Una vez aprendido lo que se tiene que hacer para producir paquetes aceptables, los fabricantes diseñarán sus futuros paquetes de acuerdo a ello; conforme se adentren en nuevas áreas (control de manufacturas, diseño asistido por computadora, procesamiento gráfico, etc.) producirán paquetes que tengan mayores oportunidades de cumplir con los requerimientos necesarios desde su liberación (como ningún otro, el fabricante de paquetes ha aprendido el precio de desarrollar paquetes exitosos - años y millones de pesos en muchos casos). El resultado deberá ser una aceptación más rápida de los nuevos paquetes y un mayor crecimiento del mercado.

Siguen apareciendo nuevas oportunidades para el soporte lógico de sistemas, y ésta industria seguirá creciendo conforme se tome ventajas de ellas. Una de las nuevas oportunidades es para los traductores de protocolos que permiten la interconexión de sistemas de computadoras y de complejas terminales de diferentes vendedores. (éstas son implementadas generalmente como combinaciones de controladores, microprocesadores dedicados y soporte lógico). Otra oportunidad es para los manejadores de canales de alta velocidad.

Quizá la oportunidad mas exitante para el soporte lógico de sistemas, la constituyan los paquetes que permiten a los usuarios, con relativamente poco conocimiento en el área de procesamiento de datos, desarrollar sus propias aplicaciones interactivas. Tales paquetes hacen a los sistemas complejos fáciles de usar, haciendo disponible un subconjunto de las funciones del soporte lógico central, por medio de diálogos dirigidos, menús etc., IBM llama a estos sistemas "generadores de aplicaciones" . Todos los grandes fabricantes de computadoras y algunas firmas independientes de soporte lógico lo estan comercializando. Estos sistemas tienen un mercado potencial muy amplio, pero en la actualidad sólo se ha cubierto una mínima parte.

### III.2.6.- UN PAR DE INQUIETUDES.

Existen algunas inquietudes respecto al tema que se está tratando. Como se hizo notar con anterioridad, se esta poniendo en disponibilidad soporte lógico, que hace a los sistemas de cómputo fáciles de usar. Los éxitos iniciales de este tipo de soporte lógico, son relevantes en todos los frentes. Los fabricantes de computadoras están emprendiendo entusiastamente nuevos desarrollos. Seguramente tal soporte lógico será mas versátil y atractivo. La instrucción en el área de cómputo también se verá incrementada conforme más y más personas tomen conciencia de su habilidad para interactuar con las computadoras. Como resultado, cada vez más usuarios finales podrán hacer sus aplicaciones en unas cuantas horas. Significa esto, que el mercado para paquetes de aplicación (e incidentalmente, también el mercado de trabajo para los programadores profesionales) puede comenzar a declinar ?

Otra inquietud la constituye un posible reflorecimiento del soporte lógico y del soporte físico. Todas las computadoras modernas contienen múltiples microprocesadores dedicados a subfunciones específicas del sistema. Los microprogramas de estas subfunciones son capaces de realizar algunos de los trabajos que anteriormente hacia el soporte lógico de sistemas, en áreas tales como: comunicación y manejo de archivos en disco. El soporte lógico de sistemas convencional, está siendo desplazado por los microprogramas de los microprocesadores que en forma creciente constituyen a las computadoras modernas. Estos microprogramas son proporcionados como parte del soporte físico por los fabricantes, por un precio combinado. A los usuarios no se les proporciona información sobre los microprogramas, ni ellos la solicitan; pocos usuarios tienen la capacidad o el deseo de modificar los microprogramas implementados en sus computadoras, cualquier vendedor independiente de soporte lógico que quisiera hacer tales modificaciones al microcódigo, tendrá dificultades para obtener la información técnica y el conocimiento que estos cambios implican; además podría impedirserlos la ley del secreto industrial o cualquier otra ley relacionada.

El microcódigo esta invadiendo también al soporte lógico de aplicaciones. Hasta ahora los paquetes de aplicación de microcódigo han sido confinados a computadoras personales y terminales orientadas a ciertos procesos, pero muchos indicadores hacen pronosticar que los "chips de aplicación" entrarán en uso general en el futuro, como ya lo hacen claro en los videojuegos.

Más cerca está la posibilidad de que el microcódigo orientado a aplicaciones sea sustituido por el microcódigo de propósito general, que está siendo liberado como parte de muchas computadoras. Hoy el microcódigo es usado para emular los conjuntos de instrucciones de máquinas antiguas, por ejemplo la IBM 360/370, de manera que los paquetes de aplicación que antes corrían en esas máquinas corren en las computadoras modernas. Si los fabricantes de computadoras descubren combinaciones de soporte lógico y microcódigo que realizan mejor una amplia variedad de aplicaciones (el procesamiento de transacciones financieras, diseño asistido por computadora, etc.), estas serán ofrecidas como sustitutos. Podrán los vendedores independientes de soporte lógico competir con ellos, dado el gran conocimiento técnico necesario, la falta de información, y (quizá) los impedimentos legales ?

Desde el punto de vista del usuario, todo esto no importa. Si los fabricantes de computadoras encuentran formas de proporcionar mejores sistemas, insertando soporte lógico dentro del soporte físico a un precio reducido, el usuario se beneficiará. Si esto no es posible, y la industria de los paquetes de soporte lógico continúa evolucionando, con los vendedores independientes mejorando por un lado, y los fabricantes de computadoras por el otro, el usuario también resultará beneficiado. Sin embargo, son los vendedores independientes los que tienen mucho que perder.

También la idea de que los usuarios finales escribirán todos los programas de la organización en su tiempo libre, es utópico. Qué hay sobre las grandes aplicaciones de soporte, que aún constituyen la mayor parte del trabajo: las nóminas, los sistemas de control financiero, los sistemas de facturas, etc. ?Podrá el usuario final entender las dificultades de los requerimientos de reportes, auditorías, etc., que tales sistemas implican ?

De cualquier forma esto pertenece al futuro. Los próximos años, al menos parece ser una época de auge para los paquetes de soporte lógico.

### III.2.7.- ADAPTACION.

Un factor importante que debe considerarse en el proceso de compra de un paquete de soporte lógico es, si el producto necesita modificaciones para satisfacer las necesidades específicas del usuario. Estas modificaciones pueden ser hechas por el vendedor, el usuario mismo o ambos.

Generalmente, los paquetes de aplicación necesitan más modificaciones que los paquetes de sistemas.

### III.3.- CONTRATACION DEL DESARROLLO DEL SOPORTE LOGICO.

Si después de hacer un análisis de los paquetes existentes en el mercado nos percatamos de que ninguno satisface nuestros requerimientos, además de tener un cierto tiempo tratando de desarrollar todo el soporte lógico que necesitamos, podemos llegar a la conclusión de que la contratación de desarrollo de soporte lógico puede ser la única forma de cubrir nuestro déficit actual de recursos. Esta opción también mantiene bajos nuestros niveles de personal.

Existen algunos elementos específicos que se deben considerar cuando se contrata el desarrollo de soporte lógico, estos elementos son:

1.- Se debe buscar una empresa confiable, que tenga varios años dentro del negocio y una reputación bien cimentada.

2.- La firma de soporte lógico que se elija debe estar obligada por contrato a producir lo que nosotros consideramos como soporte lógico de calidad.

Cuando contratamos el desarrollo del soporte lógico, estamos expuestos a una serie de riesgos, entre los que se encuentran:

a) Si surge una disputa durante el desarrollo, se pueden perder todos los pagos que se han hecho hasta la fecha, y no tendremos nada a cambio de lo que hemos pagado. Una forma de solucionar este problema es especificar los productos que puedan ser intercambiados por los pagos en cada etapa del desarrollo. Estos productos incluyen requerimientos, diseño, codificación, documentación de todo tipo y programas probados. Todo esto deberá ser especificado en el contrato.

b) Los retrasos en el calendario de desarrollo son legendarios. Estos retrasos afectan la liberación del soporte lógico y el uso final del producto. Se deben imponer penalidades por retraso al vendedor, aunque éstas causan que el vendedor libere productos no terminados y de una mala calidad.



Pueden proporcionarse bonificaciones para una liberación antes de la fecha estipulada, pero esto puede causar que se "inflen" los calendarios iniciales.

c) El vendedor puede usar nuestros recursos, gente y equipo para apoyarse en el desarrollo. El uso de estos recursos deben estar estipulado en el contrato y debe tenerse bajo control.

d) La calidad del producto final puede no cubrir nuestras necesidades. Como se estableció anteriormente, la calidad debe ser especificada tan claramente como sea posible. Puede requerirse una revisión y aceptación de los productos en cada fase.

e) El vendedor, debido a su trabajo de desarrollo, puede construir un sistema que nos proporcione un margen competitivo. Debe prohibirse al vendedor que comercialice nuestro sistema o la construcción de un sistema similar durante un período específico. Por ejemplo, un sistema de control de inventarios que puede reducir nuestras existencias en un 30%, sin afectar las ventas disminuirá los costos y nos permitirá bajar los precios, proporcionándonos una ventaja competitiva. El vendedor podría proporcionar nuestro sistema a los competidores por un precio mucho mas bajo que el original, debido a que no hay costos de desarrollo. Podría permitirse que el vendedor comercialice el producto, estipulando un pago de derechos, derivado de la venta del mismo.

La contratación del desarrollo del soporte lógico puede ser riesgosa, pero proporciona una alternativa a la compra o desarrollo por nosotros mismos del soporte lógico. Esta opción proporciona medios altamente productivos y oportunos para implantar sistemas de soporte lógico que de otra forma serían imposibles debido a las limitaciones de recursos.

**CAPITULO IV**

**ASPECTOS LEGALES DEL SOPORTE LOGICO**

#### IV.- ASPECTOS LEGALES DEL SOPORTE LOGICO.

En las secciones siguientes se hace referencia al estado que guarda la situación del soporte lógico dentro del marco legal en México. Primeramente se hace un análisis de los requisitos para contratar licencias de uso de soporte lógico. Posteriormente, se hace un análisis de las condiciones que imperan con respecto a la protección legal del soporte lógico ante las Leyes Mexicanas.

##### IV.1.- CONTRATACION DE LICENCIAS DE USO.

El Estado Mexicano al reconocer la importancia de la informática dentro del marco económico y del desarrollo tecnológico del país, ha estructurado políticas y mecanismos tendientes al control y desarrollo de esta industria en nuestro país.

La Secretaría de Comercio y Fomento Industrial tiene implantado el Programa de Manufactura de Equipos de Cómputo, sus Módulos Principales y Equipos Periféricos. Dicho Programa tiene por objeto fomentar la manufactura de dichos equipos en México, pretende además proteger e impulsar a los empresarios mexicanos que se aventuren en la fabricación de equipos, en el rango de microcomputadoras.

Este programa de manufacturas es el pilar que sostiene y estructura la política del gobierno federal en cuanto al soporte físico concierne. Debido a la importancia, cada vez más notoria del soporte lógico, resulta indispensable, para implantar una política global para la industria informática mexicana, incluir el soporte lógico dentro de los mecanismos de control legal para su reglamentación, y de ahí establecer políticas de desarrollo y fomento nacional.

Es por lo anterior que los contratos para la comercialización del soporte lógico quedaron sujetos a registro y aprobación.

El medio legal que para este propósito se encontró, fue la Ley de Transferencia de Tecnología. Sin embargo, en los contratos que permiten el uso de programas de computadora no existe generalmente ningún traspaso tecnológico, puesto que ni es el propósito del contrato, ni la intención de las partes el que nos enseñe a diseñar o a generar el programa respectivo, sino solamente que se nos permita el uso del mismo. Lo anterior ha generado incongruencias en la aplicación de la Ley, toda vez que la misma fue diseñada para regular el verdadero traspaso tecnológico, es decir, la adquisición de conocimientos, a cambio de los cuales aprendemos a hacer o a fabricar algo, y por lo cual pagamos un precio como contraprestación correspondiente.

Al imponerse la obligación de solicitar la aprobación e inscripción de los contratos por los que se comercializa el soporte lógico, se encontró que el Registro de Transferencia de Tecnología podría ser un instrumento eficaz para controlar su desarrollo en el país en todas sus facetas, esto es, su importación, fomentos de desarrollo nacionales, control de flujo de divisas al exterior, etc.

Es evidente que una política sana indica que se debe evitar la salida de divisas innecesarias y, por otra parte, proteger a los fabricantes mexicanos, e inclusive estimularlos, para avanzar en la realización de programas que cada vez cuenten con mayor contenido tecnológico y que vengan a abastecer las necesidades del mercado nacional.

La herramienta para la implantación de estas políticas se encuentra ahora en la Ley de Transferencia de Tecnología. Las facultades reglamentarias, en cuanto a la determinación de las políticas indicadas, se encuentran consignadas, en el artículo 16, fracciones I y II, mismas que han sido asentadas en los artículos 60, 61, 62, 63 y 64 del Reglamento de dicha Ley.

Ahora nos referiremos, en primer término a la disponibilidad del soporte lógico "local". Por ejemplo pensemos que se nos ocurre adquirir de la empresa "TecSoft Company" de los Estados Unidos, una nómina para nuestra empresa. Dejando a un lado de momento cuantos dólares cuesta. El contrato de licencia respectivo, debe ser inscrito, previa su aprobación, ante el Registro Nacional de Transferencia de Tecnología. Pues bien, el Registro tiene facultades legales para rechazar el contrato si se prueba que existen programas de aplicación, como la nómina de referencia, disponibles en el país, iguales o mejores quizá a la que pretendemos importar del extranjero.

Desde luego debemos recalcar que el criterio de disponibilidad debe probarse o justificarse adecuadamente ante el Registro. Es decir, ante el Registro pueden existir antecedentes de nóminas que se comercializan en el país, por empresas nacionales, y por lo tanto considerar la improcedencia de la pretendida importación, por tratarse de tecnología "disponible" en el país y que además satisfaga las necesidades del usuario.

Sin embargo, puede ser que se trate de un producto altamente especializado, y entonces se convenza al Registro de la necesidad de su importación.

Por lo tanto, una primera política es impedir la importación de soporte lógico que notoriamente pueda ser adquirido en México. Esta política, tiende dualmente a fortalecer a las empresas mexicanas, así como a evitar la fuga innecesaria de divisas. Sin embargo, el alcance de esta política no debe

sobrepasar la realidad nacional, tanto cuantitativamente como cualitativamente. Esto es, que el soporte lógico disponible en México normalmente sea consistentemente similar al que se pretende importar. En el proceso de decisión correspondiente se tiene que ser causístico, además de basarse en datos técnicos convincentes, plenamente soportados.

Se debe reconocer que en México existen técnicos calificados y desarrollos altamente complejos. El problema es su producción, comercialización y soporte a escala comercial, tres fases concatenadas, inexorablemente, para considerar la liberación comercial de un programa conforme a los estándares de la industria y con plenas garantías para el usuario.

Finalmente, en cuanto al precio del soporte lógico, en principio no se han sentido restricciones o rechazos en los precios de productos comercializados por empresas establecidas en el país. Lo anterior incluye subsidiarias de empresas extranjeras que operan en México. Así pues, tratándose de soporte lógico de importación, deberá justificarse el precio que se debe pagar, dentro del contexto de las políticas ya expresadas. En otros términos, cuánto costaría, en su lugar, hacerlo en México; adicionando lo anterior con factores, como el tiempo requerido para hacerlo localmente, su factibilidad, necesidades específicas, etc. El fundamento legal para cuestionar el importe del precio del soporte lógico de importación, se encuentra en la disposición contenida en la fracción II del artículo 16 de la Ley de Transferencia de Tecnología, que ordena que no se aprobarán los contratos respectivos "cuando la contraprestación no guarde relación con la tecnología adquirida o constituya un gravamen injustificado o excesivo para la economía nacional o para la empresa adquiriente". La reglamentación específica se encuentra consignada en los artículos 63 y 64 del reglamento de la Ley de la Materia.

Así pues, en la decisión de contratar soporte lógico de importación, el usuario deberá considerar los elementos ya expresados, a saber: la no disponibilidad local y que el precio no resulte excesivo en función del costo de su realización en México.

#### IV.2.- TRAMITES ANTE EL REGISTRO NACIONAL DE TRANSFERENCIA DE TECNOLOGIA PARA LA INSCRIPCION DE CONTRATOS DE SOPORTE LOGICO.

Los trámites de inscripción de los contratos de uso de soporte lógico, son relativamente sencillos. En efecto, en primer término debe llenarse una solicitud que acompaña a un cuestionario (ver apéndice F). Ambos documentos son formularios oficiales, emitidos por el Registro Nacional de Transferencia de Tecnología.

El trámite de referencia puede ser iniciado por cualquiera de las partes. Esto es, por el licenciante o bien por el usuario. Debe notarse que la obligación recae en ambos, aunque las sanciones más graves, las de carácter oficial, afectan más adversamente al usuario. Los datos que deben proporcionarse, tanto en el cuestionario como en la solicitud de referencia, son relativamente fáciles de llenar, por lo que en general, esta parte inicial no debe crear problemas al solicitante.

A dichos documentos oficiales acompaña un ejemplar del contrato con firmas originales. La personalidad se acredita en términos de la legislación común, esto es, con poder notarial. Una vez que se tienen listos estos documentos, se procede al pago de derechos que en la actualidad importan la suma de \$25,000.00 M.N. por la licencia inicial. Después de pagarse tales derechos, la documentación indicada se presenta ante la Oficialía de Partes del Registro. A partir de este momento, es cuando verdaderamente se inicia el trámite administrativo que nos ocupa.

Con toda esta documentación se forma e integra un expediente que primeramente es turnado al Departamento de Evaluación Jurídica. En esta fase se revisa que el contrato respectivo no contenga cláusulas que infrinjan las disposiciones de la Ley de la materia y su reglamento. Si el contrato contiene alguna o algunas cláusulas violatorias, entonces es rechazado y por lo tanto la resolución de negatoria es comunicada al peticionario, mediante un oficio que para tal efecto se expide y se le envía por correo certificado. Si por el contrario el contrato pasa exitosamente el análisis legal, se torna al departamento de Evaluación Económica. Es aquí donde muchas veces se reciben sorpresas desagradables, pues entran en juego elementos de "política económica", muchas veces ignorados por el solicitante.

Como se expresó con anterioridad, si el precio, o el tipo de programa o cualquier otra circunstancia es encontrada violatoria a la Ley, o a las políticas que en la misma se originan, el contrato es rechazado, conforme al procedimiento antes mencionado. Pero si el contrato pasa en forma satisfactoria el "examen económico", entonces es aprobado en forma definitiva y queda inscrito en el Registro Nacional de Transferencia de Tecnología, lo que igualmente se comunica en forma oficial al solicitante.

#### IV.3.- PROTECCION LEGAL DEL SOPORTE LOGICO EN MEXICO.

Dentro de los temas jurídicos que guardan relación con el mundo de la informática se presenta el problema de la protección legal del soporte lógico en México. Concretamente en referencia a la protección jurídica tradicional derivada del Derecho de Autor.

A continuación se proponen algunos antecedentes y consideraciones generales sobre la situación de la protección legal de los programas de cómputo en México.

A todos nos parece natural que el autor de una obra sienta un derecho de propiedad personal sobre la misma y que ese derecho sea oponible a todos los demás, es decir, que todo el mundo lo respete y que en caso de infracción, el titular del derecho pueda acudir a los tribunales para perseguir y sancionar al infractor.

El autor de un programa de cómputo, léase un programador, o bien, una empresa comercializadora de soporte lógico, tienen el derecho a proteger su obra y sus inversiones.

Antes de tratar los diferentes sistemas jurídicos que pueden proteger al soporte lógico de la "piratería", debemos precisar la naturaleza jurídica del soporte lógico.

En primer término es incuestionable que se trata de una creación del intelecto humano. Por lo tanto se trata de un bien intangible e incorpóreo, a pesar de que se contenga en un bien corpóreo y tangible, que es el medio magnético que lo contiene.

Así pues, una vez ubicados en el mundo de las obras intelectuales, el derecho los protege, ya sea a través del derecho de la propiedad industrial o por conducto del Derecho de Autor.

Por su parte, la propiedad industrial tiende a proteger aquellas ideas o conceptos que constituyen inventos a los que se les otorga el privilegio de la "patente", por el que se confiere al inventor el derecho exclusivo sobre la propiedad de su invento, para explotarlo por sí mismo o bien autorizar a terceros la explotación correspondiente. En el mundo de las patentes, las creaciones intelectuales deben producir una obra tendiente a un producto o desarrollo de carácter industrial y por ende las obras patentables son siempre corpóreas y tangibles. Es decir, las podemos conocer a través de nuestros sentidos, como los aparatos y las máquinas que se pueden ver y tocar.

El estado premia y estimula a los inventores, no solo reconociendo su derecho de propiedad, sino además concediéndoles un privilegio exclusivo y temporal para explotar su invento o para permitir a terceros la explotación correspondiente.

De esta manera, tratándose de bienes tangibles y corpóreos, la propiedad industrial concede derechos de protección exclusiva en las diversas categorías de patentes, marcas, nombres comerciales y avisos publicitarios.

El soporte lógico es intangible e incorpóreo. Por exclusión se ha considerado que debe ser protegido por el Derecho de Autor. El Derecho de Autor tradicionalmente ha venido protegiendo las obras del intelecto humano, pero específicamente aquéllas que por su naturaleza son intangibles e incorpóreas, tales como las obras artísticas y literarias, así como las representaciones artísticas de los intérpretes, traductores, adaptadores, etc. El Derecho de Autor se reconoce automáticamente, sin mayor requisito, y el registro de la obra es un acto administrativo, tendiente a formalizar el derecho autoral que la Ley y el Estado reconocen.

El Derecho de Autor, concede al autor privilegios exclusivos, consistentes, en primer término, al reconocimiento de la calidad, y segundo, a explotar por sí o por terceros la obra de que se trate, en forma exclusiva. Por tanto existe el derecho de perseguir ante los tribunales a los infractores del Derecho de Autor, ésto es, a los "piratas", a quienes inclusive se les puede imputar responsabilidad penal.

El trámite de registro es también un medio de prueba de la legítima titularidad sobre una obra determinada. Pero también las obras técnicas y científicas, de carácter intangible e incorpóreo, son protegidas por el Derecho de Autor.

A partir del 29 de Septiembre de 1984 se otorgaron los privilegios que concede la Ley del Derecho del Autor a los programas de cómputo según un acuerdo publicado en el Diario Oficial de la Federación el 28 de Septiembre de 1984 y el cual se reproduce a continuación:

"Acuerdo No. 114, por el que se dispone que los programas de computación podrán inscribirse en el Registro Público del Derecho de Autor ...

JESUS REYES HEROLES, Secretario de Educación Pública, con fundamento en lo dispuesto por los artículos 5o., fracciones VI, VII y XI de la Ley Federal de Educación; 7o., 119, fracción I.122 y demás aplicables a la Ley Federal de Derechos de Autor; 38, fracción XII, de la Ley Orgánica de la Administración Pública Federal, y 5o. del Reglamento Interior de esta Secretaría, y

#### CONSIDERANDO

Que los programas de computación constituyen obras producidas por autores, en los términos de las disposiciones de la Ley Federal de Derechos de Autor;

Que dichos programas de computación requieren de la protección jurídica necesaria para evitar la violación de los Derechos de Autor respecto de las mismas por parte de terceros, constituyendo su inscripción en el Registro Público del Derecho de Autor un elemento favorable para obtener la protección mencionada;



Que la producción de las obras de referencia ha tenido un notable incremento en nuestro país en los últimos años;

Que los programas de computación tienen características propias que los distinguen del resto de las obras susceptibles de protección por el Derecho de Autor, tanto por lo que se refiere a su contenido como a los diversos soportes materiales en que se encuentran incorporados, por lo que se presumirá la buena fe del solicitante de la inscripción correspondiente, y

Que de conformidad con lo dispuesto por el artículo 122 de la Ley Federal de Derechos de Autor, toda inscripción deja a salvo los derechos de tercero, he tenido a bien expedir el siguiente ACUERDO No. 114, POR EL QUE SE DISPONE QUE LOS PROGRAMAS DE COMPUTACION PODRAN INSCRIBIRSE EN EL REGISTRO PUBLICO DEL DERECHO DE AUTOR.

PRIMERO.- La Dirección General del Derecho de Autor procederá a inscribir en el Registro Público del Derecho de Autor a los programas de computación cuyo registro se solicite, una vez que se haya cumplido con los requisitos que para el efecto establecen las disposiciones legales aplicables.

SEGUNDO.- Para los efectos de dicha inscripción, el solicitante podrá presentar a su elección, las primeras y las últimas 10 hojas que corresponden al programa fuente, al programa objeto o a ambos.

TERCERO.- En todos los casos, el solicitante deberá acompañar a la solicitud correspondiente una breve explicación del contenido del programa de computación en cuestión.

CUARTO.- El solicitante podrá también presentar los ejemplares del programa de computación necesarios para el otorgamiento del registro, contenidos en cualquier tipo de soporte material. Cuando el solicitante exhiba soportes materiales diversos a su expresión impresa en papel deberá acompañar a los mismos las primeras y las últimas 10 hojas impresas del programa en cuestión, las cuales se devolverán al interesado con las anotaciones correspondientes.

QUINTO.- La Dirección General de Derechos de Autor adoptará las medidas administrativas que sean necesarias para el cumplimiento del presente acuerdo.

## TRANSITORIO

UNICO.- El presente acuerdo entrará en vigor al día siguiente de su publicación en el Diario Oficial de la Federación.

Sufragio Efectivo. No Reelección.

México, D.F., a 28 de septiembre de 1984.- El Secretario,

Jesús Reyes Heróles.- Rúbrica"

## CONTRATOS

Otro elemento es el contrato. En efecto, en la práctica lo más común es que los programas de cómputo sean comercializados a través de contratos de licencia de uso. En dichos contratos usualmente se pacta que el usuario reconoce la propiedad del licenciante sobre los programas amparados por el contrato. Adicionalmente, el usuario queda obligado a mantener estricta confidencialidad sobre el contenido de los programas, estándole prohibido revelar su contenido a persona alguna.

También es usual que se estipule que el usuario no podrá duplicar o copiar los programas, excepto un número determinado para fines de archivo. A la terminación del contrato, el usuario deberá certificar por escrito la destrucción total de los programas bajo su custodia o en su defecto regresar todas las copias o ejemplares correspondientes.

Así pues el contrato es un vehículo muy efectivo para evitar la "piratería". El contrato falla justamente cuando el programa es copiado o reproducido ilegalmente por un tercero, con quien el licenciante dueño del mismo no ha celebrado ningún contrato. Es aquí donde el Derecho de Autor debía funcionar, para perseguir y sancionar el "robo" del programa.

Subrayando lo anterior podríamos decir que si el dueño del soporte lógico detecta que el usuario con el que tiene celebrado un contrato, saca copias indebidas e inclusive pretende comercializarlo, podrá demandarle con base en el contrato. Pero en los casos en que el infractor ha hurtado el programa o lo ha obtenido en forma tal que no exista contrato alguno, el dueño del soporte lógico se verá en serias dificultades para perseguir al presunto infractor. Ello es debido a la falta de una legislación adecuada en México.

Como ejemplo de un contrato se mostrará, el del sistema LOGICAT, ver apéndice F.

**CAPITULO V**

**SISTEMA INFOCUS/PRONAL  
(INFORMACION CONACYT UNAM SISTEMATIZADA  
PROYECTOS NACIONALES)**

**CASO DE ESTUDIO**  
-----

## ANTECEDENTES

### COORDINACION DE LA INVESTIGACION CIENTIFICA

#### FUNCIONES

Ejecutar las decisiones tomadas por el Consejo Técnico de la Investigación Científica y servir de órgano de apoyo de éste, para coordinar e impulsar las labores de los institutos y centros, dentro de los lineamientos fijados por el propio Consejo. Servir de enlace con las demás dependencias universitarias y con personas o instituciones extrauniversitarias, supervisar las labores de los centros adscritos a ella y realizar diversos estudios sobre investigación que permitan optimizar todos los recursos.

La Coordinación de la Investigación Científica (CIC) fue creada por el Consejo Universitario en 1945. Las obligaciones y facultades del Coordinador, según el Estatuto General de la UNAM son:

- I. Convocar y presidir las sesiones del Consejo Técnico.
- II. Servir de órgano ejecutivo de las decisiones tomadas por éste, y
- III. Coordinar e impulsar las labores de los institutos de investigación científica dentro de los lineamientos generales que le fije el propio consejo.

Las tareas inherentes a las atribuciones señaladas, reclaman un esfuerzo cada vez mayor, debido al desarrollo que han tenido las actividades de investigación científica de los últimos años.

Aparte de las atribuciones anteriores y dado el crecimiento tanto de la propia Universidad Nacional Autónoma de México, como del país, los rectores han delegado en el coordinador ciertas actividades de investigación científica. Así, la acción de la Coordinación no se circunscribe solo a nivel intrauniversitario sino que traspasa sus muros a fin de propiciar el desarrollo de esa función universitaria.

Sus principales funciones están vinculadas con la coordinación de las labores de los institutos y de los centros del subsistema; con la promoción de investigaciones en colaboración con las facultades y escuelas del área; y con otras dependencias del subsistema de la Investigación Humanística y con gestiones de ayuda académica y económica de instituciones extrauniversitarias.

Por otra parte el Coordinador convoca y preside las sesiones del Consejo Técnico de la Investigación Científica, además de ser el órgano ejecutivo de las decisiones ahí tomadas. Es, asimismo, el enlace entre la Rectoría y las dependencias y subdependencias del subsistema de investigación científica en el establecimiento o modificación de políticas administrativas que le concierne.

Al Coordinador le compete también discutir con la Secretaría General Auxiliar y con los Institutos y Centros la política presupuestal; supervisar las actividades de los centros bajo su dependencia; concentrar, procesar y suministrar información sobre el subsistema de Investigación Científica; colaborar con los órganos de información y divulgación de la UNAM; y administrar los recursos humanos y materiales a su disposición.

Por último, atiende aquellos asuntos que le encomiendan las autoridades universitarias, para desarrollar las actividades de Investigación.

#### ORGANIZACION

Coordinador de la Investigación Científica

Secretaría Académica

Secretaría Académica para Asuntos Bibliotecarios

Secretaría Administrativa

Secretaría Técnica para Asuntos de Política y Proyectos

Secretaría Técnica de Bancos de Datos y Estadística.

#### SECRETARIA TECNICA DE LA INVESTIGACION DE LA COORDINACION CIENTIFICA

La Secretaría Técnica tiene entre sus funciones aquellas que se derivan del cumplimiento del acuerdo suscrito por el Rector de la UNAM, en el que se designa al Coordinador de la Investigación Científica como enlace oficial de nuestra Casa de Estudios con el CONACYT. Para este objeto, todas las solicitudes de subvención al CONACYT en los rubros Proyectos Nacionales, Proyectos Internacionales, Apoyos Especiales y Fortalecimiento al Posgrado requieren de la presentación Institucional a través de la CIC para su trámite.

Esta Secretaría Técnica tiene a su cargo la revisión, sistematización, trámite de presentación ante el CONACYT y archivo de toda la documentación relativa a los rubros anteriormente citados.

El proceso de control y sistematización de la información se ha venido realizando en forma manual, lo que requería una cantidad considerable de tiempo y de esfuerzo, estando sujeta a los inconvenientes que se presentan en éste tipo de sistematización no automatizada; falta de homogeneidad, errores, falta de información, dificultad para realizar búsquedas de información específica, etc.

La necesidad de automatizar el proceso surgió por el incremento que en los últimos años ha tenido el volumen de solicitudes turnadas a CONACYT; basta citar que entre 1983 y 1984 este volumen prácticamente se duplicó, además de haberse generados nuevos rubros y tipos de apoyo como en el caso de Fortalecimiento al Posgrado, que empezó a operar oficialmente en 1984.

Anteriormente se había considerado la posibilidad de automatizar el sistema, pero en ninguna ocasión llegó a iniciarse siquiera este proceso.

#### OBJETIVOS GENERALES DEL SISTEMA

Los objetivos que se desea que cumpla el sistema, pueden sintetizarse de la siguiente manera:

a) Contar con información precisa, en forma expedita, que permita el análisis, evaluación y programación del apoyo que el CONACYT brinde a la UNAM para el cumplimiento de las funciones de docencia e Investigación.

b) Disponer de un archivo en memoria de computadora, que permita preservar y consultar la información durante los períodos que se considere adecuados.

c) Poder realizar búsquedas de información específica, lo que manualmente es, en la práctica, sumamente difícil.

En resumen, se pretende poner a la disposición de las autoridades de la UNAM un instrumento que les proporcione el conocimiento de elementos importantes en materia de toma de decisiones en política científica universitaria.

## VOLUMEN DE LA INFORMACION

La información requerida para llevar el control de iniciativas y proyectos (expedientes) que se desea almacenar, consta de 962 caracteres por expediente. El número anual de expedientes a manejar se calcula entre 500 y 600 aproximadamente, se pretende que el sistema soporte la información de los últimos dos años, y debe considerarse un incremento de la información a almacenar en el transcurso del tiempo.

## FRECUENCIA DE RECUPERACION

Se requiere una recuperación de la información con las siguientes características:

1) Recuperación de la información de todos los expedientes, tres veces al año.

2) Recuperación de la información específica, de acuerdo con los requerimientos de la CIC, esta frecuencia se considera alta.

Al estudiar los objetivos y necesidades que debe satisfacer el sistema, se llegó a las siguientes conclusiones:

a) Debido a la cantidad de información a almacenar, las frecuencias de recuperación (tanto parcial, como total), se consideró que la utilización de un Sistema Manejador de Bases de Datos sería una opción deseable.

b) El sistema debe realizar las siguientes funciones:

- i) Altas de registros,
- ii) Bajas de registros,
- iii) Cambios a los registros,
- iv) Consultas y reportes no planeados y
- v) Consultas y reportes planeados.

## ALTERNATIVAS PARA LA OBTENCION DEL SOPORTE LOGICO DE APLICACION

I) Desarrollo interno de la aplicación.

El desarrollo interno del sistema (considerando como usuario a la CIC) no es posible, debido a que la Secretaría Técnica no dispone del personal especializado para realizar la aplicación.

## II) Compra de un paquete de aplicación.

Inicialmente se consideró que la implantación del sistema se hiciera en una microcomputadora APPLE II con que cuenta la Secretaría Técnica. Sin embargo, esta posibilidad se desechó debido a las limitaciones de memoria (con respecto al volumen de información a manejar) de este equipo. Quedando descartada la posibilidad de comprar un paquete de aplicación que corriera en este equipo.

## III) Contratación para el desarrollo de la aplicación.

La Dirección General de Servicios de Cómputo Académico (DGSCA) brinda a las dependencias universitarias apoyo para el desarrollo de sus aplicaciones, por lo cual la "contratación" ofrece las siguientes ventajas:

a) Prácticamente el costo de desarrollo de la aplicación sería nulo para la CIC.

b) La Coordinación tiene establecida una línea de teleproceso con el equipo B7800 instalado en la DGSCA.

c) Con la configuración B7800 no se tendrán las limitaciones de almacenamiento de información.

d) Disponibilidad del Sistema Manejador de Bases de Datos DMS-II.

e) Disponibilidad de utilizar para el desarrollo lenguajes de Tercera y Cuarta Generación.

Por lo expuesto anteriormente se decide "contratar" a la DGSCA, específicamente a la Dirección de Cómputo para la Investigación para el desarrollo del sistema.

## METODOLOGIA DE DESARROLLO

La metodología que se eligió para desarrollar el sistema es la expuesta en el capítulo III.

A continuación se hará una descripción del proceso de desarrollo del sistema mediante la metodología mencionada, la cual se divide en tres fases: la fase de Planeación, la fase de Desarrollo y la fase de Mantenimiento.



## FASE DE PLANEACION

### Especificación del sistema.

El sistema deberá realizar las siguientes funciones sobre los registros contenidos en una base de datos:

- 1) Altas,
- 2) Bajas,
- 3) Consultas,
- 4) Consultas y reportes no planeados y
- 5) Consultas y reportes planeados.

El sistema facilitará el manejo de la información (operación que se realiza en forma manual y que representa una carga de trabajo muy grande, así como también limitaciones considerables) que la CIC requiere para cumplir con sus funciones en la política científica universitaria.

El tiempo asignado a este proyecto es de 3 meses. El equipo de trabajo esta constituido por 4 personas, y se usará el equipo Burroughs 7800 instalado en la Dirección General de Servicios de Cómputo Académico de la UNAM para desarrollar el sistema.

Los reportes planeados que se requieren son los siguientes:

1) Listado de iniciativas de proyectos nacionales por institución universitaria, por periodo específico.

#### FORMATO:

periodo/ nombre y clave del proyecto/investigador responsable/dirección adjunta/ presupuesto total solicitado a CONACYT.

2) Listado de convenios de proyectos nacionales por institución universitaria, por periodo específico.

#### FORMATO:

periodo/nombre y clave del proyecto/investigador responsable/dirección adjunta/ fecha de firma del convenio/duración en meses/presupuesto total asignado por CONACYT al proyecto.

3) Listado de montos solicitados al CONACYT por instituciones universitarias.

FORMATO:

período/institución universitaria/número de  
iniciativas/monto solicitado 1er. año/monto solicitado 2o.  
año/monto solicitado 3er. año/monto solicitado 4o.  
año/monto total solicitado.

4) Listado de montos convenidos con CONACYT por institución universitaria.

FORMATO:

período/número de convenios/monto comprometido 1er.  
año/monto comprometido 2o. año/monto comprometido 3er.  
año/monto comprometido 4o. año/monto total comprometido por  
CONACYT.

PROGRAMA DE TRABAJO

I) Diseño de la forma de captura y elaboración del manual para la codificación de la información.

II) Transcripción de la información a las formas de captura. Esto se realizará debido a que la documentación es estrictamente confidencial.

III) Almacenar la información en la computadora. Para ello se desarrollará un programa de captura(para el caso de almacenamiento masivo de información).

IV) Realizar un programa que almacene, en un formato dado por el usuario, la información de las iniciativas y proyectos. Dicho programa debe generar diagnósticos de la información faltante.

V) Validar la información capturada, así como su actualización; mediante otro programa.

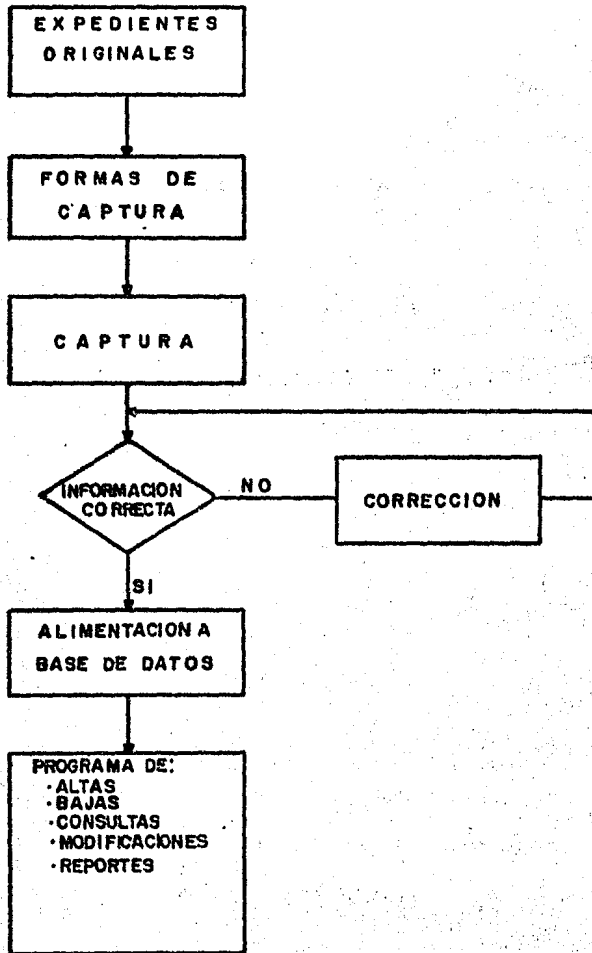
VI) Teniendo la información debidamente validada, se hará la carga a la base de datos diseñada para este propósito.

VII) Elaboración del sistema que realice: altas, bajas, cambios, consultas, validación, etc. Se pretende utilizar un lenguaje de Cuarta Generación y el Sistema Manejador de Base de Datos DMS-II de Burroughs.

Algunas de estas etapas se realizarán en paralelo, por ejemplo la etapa VII se comenzará desde el inicio de la etapa III.

El programa de captura masiva solo se usará temporalmente para validar la información de que se dispone. Posteriormente todo el proceso será realizado por el sistema elaborado en el paso VII.

# FLUJO DE LA INFORMACION



## DISEÑO DE PROGRAMAS PARA EL MANEJO DE INFORMACION MASIVA

Programas temporales para los procesos de: CAPTURA, ACTUALIZACION E IMPRESION de la información de proyectos nacionales realizados por la UNAM (estos programas serán utilizados solo al inicio).

### Objetivos Generales:

Realizar los procesos anteriormente mencionados para el manejo de información masiva.

El lenguaje de programación, que se utilizó fue: ALGOL, debido a que es uno de los mas poderosos en el sistema B7800 y porque se tiene la familiaridad con él.

El proceso más laborioso, es el de reconstrucción e impresión de la información, ya que tiene lo siguiente:

#### - 4 extensos catálogos:

Programas indicativos

Disciplinas

Dependencias Universitarias y extrauniversitarias

Contraparte Internacional.

- Fechas y cantidades faltantes, existentes y no existentes.

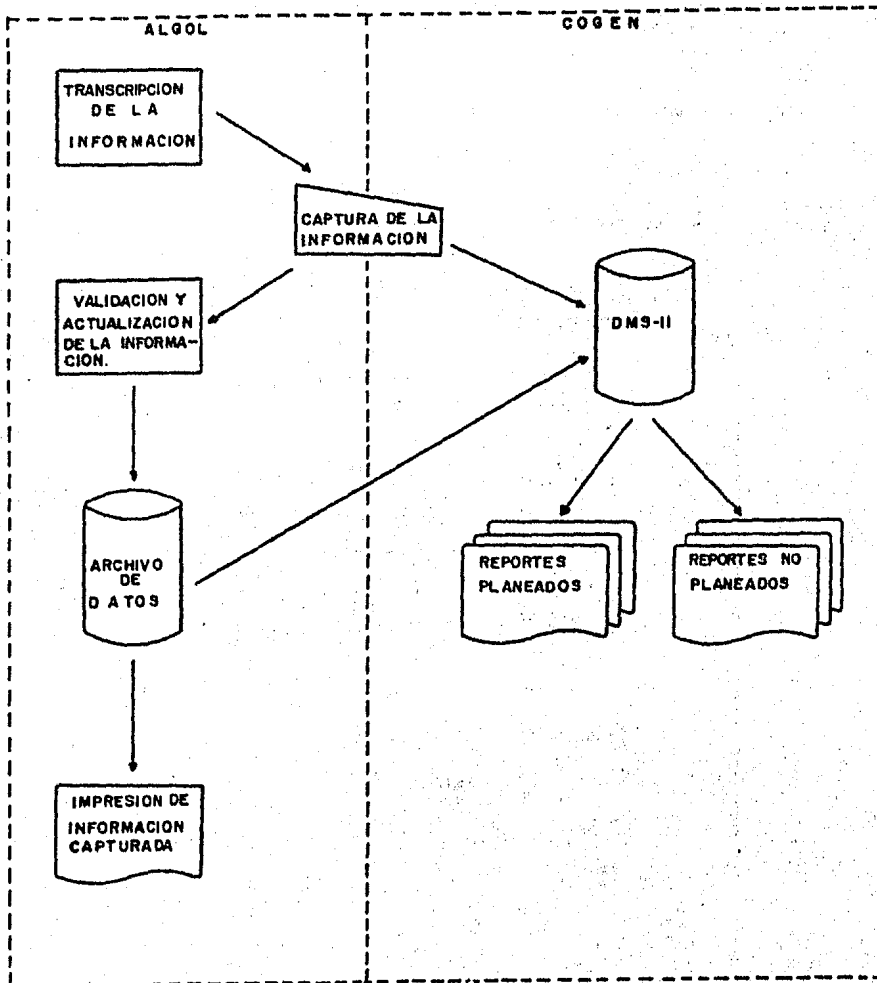
- Reportes de información faltante

- Procedimientos de formateo en la impresión

- Impresión prevista a 1/6 de pulgada y tamaño carta.

El tiempo en el que se realizaron los programas fue de tres semanas y media.

# DIAGRAMA DE BLOQUES



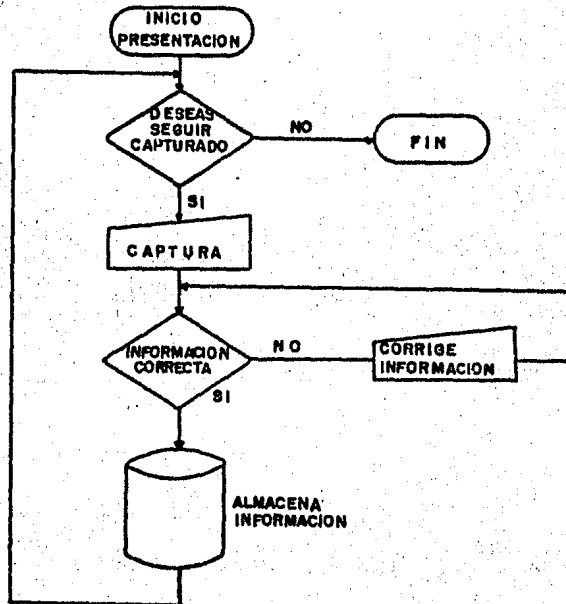
Programa de captura: CAPTURA/PRONAL

Objetivos:

Permitir realizar el proceso de captura masiva de proyectos nacionales que maneja la CIC.

El programa contiene nueve subrutinas, utiliza caracteres de control que permiten un versátil y agradable proceso de captura, así como también tiene un manejo dinámico de archivos.

### DIAGRAMA DE FLUJO



En este proceso la imagen del registro almacenado es la siguiente:

REGISTRO INFOCUS/PRONAL

-----

I Elemento	I N. de C. I	Posición	I
1.- Clave del proyecto	13	A(0)	P(A)
2.- Dirección Adjunta CONACYT	1	A(2)+2	P(A)+14
3.- Programa indicativo	2	A(2)+4	P(A)+16
4.- Disciplina	4	A(3)+1	P(A)+19
5.- Tipo de proyecto	1	A(3)+5	P(A)+23
6.- Dep. universitaria solicitante	5	A(4)+1	P(A)+25
7.- Inst. nal. participante	5	A(5)+1	P(A)+31
8.- Otra Inst. nal. participante	5	A(6)+1	P(A)+37
9.- Otra Inst. nal. participante	5	A(7)+1	P(A)+43
10.- Contraparte internacional	4	A(8)+2	P(A)+50
11.- Otra Contraparte internacional	4	A(9)+2	P(A)+56
12.- Nombre del proyecto	180	A(10)	P(A)+60
13.- Nombre del investigador responsable	40	A(40)	P(A)+240
14.- Grado académico	1	A(46)+5	P(A)+281
15.- Fecha de presentación iniciativa	6	A(47)	P(A)+282
16.- Número de oficio	4	A(48)+2	P(A)+290
17.- Fecha propuesta de inicio	6	A(49)	P(A)+294
18.- Duración en meses	2	A(50)+2	P(A)+302
19.- Cantidad solc. a CONACYT 1 año	6	A(51)	P(A)+306
20.- Cantidad solc. a CONACYT 2 año	6	A(52)	P(A)+312
21.- Cantidad solc. a CONACYT 3 año	6	A(53)	P(A)+318
22.- Cantidad solc. a CONACYT 4 año	6	A(54)	P(A)+324
23.- Cantidad solc. a CONACYT total	6	A(55)	P(A)+330
24.- Cantidad aport. por UNAM 1 año	6	A(56)	P(A)+336
25.- Cantidad aport. por UNAM 2 año	6	A(57)	P(A)+342
26.- Cantidad aport. por UNAM 3 año	6	A(58)	P(A)+348
27.- Cantidad aport. por UNAM 4 año	6	A(59)	P(A)+354
28.- Cantidad aport. por UNAM total	6	A(60)	P(A)+360
29.- Cantidad aport. por otros 1 año	6	A(61)	P(A)+366
30.- Cantidad aport. por otros 2 año	6	A(62)	P(A)+372
31.- Cantidad aport. por otros 3 año	6	A(63)	P(A)+378
32.- Cantidad aport. por otros 4 año	6	A(64)	P(A)+384
33.- Cantidad aport. por otros total	6	A(65)	P(A)+390
34.- Resultado de la evaluación	1	A(66)+1	P(A)+397
35.- Addendum	1	A(66)+3	P(A)+399
36.- Fecha de firma del convenio	6	A(67)	P(A)+402
37.- Fecha de terminación del convenio	6	A(68)	P(A)+408
38.- Cantidad aport. por CONACYT 1 año	6	A(69)	P(A)+414
39.- Cantidad aport. por CONACYT 2 año	6	A(70)	P(A)+420
40.- Cantidad aport. por CONACYT 3 año	6	A(71)	P(A)+426
41.- Cantidad aport. por CONACYT 4 año	6	A(72)	P(A)+432
42.- Cantidad aport. por CONACYT total	6	A(73)	P(A)+438
43.- Cantidad aport. por UNAM 1 año	6	A(74)	P(A)+444



44.- Cantidad aport. por	UNAM	2 año	6	A(75)	P(A)+450
45.- Cantidad aport. por	UNAM	3 año	6	A(76)	P(A)+456
46.- Cantidad aport. por	UNAM	4 año	6	A(77)	P(A)+462
47.- Cantidad aport. por	UNAM	total	6	A(78)	P(A)+468
48.- Cantidad aport. por	otros	1 año	6	A(79)	P(A)+474
49.- Cantidad aport. por	otros	2 año	6	A(80)	P(A)+480
50.- Cantidad aport. por	otros	3 año	6	A(81)	P(A)+486
51.- Cantidad aport. por	otros	4 año	6	A(82)	P(A)+492
52.- Cantidad aport. por	otros	total	6	A(83)	P(A)+498
53.- Resumen del proyecto			480	A(84)	P(A)+504

FORMA DE CAPTURA



COORDINACION DE LA INVESTIGACION CIENTIFICA

SECRETARIA TECNICA

FORMA DE CAPTURA PARA PROYECTOS NACIONALES

1: CLAVE DEL PROYECTO [ ]

2: DIRECCION ADJUNTA CONACYT [ ]

3: PROGRAMA INDICATIVO [ ]

4: DISCIPLINA [ ]

5: TIPO DE PROYECTO [ ]

6: DEPENDENCIA UNIVERSITARIA SOLICITANTE [ ]

7: INSTITUCION NACIONAL PARTICIPANTE [ ]

8: OTRA INSTITUCION NACIONAL PARTICIPANTE [ ]

9: OTRA INSTITUCION NACIONAL PARTICIPANTE [ ]

10: CONTRAPARTE INTERNACIONAL [ ]

11: OTRA CONTRAPARTE INTERNACIONAL [ ]

12: NOMBRE DEL PROYECTO

[ ]  
[ ]  
[ ]

13: NOMBRE DEL INVESTIGADOR RESPONSABLE

[ ]

14: GRADO ACADEMICO [ ]



53 : RESUMEN DEL PROYECTO

Blank lined area for project summary.

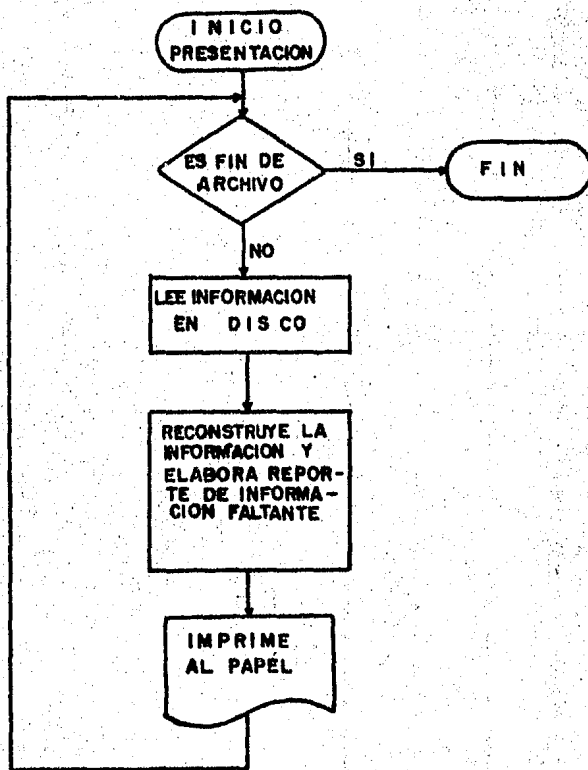
Programa para la reconstrucción de la información almacenada en disco: CONSTRUYE/PRONAL

Objetivos:

Presentar en hojas tamaño carta la información de los proyectos, que queda almacenada durante el proceso de captura y/o proceso de actualización o modificación. Esto es con el propósito de que la CIC tenga la información requerida para su debido control.

Hacer un reporte de la información faltante para cada uno de los proyectos.

### DIAGRAMA DE FLUJO



REPORTES GENERADOS POR EL PROGRAMA  
CONSTRUYE PRONAL

**EJEMPLO DE REPORTE GENERADO PARA UN  
CONVENIO CON INFORMACION FALTANTE**



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
COORDINACION DE LA INVESTIGACION CIENTIFICA  
SECRETARIA TECNICA

PROGRAMA UNAM-CONACYT  
PROYECTOS NACIONALES

CLAVE: IC32301NA8404

INSTITUCION UNIVERSITARIA SOLICITANTE:  
INSTITUTO DE INGENIERIA

NOMBRE DEL PROYECTO:  
MODIFICACION DE UN REFRIGERADOR SOLAR AUTONOMO PARA AMPLIAR SU  
OPERACION A CLIMAS CON ELEVADA TEMPERATURA AMBIENTE.

NOMBRE DEL INVESTIGADOR RESPONSABLE:  
GUTIERREZ FILIBERTO

GRADO ACADEMICO: MAESTRIA

DISCIPLINA: INGENIERIA MECANICA

PRESENTACION A CONACYT:  
FECHA: 22 DE MAYO DE 1984 OFICIO NUMERO: 0376

DIRECCION ADJUNTA DE CONACYT: DESARROLLO TECNOLOGICO

PROGRAMA NACIONAL INDICATIVO: \_\_\_

TIPO DE PROYECTO: NUEVO

FECHA PROPUESTA DE INICIO: \_\_\_\_\_

DURACION: \_\_\_

OTRAS INSTITUCIONES NACIONALES PARTICIPANTES:

- 1.- \_\_\_\_\_
- 2.- \_\_\_\_\_
- 3.- \_\_\_\_\_

CONTRAPARTE INTERNACIONAL:

- 1.- \_\_\_\_\_
- 2.- \_\_\_\_\_

PRESUPUESTO (EN MILES DE PESOS)

	SOLICITADO A CONACYT	APORTADO POR UNAM	APORTADO POR OTROS
PRIMER AÑO:	\$ _____	\$ _____	\$ _____
SEGUNDO AÑO:	\$ _____	\$ _____	\$ _____
TERCER AÑO:	\$ _____	\$ _____	\$ _____
CUARTO AÑO:	\$ _____	\$ _____	\$ _____
TOTAL:	\$ _____	\$ _____	\$ _____

CLAVE: IC32301NA8404

RESULTADO DE LA EVALUACION: \_

RESUMEN DEL PROYECTO:

---

---

---

\*\*\*\*\* REPORTE DE INFORMACION FALTANTE \*\*\*\*\*

- 01.- PROGRAMA INDICATIVO
- 02.- INSTITUCIONES NACIONALES PARTICIPANTES
- 03.- CONTRAPARTE INTERNACIONAL
- 04.- FECHA PROPUESTA DE INICIO DEL PROYECTO
- 05.- DURACION EN MESES DEL PROYECTO
- 06.- DESGLOSE DE CANTIDADES SOLICITADAS Y/O APORTADAS
- 07.- RESULTADO DE LA EVALUACION
- 08.- RESUMEN DEL PROYECTO

EJEMPLO DE REPORTE GENERADO PARA UNA  
INICIATIVA CON INFORMACION FALTANTE

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
COORDINACION DE LA INVESTIGACION CIENTIFICA  
SECRETARIA TECNICA

PROGRAMA UNAM-CONACYT  
PROYECTOS NACIONALES

CLAVE: PCSABNA022845

INSTITUCION UNIVERSITARIA SOLICITANTE:  
INSTITUTO DE INVESTIGACIONES BIOMEDICAS

NOMBRE DEL PROYECTO:  
TRANSFERENCIA DE TECNOLOGIA PARA LA FORMACION DE RECURSOS HUMANOS  
PARA REALIZAR EL INMUNODIAGNOSTICO DE LA CISTICERCOSIS

NOMBRE DEL INVESTIGADOR RESPONSABLE:  
FLISSER ANA

GRADO ACADEMICO: DOCTORADO

DISCIPLINA: INMUNOLOGIA

PRESENTACION A CONACYT:  
FECHA: 03 DE MAYO DE 1984 OFICIO NUMERO: 0292

DIRECCION ADJUNTA DE CONACYT: DESARROLLO TECNOLOGICO

PROGRAMA NACIONAL INDICATIVO: INVESTIGACION SOBRE NUTRICION Y SALUD.

TIPO DE PROYECTO: NUEVO

FECHA PROPUESTA DE INICIO: 01 DE MAYO DE 1984

DURACION: 36 MESES

OTRAS INSTITUCIONES NACIONALES PARTICIPANTES:  
1.- GERENCIA GENERAL DE BIOLOGICAS Y REACTIVOS

PRESUPUESTO (EN MILES DE PESOS)

	SOLICITADO A CONACYT	APORTADO POR UNAM	APORTADO POR OTROS
PRIMER AÑO:	\$ 1500	\$ 2900	\$ 600
SEGUNDO AÑO:	\$ 500	\$ 2900	\$ 600
TERCER AÑO:	\$ 1500	\$ 2900	\$ 600
TOTAL:	\$ 3500	\$ 8700	\$ 1800

RESULTADO DE LA EVALUACION: POSITIVO

FECHA DE FIRMA DEL CONVENIO: 30 DE OCTUBRE DE 1984

FECHA DE TERMINACION DEL CONVENIO: 29 DE OCTUBRE DE 1987

PRESUPUESTO CONVENIDO (EN MILES DE PESOS) APORTADO POR

	CONACYT	UNAM	OTROS
PRIMER AÑO:	\$ 3150	\$ 2900	\$ 600
SEGUNDO AÑO:	\$ 0	\$ 2900	\$ 600
TERCER AÑO:	\$ 350	\$ 2900	\$ 600
TOTAL:	\$ 3500	\$ 8700	\$ 1800

## RESUMEN DEL PROYECTO:

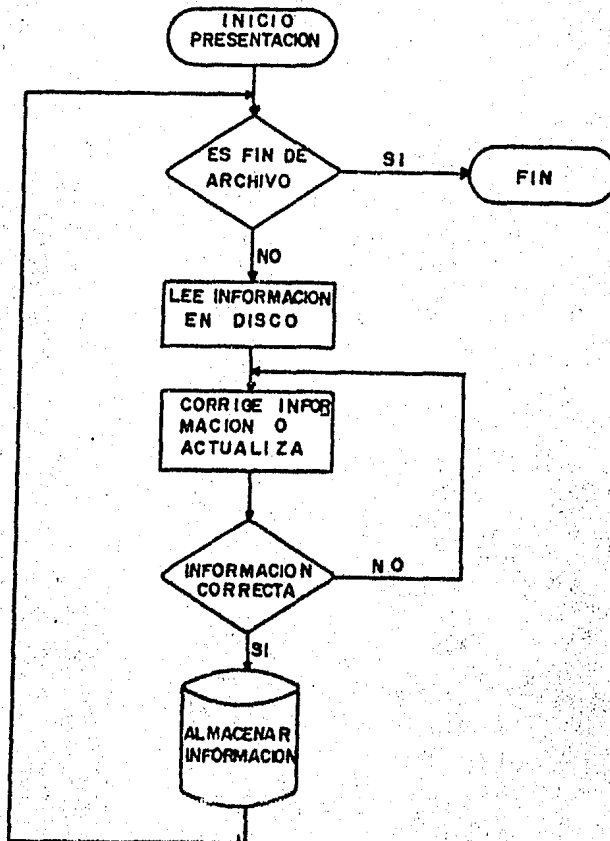
LA CISTICERCOSIS ES UN PROBLEMA IMPORTANTE DE SALUD PUBLICA EN VARIOS PAISES DE LATINOAMERICA, AFRICA Y ASIA DEBIDO A SU FRECUENCIA ELEVADA Y A LA GRAVEDAD DE LA ENFERMEDAD (SCHENONE, 1982; MAHAHAN, 1982). TAMBIEN TIENE TRASCENDENCIA SOBRE LA ECONOMIA GANADERA POR EL DECOMISO DE GANADO. EN MEXICO LA CISTICERCOSIS AFECTA PRINCIPALMENTE A LOS SERES HUMANOS Y AL GANADO PORCINO Y SE DEBE AL ESTABLECIMIENTO DE LA LARVA DE LA TAENIA SOLIUM EN DIVERSOS ORGANOS. LA ...

Programa para modificar y actualizar la información:  
ACTUALIZA/PRONAL

Objetivos:

Actualizar y/o modificar en caso necesario, la información que queda almacenada durante el proceso de captura de la información.

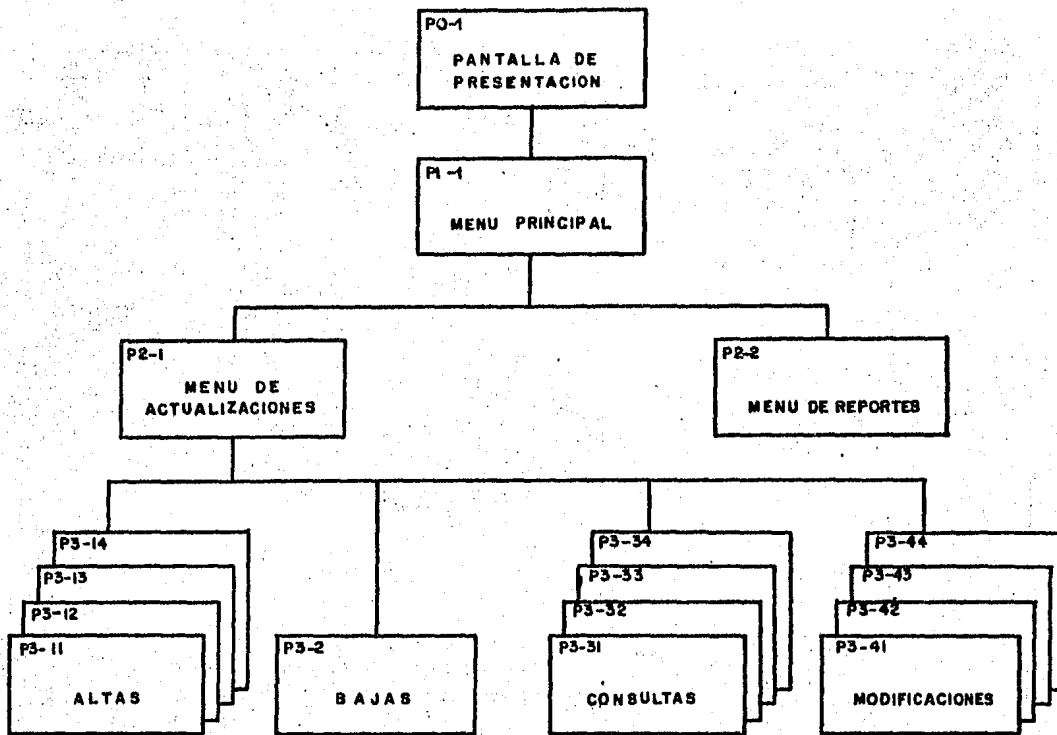
### DIAGRAMA DE FLUJO



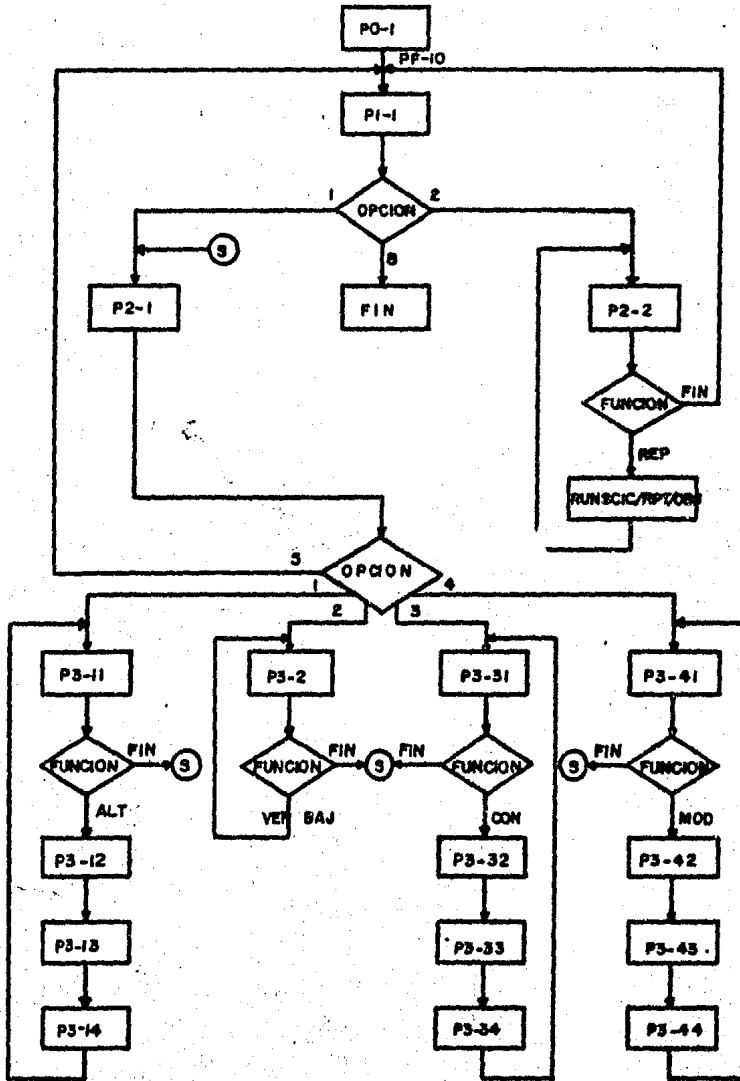
**ESTRUCTURA Y FLUJO DE  
PANTALLAS DEL SISTEMA**



# ESTRUCTURA JERARQUICA DE LAS PANTALLAS DEL SISTEMA



# DIAGRAMA DE FLUJO ENTRE PANTALLAS



DESCRIPCION:

Cada pantalla se identifica por la siguiente clave:

P = PANTALLA

N = NIVEL

U = NUMERO DE LA PANTALLA EN EL NIVEL INDICADO POR EL CAMPO PRECEDENTE (DE IZQUIERDA A DERECHA).

Indice de pantalla

PANTALLA	DESCRIPCION
PO-1	Pantalla de presentación del sistema
P1-1	Menú principal, indica las funciones generales a realizar: actualizaciones y reportes.
P2-1	Menú de actualizaciones, indica las funciones a realizar: altas, bajas, cambios y consultas
P2-2	Menú de reportes planeados. En esta pantalla se presenta la opción de elegir entre 4 tipos de reportes planeados, solo se indica la clave de dependencia y el período considerado.
P3-11	Desde esta pantalla se activa un programa fuera de línea que realiza los reportes requeridos.
P3-11	Altas (hoja 1). Se verifica que la clave no este registrada para darla de alta, en caso contrario se emite un mensaje de error.
P3-12	Altas (hoja 2)
P3-13	Altas (hoja 3)
P3-14	Altas (hoja 4)
P3-2	Bajas. Se realiza una consulta a la base de datos para verificar que la clave este dada de alta, si lo esta presenta la información para su verificación antes de hacer este proceso.
P3-31	Consultas (hoja 1). Se hace una consulta a la base de datos para ver la información que contiene un expediente específico.
P3-32	Consultas (hoja 2)
P3-33	Consultas (hoja 3)
P3-34	Consultas (hola 4)

- P3-41 ..... Cambios (hoja 1). Se hace una consulta a la base de datos, si el expediente esta dado de alta, se hace una lectura. Después de las modificaciones se actualiza la base de datos. Si la clave no esta registrada se emite un mensaje de error.
- P3-42 ..... Cambios (hoja 2)
- P3-43 ..... Cambios (hoja 3)
- P3-44 ..... Cambios (hoja 4)

PO-1

UNAM-CIC

UNAM - COORDINACION DE LA INVESTIGACION  
CIENTIFICA

SECRETARIA TECNICA

SISTEMA DE FINANCIAMIENTO

-----  
UNAM-CONICYT  
-----

OPRIMA F10 PARA CONTINUAR

```
*****
* P1-1 UNAM-CIC *
*
* MENU PRINCIPAL *
*
* OPCIONES: *
*
* 1.- ACTUALIZACIONES *
* 2.- REPORTES *
*
* OPCION ELEGIDA: ( ) *
*
* 8.- FIN *
* MSG: ( *
*****
```

P2-1

UNAM-CIC

MENU DE ACTUALIZACIONES

OPCIONES:

- 1.- ALTAS
- 2.- BAJAS
- 3.- CONSULTAS
- 4.- CAMBIOS

OPCION ELEGIDA: ( )

5.- MENU PRINCIPAL

MSG: (

)

```
*****
* P2-2 FUNCION: ( ) UNAM-CIC *
*
* INSTITUCION UNIVERSITARIA: ( ) *
* PERIODO: ( ) ( ) *
*
* 1.- REPORTE DE INICIATIVAS DE PROYECTOS NACIONALES *
* POR INSTITUCION UNIVERSITARIA. *
* 2.- REPORTE DE CONVENIOS DE PROYECTOS NACIONALES *
* POR INSTITUCION UNIVERSITARIA. *
* 3.- REPORTE DE MONTOS SOLICITADOS AL CONACYT POR *
* INSTITUCION UNIVERSITARIA. *
* 4.- REPORTE DE MONTOS CONVENIDOS CON CONACYT POR *
* INSTITUCION UNIVERSITARIA. *
*
* OPCION: ( ) *
*
* REP.-REPORTES FIN.-MENU PRINCIPAL *
* MSG: ( ) *
*****
```

116



```

*****
* ( ) FUNCION: ( ) CLAVE DEL PROYECTO: ( ) UNAM-CIC *
*
* 2.- DIRECCION ADJUNTA: ( ) *
* 3.- PROGRAMA INDICATIVO: ( ) *
* 4.- DISCIPLINA: ( ) *
* 5.- CARACTER DEL PROYECTO: ( ) *
* 6.- DEP. UNIVERSITARIA SOLICITANTE: ( ) *
* 7.- INST. NACIONAL SOLICITANTE: ( ) *
* 8.- OTRA INST. NAL. PARTICIPANTE: ( ) *
* 9.- OTRA INST. NAL. PARTIFIPANTE: ( ) *
* 10.- CONTRAPARTE INTERNACIONAL: ( ) *
* 11.- OTRA CONTRAPARTE INTERNACIONAL:( ) *
* 12.- NOMBRE DEL PROYECTO: ( ) *
*
* 13.- NOMBRE DEL INVEST. RESPONSABLE: ( ) *
* ( ) *
* 14.- GRADO ACADEMICO: ( ) *
* 15.- FECHA PRESENT. DE INICIATIVA: ( ) *
* 16.- NUMERO DE OFICIO: ( ) *
*
* FIN: MENU DE ACTUALIZACION *
* MSG: ( ) *
*****

```

```

*****
* ( ) FUNCION: ( ) CLAVE: ( ) UNAM-CIC *
*
* 17.- FECHA PROPUESTA DE INICIO: ( ) *
* 18.- DURACION EN MESES: ( ) *
* 19.- CANTIDAD SOLICITADA A CONACYT 1 A&O:( ) *
* 20.- CANTIDAD SOLICITADA A CONACYT 2 A&O:( ) *
* 21.- CANTIDAD SOLICITADA A CONACYT 3 A&O:( ) *
* 22.- CANTIDAD SOLICITADA A CONACYT 4 A&O:( ) *
* 23.- CANTIDAD TOTAL SOLICITADA A CONACYT:( ) *
* 24.- CANTIDAD APORTADA POR UNAM 1 A&O: ( ) *
* 25.- CANTIDAD APORTADA POR UNAM 2 A&O: ( ) *
* 26.- CANTIDAD APORTADA POR UNAM 3 A&O: ( ) *
* 27.- CANTIDAD APORTADA POR UNAM 4 A&O: ( ) *
* 28.- CANTIDAD TOTAL APORTADA POR UNAM: ( ) *
* 29.- CANTIDAD APORTADA POR OTROS 1 A&O: ( ) *
* 30.- CANTIDAD APORTADA POR OTROS 2 A&O: ( ) *
* 31.- CANTIDAD APORTADA POR OTROS 3 A&O: ( ) *
* 32.- CANTIDAD APORTADA POR OTROS 4 A&O: ( ) *
* 33.- CANTIDAD TOTAL APORTADA POR OTROS: ( ) *
* 34.- RESULTADO DE LA EVALUACION: ( ) *
*
* MSG: ( *
*****

```

119

```

*****
*      (      )      FUNCION: (      )      CLAVE: (      )      UNAM-CIC      *
*
*      35.- ADDENDUM: (      )
*      36.- FECHA DE FIRMA DEL CONVENIO: (      )
*      37.- FECHA DE TERMINACION DEL CONVENIO: (      )
*      38.- CANTIDAD APORTADA POR CONACYT 1 A&O:(      )
*      39.- CANTIDAD APORTADA POR CONACYT 2 A&O:(      )
*      40.- CANTIDAD APORTADA POR CONACYT 3 A&O:(      )
*      41.- CANTIDAD APORTADA POR CONACYT 4 A&O:(      )
*      42.- CANTIDAD TOTAL APORTADA POR CONACYT:(      )
*      43.- CANTIDAD APORTADA POR UNAM 1 A&O: (      )
*      44.- CANTIDAD APORTADA POR UNAM 2 A&O: (      )
*      45.- CANTIDAD APORTADA POR UNAM 3 A&O: (      )
*      46.- CANTIDAD APORTADA POR UNAM 4 A&O: (      )
*      47.- CANTIDAD TOTAL APORTADA POR UNAM: (      )
*      48.- CANTIDAD APORTADA POR OTROS 1 A&O: (      )
*      49.- CANTIDAD APORTADA POR OTROS 2 A&O: (      )
*      50.- CANTIDAD APORTADA POR OTROS 3 A&O: (      )
*      51.- CANTIDAD APORTADA POR OTROS 4 A&O: (      )
*      52.- CANTIDAD TOTAL APORTADA POR OTROS: (      )
*
*      MSG: (      )
*****

```





## SELECCION DEL LENGUAJE

Para desarrollar el sistema, de entre las diferentes alternativas disponibles (COBOL, ALGOL, COGEN) se seleccionó el lenguaje de Cuarta Generación COGEN, el cual se encuentra disponible en la Dirección General de Servicios de Cómputo Académico.

Este lenguaje de Cuarta Generación fue elegido debido a las facilidades que proporciona para la generación de aplicaciones. Entre las facilidades más importantes están: una disminución en el tiempo y costo de desarrollo, requiere poco tiempo de entrenamiento, estandariza el desarrollo de aplicaciones, facilidades para el diseño de pantallas y reportes, etc. Otra razón importante para la selección de un generador de aplicaciones fue la oportunidad de trabajar con una tecnología relativamente nueva en México y hacer una comparación práctica de los resultados obtenidos con un lenguaje de Cuarta Generación y las experiencias anteriores con lenguajes de Tercera Generación.

En los párrafos que siguen se hace una breve descripción de las características principales de COGEN.

COGEN (CObol GEnerative Programming System) es un producto de Jacksonville Software Development Corporation para computadoras Burroughs, este producto genera programas de aplicación cuyo código fuente se encuentra en COBOL, completos y confiables a partir de especificaciones no procedurales, las cuales describen el problema de aplicación. COGEN también genera las instrucciones de WFL (Work Flow Language) necesarias para la compilación de los programas generados.

Los beneficios que se obtienen al usar este generador de aplicaciones son:

- 1) Hace posible una reducción del tiempo de desarrollo y del costo de la aplicación.
- 2) Los programas generados son más confiables que los codificados a mano.
- 3) Los programas generados pueden cambiarse o expandirse fácilmente.
- 4) Requiere poco tiempo de entrenamiento.
- 5) Permite la estandarización de las aplicaciones generadas.

El proceso para la generación de una aplicación usando COGEN es el siguiente:

#### a) Especificación de la aplicación.

Los detalles de la aplicación para la generación de un programa son codificadas en especificaciones no procedurales simples. Algunas especificaciones definen el medio ambiente general del programa. Estas especificaciones globales describen detalles tales como: cual base de datos y cuales archivos usará el programa, que pantallas manejará, que registros se manejan, etc.

Siguiendo a las especificaciones globales hay uno o más grupos de especificaciones las cuales describen las transacciones individuales a los procesos que deberá realizar el programa generado. Aquí se describen detalles tales como que registros serán actualizados, como se localizará cada registro y los criterios de edición a usar.

Todos las especificaciones para un programa deseado son codificadas en un archivo llamado "archivo de especificaciones" de COGEN.

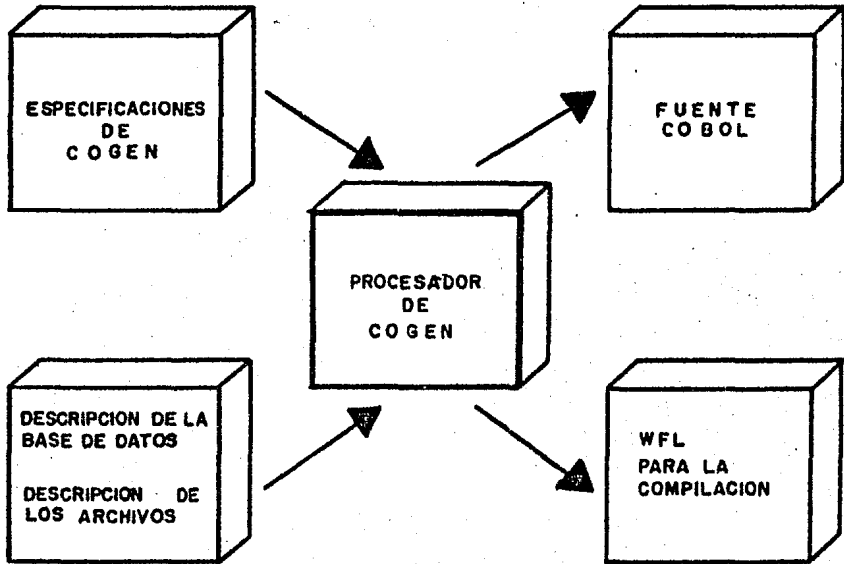
#### b) Generación de la aplicación.

Una vez que el archivo de especificaciones ha sido preparado, se ejecuta el procesador de COGEN. El procesador analizará todas las especificaciones contenidas en el respectivo archivo generando un programa fuente completo en COBOL, así como las instrucciones del lenguaje de control (WFL) necesarias para su compilación. En este instante el usuario puede iniciar el WFL generado para la compilación y obtener un programa fuente ejecutable.

En la siguiente figura se muestra el proceso de generación:

**ARCHIVOS DE ENTRADA  
PROPORCIONADOS POR  
EL USUARIO**

**ARCHIVOS GENERADORES**



El rango de capacidades disponibles por medio de las especificaciones de COGEN es muy completo. Sin embargo, pueden requerirse ciertas porciones de código escrito en COBOL. Considerando estas situaciones COGEN proporciona la capacidad, por medio de las especificaciones, de copiar código fuente COBOL escrito por el usuario y ponerlo en el lugar elegido. Por ejemplo, esta facilidad permite codificar cálculos complejos en COBOL y tener este código automáticamente incluido en el programa fuente generado.

La documentación de la aplicación está dada directamente por las especificaciones. Cada generación produce un listado del archivo de especificaciones junto con información adicional proporcionada por el procesador de COGEN. Este reporte sirve como una documentación completa del programa generado. Cualquier persona familiarizada con COGEN podrá leer el listado de las especificaciones y determinar el propósito y las capacidades del programa de aplicación.

Los recursos requeridos por COGEN para generar programas de aplicación son reducidos. Por ejemplo, para un sistema grande Burroughs (modelo B5900 en adelante) una generación típica hará uso de los siguientes recursos:



Memoria 12-15 K SAVE, (50-60 k total)

Disco 10-20 K segmentos

Tiempo 60 segundos de UCP por 3000-5000 líneas de código fuente generado.

El tiempo de ejecución requerido por un programa generado es comparable al requerido por un programa escrito a mano que realice las mismas funciones. Con COGEN no es necesario soporte físico adicional para tener los beneficios de la generación.

ESPECIFICACIONES DE COGEN  
DEL SUBSISTEMA ONLINE

\*\*\*\*\*  
 COGEN PROCESSOR VERSION 4.7.19                    06/11/85 09:55:37                    EXPIRES ON  
 SPECS FILE IS CIC/SPECS.  
 SUBSYS FILE IS GEN/ONLINE.  
 \*\*\*\*\*

000100%\*\*\*\*\*  
 000200%  
 000300%                    ESPECIFICACIONES GLOBALES                    %  
 000400%                    %  
 000500%\*\*\*\*\*  
 000600 RECORD(CIC/RECORDS)  
   P1-1-REC LOADING  
   P1-1-REC-OUT LOADING  
   P2-1-REC LOADING  
   P2-1-REC-OUT LOADING  
   P0-1-REC LOADING  
   P3-11-REC LOADING  
   P3-11-REC-OUT LOADING  
   P3-12-REC LOADING  
   P3-12-REC-OUT LOADING  
   P3-13-REC LOADING  
   P3-13-REC-OUT LOADING  
   P3-14-REC LOADING  
   P3-14-REC-OUT LOADING  
   P3-2-REC LOADING  
   P3-2-REC-OUT LOADING  
   P2-2-REC LOADING  
   P2-2-REC-OUT LOADING  
   P3-5-REC LOADING  
   P3-5-REC-OUT LOADING  
   P0-1-REC-OUT LOADING  
 000600 RECORD(CIC/RECORDS)  
 000700 RECORD(CIC/RECORDS/PARAMETROS)  
   PARAMETROS-REC LOADING  
 000700 RECORD(CIC/RECORDS/PARAMETROS)  
 000800 RECORD  
   PARAMETROS-ZIP LOADING  
 001200 WORKRECORDS(PARAMETROS-ZIP)  
 001300 DATABASE(CICDB,UNAM3)  
   RESTART STRUCTURE 2 RESTART-DSET LOADED  
   RESTART SET STRUCTURE 3 RESTART-SET LOADED  
 001400 RESTART(RST-USERAREA,2,"RST-PROGRAM-COM","RST-PROGRAM-ID")  
 001500 IDENT(CIC-PROYECTOS-NACIONALES)  
 001600 MASTER(CIC/MASTER/PARAMETROS,PARAMETROS-REC)  
 001700 SEQUENTIAL(PARAMETROS)  
 001800 REMOTE(,P1-1-OPCION=8)  
 001900 SCREENFILE(CIC/SCREENS)  
 002000 FIRSTSCREEN("PO-1")  
 002010%\*\*\*\*\*  
 002020%                    FIN DE ESPECIFICACIONES GLOBALES                    %  
 002030%\*\*\*\*\*  
 002100%\*\*\*\*\*  
 002200%                    %  
 002300%                    PANTALLA DE PRESENTACION                    %  
 002400%                    %  
 002500%\*\*\*\*\*

```

002600TRANSACTION (PO-1-IDPO-1="PO-1",PO-1-REC)
***** TRANSACTION # 1
002700  RESPONSE("P1-1",P1-1-REC-OUT)
002800    MOVE(1,P1-1-OPCION-OUT)
002900XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
003000%  FIN DE PANTALLA DE PRESENTACION  %
003100XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
003200XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
003300%  %
003400%  MENU PRINCIPAL  %
003500%  %
003600XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
003700TRANSACTION(P1-1-IDP1-1="P1-1",P1-1-REC)
***** TRANSACTION # 2
003800  PREACCESSEDEITS(P1-1-MSG)
003900  REQUIRED(P1-1-OPCION)
004000  ERRORMSG("OPCION REQUERIDA")
004100  NUMERIC(P1-1-OPCION)
004200  ERRORMSG("CAMPO NUMERICO")
004300  EDITERROR(P1-1-OPCION 1 OR P1-1-OPCION 8)
004400  ERRORMSG("OPCION FUERA DE RANGO")
004500  ENDEDITS
004600  RESPONSE("P2-1",P2-1-REC-OUT,P1-1-OPCION=1)
004700    MOVE("P2-1",P2-1-IDP2-1-OUT)
004800    MOVE(1,P2-1-OPCION-OUT)
004900  RESPONSE("P2-2",P2-2-REC-OUT,P1-1-OPCION=2)
005000    MOVE("P2-2",P2-2-IDP2-2-OUT)
005100    MOVE("REP",P2-2-FUNCION-OUT)
005200XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
005300%  FIN DE MENU PRINCIPAL  %
005400XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
005500XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
005600%  %
005700%  MENU DE REPORTES  %
005800%  %
005900XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
006000TRANSACTION(P2-2-IDP2-2="P2-2",P2-2-REC)
***** TRANSACTION # 3
006100  PREACCESSEDEITS(P2-2-MSG)
006200  REQUIRED(P2-2-FUNCION)
006300  ERRORMSG("FUNCION REQUERIDA")
006400  ALPHA(P2-2-FUNCION)
006500  ERRORMSG("LA FUNCION ES ALFABETICA ")
006600  SUBTRAN(P2-2-FUNCION ="REP")
006700  REQUIRED(P2-2-DEP-UNI)
006800  ERRORMSG("CLAVE DE DEPENDENCIA UNIVERSITARIA REQUERIDA")
006900  NUMERIC(P2-2-DEP-UNI)
007000  ERRORMSG("CAMPO NUMERICO")
007100  REQUIRED(P2-2-LSPER)
007200  ERRORMSG("FECHA FINAL REQUERIDA")
007300  NUMERIC(P2-2-LSPER)
007400  ERRORMSG("CAMPO NUMERICO")
007500  REQUIRED(P2-2-LIPER)
007600  ERRORMSG("FECHA INICIAL REQUERIDA")
007700  NUMERIC(P2-2-LIPER)
007800  ERRORMSG("CAMPO NUMERICO")
007900  REQUIRED(P2-2-OPCION)

```

```

008000      ERRORMSG("OPCION REQUERIDA")
008100      NUMERIC(P2-2-OPCION)
008200      ERRORMSG("CAMPO NUMERICO")
008300      EDITERROR(P2-2-OPCION 1 OR P2-2-OPCION 4)
008400      ERRORMSG("OPCION FUERA DE RANGO")
008500      ENDSUBTRAN
008600      ENDEDITS
008700      SUBTRAN(P2-2-FUNCION ="REP")
008800      DMSACCESS(CIC-INS-PAR-SET,P2-2-DEP-UNI)
      STRUCTURE 4 CIC-DSET LOADED
      STRUCTURE 7 CIC-INS-PAR-SET LOADED
008900      IFPRESENT(READ)
009000      IFNOTPRESENT(ERROR,"CLAVE DE DEPENDENCIA NO REGISTRADA")
009100      MOVE(LIKEIDS(P2-2-?,P2-2-REC,PARAMETRO-?,PARAMETROS-REC,
009110      P2-2-LSPER,P2-2-LIPER)
      THE FOLLOWING DATA ITEM(S) WILL BE MOVED
      P2-2-DEP-UNI
      P2-2-OPCION
009120      CONVERTDATE(DDMMYY,P2-2-LIPER,YYMMDD,
009130      PARAMETRO-LIPER)
009140      CONVERTDATE(DDMMYY,P2-2-LSPER,YYMMDD,
009150      PARAMETRO-LSPER)
009200      WRITE(PARAMETROS)
009300      ENDSUBTRAN
009700      RESPONSE("P2-2",P2-2-REC-OUT,P2-2-FUNCION="REP")
009800      MOVE(" ",P2-2-FUNCION-OUT)
009900      MOVE(1,P2-2-OPCION-OUT)
010000      MOVE("EDICION PROCESADA",P2-2-MSG-OUT)
010100      RESPONSE("P1-1",P1-1-REC-OUT,P2-2-FUNCION,"FIN")
010200      MOVE(1,P1-1-OPCION-OUT)
010300      MOVE("EDICION DE REPORTES COMPLETA",P1-1-MSG-OUT)
010400%*****
010500%      FIN DE MENU DE REPORTES      %
010600%*****
010700%*****
010800%      %
010900%      MENU DE ACTUALIZACIONES      %
011000%      %
011100%*****
011200TRANSACTION(P2-1-IDP2-1="P2-1",P2-1-REC)
*****
011300      PREACCESSEDEITS(P2-1-MSG)
011400      REQUIRED(P2-1-OPCION)
011500      ERRORMSG("OPCION REQUERIDA")
011600      NUMERIC(P1-1-OPCION)
011700      ERRORMSG("CAMPO NUMERICO")
011800      EDITERROR(P1-1-OPCION 1 OR P1-1-OPCION 5)
011900      ERRORMSG("OPCION FUERA DE RANGO")
012000      ENDEDITS
012100      RESPONSE("P3-11",P3-11-REC-OUT,P2-1-OPCION=1)
012200      MOVE("P3-11",P3-11-IDP3-11-OUT)
012300      MOVE("ALT",P3-11-FUNCION-OUT)
012400      RESPONSE("P3-2",P3-2-REC-OUT,P2-1-OPCION=2)
012500      MOVE("VER",P3-2-FUNCION-OUT)
012600      RESPONSE("P3-11",P3-11-REC-OUT,P2-1-OPCION=3)
012700      MOVE("P3-31",P3-11-IDP3-11-OUT)
012800      MOVE("CON",P3-11-FUNCION-OUT)

```

```

012900  RESPONSE("P3-11",P3-11-REC-OUT,P2-1-OPCION=4)
013000  MOVE("P3-41",P3-11-IDP3-11-OUT)
013100  MOVE("CAM",P3-11-FUNCION-OUT)
013200  RESPONSE("P1-1",P1-1-REC-OUT,P2-1-OPCION=5)
013300  MOVE("P1-1",P1-1-IDP1-1-OUT)
013400  MOVE(1,P1-1-OPCION-OUT)
013500  RESPONSE("P2-1",P2-1-REC-OUT,P2-1-OPCION=0)
013600  MOVE("P2-1",P2-1-IDP2-1-OUT)
013700  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
013800%      FIN DE MENU DE ACTUALIZACION      %
013900  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
014000  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
014100%      %
014200%      ALTAS      %
014300%      %
014400  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
014500  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
014600  %      PANTALLA UNO ALTAS      %
014700  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
014800  TRANSACTION(P3-11-IDP3-11="P3-11",P3-11-REC)
***** TRANSACTION # 5
014900  PREACCESSEDITS(P3-11-MSG)
015000  REQUIRED(P3-11-FUNCION)
015100  ERRORMSG("FUNCION REQUERIDA")
015200  ALPHA(P3-11-FUNCION)
015300  ERRORMSG("CAMPO ALFANUMERICO")
015400  SUBTRAN (P3-11-FUNCION ="ALT")
015500  REQUIRED(P3-11-CLAVE)
015600  ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
015700  KEYNOTPRESENT(CIC-CLAVE-SET,P3-11-CLAVE)
STRUCTURE 4 CIC-DSET LOADED
STRUCTURE 5 CIC-CLAVE-SET LOADED
015800  ERRORMSG("CLAVE YA REGISTRADA")
015900  ENDSUBTRAN
016000  ENDEDITS
016100  SUBTRAN(P3-11-FUNCION ="ALT")
016200  DMSWRITE(CIC-DSET)
016300  MOVELIKEIDS(P3-11-7,P3-11-REC,CIC-7,CIC-DSET,
016400  P3-11-FEC-PRE-INI)
THE FOLLOWING DATA ITEM(S) WILL BE MOVED
P3-11-CLAVE
P3-11-DIR-ADJ
P3-11-PRO-IND
P3-11-DIS
P3-11-CAR-PRO
P3-11-DEP-UNI
P3-11-INS-NAL1
P3-11-INS-NAL2
P3-11-INS-NAL3
P3-11-CON-INT1
P3-11-CON-INT2
P3-11-NOM-PRO
P3-11-NOM-INV
P3-11-GRA-ACA
P3-11-NUM-OFI
016500  CONVERTDATE(DDMMYY,P3-11-FEC-PRE-INI,YYMDD,
016600  CIC-FEC-PRE-INI)

```

```

016700   ENDSUBTRAN
016800   RESPONSE("P3-12",P3-12-REC-OUT,P3-11-FUNCION="ALT")
016850     MOVE("P3-12",P3-12-IDP3-12-OUT)
016900     MOVE("ALT",P3-12-FUNCION-OUT)
017000     MOVE(P3-11-CLAVE,P3-12-CLAVE-OUT)
017100     MOVE("PRIMERA PAGINA GRABADA",P3-12-MSG-OUT)
017200     RESPONSE("P2-1",P2-1-REC-OUT,P3-11-FUNCION="FIN")
017300     MOVE(1,P2-1-OPCION-OUT)
017400     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
017500     %           PANTALLA DOS ALTAS           %
017600     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
017700     TRANSACTION(P3-12-IDP3-12="P3-12",P3-12-REC)
***** TRANSACTION # 6
017800     PREACCESSEDTITS(P3-12-MSG)
017900     REQUIRED(P3-12-FUNCION)
018000     ERRORMSG("FUNCION REQUERIDA")
018100     SUBTRAN (P3-12-FUNCION ="ALT")
018200     REQUIRED(P3-12-CLAVE)
018300     ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
018600     ENDSUBTRAN
018700     ENDEDITS
018800     SUBTRAN(P3-12-FUNCION ="ALT")
018900     DMSACCESS(CIC-CLAVE-SET,P3-12-CLAVE)
019000     IFPRESENT(UPDATE)
019100     IFNOTPRESENT(ERROR,"CLAVE NO REGISTRADA",P3-12-MSG)
019200     MOVELIKEIDS(P3-12-?,P3-12-REC,CIC-?,CIC-DSET,
019220     P3-12-FEC-PRO-INI)
THE FOLLOWING DATA ITEM(S) WILL BE MOVED
P3-12-CLAVE
P3-12-DUR-MES
P3-12-SOL-CYT-A1
P3-12-SOL-CYT-A2
P3-12-SOL-CYT-A3
P3-12-SOL-CYT-A4
P3-12-TOT-SOL-CYT
P3-12-SOL-UNAM-A1
P3-12-SOL-UNAM-A2
P3-12-SOL-UNAM-A3
P3-12-SOL-UNAM-A4
P3-12-TOT-SOL-UNAM
P3-12-SOL-OTROS-A1
P3-12-SOL-OTROS-A2
P3-12-SOL-OTROS-A3
P3-12-SOL-OTROS-A4
P3-12-TOT-SOL-OTROS
P3-12-RES-EVA
019240     CONVERTDATE(DDMMYY,P3-12-FEC-PRO-INI,YMMDD,
019260     CIC-FEC-PRO-INI)
019300   ENDSUBTRAN
019400   RESPONSE("P3-13",P3-13-REC-OUT,P3-12-FUNCION="ALT" AND
019500     P3-12-RES-EVA=1)
019550     MOVE("P3-13",P3-13-IDP3-13-OUT)
019600     MOVE("ALT",P3-13-FUNCION-OUT)
019700     MOVE(P3-12-CLAVE,P3-13-CLAVE-OUT)
019800     MOVE("SEGUNDA PAGINA GRABADA",P3-13-MSG-OUT)
019900     RESPONSE("P3-14",P3-14-REC-OUT,P3-12-FUNCION="ALT" AND
020000     P3-12-RES-EVA NOT=1)

```

```

020050      MOVE("P3-14-",P3-14-IDP3-14-OUT)
020100      MOVE("ALT",P3-14-FUNCION-OUT)
020200      MOVE(P3-12-CLAVE,P3-14-CLAVE-OUT)
020300      MOVE("SEGUNDA PAGINA GRABADA",P3-14-MSG-OUT)
020400      %%%%%%%%%%%
020500      %      PANTALLA TRES ALTAS      %
020600      %%%%%%%%%%%
020700      TRANSACTION(P3-13-IDP3-13="P3-13",P3-13-REC)
***** TRANSACTION # 7
020800      PREACCESEDITS(P3-13-MSG)
020900      REQUIRED(P3-13-FUNCION)
021000      ERRORMSG("FUNCION REQUERIDA")
021100      SUBTRAN (P3-13-FUNCION ="ALT")
021200      REQUIRED(P3-13-CLAVE)
021300      ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
021600      ENDSUBTRAN
021700      ENDEDITS
021800      SUBTRAN(P3-13-FUNCION ="ALT")
021900      DMSACCESS(CIC-CLAVE-SET,P3-13-CLAVE)
022000      IFPRESENT(UPDATE)
022100      IFNOTPRESENT(ERROR,"CLAVE NO REGISTRADA",P3-13-MSG)
022200      MOVELIKEIDS(P3-13-?,P3-13-REC,CIC-?,CIC-DSET,
022300      P3-13-FEC-FIR-CON,FEC-TER-CON)
THE FOLLOWING DATA ITEM(S) WILL BE MOVED
P3-13-CLAVE
P3-13-ADDENDUM
P3-13-FEC-TER-CON
P3-13-APO-CYT-A1
P3-13-APO-CYT-A2
P3-13-APO-CYT-A3
P3-13-APO-CYT-A4
P3-13-TOT-APO-CYT
P3-13-APO-UNAM-A1
P3-13-APO-UNAM-A2
P3-13-APO-UNAM-A3
P3-13-APO-UNAM-A4
P3-13-TOT-APO-UNAM
P3-13-APO-OTROS-A1
P3-13-APO-OTROS-A2
P3-13-APO-OTROS-A3
P3-13-APO-OTROS-A4
P3-13-TOT-APO-OTROS
022400      CONVERTDATE(DDMMYY,P3-13-FEC-FIR-CON,YMMDD,
022500      CIC-FEC-FIR-CON)
022600      CONVERTDATE(DDMMYY,P3-13-FEC-TER-CON,YMMDD,
022700      CIC-FEC-TER-CON)
022800      ENDSUBTRAN
022900      RESPONSE("P3-14",P3-14-REC-OUT,P3-14-FUNCION="ALT")
022950      MOVE("P3-14",P3-14-IDP3-14-OUT)
023000      MOVE("ALT",P3-14-FUNCION-OUT)
023100      MOVE(P3-13-CLAVE,P3-14-CLAVE-OUT)
023200      MOVE("SEGUNDA PAGINA GRABADA",P3-14-MSG-OUT)
023300      %%%%%%%%%%%
023400      %      PANTALLA CUATRO ALTAS      %
023500      %%%%%%%%%%%
023600      TRANSACTION(P3-14-IDP3-14="P3-14",P3-14-REC)
***** TRANSACTION # 8

```



```

023700  PREACCESSEDTITS(P3-14-MSG)
023800  REQUIRED(P3-14-FUNCION)
023900  ERRORMSG("FUNCION REQUERIDA")
024000  SUBTRAN (P3-14-FUNCION ="ALT")
024100  REQUIRED(P3-14-CLAVE)
024200  ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
024500  ENDSUBTRAN
024600  ENDEDITS
024700  SUBTRAN(P3-14-FUNCION ="ALT")
024800  DMSACCESS(CIC-CLAVE-SET,P3-14-CLAVE)
024900  IFPRESENT(UPDATE)
025000  IFNOTPRESENT(ERROR,"CLAVE NO REGISTRADA",P3-14-MSG)
025100  MOVELIKEIDS(P3-14-?,P3-14-REC,CIC-?,CIC-DSET)
      THE FOLLOWING DATA ITEM(S) WILL BE MOVED
      P3-14-CLAVE
      P3-14-RES-PRO
025200  ENDSUBTRAN
025300  MOVE("ALT",P3-14-FUNCION-OUT)
025400  RESPONSE("P3-11",P3-11-REC-OUT,ALL)
025450  MOVE("P3-11",P3-11-IDP3-11-OUT)
025500  MOVE(" ",P3-11-FUNCION-OUT)
025600  MOVE(" ",P3-11-CLAVE-OUT)
025700  MOVE("SE DIO DE ALTA UN NUEVO REGISTRO",P3-11-MSG-OUT)
025800%*****%
025900%          FIN DE ALTAS          %
026000%*****%
026100%*****%
026200%*****%
026300%          BAJAS          %
026400%*****%
026500%*****%
026600  %*****%
026700  %          PANTALLA UNO BAJAS          %
026800  %*****%
026900  INQUIRY(P3-2-FUNCION="VER",P3-2-REC,"P3-2",P3-2-REC-OUT)
      ***** INQUIRY # 1
027000  DMSACCESS(CIC-CLAVE-SET,P3-2-CLAVE)
027100  MISSING(ERROR,"CLAVE NO REGISTRADA",P3-2-MSG)
027200  MOVELIKEIDS(CIC-?,CIC-DSET,P3-2-?-OUT,P3-2-REC-OUT)
      THE FOLLOWING DATA ITEM(S) WILL BE MOVED
      CIC-CLAVE
      CIC-DEP-UNI
      CIC-NOM-PRO
      CIC-NOM-INV
      CIC-GRA-ACA
027300  MOVE("INFORMACION CONTENIDA EN EL REGISTRO CONSULTADO",
027400  P3-2-MSG-OUT)
027500  TRANSACTION(P3-2-IDP3-2="P3-2",P3-2-REC)
      ***** TRANSACTION # 9
027600  SUBTRAN(P3-2-FUNCION ="BAJ")
027700  DMSACCESS(CIC-CLAVE-SET,P3-2-CLAVE)
027800  IFPRESENT(DELETE)
027900  IFNOTPRESENT(ERROR,"*CLAVE NO REGISTRADA",P3-2-MSG)
028000  ENDSUBTRAN
028100  RESPONSE("P3-2",P3-2-REC-OUT,P3-2-FUNCION="BAJ")
028200  MOVE("REGISTRO DADO DE BAJA",P3-2-MSG-OUT)
028300  MOVE("VER",P3-2-FUNCION-OUT)

```

```

028400 MOVE(" ", P3-2-CLAVE-OUT)
028500 RESPONSE("P2-1", P2-1-REC-OUT, P3-2-FUNCION="FIN")
028600 MOVE(1, P2-1-OPCION-OUT)
028700%
028800% FIN DE BAJAS %
028900%
029000%
029100%
029200% CONSULTA %
029300%
029400%
029500 %
029600 % PANTALLA UNO CONSULTA %
029700 %
029800 TRANSACTION(P3-11-IDP3-11="P3-31", P3-11-REC)
***** TRANSACTION # 10
029900 PREACCESSEDITS(P3-11-MSG)
030000 REQUIRED(P3-11-FUNCION)
030100 ERRORMSG("FUNCION REQUERIDA")
030200 SUBTRAN(P3-11-FUNCION="CON")
030300 REQUIRED(P3-11-CLAVE)
030400 ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
030500 ENDSUBTRAN
030600 ENDEDITS
030700 SUBTRAN(P3-11-FUNCION="CON")
030800 DMSACCESS(CIC-CLAVE-SET, P3-11-CLAVE)
030900 IFPRESENT(READ)
031000 IFNOTPRESENT(ERROR, "*CLAVE NO REGISTRADA", P3-11-MSG)
031100 MOVELIKEIDS(CIC-?, CIC-DSET, P3-11-7-OUT, P3-11-REC-OUT,
031200 CIC-FEC-PRE-INI)
THE FOLLOWING DATA ITEM(S) WILL BE MOVED
CIC-CLAVE
CIC-DIR-ADJ
CIC-PRO-IND
CIC-DIS
CIC-CAR-PRO
CIC-DEP-UNI
CIC-INS-NAL1
CIC-INS-NAL2
CIC-INS-NAL3
CIC-CON-INT1
CIC-CON-INT2
CIC-NOM-PRO
CIC-NOM-INV
CIC-GRA-ACA
CIC-NUM-OFI
031300 CONVERTDATE(YMMDD, CIC-FEC-PRE-INI, DDDMMYY,
031400 P3-11-FEC-PRE-INI-OUT)
031500 ENDSUBTRAN
031600 RESPONSE("P3-11", P3-11-REC-OUT, P3-11-FUNCION="CON")
031700 MOVE("P3-31", P3-11-IDP3-11-OUT)
031800 MOVE("CO1", P3-11-FUNCION-OUT)
031900 MOVE("PRIMERA PAGINA DEL REGISTRO", P3-11-MSG-OUT)
032000 RESPONSE("P3-12", P3-12-REC-OUT, P3-11-FUNCION="CO1")
032100 MOVE("P3-32", P3-12-IDP3-12-OUT)
032200 MOVE("CO1", P3-12-FUNCION-OUT)
032300 MOVE(P3-11-CLAVE, P3-12-CLAVE-OUT)

```

```

032400 RESPONSE("P2-1",P2-1-REC-OUT,P3-11-FUNCION="FIN")
032500 MOVE(1,P2-1-OPCION-OUT)
032600 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
032700 % PANTALLA DOS CONSULTA %
032800 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
032900 TRANSACTION(P3-12-IDP3-12="P3-32",P3-12-REC)
***** TRANSACTION # 11

```

```

033000 PREACCESSEDTITS(P3-12-MSG)
033100 REQUIRED(P3-12-FUNCION)
033200 ERRORMSG("FUNCION REQUERIDA")
033300 SUBTRAN(P3-12-FUNCION="CO1")
033400 REQUIRED(P3-12-CLAVE)
033500 ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
033600 ENDSUBTRAN
033700 ENDEDITS
033800 SUBTRAN(P3-12-FUNCION="CO1")
033900 DMSACCESS(CIC-CLAVE-SET,P3-12-CLAVE)
034000 IFPRESENT(READ)
034100 IFNOTPRESENT(ERROR,"*CLAVE NO REGISTRADA",P3-12-MSG)
034200 MOVELIKEIDS(CIC-?,CIC-DSET,P3-12-?-OUT,P3-12-REC-OUT,
034220 CIC-FEC-PRO-INI)

```

THE FOLLOWING DATA ITEM(S) WILL BE MOVED

```

CIC-CLAVE
CIC-DUR-MES
CIC-SOL-CYT-A1
CIC-SOL-CYT-A2
CIC-SOL-CYT-A3
CIC-SOL-CYT-A4
CIC-TOT-SOL-CYT
CIC-SOL-UNAM-A1
CIC-SOL-UNAM-A2
CIC-SOL-UNAM-A3
CIC-SOL-UNAM-A4
CIC-TOT-SOL-UNAM
CIC-SOL-OTROS-A1
CIC-SOL-OTROS-A2
CIC-SOL-OTROS-A3
CIC-SOL-OTROS-A4
CIC-TOT-SOL-OTROS
CIC-RES-EVA

```

```

034240 CONVERTDATE(YMMDD,CIC-FEC-PRO-INT,DDMMYY,
034260 P3-12-FEC-PRO-INT-OUT)
034300 ENDSUBTRAN
034400 RESPONSE("P3-12",P3-12-REC-OUT,P3-12-FUNCION="CO1")
034500 MOVE("P3-32",P3-12-IDP3-12,"T")
034600 MOVE("CO2",P3-12-FUNCION-OUT)
034700 MOVE("SEGUNDA PAGINA DEL RI",P3-12-MSG-OUT)
034800 RESPONSE("P3-13",P3-13-REC-OUT,P3-13-FUNCION="CO2")
034900 MOVE("P3-33",P3-13-IDP3-13,"T")
035000 MOVE("CO2",P3-13-FUNCION-OUT)
035100 MOVE(P3-12-CLAVE,P3-13-CLAVE-OUT)
035200 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
035300 % PANTALLA TRES CONSULTA %
035400 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
035500 TRANSACTION(P3-13-IDP3-13="P3-33",P3-13-REC)
***** TRANSACTION # 11

```

```

035600 PREACCESSEDTITS(P3-13-MSG)

```

```

035700      REQUIRED(P3-13-FUNCION)
035800      ERRORMSG("FUNCION REQUERIDA")
035900      SUBTRAN(P3-13-FUNCION="CO2")
036000      REQUIRED(P3-13-CLAVE)
036100      ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
036200      ENDSUBTRAN
036300      ENDEDITS
036400      SUBTRAN(P3-13-FUNCION="CO2")
036500      DMSACCESS(CIC-CLAVE-SET,P3-13-CLAVE)
036600      IFPRESENT(READ)
036700      IFNOTPRESENT(ERROR,"*CLAVE NO REGISTRADA",P3-13-MSG)
036800      MOVELIKEIDS(CIC-?,CIC-DSET,P3-13-?-OUT,P3-13-REC-OUT,
036900      CIC-FEC-FIR-CON,CIC-FEC-TER-CON)

```

THE FOLLOWING DATA ITEM(S) WILL BE MOVED

```

CIC-CLAVE
CIC-ADDENDUM
CIC-APO-CYT-A1
CIC-APO-CYT-A2
CIC-APO-CYT-A3
CIC-APO-CYT-A4
CIC-TOT-APO-CYT
CIC-APO-UNAM-A1
CIC-APO-UNAM-A2
CIC-APO-UNAM-A3
CIC-APO-UNAM-A4
CIC-TOT-APO-UNAM
CIC-APO-OTROS-A1
CIC-APO-OTROS-A2
CIC-APO-OTROS-A3
CIC-APO-OTROS-A4
CIC-TOT-APO-OTROS

```

```

037000      CONVERTDATE(YMMMDD,CIC-FEC-FIR-CON,DDMMYY,
037100      P3-13-FEC-FIR-CON-OUT)
037200      CONVERTDATE(YMMMDD,CIC-FEC-TER-CON,DDMMYY,
037300      P3-13-FEC-TER-CON-OUT)
037400      ENDSUBTRAN
037500      RESPONSE("P3-13",P3-13-REC-OUT,P3-13-FUNCION="CO2")
037600      MOVE("P3-33",P3-13-IDP3-13-OUT)
037700      MOVE("CO3",P3-13-FUNCION-OUT)
037800      MOVE("TERCERA PAGINA DEL REGISTRO",P3-13-MSG-OUT)
037900      RESPONSE("P3-14",P3-14-REC-OUT,P3-13-FUNCION="CO3")
038000      MOVE("P3-34",P3-14-IDP3-14-OUT)
038100      MOVE("CO3",P3-14-FUNCION-OUT)
038200      MOVE(P3-13-CLAVE,P3-14-CLAVE-OUT)
038300      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
038400      %      PANTALLA CUATRO CONSULTA      %
038500      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
038600      TRANSACTION(P3-14-IDP3-14="P3-34",P3-14-REC)

```

\*\*\*\*\* TRANSACTION # 13

```

038700      PREACCESSEDTITS(P3-14-MSG)
038800      REQUIRED(P3-14-FUNCION)
038900      ERRORMSG("FUNCION REQUERIDA")
039000      SUBTRAN(P3-14-FUNCION="CO3")
039100      REQUIRED(P3-14-CLAVE)
039200      ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
039300      ENDSUBTRAN
039400      ENDEDITS

```

039500 SUBTRAN(P3-14-FUNCION="CO3")  
039600 DMSACCESS(CIC-CLAVE-SET,P3-14-CLAVE)  
039700 IFPRESENT(READ)  
039800 IFNOTPRESENT(ERROR,"\*CLAVE NO REGISTRADA",P3-14-MSG)  
039900 MOVELIKEIDS(CIC-?,CIC-DSET,P3-14-?-OUT,P3-14-REC-OUT)

THE FOLLOWING DATA ITEM(S) WILL BE MOVED

CIC-CLAVE  
CIC-RES-PRO

040000 ENDSUBTRAN  
040100 RESPONSE("P3-14",P3-14-REC-OUT,P3-14-FUNCION="CO3")  
040200 MOVE("P3-34",P3-14-IDP3-14-OUT)  
040300 MOVE("CON",P3-14-FUNCION-OUT)  
040400 MOVE("CUARTA PAGINA DEL REGISTRO",P3-14-MSG-OUT)  
040500 RESPONSE("P3-11",P3-11-REC-OUT,P3-14-FUNCION="CON")  
040600 MOVE("P3-31",P3-11-IDP3-11-OUT)  
040700 MOVE("CON",P3-11-FUNCION-OUT)  
040800 MOVE(" ",P3-11-CLAVE-OUT)  
040900%%%%%%%%%%  
041000% FIN DE CONSULTA %  
041100%%%%%%%%%%  
041200%%%%%%%%%%  
041300%  
041400% CAMBIOS %  
041500%  
041600%%%%%%%%%%  
041700 %%%%%%%%%%%  
041800 % PANTALLA UNO CAMBIOS %  
041900 %%%%%%%%%%%  
042000 INQUIRY(P3-11-FUNCION="CAM",P3-11-REC,"P3 11",P3-11-REC-OUT)  
..... INQUIRY # 2  
042100 DMSACCESS(CIC-CLAVE-SET,P3-11-CLAVE)  
042200 MISSING(ERROR,"CLAVE NO REGISTRADA",P3-11-MSG)  
042300 MOVELIKEIDS(CIC-?,CIC-DSET,P3-11-?-OUT,P3-11-REC-OUT,  
CIC-FEC-PRE-INI)

THE FOLLOWING DATA ITEM(S) WILL BE MOVED

CIC-CLAVE  
CIC-DIR-ADJ.  
CIC-PRO-IND  
CIC-DIS  
CIC-CAR-PRO  
CIC-DEP-UNI  
CIC-INS-NAL1  
CIC-INS-NAL2  
CIC-INS-NAL3  
CIC-CON-INT1  
CIC-CON-INT2  
CIC-NOM-PRO  
CIC-NOM-INV  
CIC-GRA-ACA  
CIC-NUM-OFI

042500 CONVERTDATE(YMMDD,CIC-FEC-PRE-INI,DDMMYY,  
042600 P3-11-FEC-PRE-INI-OUT)  
042700 MOVE("P3-41",P3-11-IDP3-11-OUT)  
042800 MOVE("CA1",P3-11-FUNCION-OUT)  
042900 MOVE("INFORMACION EN LA PAGINA UNO",P3-11-MSG-OUT)  
043000 TRANSACTION(P3-11-IDP3-11="P3-41",P3-11-REC)  
..... TRANSACTION # 14

```

043100   PREACCESSEDITS(P3-11-MSG)
043200   REQUIRED(P3-11-FUNCION)
043300   ERRORMSG("FUNCION REQUERIDA")
043400   SUBTRAN(P3-11-FUNCION="CA1")
043500   REQUIRED(P3-11-CLAVE)
043600   ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
043700   ENDSUBTRAN
043800   ENDEDITS
043900   SUBTRAN(P3-11-FUNCION="CA1")
044000   DMSACCESS(CIC-CLAVE-SET,P3-11-CLAVE)
044100   IFPRESENT(UPDATE)
044200   IFNOTPRESENT(ERROR,"CLAVE NO REGISTRADA",P3-11-MSG)
044300   MOVELIKEIDS(P3-11-?,P3-11-REC,CIC-?,CIC-DSET,
044400   P3-11-FEC-PRE-INI)

```

THE FOLLOWING DATA ITEM(S) WILL BE MOVED

```

P3-11-CLAVE
P3-11-DIR-ADJ
P3-11-PRO-IND
P3-11-DIS
P3-11-CAR-PRO
P3-11-DEP-UNI
P3-11-INS-NAL1
P3-11-INS-NAL2
P3-11-INS-NAL3
P3-11-CON-INT1
P3-11-CON-INT2
P3-11-NOM-PRO
P3-11-NOM-INV
P3-11-GRA-ACA
P3-11-NUM-OFI

```

```

044500   CONVERTDATE(DDMMYY,P3-11-FEC-PRE-INI,YMMMDD,
044600   CIC-FEC-PRE-INI)
044700   ENDSUBTRAN
044800   RESPONSE("P3-12",P3-12-REC-OUT,P3-11-FUNCION="CA1")
044900   MOVE("P3-42",P3-12-IDP3-12-OUT)
045000   MOVE("CA1",P3-12-FUNCION-OUT)
045100   MOVE(P3-11-CLAVE,P3-12-CLAVE-OUT)
045200   MOVE("PAGINA UNO ACTUALIZADA",P3-12-MSG-OUT)
045300   RESPONSE("P2-1",P2-1-REC-OUT,P3-11-FUNCION="FIN")
045400   MOVE(1,P2-1-OPCION-OUT)
045500   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
045600   % PANTALLA DOS CAMBIOS %
045700   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
045800   INQUIRY(P3-12-FUNCION="CA1",P3-12-REC,"P3-12",P3-12-REC-OUT)
***** INQUIRY # 3
045900   DMSACCESS(CIC-CLAVE-SET,P3-12-CLAVE)
046000   MISSING(ERROR,"CLAVE NO REGISTRADA",P3-12-MSG)
046100   MOVELIKEIDS(CIC-?,CIC-DSET,P3-12-?-OUT,P3-12-REC-OUT,
046120   CIC-FEC-PRO-INI)

```

THE FOLLOWING DATA ITEM(S) WILL BE MOVED

```

CIC-CLAVE
CIC-DUR-MES
CIC-SOL-CYT-A1
CIC-SOL-CYT-A2
CIC-SOL-CYT-A3
CIC-SOL-CYT-A4
CIC-TOT-SOL-CYT

```

CIC-SOL-UNAM-A1  
CIC-SOL-UNAM-A2  
CIC-SOL-UNAM-A3  
CIC-SOL-UNAM-A4  
CIC-TOT-SOL-UNAM  
CIC-SOL-OTROS-A1  
CIC-SOL-OTROS-A2  
CIC-SOL-OTROS-A3  
CIC-SOL-OTROS-A4  
CIC-TOT-SOL-OTROS  
CIC-RES-EVA

046140 CONVERTDATE(YMMDD,CIC-FEC-PRO-INI,DDMMYY,  
046160 P3-12-FEC-PRO-INI-OUT)  
046200 MOVE("P3-42",P3-12-IDP3-12-OUT)  
046300 MOVE("CA2",P3-12-FUNCION-OUT)  
046400 MOVE("INFORMACION EN LA PAGINA DOS",P3-12-MSG-OUT)  
046500 TRANSACTION(P3-12-IDP3-12="P3-42",P3-12-REC)

\*\*\*\*\* TRANSACTION # 1:

046600 PREACCESSEDTITS(P3-12-MSG)  
046700 REQUIRED(P3-12-FUNCION)  
046800 ERRORMSG("FUNCION REQUERIDA")  
046900 SUBTRAN(P3-12-FUNCION="CA2")  
047000 REQUIRED(P3-12-CLAVE)  
047100 ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")  
047200 ENDSUBTRAN  
047300 ENDEDITS  
047400 SUBTRAN(P3-12-FUNCION="CA2")  
047500 DMSACCESS(CIC-CLAVE-SET,P3-12-CLAVE)  
047600 IFPRESENT(UPDATE)  
047700 IFNOTPRESENT(ERROR,"CLAVE NO REGISTRADA",P3-12-MSG)  
047800 MOVELIKEIDS(P3-12-?,P3-12-REC,CIC-?,CIC-DSET,  
047820 P3-12-FEC-PRO-INI)

THE FOLLOWING DATA ITEM(S) WILL BE MOVED

P3-12-CLAVE  
P3-12-DUR-MES  
P3-12-SOL-CYT-A1  
P3-12-SOL-CYT-A2  
P3-12-SOL-CYT-A3  
P3-12-SOL-CYT-A4  
P3-12-TOT-SOL-CYT  
P3-12-SOL-UNAM-A1  
P3-12-SOL-UNAM-A2  
P3-12-SOL-UNAM-A3  
P3-12-SOL-UNAM-A4  
P3-12-TOT-SOL-UNAM  
P3-12-SOL-OTROS-A1  
P3-12-SOL-OTROS-A2  
P3-12-SOL-OTROS-A3  
P3-12-SOL-OTROS-A4  
P3-12-TOT-SOL-OTROS  
P3-12-RES-EVA

047840 CONVERTDATE(DDMMYY,P3-12-FEC-PRO-INI,YMMDD,  
047860 CIC-FEC-PRO-INI)  
047900 ENDSUBTRAN  
048000 RESPONSE("P3-13",P3-13-REC-OUT,P3-12-FUNCION="CA2")  
048100 MOVE("P3-43",P3-13-IDP3-13-OUT)  
048200 MOVE("CA2",P3-13-FUNCION-OUT)

```

048300      MOVE(P3-12-CLAVE,P3-13-CLAVE-OUT)
048400      MOVE("PAGINA DOS ACTUALIZADA",P3-13-MSG-OUT)
048500      %%%%%%%%%%%
048600      %      PANTALLA TRES CAMBIOS      %
048700      %%%%%%%%%%%
048800      INQUIRY(P3-13-FUNCION="CA2",P3-13-REC,"P3-13",P3-13-REC-OUT)
***** INQUIRY # 4
048900      DMSACCESS(CIC-CLAVE-SET,P3-13-CLAVE)
049000      MISSING(ERROR,"CLAVE NO REGISTRADA",P3-13-MSG)
049100      MOVELIKEIDS(CIC-?,CIC-DSET,P3-13-?-OUT,P3-13-REC-OUT,
049200      CIC-FEC-FIR-CON,CIC-FEC-TER-CON)
THE FOLLOWING DATA ITEM(S) WILL BE MOVED
CIC-CLAVE
CIC-ADDENDUM
CIC-APO-CYT-A1
CIC-APO-CYT-A2
CIC-APO-CYT-A3
CIC-APO-CYT-A4
CIC-TOT-APO-CYT
CIC-APO-UNAM-A1
CIC-APO-UNAM-A2
CIC-APO-UNAM-A3
CIC-APO-UNAM-A4
CIC-TOT-APO-UNAM
CIC-APO-OTROS-A1
CIC-APO-OTROS-A2
CIC-APO-OTROS-A3
CIC-APO-OTROS-A4
CIC-TOT-APO-OTROS
049300      CONVERTDATE(YMMDD,CIC-FEC-FIR-CON,DDMMYY,
049400      P3-13-FEC-FIR-CON-OUT)
049500      CONVERTDATE(YMMDD,CIC-FEC-TER-CON,DDMMYY,
049600      P3-13-FEC-TER-CON-OUT)
049700      MOVE("P3-43",P3-13-IDP3-13-OUT)
049800      MOVE("CA3",P3-13-FUNCION-OUT)
049900      MOVE("INFORMACION EN LA PAGINA TRES",P3-13-MSG-OUT)
050000      TRANSACTION(P3-13-IDP3-13="P3-43",P3-13-REC)
***** TRANSACTION # 1
050100      PREACCESSEDS(P3-13-MSG)
050200      REQUIRED(P3-13-FUNCION)
050300      ERRORMSG("FUNCION REQUERIDA")
050400      SUBTRAN(P3-13-FUNCION="CA3")
050500      REQUIRED(P3-13-CLAVE)
050600      ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")
050700      ENDSUBTRAN
050800      ENDEDITS
050900      SUBTRAN(P3-13-FUNCION="CA3")
051000      DMSACCESS(CIC-CLAVE-SET,P3-13-CLAVE)
051100      IFPRESENT(UPDATE)
051200      IFNOTPRESENT(ERROR,"CLAVE NO REGISTRADA",P3-13-MSG)
051300      MOVELIKEIDS(P3-13-?,P3-13-REC,CIC-?,CIC-DSET,
051400      P3-13-FEC-FIR-CON,P3-13-FEC-TER-CON)
THE FOLLOWING DATA ITEM(S) WILL BE MOVED
P3-13-CLAVE
P3-13-ADDENDUM
P3-13-APO-CYT-A1
P3-13-APO-CYT-A2

```



P3-13-APO-CYT-A3  
P3-13-APO-CYT-A4  
P3-13-TOT-APO-CYT  
P3-13-APO-UNAM-A1  
P3-13-APO-UNAM-A2  
P3-13-APO-UNAM-A3  
P3-13-APO-UNAM-A4  
P3-13-TOT-APO-UNAM  
P3-13-APO-OTROS-A1  
P3-13-APO-OTROS-A2  
P3-13-APO-OTROS-A3  
P3-13-APO-OTROS-A4  
P3-13-TOT-APO-OTROS

051500 CONVERTDATE(DDMMYY,P3-13-FEC-FIR-CON,YMDD,  
051600 CIC-FEC-FIR-CON)  
051700 CONVERTDATE(DDMMYY,P3-13-FEC-TER-CON,YMDD,  
051800 CIC-FEC-TER-CON)  
051900 ENDSUBTRAN  
052000 RESPONSE("P3-14",P3-14-REC-OUT,P3-13-FUNCION="CA3")  
052100 MOVE("P3-44",P3-14-IDP3-14-OUT)  
052200 MOVE("CA3",P3-14-FUNCION-OUT)  
052300 MOVE(P3-13-CLAVE,P3-14-CLAVE-OUT)  
052400 MOVE("PAGINA TRES ACTUALIZADA",P3-14-MSG-OUT)  
052500 %%%  
052600 % PANTALLA CUATRO CAMBIOS %  
052700 %%%  
052800 %%%  
052900 INQUIRY(P3-14-FUNCION="CA3",P3-14-REC,"P3-14",P3-14-REC-OUT)  
\*\*\*\*\* INQUIRY # 5  
053000 DMSACCESS(CIC-CLAVE-SET,P3-14-CLAVE)  
053100 MISSING(ERROR,"CLAVE NO REGISTRADA",P3-14-MSG)  
053200 MOVELIKEIDS(CIC-?,CIC-DSET,P3-14-?-OUT,P3-14-REC-OUT)  
THE FOLLOWING DATA ITEM(S) WILL BE MOVED  
CIC-CLAVE  
CIC-RES-PRO  
053300 MOVE("P3-44",P3-14-IDP3-14-OUT)  
053400 MOVE("CA4",P3-14-FUNCION-OUT)  
053500 MOVE("INFORMACION EN LA PAGINA CUATRO",P3-14-MSG-OUT)  
053600 TRANSACTION(P3-14-IDP3-14="P3-44",P3-14-REC)  
\*\*\*\*\* TRANSACTION # 1  
053700 PREACCESSEDTITS(P3-14-MSG)  
053800 REQUIRED(P3-14-FUNCION)  
053900 ERRORMSG("FUNCION REQUERIDA")  
054000 SUBTRAN(P3-14-FUNCION="CA4")  
054100 REQUIRED(P3-14-CLAVE)  
054200 ERRORMSG("CLAVE DEL PROYECTO REQUERIDA")  
054300 ENDSUBTRAN  
054400 ENDEDITS  
054500 SUBTRAN(P3-14-FUNCION="CA4")  
054600 DMSACCESS(CIC-CLAVE-SET,P3-14-CLAVE)  
054700 IFPRESENT(UPDATE)  
054800 IFNOTPRESENT(ERROR,"CLAVE NO REGISTRADA",P3-14-MSG)  
054900 MOVELIKEIDS(P3-14-?,P3-14-REC,CIC-?,CIC-DSET)  
THE FOLLOWING DATA ITEM(S) WILL BE MOVED  
P3-14-CLAVE  
P3-14-RES-PRO  
055000 ENDSUBTRAN

```

055100     RESPONSE("P3-11",P3-11-REC-OUT,P3-14-FUNCION="CA4")
055200     MOVE("P3-41",P3-11-IDP3-11-OUT)
055300     MOVE("CAM",P3-11-FUNCION-OUT)
055400     MOVE("                ",P3-11-CLAVE)
055500     MOVE("REGISTRO MODIFICADO",P3-11-MSG-OUT)
055600%*****
055700%                FIN DE CAMBIOS                %
055800%*****
055900INITIALIZE
056000 GENERATE(CIC/FUENTE,CIC/OBJ,CIC/WFL,,2)
        START ONLINE GENERATION
        GENERATING DATA DIVISION
        GENERATING PROCEDURE DIVISION
        GENERATING INQUIRY 1
        GENERATING INQUIRY 2
        GENERATING INQUIRY 3
        GENERATING INQUIRY 4
        GENERATING INQUIRY 5
        GENERATING TRANSACTION 1
        GENERATING TRANSACTION 2
        GENERATING TRANSACTION 3
        GENERATING TRANSACTION 4
        GENERATING TRANSACTION 5
        GENERATING TRANSACTION 6
        GENERATING TRANSACTION 7
        GENERATING TRANSACTION 8
        GENERATING TRANSACTION 9
        GENERATING TRANSACTION 10
        GENERATING TRANSACTION 11
        GENERATING TRANSACTION 12
        GENERATING TRANSACTION 13
        GENERATING TRANSACTION 14
        GENERATING TRANSACTION 15
        GENERATING TRANSACTION 16
        GENERATING TRANSACTION 17
        ONLINE GENERATION COMPLETE
        NO COGEN ERRORS ENCOUNTERED

```

FRAGMENTO DEL PROGRAMA FUENTE  
EN COBOL GENERADO POR (OGEN  
(ONLINE)

\*\*\*\*\*  
\* JACKSONVILLE SOFTWARE DEVELOPMENT CORPORATION \*  
\*\*\*\*\*

\* BELIEVES THAT THE SOFTWARE GENERATED HERewith ACCORDING \*  
\* TO THE USER PROVIDED SPECIFICATIONS IS ACCURATE AND \*  
\* RELIABLE, AND MUCH CARE HAS BEEN TAKEN TO INSURE ITS \*  
\* ACCURACY AND RELIABILITY. HOWEVER, NO RESPONSIBILITY \*  
\* FINANCIAL OR OTHERWISE CAN BE ACCEPTED FOR ANY CONSEQUENCES \*  
\* ARISING OUT OF THE USE OF THIS PROGRAM, INCLUDING LOSS \*  
\* OF PROFIT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES. \*  
\* IT IS THE USERS RESPONSIBILITY TO FULLY TEST ALL GENERATED \*  
\* SOFTWARE TO INSURE THAT IT PERFORMS AS DESIRED. \*  
\*\*\*\*\*

IDENTIFICATION DIVISION.  
\*\*\*\*\*

PROGRAM-ID. CIC-PROYECTOS-NACIONALES.  
AUTHOR. COGEN GENERATOR - ONLINE SUBSYSTEM (VERSION4.7.153).  
DATE-WRITTEN. 06/11/85 09:59:56.  
DATE-COMPILED.  
\*\*\*\*\*

ENVIRONMENT DIVISION.  
\*\*\*\*\*

CONFIGURATION SECTION.  
SOURCE-COMPUTER. BURROUGHS LARGE SYSTEM.  
OBJECT-COMPUTER. BURROUGHS LARGE SYSTEM.  
INPUT-OUTPUT SECTION.  
\*\*\*\*\*

FILE-CONTROL.  
SELECT REMOTEQUE ASSIGN TO REMOTE  
ACTUAL KEY G-REMOTE-KEY.  
SELECT SCREENFILE ASSIGN TO DISK  
RESERVE NO ALTERNATE AREAS  
ACCESS RANDOM  
ACTUAL KEY G-SCREEN-KEY.  
SELECT PARAMETROS ASSIGN TO DISK  
ACTUAL KEY G-MASKEY-1.  
/

\*\*\*\*\*  
DATA DIVISION.  
\*\*\*\*\*

FILE SECTION.  
\*\*\*\*\*

FD REMOTEQUE  
RECORD CONTAINS 1 TO 1920 CHARACTERS.  
01 G-REMOTE-REC  
SIZE 1 TO 1920 DEPENDING ON G-MSG-SIZE.  
FD SCREENFILE  
VALUE OF ID IS "CIC"/"SCREENS"  
RECORD CONTAINS 1980 CHARACTERS.  
01 G-SCREEN-REC SIZE 1980.  
05 G-SCREEN-NAME PIC X(12).  
05 G-SCREEN-SIZE PIC 9(6).  
05 G-SCREEN PIC X(1962).  
FD PARAMETROS  
VALUE OF ID IS "CIC"/"MASTER"/"PARAMETROS".  
01 G-PARAMETROS-REC PIC X(18).  
01 PARAMETROS-REC.

```

05 PARAMETRO-OPCION      PIC 9(1).
05 PARAMETRO-DEP-UNI     PIC 9(5).
05 PARAMETRO-LIPER       PIC 9(6).
05 PARAMETRO-LSPER       PIC 9(6).

```

```

/
*****
DATA-BASE SECTION.
*****

```

```

DB CICDB.
01 RESTART-DSET USING RESTART-SET.
01 CIC-DSET USING CIC-INS-PAR-SET, CIC-CLAVE-SET.
/

```

```

*****
WORKING-STORAGE SECTION.
*****

```

```

77 G-MSG-SIZE             PIC S9(11) COMP-2.
77 G-READ-ERROR          PIC 9(6) VALUE ZERO.
77 G-STOP-FLAG           PIC 9(6) VALUE ZERO.
77 G-MASKEY-1            PIC S9(11) COMP-2.
77 G-LASTKEY-1          PIC S9(11) COMP-2.
77 G-TRANS-STATE-FLAG   PIC 9(6) VALUE ZERO.
77 G-DMS-RESTART-SW     PIC 9(6) VALUE ZERO.
77 G-EDIT-FLAG          PIC 9(6).
77 G-BLK-EDIT-FLAG      PIC 9(6).
77 G-NUM-RESP           PIC 9(6).
77 G-FILE-STATUS        PIC 9(6) VALUE ZERO.
88 G-NOT-FOUND          VALUE 1.
88 G-FOUND              VALUE 2.
77 G-REC-ADDR-1         PIC S9(11) COMP-2.
77 G-REC-ADDR-2         PIC S9(11) COMP-2.
77 G-STATION            PIC 9(6).
77 G-SAVE-STATION-ID    PIC X(18).
77 G-SAVE-REMOTE-KEY    PIC S9(11) COMP-2.
77 G-REMOTE-KEY         PIC S9(11) COMP-2.
77 G-SCREEN-KEY         PIC S9(11) COMP-2.
77 G-SCREEN-SUB         PIC S9(11) COMP-2.
77 G-LOWV              PIC S9(11) COMP-2.
77 G-HIGHV             PIC S9(11) COMP-2.
77 G-SCREEN-ERROR       PIC 9 VALUE ZERO.
77 G-STATE              PIC S9(11) COMP-1.
77 G-SUB                PIC S9(11) COMP-2.
77 G-TEMP               PIC S9(11) COMP.
77 G-TRANS-COUNT        PIC S9(11) COMP-2.
77 G-PROCESSED-TRX     PIC 9.
*****
01 G-TIME-COMP-MSG.
03 FILLER                PIC X(10)          VALUE
   " COMPILED ".
03 G-TIME-DATE            PIC 99/99/99.
03 FILLER                PIC XX           VALUE SPACES
03 G-TIME-HOUR           PIC 99.
03 FILLER                PIC X           VALUE ":".
03 G-TIME-MIN           PIC 99.
01 G-TIME                PIC 9(10).
01 G-INDEX-STATUS       PIC XX.
01 G-DELETED            PIC X(4).
01 G-ID                 PIC X(12).

```

```

01 G-SCREEN-ID PIC X(12).
01 G-STATION-ID PIC X(12).
01 G-ACCESS-STATES.
05 G-ACCESS-STATE-1 PIC 9.
05 G-ACCESS-STATE-2 PIC 9.
***** LIBRARY INTERFACE *****
01 G-LIB-RECORD.
05 G-LIB-FUNC PIC 9.
05 G-LIB-STATION-ID PIC X(18).
05 G-LIB-STATION-TYPE PIC 9.
05 G-LIB-SCREEN-ID PIC X(12).
05 G-LIB-DUPL-NO PIC S9(4).
05 G-LIB-DATA-SZ PIC 9(4).
05 G-LIB-ERRORS PIC X(1920).
***** USER RECORD BY STATION *****
01 G-USER-REC.
03 FILLER PIC X(6).
***** SCREEN CONTROL CHARACTERS *****
01 G-CLEAR-SCREEN.
03 G-FF-CHAR PIC S9(11) COMP VALUE 12.
01 G-CLEAR-SCREEN-2 REDEFINES G-CLEAR-SCREEN.
03 FILLER PIC X(5).
03 G-CLEAR-CHAR PIC X.
*****

```

```

.....
PROCEDURE DIVISION.
.....
G-COMPILE-DISPLAY SECTION.
G-DISPLAY.
    MOVE COMPILETIME(1)          TO G-TIME.
    DIVIDE G-TIME BY 216000
        GIVING G-TIME-HOUR
        REMAINDER G-TIME.
    COMPUTE G-TIME-MIN = G-TIME DIV 3600.
    MOVE COMPILETIME(15)        TO G-TIME-DATE.,
    DISPLAY G-TIME-COMP-MSG.
.....
G-RESTART-DMS SECTION.
G-RESTART-DMS-RESIDENT.
    OPEN UPDATE CICDB.
G-RECOVER-LOCK.
    LOCK LAST RESTART-SET
        AT RST-PROGRAM-COM = "RST-PROGRAM-COM"
        AND RST-PROGRAM-ID = "RST-PROGRAM-ID"
    ON EXCEPTION
        IF DMSTATUS (DEADLOCK) GO TO G-RECOVER-LOCK
        ELSE
            IF DMSTATUS (NOTFOUND) OR DMSTATUS (NORECORD)
                GO TO G-RECOVER-CREATE
            ELSE
                CALL DMTERMINATE.
    BEGIN-TRANSACTION NO-AUDIT RESTART-DSET
    ON EXCEPTION
        IF DMSTATUS (DEADLOCK) GO TO G-RECOVER-LOCK
        ELSE
            IF DMSTATUS (ABORT)
                DISPLAY "DMS ABORT DETECTED"
                GO TO G-RECOVER-LOCK
            ELSE
                CALL DMTERMINATE.
    DELETE RESTART-DSET.
G-RECOVER-DELETE-END.
    END-TRANSACTION AUDIT RESTART-DSET
    ON EXCEPTION
        IF DMSTATUS (ABORT)
            DISPLAY "DMS ABORT DETECTED"
            GO TO G-RECOVER-LOCK
        ELSE
            CALL DMTERMINATE.
G-RECOVER-CREATE.
    CREATE RESTART-DSET.
    MOVE "RST-PROGRAM-COM"      TO RST-PROGRAM-COM.
    MOVE "RST-PROGRAM-ID"      TO RST-PROGRAM-ID.
.....
G-BUILD-SCREEN-TABLE SECTION.
G-SCREEN-TABLE.
    OPEN INPUT SCREENFILE.
    MOVE 1                      TO G-SCREEN-KEY.
    READ SCREENFILE INTO G-SCREEN-INDEX
    INVALID KEY
        DISPLAY "SCREENFILE - INVALID KEY"

```

```

        SET MYSELF (STATUS) TO -1
        WAIT 5.
    IF G-SCREEN-NO 1 OR G-SCREEN-NO 100
        DISPLAY "INVALID SCREEN FILE"
        SET MYSELF(STATUS) TO -1
        WAIT 5.
.....
G-INITIALIZE-PROCESS SECTION.
.....
G-INITIALIZE.
    IF PARAMETROS(RESIDENT) = ZERO
        SET PARAMETROS(PROTECTION) TO VALUE(SAVE)
        OPEN OUTPUT PARAMETROS
        MOVE ZERO                TO G-LASTKEY-1
    ELSE
        SET PARAMETROS(FILETYPE) TO 7
        OPEN INPUT-OUTPUT PARAMETROS
        MOVE PARAMETROS(LASTRECORD) TO G-LASTKEY-1
        ADD 2 TO G-LASTKEY-1
        MOVE G-LASTKEY-1 TO G-MASKEY-1
        SEEK PARAMETROS.
    OPEN INPUT-OUTPUT REMOTEQUE.
    MOVE SPACES TO G-LIB-RECORD G-USER-REC.
    MOVE ZERO TO G-LIB-STATION-TYPE
                G-LIB-DUPL-NO
                G-LIB-DATA-SZ.
    MOVE "PO-1"                TO G-ID.
    MOVE 1 TO G-STATION.
    PERFORM G-READ-SCREENFILE THRU G-READ-SCREENFILE-EXIT.
    IF G-SCREEN-ERROR = 1
        MOVE "PO-1" TO G-LINE25-MISC-DATA
        MOVE " FIRST SCREEN NOT FOUND" TO G-LINE25-MISC-LIT
        PERFORM G-WRITE-LINE25 THRU G-WRITE-LINE25-EXIT
        GO TO G-FIRST-SCREEN-EXIT.
    MOVE 2 TO G-LIB-FUNC.
    MOVE "PO-1" TO G-LIB-SCREEN-ID.
G-FIRST-SCREEN.
    MOVE G-STATION TO G-REMOTE-KEY.
    MOVE 1 TO G-MSG-SIZE.
    MOVE G-CLEAR-CHAR TO G-REMOTE-REC.
    WRITE G-REMOTE-REC
        INVALID KEY
        GO TO G-FIRST-SCREEN-CONT.
    MOVE REMOTEQUE(TITLE,G-REMOTE-KEY) TO G-LIB-STATION-ID.
    CALL "COGENLIB OF GEN/LIBRARY"
    USING G-LIB-RECORD, G-INPUT-MSG, G-SCREEN-REC,
        G-USER-REC.
    MOVE G-SCREEN-SIZE TO G-MSG-SIZE.
    MOVE G-SCREEN TO G-REMOTE-REC.
    PERFORM G-WRITE-MESSAGE THRU G-WRITE-MSG-EXIT.
G-FIRST-SCREEN-CONT.
    IF G-STATION 1
        ADD 1 TO G-STATION
        GO TO G-FIRST-SCREEN.
G-FIRST-SCREEN-EXIT.
    MOVE ZERO                TO G-TRANS-COUNT.
    MOVE G-HOME-CHAR TO G-FORMS-HOME.

```



MOVE G-ESC-CHAR TO G-FORMS-ESC.  
MOVE G-ETX-CHAR TO G-FORMS-ETX.

.....  
G-TRANSACTION-PROCESSOR SECTION.  
.....

G-NEXT-TRANSACTION.

PERFORM G-READ-TRANSACTION THRU G-READ-TRANS-EXIT.  
IF G-INPUT-ASTERISK = "\*" "/" "?" OR ""  
    MOVE G-INPUT-SCREEN-ID TO G-ID  
    PERFORM G-READ-SCREENFILE THRU G-READ-SCREENFILE-EXIT  
    IF G-SCREEN-ERROR = ZERO  
        MOVE 2 TO G-LIB-FUNC  
        MOVE G-ID TO G-LIB-SCREEN-ID  
        CALL "COGENLIB OF GEN/LIBRARY"  
        USING G-LIB-RECORD, G-REMOTE-REC, G-SCREEN-REC,  
            G-USER-REC  
        MOVE G-SCREEN TO G-REMOTE-REC  
        MOVE G-SCREEN-SIZE TO G-MSG-SIZE  
        PERFORM G-WRITE-MESSAGE THRU G-WRITE-MSG-EXIT  
        GO TO G-NEXT-TRANSACTION  
    ELSE  
        MOVE G-ID TO G-LINE25-MISC-DATA  
        MOVE " UNKNOWN SCREEN ID" TO G-LINE25-MISC-LIT  
        PERFORM G-WRITE-LINE25 THRU G-WRITE-LINE25-EXIT  
        GO TO G-NEXT-TRANSACTION.  
MOVE G-LIB-SCREEN-ID TO G-SCREEN-ID.  
MOVE SPACES TO G-SCREEN-REC.  
ADD 1 TO G-TRANS-COUNT.  
MOVE ZERO TO G-PROCESSED-TRX.  
IF ( P1-1-OPCION=8 ) OR G-STOP-FLAG = 1  
    GO TO G-END-TRANSACTION-PROCESSOR.  
IF PO-1-IDPO-1="PO-1"  
    PERFORM G-PROCESS-TRANSACTION-1 THRU  
        G-EXIT-TRANSACTION-1  
    GO TO G-NEXT-TRANSACTION.  
IF P1-1-IDP1-1="P1-1"  
    PERFORM G-PROCESS-TRANSACTION-2 THRU  
        G-EXIT-TRANSACTION-2  
    GO TO G-NEXT-TRANSACTION.  
IF P2-2-IDP2-2="P2-2"  
    PERFORM G-PROCESS-TRANSACTION-3 THRU  
        G-EXIT-TRANSACTION-3  
    GO TO G-NEXT-TRANSACTION.  
IF P2-1-IDP2-1="P2-1"  
    PERFORM G-PROCESS-TRANSACTION-4 THRU  
        G-EXIT-TRANSACTION-4  
    GO TO G-NEXT-TRANSACTION.  
IF P3-11-IDP3-11="P3-11"  
    PERFORM G-PROCESS-TRANSACTION-5 THRU  
        G-EXIT-TRANSACTION-5  
    GO TO G-NEXT-TRANSACTION.  
IF P3-12-IDP3-12="P3-12"  
    PERFORM G-PROCESS-TRANSACTION-6 THRU  
        G-EXIT-TRANSACTION-6  
    GO TO G-NEXT-TRANSACTION.  
IF P3-13-IDP3-13="P3-13"

```

PERFORM G-PROCFGS-TRANSACTION-7 THRU
G-EXIT-TRANSACTION-7
GO TO G-NEXT-TRANSACTION.
IF P3-14-IDP3-14="P3-14"
PERFORM G-PROCESS-TRANSACTION-8 THRU
G-EXIT-TRANSACTION-8
GO TO G-NEXT-TRANSACTION.
IF P3-2-FUNCION="VER"
PERFORM G-PROCESS-INQUIRY-1 THRU
G-EXIT-INQUIRY-1
GO TO G-NEXT-TRANSACTION.
IF P3-2-IDP3-2="P3-2"
PERFORM G-PROCESS-TRANSACTION-9 THRU
G-EXIT-TRANSACTION-9
GO TO G-NEXT-TRANSACTION.
IF P3-11-IDP3-11="P3-31"
PERFORM G-PROCESS-TRANSACTION-10 THRU
G-EXIT-TRANSACTION-10
GO TO G-NEXT-TRANSACTION.
IF P3-12-IDP3-12="P3-32"
PERFORM G-PROCESS-TRANSACTION-11 THRU
G-EXIT-TRANSACTION-11
GO TO G-NEXT-TRANSACTION.
IF P3-13-IDP3-13="P3-33"
PERFORM G-PROCESS-TRANSACTION-12 THRU
G-EXIT-TRANSACTION-12
GO TO G-NEXT-TRANSACTION.
IF P3-14-IDP3-14="P3-34"
PERFORM G-PROCESS-TRANSACTION-13 THRU
G-EXIT-TRANSACTION-13
GO TO G-NEXT-TRANSACTION.
IF P3-11-FUNCION="CAM"
PERFORM G-PROCESS-INQUIRY-2 THRU
G-EXIT-INQUIRY-2
GO TO G-NEXT-TRANSACTION.
IF P3-11-IDP3-11="P3-41"
PERFORM G-PROCESS-TRANSACTION-14 THRU
G-EXIT-TRANSACTION-14
GO TO G-NEXT-TRANSACTION.
IF P3-12-FUNCION="CA1"
PERFORM G-PROCESS-INQUIRY-3 THRU
G-EXIT-INQUIRY-3
GO TO G-NEXT-TRANSACTION.
IF P3-12-IDP3-12="P3-42"
PERFORM G-PROCESS-TRANSACTION-15 THRU
G-EXIT-TRANSACTION-15
GO TO G-NEXT-TRANSACTION.
IF P3-13-FUNCION="CA2"
PERFORM G-PROCESS-INQUIRY-4 THRU
G-EXIT-INQUIRY-4
GO TO G-NEXT-TRANSACTION.
IF P3-13-IDP3-13="P3-43"
PERFORM G-PROCESS-TRANSACTION-16 THRU
G-EXIT-TRANSACTION-16
GO TO G-NEXT-TRANSACTION.
IF P3-14-FUNCION="CA3"
PERFORM G-PROCESS-INQUIRY-5 THRU

```

```

G-EXIT-INQUIRY-5
GO TO G-NEXT-TRANSACTION.
IF P3-14-IDP3-14="P3-44"
PERFORM G-PROCESS-TRANSACTION-17 THRU
G-EXIT-TRANSACTION-17
GO TO G-NEXT-TRANSACTION.
IF G-PROCESSED-TRX = ZERO
MOVE "UNKNOWN TRANSACTION" TO G-LINE25-MSG
PERFORM G-WRITE-LINE25 THRU G-WRITE-LINE25-EXIT.
IF G-TRANS-STATE-FLAG = 1
MOVE 4 TO G-MSG-SIZE
MOVE G-FORMS-MSG TO G-REMOTE-REC
PERFORM G-WRITE-MESSAGE THRU G-WRITE-MSG-EXIT.
GO TO G-NEXT-TRANSACTION.
*****
G-END-TRANSACTION-PROCESSOR SECTION.
*****
G-CLOSE-FILES.
CLOSE PARAMETROS LOCK.
STOP RUN.
$PAGE
*****
G-READ-TRANSACTION.
*****
MOVE 1920 TO G-MSG-SIZE.
MOVE ZERO TO G-REMOTE-KEY G-STATION G-EDIT-FLAG.
MOVE SPACES TO G-OUTPUT-MSG G-LIB-ERRORS.
IF G-READ-ERROR 25
DISPLAY "REMOTEQUE - INVALID KEY ON READ"
SET MYSELF (STATUS) TO -1
WAIT 5.
READ REMOTEQUE INTO G-INPUT-MSG
INVALID KEY
ADD 1 TO G-READ-ERROR
GO TO G-READ-TRANSACTION.
IF G-DMS-RESTART-SW = 1
DISPLAY "DMS RESTART PERFORMED"
MOVE ZERO TO G-DMS-RESTART-SW
MOVE "DMS RESTART PERFORMED" TO G-LINE25-MSG
PERFORM G-WRITE-LINE25 THRU G-WRITE-LINE25-EXIT
GO TO G-READ-TRANSACTION.
G-CLEAR-SCREEN-ROUTINE.
MOVE REMOTEQUE (STATE) TO G-STATE.
MOVE ZERO TO G-TEMP.
MOVE G-STATE TO G-TEMP |47:19:20|.
MOVE G-TEMP TO G-STATE.
G-CLEAR-MOVE-SPACES.
IF G-TEMP MOD 80 NOT EQUAL 0
ADD 1 TO G-TEMP
MOVE SPACES TO G-INPUT-CLEAR-CHAR (G-TEMP)
GO TO G-CLEAR-MOVE-SPACES.
COMPUTE G-TEMP = G-TEMP DIV 80.
G-CLEAR-MOVE-SPACES-LINES.
IF G-TEMP 24
ADD 1 TO G-TEMP
MOVE ALL SPACES TO G-INPUT-CLEAR-LINE (G-TEMP)
GO TO G-CLEAR-MOVE-SPACES-LINES.

```

```
MOVE 1 TO G-LIB-FUNC.  
MOVE REMOTEQUE(TITLE,G-REMOTE-KEY) TO G-LIB-STATION-ID  
G-STATION-ID.  
CALL "COGENLIB OF GEN/LIBRARY"  
USING G-LIB-RECORD, G-INPUT-MSG, G-SCREEN-REC,  
G-USER-REC.  
G-READ-TRANS-EXIT.  
EXIT.
```

\*\*\*\*\*

**ESPECIFICACIONES DE COGEN  
DEL SUBSISTEMA DBPROCESS**



```

002100 WORKRECORDS(NOM-PRO-REC)
002200 WORKRECORDS(CONTADOR-REC)
002300 IDENT(CIC-PRT-PROYECTOS-NACIONALES)
002400 DATABASE(CICDB,UNAM3)
        RESTART STRUCTURE 2 RESTART-DSET LOADED
        RESTART SET STRUCTURE 3 RESTART-SET LOADED
002500 MASTER(CIC/MASTER/PARAMETROS,PARAMETROS-REC)
002600 SEQUENTIAL(PARAMETROS)
002700%%%%%%%%%%
002800% FIN DE ESPECIFICACIONES GLOBALES %
002900%%%%%%%%%%
003000%%%%%%%%%%
003100%
003200% PROCEDURE FORMATEA FECHAS %
003300%
003400%%%%%%%%%%
003500 PROCEDURE(FORMATEA-FECHAS)
003600 MOVE(ANO-I,ANO-INI)
003700 ADD(1900,ANO-INI)
003800 MOVE(DIA-I,DIA-INI)
003900 SUBTRAN(MES-I=1)
004000 MOVE("ENERO ",MES-INI)
004100 ELSESUBTRAN
004200 SUBTRAN(MES-I=2)
004300 MOVE("FEBRERO ",MES-INI)
004400 ELSESUBTRAN
004500 SUBTRAN(MES-I=3)
004600 MOVE("MARZO ",MES-INI)
004700 ELSESUBTRAN
004800 SUBTRAN(MES-I=4)
004900 MOVE("ABRIL ",MES-INI)
005000 ELSESUBTRAN
005100 SUBTRAN(MES-I=5)
005200 MOVE("MAYO ",MES-INI)
005300 ELSESUBTRAN
005400 SUBTRAN(MES-I=6)
005500 MOVE("JUNIO ",MES-INI)
005600 ELSESUBTRAN
005700 SUBTRAN(MES-I=7)
005800 MOVE("JULIO ",MES-INI)
005900 ELSESUBTRAN
006000 SUBTRAN(MES-I=8)
006100 MOVE("AGOSTO ",MES-INI)
006200 ELSESUBTRAN
006300 SUBTRAN(MES-I=9)
006400 MOVE("SEPTIEMBRE",MES-INI)
006500 ELSESUBTRAN
006600 SUBTRAN(MES-I=10)
006700 MOVE("OCTUBRE ",MES-INI)
006800 ELSESUBTRAN
006900 SUBTRAN(MES-I=11)
007000 MOVE("NOVIEMBRE ",MES-INI)
007100 ELSESUBTRAN
007200 SUBTRAN(MES-I=12)
007300 MOVE("DICIEMBRE ",MES-INI)
007400 ENDSUBTRAN
007500 ENDSUBTRAN

```

```

007600         ENDSUBTRAN
007700         ENDSUBTRAN
007800         ENDSUBTRAN
007900         ENDSUBTRAN
008000         ENDSUBTRAN
008100         ENDSUBTRAN
008200         ENDSUBTRAN
008300         ENDSUBTRAN
008400         ENDSUBTRAN
008500         ENDSUBTRAN
008600         MOVE(ANO-F,ANO-FIN)
008700         ADD(1900,ANO-FIN)
008800         MOVE(DIA-F,DIA-FIN)
008900         SUBTRAN(MES-F=1)
009000         MOVE("ENERO      ",MES-FIN)
009100         ELSESUBTRAN
009200         SUBTRAN(MES-F=2)
009300         MOVE("FEBRERO    ",MES-FIN)
009400         ELSESUBTRAN
009500         SUBTRAN(MES-F=3)
009600         MOVE("MARZO      ",MES-FIN)
009700         ELSESUBTRAN
009800         SUBTRAN(MES-F=4)
009900         MOVE("ABRIL       ",MES-FIN)
010000         ELSESUBTRAN
010100         SUBTRAN(MES-F=5)
010200         MOVE("MAYO        ",MES-FIN)
010300         ELSESUBTRAN
010400         SUBTRAN(MES-F=6)
010500         MOVE("JUNIO       ",MES-FIN)
010600         ELSESUBTRAN
010700         SUBTRAN(MES-F=7)
010800         MOVE("JULIO        ",MES-FIN)
010900         ELSESUBTRAN
011000         SUBTRAN(MES-F=8)
011100         MOVE("AGOSTO       ",MES-FIN)
011200         ELSESUBTRAN
011300         SUBTRAN(MES-F=9)
011400         MOVE("SEPTIEMBRE",MES-FIN)
011500         ELSESUBTRAN
011600         SUBTRAN(MES-F=10)
011700         MOVE("OCTUBRE    ",MES-FIN)
011800         ELSESUBTRAN
011900         SUBTRAN(MES-F=11)
012000         MOVE("NOVIEMBRE ",MES-FIN)
012100         ELSESUBTRAN
012200         SUBTRAN(MES-F=12)
012300         MOVE("DICIEMBRE ",MES-FIN)
012400         ENDSUBTRAN
012500         ENDSUBTRAN
012600         ENDSUBTRAN
012700         ENDSUBTRAN
012800         ENDSUBTRAN
012900         ENDSUBTRAN
013000         ENDSUBTRAN
013100         ENDSUBTRAN
013200         ENDSUBTRAN

```



```

013300      ENDSUBTRAN
013400      ENDSUBTRAN
013500      ENDSUBTRAN
013600      MOVE(1,PLINE2)
013700      LOOP(PLINE1,1,59,1)
013800      SUBTRAN(LF-FDOS(PLINE1) NOT=" ")
013900      MOVE(LF-FDOS(PLINE1),LF-FTRES(PLINE2))
014000      ADD(1,PLINE2)
014100      ELSESUBTRAN
014200      SUBTRAN(LF-FDOS(PLINE1)=" " AND
014300      LF-FDOS(PLINE1 + 1) NOT=" ")
014400      MOVE(LF-FDOS(PLINE1),LF-FTRES(PLINE2))
014500      ADD(1,PLINE2)
014600      ENDSUBTRAN
014700      ENDSUBTRAN
014800      ENDOLOOP
014900      ENDPROCEDURE
015000%*****
015100%   FIN DEL PROCEDURE FORMATEA FECHAS %
015200%*****
015300%*****
015400%***** %
015500%   PROCEDURE FORMATEA FECHA %
015600%***** %
015700%*****
015800%   PROCEDURE(FORMATEA-FECHA)
015900%   MOVE(ANO-I,ANO-IN4)
016000%   ADD(1900,ANO-IN4)
016100%   MOVE(DIA-I,DIA-IN4)
016200%   MOVE(CIC-DUR-MES,DUR-MES)
016300%   SUBTRAN(MES-I=1)
016400%   MOVE("ENERO ",MES-IN4)
016500%   ELSESUBTRAN
016600%   SUBTRAN(MES-I=2)
016700%   MOVE("FEBRERO ",MES-IN4)
016800%   ELSESUBTRAN
016900%   SUBTRAN(MES-I=3)
017000%   MOVE("MARZO ",MES-IN4)
017100%   ELSESUBTRAN
017200%   SUBTRAN(MES-I=4)
017300%   MOVE("ABRIL ",MES-IN4)
017400%   ELSESUBTRAN
017500%   SUBTRAN(MES-I=5)
017600%   MOVE("MAYO ",MES-IN4)
017700%   ELSESUBTRAN
017800%   SUBTRAN(MES-I=6)
017900%   MOVE("JUNIO ",MES-IN4)
018000%   ELSESUBTRAN
018100%   SUBTRAN(MES-I=7)
018200%   MOVE("JULIO ",MES-IN4)
018300%   ELSESUBTRAN
018400%   SUBTRAN(MES-I=8)
018500%   MOVE("AGOSTO ",MES-IN4)
018600%   ELSESUBTRAN
018700%   SUBTRAN(MES-I=9)
018800%   MOVE("SEPTIEMBRE",MES-IN4)
018900%   ELSESUBTRAN

```

```

019000          SUBTRAN(MES-I=10)
019100          MOVE("OCTUBRE ",MES-IN4)
019200          ELSESUBTRAN
019300          SUBTRAN(MES-I=11)
019400          MOVE("NOVIEMBRE ",MES-IN4)
019500          ELSESUBTRAN
019600          SUBTRAN(MES-I=12)
019700          MOVE("DICIEMBRE ",MES-IN4)
019800          ENDSUBTRAN
019900          ENDSUBTRAN
020000          ENDSUBTRAN
020100          ENDSUBTRAN
020200          ENDSUBTRAN
020300          ENDSUBTRAN
020400          ENDSUBTRAN
020500          ENDSUBTRAN
020600          ENDSUBTRAN
020700          ENDSUBTRAN
020800          ENDSUBTRAN
020900          ENDSUBTRAN
021000          MOVE(1,PLINE2)
021100          LOOP(PLINE1,1,33,1)
021200          SUBTRAN(LF-FCUATRO(PLINE1) NOT=" ")
021300          MOVE(LF-FCUATRO(PLINE1),LF-FTRES(PLINE2))
021400          ADD(1,PLINE2)
021500          ELSESUBTRAN
021600          SUBTRAN(LF-FCUATRO(PLINE1)=" " AND
021700          LF-FCUATRO(PLINE1 + 1) NOT=" ")
021800          MOVE(LF-FCUATRO(PLINE1),LF-FTRES(PLINE2))
021900          ADD(1,PLINE2)
022000          ENDSUBTRAN
022100          ENDSUBTRAN
022200          ENDLOOP
022300          ENDPROCEDURE
022400%%%%%%%%%%
022500          FIN DEL PROCEDURE FORMATEA FECHA %
022600%%%%%%%%%%
022700%%%%%%%%%%
022800          %
022900          PROCEDURE PROCESO 1 %
023000          %
023100%%%%%%%%%%
023200          PROCEDURE(PROCESO1)
023300          MOVE(SPACES,LO1-LB)
023400          MOVE(CIC-CLAVE,LO2-CLAVE)
023500          RPTWRITE(CICFORMA1,LO2-REC,2)
023600          MOVE(CIC-NOM-PRO,NOM-PRO-FUNO)
023700          MOVE(SPACES,NOM-PRO-FDOS)
023800          %%%%%%%%%%%
023900          % %
024000          % FORMATEO DEL NOMBRE DEL PROYECTO %
024100          % %
024200          %%%%%%%%%%%
024300          MOVE(1,LS)
024400          LOOP(I,1,4,1)
024500          SUBTRAN(I = 3 OR I 3)
024600          MOVE(LS,LI)

```

```

024700 COMPUTE(LS,LI + 59)
024800 LOOP(LN-FUNO(LS) = " ")
024900 ADD(-1,LS)
025000 ENDLLOOP
025100 MOVE(1,IND)
025200 SUBTRAN(I = 1)
025300 LOOP(J,LI,LS - 1,1)
025400 MOVE(LN-FUNO(J),LN-FDOS(IND))
025500 ADD(1,IND)
025600 ENDLLOOP
025700 ELSESUBTRAN
025800 LOOP(J,LI + 1,LS - 1,1)
025900 MOVE(LN-FUNO(J),LN-FDOS(IND))
026000 ADD(1,IND)
026100 ENDLLOOP
026200 ENDSUBTRAN
026300 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
026400 % FORMATEO POR LINEA %
026500 % %
026600 % %
026700 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
026800 SUBTRAN((LN-FDOS(1) NOT=" ") AND
026900 (LN-FDOS(2) NOT=" ") AND
027000 (LN-FDOS(3) NOT=" "))
027100 MOVE(SPACES,NOM-PRO-FTRES)
027200 MOVE(NOM-PRO-FDOS,NOM-PRO-FTRES)
027300 MOVE(SPACES,NOM-PRO-FDOS)
027400 MOVE(60,LSFL)
027500 LOOP(LN-FTRES(LSFL) NOT=" ")
027600 ADD(-1,LSFL)
027700 ENDLLOOP
027800 SUBTRAN(LSFL=51 OR LSFL 51)
027900 MOVE(1,PLINE1)
028000 MOVE(1,PLINE2)
028100 MOVE(0,BLANCOS)
028200 LOOP(LN-FDOS(60)NOT=" ")
028300 SUBTRAN(PLINE1=(LSFL + 1))
028400 MOVE(SPACES,NOM-PRO-FTRES)
028500 MOVE(NOM-PRO-FDOS,NOM-PRO-FTRES)
028600 MOVE(SPACES,NOM-PRO-FDOS)
028700 MOVE(1,PLINE1)
028800 MOVE(1,PLINE2)
028900 ADD(BLANCOS,LSFL)
029000 MOVE(0,BLANCOS)
029100 ENDSUBTRAN
029200 SUBTRAN(LN-FTRES(PLINE1)=" ")
029300 MOVE(LN-FTRES(PLINE1),LN-FDOS(PLINE2))
029400 ADD(1,PLINE2)
029500 MOVE(" ",LN-FDOS(PLINE2))
029600 ADD(1,BLANCOS)
029700 ADD(1,PLINE1)
029800 ADD(1,PLINE2)
029900 MOVE(PLINE1,JFL)
030000 LOOP(IFL,PLINE2,60,1)
030100 MOVE(LN-FTRES(JFL),LN-FDOS(IFL))
030200 ADD(1,JFL)
030300 ENDLLOOP

```

```

030400         ELSESUBTRAN
030500             MOVE(LN-FTRES(PLINE1),LN-FDOS(PLINE2))
030600             ADD(1,PLINE1)
030700             ADD(1,PLINE2)
030800         ENDSUBTRAN
030900     ENDLOOP
031000 ELSESUBTRAN
031100     LOOP(IFL,1,51,1)
031200         MOVE(LN-FTRES(IFL),LN-FDOS(IFL))
031300     ENDLOOP
031400 ENDSUBTRAN
031500 ENDSUBTRAN
031600 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
031700 %     FIN DE FORMATEO POR LINEA
031800 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
031900 SUBTRAN(I = 1)
032000     MOVE(NOM-PRO-FDOS,LO3-NOM-PRO)
032100     MOVE(SPACES,NOM-PRO-FDOS)
032200     RPTWRITE(CICFORMA1,LO3-REC)
032300 ELSESUBTRAN
032400     SUBTRAN(I = 2)
032500         MOVE(NOM-PRO-FDOS,LO3-NOM-PRO)
032600         MOVE(SPACES,NOM-PRO-FDOS)
032700         RPTWRITE(CICFORMA1,LO3-REC)
032800 ELSESUBTRAN
032900     SUBTRAN(I = 3)
033000         MOVE(NOM-PRO-FDOS,LO3-NOM-PRO)
033100         MOVE(SPACES,NOM-PRO-FDOS)
033200         RPTWRITE(CICFORMA1,LO3-REC)
033300     ENDSUBTRAN
033400 ENDSUBTRAN
033500 ENDSUBTRAN
033600 ELSESUBTRAN
033700     SUBTRAN(I = 4)
033800         MOVE(LS,LI)
033900         LOOP(J,LI,180,1)
034000             MOVE(LN-FUNO(J),LN-FDOS(J))
034100         ENDLOOP
034200         MOVE(NOM-PRO-FDOS,LO3-NOM-PRO)
034300         MOVE(SPACES,NOM-PRO-FDOS)
034400     ENDSUBTRAN
034500 ENDSUBTRAN
034600 ENDLOOP
034700 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
034800 % FIN DE FORMATO DE NOMBRE DEL PROYECTO %
034900 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
035000 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
035100 %
035200 %     FORMATEO DEL NOMBRE DEL INVESTIGADOR %
035300 %
035400 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
035500 MOVE(SPACES,NOM-INV-FUNO)
035600 MOVE(SPACES,NOM-INV-FDOS)
035700 MOVE(ZEROS,CCAR1)
035800 MOVE(ZEROS,CCAR2)
035900 MOVE(ZEROS,ASTERISCO)
036000 MOVE(CIC-NOM-INV,NOM-INV-FUNO)

```

```

036100 LOOP(I,1,40,1)
036200     SUBTRAN(LET-FUNO(I)="*")
036300     ADD(1,ASTERISCO)
036400     SUBTRAN(ASTERISCO=2)
036500     COMPUTE(INDL1,CCAR1 + 3)
036600     LOOP(J,INDL1,40,1)
036700         SUBTRAN(LET-FUNO(J) =" " AND
036800             LET-FUNO(J + 1)=" " AND
036900             LET-FUNO(J + 2)=" ")
037000     ELSESUBTRAN
037100         ADD(1,CCAR2)
037200         MOVE(LET-FUNO(J),LET-FDOS(J - (CCAR1 + 2)))
037300     ENDSUBTRAN
037400     ENDLOOP
037500     COMPUTE(INDL2,CCAR2 + 2)
037600     LOOP(J,1,CCAR1 + 1,1)
037700         SUBTRAN(LET-FUNO(J) NOT="*")
037800         MOVE(LET-FUNO(J),LET-FDOS((CCAR2 + 1) + J))
037900     ELSESUBTRAN
038000         MOVE(" ",LET-FDOS((CCAR2 + 1) + J))
038100     ENDSUBTRAN
038200     ENDLOOP
038300     ENDSUBTRAN
038400     ELSESUBTRAN
038500         ADD(1,CCAR1)
038600     ENDSUBTRAN
038700     ENDLOOP
038800     MOVE(ZEROS,CCAR1)
038900     MOVE(SPACES,NOM-INV-FUNO)
039000     MOVE(NOM-INV-FDOS,NOM-INV-FUNO)
039100     MOVE(SPACES,NOM-INV-FDOS)
039200     %%%%%%%%%%%
039300     %
039400     %     CASE DEL GRADO ACADEMICO     %
039500     %
039600     %%%%%%%%%%%
039700     MOVE(CIC-GRA-ACA, GRA-ACA)
039800     SUBTRAN(GRA-ACA=1)
039900         MOVE("L",LET-FDOS(1))
040000         MOVE("I",LET-FDOS(2))
040100         MOVE("C",LET-FDOS(3))
040200         MOVE(".",LET-FDOS(4))
040300         MOVE(5,CCAR1)
040400     ELSESUBTRAN
040500         SUBTRAN(GRA-ACA=2)
040600             MOVE("M",LET-FDOS(1))
040700             MOVE(".",LET-FDOS(2))
040800             MOVE(3,CCAR1)
040900         ELSESUBTRAN
041000             SUBTRAN(GRA-ACA=3)
041100                 MOVE("D",LET-FDOS(1))
041200                 MOVE("R",LET-FDOS(2))
041300                 MOVE(".",LET-FDOS(3))
041400                 MOVE(4,CCAR1)
041500             ENDSUBTRAN
041600         ENDSUBTRAN
041700     ENDSUBTRAN

```

```

041800 %%%%%%%%%%%
041900 % FIN DEL CASE DEL GRADO ACADEMICO %
042000 %%%%%%%%%%%
042100 LOOP(I,1,40,1)
042200 MOVE(LET-FUNO(I),LET-FDOS(I + CCAR1))
042300 ENDLLOOP
042400 MOVE(NOM-INV-FDOS,LO4-NOM-INV)
042500 %%%%%%%%%%%
042600 % FIN DE FORMATEO DEL NOM. DEL INV. %
042700 %%%%%%%%%%%
042800 RPTWRITE(CICFORMA1,LO4-REC)
042900 %%%%%%%%%%%
043000 % %
043100 % FECHA DE FIRMA DEL CONVENIO %
043200 % %
043300 %%%%%%%%%%%
043400 SUBTRAN(PARAMETRO-OPCION=2)
043500 MOVE(SPACES,DET-FECHA-FTRES)
043600 MOVE(CIC-FEC-FIR-CON,FECHA-I)
043700 PERFORM(FORMATEA-FECHA)
043800 MOVE(DET-FECHA-FTRES,LO4-NOM-INV)
043900 RPTWRITE(CICFORMA1,LO4-REC)
044000 ENDSUBTRAN
044100 %%%%%%%%%%%
044200 % FIN DE FECHA DE FIRMA %
044300 %%%%%%%%%%%
044400 MOVE(CIC-DIR-ADJ,DIR-ADJ)
044500 SUBTRAN(DIR-ADJ = 1)
044600 MOVE("DESARROLLO CIENTIFICO " ,LO5-DIR-ADJ)
044700 ELSESUBTRAN
044800 SUBTRAN(DIR-ADJ = 2)
044900 MOVE("DESARROLLO TECNOLOGICO " ,LO5-DIR-ADJ)
045000 ELSESUBTRAN
045100 SUBTRAN(DIR-ADJ = 3)
045200 MOVE("FORMACION DE RECURSOS HUMANOS",LO5-DIR-ADJ)
045300 ENDSUBTRAN
045400 ENDSUBTRAN
045500 ENDSUBTRAN
045600 MOVE(CIC-TOT-SOL-CYT,LO5-TOT-SOL-CYT)
045700 ADD(CIC-TOT-SOL-CYT,TOTAL-SOL-CYT)
045800 RPTWRITE(CICFORMA1,LO5-REC)
045900 RPTWRITE(CICFORMA1,L15-REC)
046000 ENDPROCEDURE
046100 %%%%%%%%%%%
046200 % FIN DEL PROCEDURE PROCESO 1 %
046300 %%%%%%%%%%%
046400 READLOOP(PARAMETROS)
046500 MOVE(ZEROS,L16-TOTAL-SOL-CYT)
046600 MOVE(ZEROS,TOTAL-SOL-CYT)
046700 %%%%%%%%%%%
046800 % %
046900 % IMPRESION DE ENCABEZADOS %
047000 % %
047100 %%%%%%%%%%%
047200 MOVE(SPACES,L13-DEP-UNI)
047300 MOVE(SPACES,LO1-LB)
047400 MOVE(ZEROS,DEP-UNI-FUNO)

```

```

047500 RPTWRITE(CICFORMA1,LO6-REC,CHANNEL 1)
047600 RPTWRITE(CICFORMA1,LO7-REC,,2)
047700 RPTWRITE(CICFORMA1,LO8-REC)
047800 RPTWRITE(CICFORMA1,LO9-REC)
047900 RPTWRITE(CICFORMA1,L10-REC,2,1)
048000 RPTWRITE(CICFORMA1,L11-REC)
048100 RPTWRITE(CICFORMA1,LO1-REC)
048200 RPTWRITE(CICFORMA1,L12-REC,,4)
048300 MOVE(SPACE,LO2-CLAVE)
048400 SUBTRAN(PARAMETRO-OPCION=1 OR
048500 PARAMETRO-OPCION=3)
048600 MOVE("INICIATIVAS ",LO2-CLAVE)
048700 ELSESUBTRAN
048800 SUBTRAN(PARAMETRO-OPCION=2 OR
048900 PARAMETRO-OPCION=4)
049000 MOVE("PROYECTOS ",LO2-CLAVE)
049100 ENDSUBTRAN
049200 ENDSUBTRAN
049300 RPTWRITE(CICFORMA1,LO2-REC,,1)
049400 MOVE(SPACES,LO2-CLAVE)
049500 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
049600 % FIN DE IMPRESION DE ENCABEZADOS %
049700 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
049800 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
049900 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
050000 % DEPENDENCIA UNIV. SOLICITANTE %
050100 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
050200 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
050300 MOVE(PARAMETRO-DEP-UNI,DEP-UNI-FUNO)
050400 SUBTRAN(DOS-DIG=21)
050500 ELSESUBTRAN
050600 SUBTRAN(DOS-DIG=22)
050700 ELSESUBTRAN
050800 SUBTRAN(DOS-DIG=31)
050900 ELSESUBTRAN
051000 SUBTRAN(DOS-DIG=32)
051100 SUBTRAN(TRES-DIG=500)
051200 MOVE("INSTITUTO DE CIENCIAS DEL MAR Y LIMNOLOGIA"
051300 L13-DEP-UNI)
051400 ENDSUBTRAN
051500 ELSESUBTRAN
051600 SUBTRAN(DOS-DIG=41)
051700 ELSESUBTRAN
051800 SUBTRAN(DOS-DIG=42)
051900 ELSESUBTRAN
052000 SUBTRAN(DOS-DIG=44)
052100 ENDSUBTRAN
052200 ENDSUBTRAN
052300 ENDSUBTRAN
052400 ENDSUBTRAN
052500 ENDSUBTRAN
052600 ENDSUBTRAN
052700 ENDSUBTRAN
052800 RPTWRITE(CICFORMA1,L13-REC,,1)
052900 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
053000 % FIN DE DEP. UNIV. SOLICITANTE %
053100 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

053200 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
053300 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
053400 %   FORMATEO DEL PERIODO DE CONSULTA   %
053500 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
053600 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
053700 MOVE (SPACES,DET-FECHA-FTRES)
053800 MOVE (PARAMETRO-LIPER,FECHA-I)
053900 MOVE (PARAMETRO-LSPER,FECHA-F)
054000 PERFORM (FORMATEA-FECHAS)
054100 MOVE (DET-FECHA-FTRES,L14-PER)
054200 RPTWRITE (CICFORMA1,L14-REC)
054300 RPTWRITE (CICFORMA1,L15-REC)
054400 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
054500 %   FIN DEL FORMATEO DEL PERIODO   %
054600 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
054700 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
054800 %   INICIATIVAS   %
054900 %   OPCION=1   %
055000 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
055100 SUBTRAN (PARAMETRO-OPCION=1)
055200 DMSREADLOOP (CIC-INS-PAR-SET)
STRUCTURE 4 CIC-DSET LOADED
STRUCTURE 7 CIC-INS-PAR-SET LOADED
055300 SELECT (CIC-DEP-UNI=PARAMETRO-DEP-UNI,NEXT)
055400 SELECT (CIC-RES-EVA NOT=1,NEXT)
055500 SELECT ((CIC-FEC-PRE-INI PARAMETRO-LIPER OR
055600 CIC-FEC-PRE-INI = PARAMETRO-LIPER) AND
055700 (CIC-FEC-PRE-INI PARAMETRO-LSPER OR
055800 CIC-FEC-PRE-INI = PARAMETRO-LSPER),NEXT)
055900 PERFORM (PROCESO1)
056000 ENDLLOOP
056100 ENDSUBTRAN
056200 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
056300 %   FIN DE INICIATIVAS   %
056400 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
056500 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
056600 %   PROYECTOS   %
056700 %   OPCION=2   %
056800 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
056900 SUBTRAN (PARAMETRO-OPCION=2)
057000 DMSREADLOOP (CIC-INS-PAR-SET)
057100 SELECT (CIC-DEP-UNI=PARAMETRO-DEP-UNI,NEXT)
057200 SELECT (CIC-RES-EVA =1,NEXT)
057300 SELECT ((CIC-FEC-PRE-INI PARAMETRO-LIPER OR
057400 CIC-FEC-PRE-INI = PARAMETRO-LIPER) AND
057500 (CIC-FEC-PRE-INI PARAMETRO-LSPER OR
057600 CIC-FEC-PRE-INI = PARAMETRO-LSPER),NEXT)
057700 PERFORM (PROCESO1)
057800 ENDLLOOP
057900 ENDSUBTRAN
058000 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
058100 %   FIN DE PROYECTOS   %
058200 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
058300 MOVE (TOTAL-SOL-CYT,L16-TOTAL-SOL-CYT)
058400 RPTWRITE (CICFORMA1,L15-REC,,1)
058500 RPTWRITE (CICFORMA1,L16-REC)
058600 ENDLLOOP

```



058700 INITIALIZE  
058800 GENERATE(CIC/RPT/FTE,CIC/RPT/OBJ,CIC/RPT/WFL,,2)  
START DATA BASE PROCESSING GENERATION  
GENERATING DATA DIVISION  
GENERATING PROCEDURE DIVISION  
DATA BASE PROCESSING GENERATION COMPLETE  
NO COGEN ERRORS ENCOUNTERED

EJEMPLO DE LOS REPORTES FINALES  
GENERADOS POR EL SISTEMA PARA  
INICIATIVAS Y CONVENIOS

UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

COORDINACION DE LA INVESTIGACION  
CIENTIFICA

SECRETARIA TECNICA

PROGRAMA UNAM-CONACYT  
PROYECTOS NACIONALES

INICIATIVAS

INSTITUTO DE CIENCIAS DEL MAR Y LIMNOLOGIA  
PERIODO: DEL 01 DE MARZO DE 1984 AL 01 DE ABRIL DE 1985

---

ICECXNAO21923  
BIOLOGIA Y ECOLOGIA DE FASES JUVENILES DEL LANGOSTINO MACHO  
BRACHIUM ACANTHURUS EN AREAS ADYACENTES A LA LAGUNA DE  
TERMINOS, CAMPECHE  
M. RAMIRO ROHAN CONTRERAS  
DIRECCION ADJUNTA DE DESARROLLO CIENTIFICO 5327

---

ICECXNAO21927  
BIOLOGIA Y DINAMICA POBLACIONAL DE LAS LANGOSTAS DEL CARIBE  
MEXICANO  
M. ENRIQUE LOZANO A.  
DIRECCION ADJUNTA DE DESARROLLO CIENTIFICO 21299

---

ICECXNAO22064  
INFLUENCIA DEL CLIMA OCEANICO EN LA BIOLOGIA Y ECOLOGIA DEL  
CALAMAR "JUMBO" DOSIDICUS GIGAS EN EL GOLFO DE CALIFORNIA  
DR. ANTONIO GARCIA CUBAS GUTIERREZ  
DIRECCION ADJUNTA DE DESARROLLO CIENTIFICO 18918

---

ICECXNAO21929  
ESTUDIO Y CARTOGRAFIA REGIONALES E INTEGRADCS DE LA  
LITOSFERA EN MEXICO.  
M. JOSE HECTOR SANDOVAL OCHOA  
DIRECCION ADJUNTA DE DESARROLLO CIENTIFICO 9840

---

CANTIDAD TOTAL SOLICITADA A CONACYT: 55384

UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

COORDINACION DE LA INVESTIGACION  
CIENTIFICA

SECRETARIA TECNICA

PROGRAMA UNAM-CONACYT  
PROYECTOS NACIONALES

CONVENIOS

INSTITUTO DE CIENCIAS DEL MAR Y LIMNOLOGIA  
PERIODO: DEL 01 DE MARZO DE 1984 AL 01 DE ABRIL DE 1985

---

PCCBBNAO21928  
ESTUDIO REGIONAL DE LOS ARRECIFES CORALINOS DEL MAR CARIBE  
MEXICANO: SU POTENCIAL DE USO

DR. ERIC JORDAN DAHLGREN  
14 DE SEPTIEMBRE DE 1984/24 MESES  
DIRECCION ADJUNTA DE DESARROLLO CIENTIFICO 14909

---

PCCBXNAO21434  
PROSPECCION CITOGENETICA Y ELECTROFORETICA DE PECES DE  
IMPORTANCIA ECONOMICA Y ALIMENTARIA.

DR. MANUEL URIBE ALCOCER  
20 DE OCTUBRE DE 1984/12 MESES  
DIRECCION ADJUNTA DE DESARROLLO CIENTIFICO 2000

---

CANTIDAD TOTAL COMPROMETIDA POR CONACYT: 16909

**RESULTADOS**  
**Y**  
**CONCLUSIONES**

## RESULTADOS

Al finalizar el período asignado al proyecto, se tiene un prototipo operable de la aplicación. El lenguaje de Cuarta Generación (COGEN) utilizado para realizar el proyecto contribuyó a la obtención de éste resultado al disminuir el esfuerzo y tiempo necesarios en la etapa de programación, permitiendo realizar una aplicación con cierto grado de complejidad al utilizarse un Sistema Manejador de Bases de Datos (DMS-II) y a la posibilidad de utilizar un manejador de video.

La experiencia previa en el manejo de COGEN y DMS-II por parte de los miembros del equipo de desarrollo era mínima. Sin embargo, al finalizarse el desarrollo de la aplicación se obtuvo un conocimiento aceptable en su uso.

Los resultados obtenidos al término del desarrollo de la aplicación fueron los siguientes:

- 1) Se obtuvo una aplicación operable dentro del período asignado.
- 2) Una aplicación documentada.
- 3) Una aplicación fácilmente modificable.
- 4) Una estandarización de la aplicación.
- 5) Se ganó experiencia en la planeación y diseño de aplicaciones con sistemas de soporte lógico de Cuarta Generación.
- 6) Se evaluó COGEN.

Las líneas en COBOL generadas por especificación fueron en promedio las siguientes:

Total de especificaciones COGEN = 1040 líneas.

Total de líneas COBOL generadas = 6414.

Líneas generadas por especificación = 6 líneas.

## CONCLUSIONES

El adoptar una metodología para el desarrollo de aplicaciones es necesario para realizar aplicaciones coherentes, completas, operables y mantenibles, dentro de los límites de tiempo y costos razonables que permitan satisfacer las necesidades de proceso de cómputo.

Muchas de las funciones comprendidas en una metodología requieren de experiencia previa. Una gran parte de las estimaciones necesarias en la fase de planeación se realizan sobre la base de experiencias anteriores, tales como la estimación del número de líneas de código, la estimación del costo, los recursos a utilizar, etc.

Por lo anterior expuesto, presenta cierta dificultad el tratar de aplicar por vez primera una metodología ya establecida. Además debe tenerse el criterio suficiente (basado en experiencias pasadas) para decidir que metodología y con que restricciones se debe aplicar al desarrollar una aplicación, debido a que no todas las metodologías son aplicables en todas las áreas ya que dependen mucho de la magnitud de la aplicación y del esfuerzo a realizar.

La metodología de desarrollo adoptada en el presente trabajo, se aplicó en sus bases fundamentales, y se espera que conforme se adquiriera mayor experiencia en el desarrollo de sistemas se logrará un mayor provecho de su aplicación.

**APENDICES**



APENDICE A

En este apéndice se presenta una lista de los Sistemas Manejadores de Bases de Datos más importantes, así como sus características más sobresalientes, también se indica el año en que se desarrollaron, así como la empresa que los realizó.

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
ACCENT R	1981	National Information Syst, Cupertino CA	DEC10/20	Com DBMS slc rel stq sch
ADABAS	1971	software ag Darmstadt FRG	IBM360/370 Siemens	Com DBMS hlc stq sch drf cip pri rec Atra <sup>60</sup>
ADEPT	1969	Systems Dev Corp Santa Monica CA	IBM360-50	Exp DBMS slc nlq(CONVERSE) pri Weissman <sup>60</sup>
ADMINS	1966	MIT&ADMINS inc Cambridge MA	DEC 11, VAX	DBMS slc sch rel trf McIntosh <sup>60</sup>
ALPHA	1971	IBM Research San Jose CA		Pro DBL slc rel Codd in Codd <sup>71</sup>
AMBASE	1979	Amcort Computer Corp Louisville KY	DEC 11 (RSTS)	COM DBMS sch lsf
AMIGOS	1970	Com ress Inc Rockville MD	IBM360/370	Com FMS hlc lsf
APPEL IV	1974	SIS Paris France	IBM360/370 Burr 500/700	Com DBMS hlc hie drf lsf
ASI/INQ	1975	Applications Softw. Torrance CA	IBM360/370	Com QUS para DL1 slc hie stq
ASI/ST	1969	Applications Softw. Torrance CA	IBM360/370 Univac70	Com QUS slc rpg tbq sch para sqf, lsf, IMS, TOTAL
Ass.PL	1967	General Motors Warren MI	IBM360-87	Inst DBMS hlp(PUI) rnf gra Dodd <sup>60</sup>
AUTONOTE	1969	Univ of Michigan Ann Arbor MI	IBM360/370 (MTS)	Exp SATDBMS slc txt sch sqf drf Reitman <sup>60</sup>
BASIS	1970	Estelle Mem Labs Columbus OH	CDC6400 DEC 10/20, VAX IBM 870 UNIVAC1100	Inst IRS stq bib rpg Fried en Walker <sup>71</sup>
BEAST	1968	Brookings Inst Washington DC	DEC PDP10	Inst SATDBMS slc sch sqf Kidd <sup>60</sup>

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
BIS	1967	Am Tel & Tel New York NY	IBM360	Inst DBMS hlc hie sch Benner <sup>67</sup>
CAFS	1976	ICL Stevenage UK		Dev DBCMP real Babb <sup>70</sup>
CASSM	1975	Univ. of Florida Gainesville FL		Exp DBCMP Su <sup>79</sup> Hawthorn <sup>82</sup>
CDMS	1969	System Dev Corp Santa Monica CA	IBM360/370	Com DBMS service slc lrc lxf véase TDMS
CDMS	1974	Digital Eq Corp Maynard MA	DEC 11	Com DBMS slc hie trf véase MUMPS
CFS	1960	Carnegie-Mellon U. Pittsburgh PA	DEC LSI-11e	Exp DFMS
CIA	1982	Computer Invest. Adv. Sewickley PA	Apple	Com DBMS 1-rel alg hlc (BASIC) rpg sch lxf
COGENT	1969	Comp Sciences Corp Los Angeles CA	IBM7090, 370 Univac1100	Com DPG hlp(cosol) sch hie lxf
CONVERSE	1967	System Dev Corp Santa Monica CA	IBM360-67 ANFSQ32	Exq QUS net nlq vrf Kallogg <sup>66</sup> , en SIGIR <sup>71</sup>
COSTAR	1978	Mass Gen Hospital Boston MA	DEC PDP 15,11 Tandem	SATBMS slc trf(MUMPS) Barnett <sup>70</sup>
CREATE	1975	Complete Computer Sys, Horsham PA	DataGeneral	Com DPG lxf pri
CREATE/3000	1977	CRI Inc Mountain View CA	HP 3000	Com DBMS hlc rel stq lxf
CZAR	1970	Crown Zellerbach San Francisco CA	IBM360/370	Inst QUS hlc sch lxf Palmer <sup>75</sup>
DATA ANALYZER	1971	Program Prod. Nanuet NY	IBM360/370	Com IRS slc rpg sch sqf lxf

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
DATACATALOG	1974	Synergetics Bedford MA	IBM370 OS,DOS UNIVAC	Com DDICT s/c rpg sch lsf DMS1100 IDMS IMS S2000
DATACOM	1970	Applied Data Res. Dallas TX	IBM360/370	Com DBMS h/c sch sqf ixf cpr
DATA COMPUTER	1971	Comp Corp of Am Cambridge MA	PDP10 (TENEX)	Dev DBMS h/c h/c stq vrf para ARPAnet
DATAMAN	1975	Dataman Ltd Calgary Alberta	IBM360/370	Com FMS s/c rpg sch sqf
DATAMANAGER	1978	MSP London UK & Lexington MA	IBM360/370	Com DDICT s/c rpg sch ADABAS IMS MARKIV SYSTEM 2000 TOTAL
DATAMASTER	1980	Microsoft Seattle WA	Apple 8080	Com FMS rpg sch sqf
DATASAB	1974	Saab-Scania AB Linkoping Sweden	SAAB D22/D23	Com DBMS hlp(COBOL) net sch rnf pri Bubenko <sup>76</sup>
dBASE II	1981	Ashon-Tate Culver City CA	Z-80 CP/M	Com FMS, Join stq rpg ixf(1 actualizado)
DBC	1978	Ohio State&UNIVAC Columbus OH		Dev DBCMP sch ixf Banerjee <sup>79</sup> Hawthorn <sup>82</sup>
DBMS	1977	Prime Computer Inc Wellesley Hills MA	Prime	Com DBMS hlp net sch stq(IQL) rnf rec pri
DBMS10/20	1973	Digital Eq Corp Marlboro MA	DEC 10/20	Com DBMS h/c net(1973) sch stq(IQL) rnf rec pri
DBMS11	1979	Digital Eq Corp Marlboro MA	DEC 11	Com DBMS h/c net(1973) sch rnf
DBMS990	1980	Texas Instruments Austin TX	T1990	Com FMS h/c(PASCAL CO- BOL FORTRAN) h/c-tbq lsf

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
DBMS1900	1974	ICL London England	ICL 1903	Com DBMS hlc sch lsf pri rec
DBOMP	1970	IBM White Plains NY	IBM360/370	Com BOMP net stq DL/I files; CVMS para lsf
DBS90	1972	Sperry Rand GmbH Frankfurt a/M FRG	Univac90	Com BOMP hlc (COBOL) net
DB/DC	1975	IBM White Plains NY	IBM360/370 IMS	Com DDICT slc sch rpg (CAMS para manufactura)
DIALOG	1967	Lockheed Res Corp Palo Alto CA	IBM360	Com IRS servicio slq irq hlc bib Walker <sup>71</sup>
DIRECT	1977	Northwestern Univ Evanston IL		Pro DBCMP DeWitt <sup>70</sup> Hawthorn <sup>62</sup>
DISAM	1975	Four Phase Systems Cupertino CA	4phase70	Com FS para DDBMS hlp(COBOL) lsf lxf
DL/I	1968	IBM White Plains NY	IBM360/370	Com FMS hlc sch hlc sqf lsf drf stq (CICS)
DMI, 5 también SC-1	1968	Auerbach Philadelphia PA	Univac418 IBM360/370	Com DBMS slc hlc rpg sch lsf CODASYL <sup>71A</sup>
DMS1100, 90	1971	Univac Minneapolis MN	Univac1100 Univac90	Com DBMS hlp (COBOL) net(1969) rnf rec vie
DMS170	1977	Control Data Corp Minneapolis MN	Cyber170	Com DBMS hlc sch sqf obtenido de (MARS) lsf drf
DMS/1700	1975	Dedicated Systems Chicago IL	Burr. 1700	Com FMS lxf
DMS II	1972	Burroughs Pasadena CA	B1700 a 7700	Com DBMS hlp(COBOL, ALGOL) net sch sqf lsf lxf cpr rec

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
DMS IV basada en IDS	1972	Honeywell Inf Sys Phoenix AZ	H60	Com DBMS hlp(COBOL) net(1973) rpg rnf rec
DPL también IPL	1976	National Information Syst, Cupertino CA	DEC10/20	Com DBMS hlc(FORTRAN COBOL) sch lsf pri
DYL260	1971	Dylakor Comp Syst Encino CA	IBM360/370	Com FMS slc rpg stq sqf lsf
EDEN	1982	Univ of Washington Seattle WA	DEC VAX	Exp DFMS obj Jessop en Wiederhold <sup>62</sup>
EDMS	1969	Control Data Corp Brussels Belgium	CDC6400 Cyber	Com DBMS hlc(FORTRAN) stqsch(ANSII) ls Nijssen <sup>7</sup>
FACETS was PRISM	1981	Synergetics Bedford MA	IBM 370	Com DDICT slc rpg
FOCUS		Information Builders New York 10001 NY	IBM 370 CMS, TSO	Com QUS irq rpg lsf, DL/1, IDMS pri
FORDATA	1974	CSIRO Canberra Australia	CDC CYBER76	Inst DBMS hlp(FORTRAN) net sch
FORIMS	1970	Nippon Univac Tokyo Japan	Univac1100	Dev DBMS hlc(FORTRAN) net lsf rec
FORTE	1959	Burroughs Corp Paoli PA	B2500/3500 B1700-7700	Com FMS hlc(COBOL) sqf lsf drf lxf rnf Chapin <sup>60</sup>
FRAMIS	1977	Lawrence Liv.Lab. Livermore CA	CDC 7600 CRAY VEC VAX(VMS)	Dev DBMS rel alg stq rnf(CODASYL)
GIM	1967	TRW Systems Redondo Beach CA	IBM7094,360/ 370 Univac1100 Honeywell6000	Com QUS slc stq sch rec drf rng Nelson <sup>67</sup>
GIS	1966	IBM White Plains NY	IBM360/370	Com QUS hlc(COBOL, PL/I) hlc stq sch sqf lsf

42

Nombre	Año Lugar de desarrollo	Computadora	Tipo y característica
GMIS	1975 MIT Sloan School & IBM, Cambridge MA	IBM370 (XRM)	Dev SATDBMS(Dec.supp) rel vrf Donovan <sup>76</sup>
IDBP 86/440	1982 Intel-MRI Austin TX	link to IEEE 488, ETHERNET	Com DBCMP rel pri rec
HOPS	1975 Technicon Haifa Israel	Burroughs 128	Exp DBMS hie trf Reiter in Kerr <sup>76</sup>
IDM 500	1981 Britton-Lee Los Gatos CA	VAX on IEEE-488 o intlgnt term.	Com DBCMP rel inf,xf pri
IDMS	1972 Cullinane Corp Westwood MA	IBM360/370 ICL1902 Univac70,90 Siemens4004	Com DBMS hlp(COBOL) nlq(ROBOT) rpg (CULPRIT) net(1973+) sch DDICT drf rec
IDS I, II	1962 Honeywell Inf Sys Phoenix AZ	H200, H60, H6000	Com DBMS hlp(COBOL) net(73) rnf rec Bachman <sup>80</sup>
IDP EDMD	1978 Honeywell was XDS Los Angeles 45 CA	H88 Sigma 6,7,9	Com DBMS hlc(COBOL) net rnf lxf sch rec
IFIP también RIM	1978 Boeing Computer Co (IPAD) Seattle WA	DEC VAX IBM	SATDBMS para CAD/CAM net(1978) hlp(FORTRAN)
IMAGE	1974 Hewlett-Packard Santa Clara CA	HP3000, HP2100	Com DBMS hlc sch pri stq net(2 niveles: drf, rnf)
IMARS	1971 Computeria Inc Braintree MA	DEC PDP10	Com QUS sic stq rpg rec
IMS-2VS	1968 IBM White Plains NY	IBM360/370	Com DBMS hlc mu/d-hie sch(DL/1) stq rec-VS)
INFOS	1975 Data General Southboro MA	DG Nova, Eclipse	Com FMS hlc hie lxf lxf stq
INGRES	1973 Un.of CA & Relational Technology, Berkeley	DEC 11, VAX (UNIX, VMS)	Dev DBMS sic hlp(C, es.) rel stq gra pri Held <sup>79</sup>

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y características
INQUIRE	1969	Infodata Systems Falls Church VA	IBM360/370	Com IRS hlc rpg stq ech pri; Dev DDBMS(IQNET)
INTREX	1966	Mass Inst. Tech Cambridge MA	IBM7094, IBM360	Inst IRS irq/stq bib lxf Walker <sup>71</sup>
ISI/ <i>después</i> PRTV	1971	IBM UK Research Peterlee UK	IBM360/370	Dev DBMS hip(PL/I) rel alg vie com stq Todd <sup>70</sup>
ISAM70	1974	Software70 Anaheim CA	Cualquier sistema FORTRAN	Com FS hlc(FORTRAN) lxf
LADDER	1977	SRI International Menlo Park CA	DEC PDP10	Exp IRS niq net(DBMS20) Hendrix <sup>78</sup>
LEADER	1967	Lehigh Univ Bethlehem PA	CDC6400	Inst IRS irq bib lxf Hillman <sup>69</sup>
LEXICON	1976	Arthur Anderson Chicago IL	IBM360/370, System 8	Com DDICT alc IDMS IMS TOTAL
LEXIS	1975	Mead Data Central Lexia: New York NY	IBM370	Com IRS <i>servicio alc legal,</i> <i>bases económicas de datos</i>
LUNAR	1972	Bolt Beranek Newman Cambridge MA	DEC PDP10 TENEX	Inst IRS niq sqf lxf Woods <sup>78</sup>
MADAM	1970	MIT MacAIMS Proj. Cambridge MA	H6000 (MULTICS)	Dev DBMS <i>primero</i> rel hip (PL/I) stq rpg vrf Strnad <sup>71</sup>
MAGNUM	1975	Tymshare Cupertino CA	DEC PDP10	Com DBMS alc rel stq ech
MARKIV	1967	Informatics Canoga Park CA	IBM360/370 Univac900	COM FMS alc rpg tbq hie lxf CODASYL <sup>71A</sup>
MARS	1969	Control Data Corp Sunnyvale CA	CDC6400	Com DBMS hie ech hlc(FORTRAN) sqf lxf

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
MDBS	1980	Micro Data Base Syst Lafayette IN	Z80,8080-sistemas basados en CP/M	Com DBMS alc stq net sch
MEDLARS también ELHILL	1983	National Lib Med Bethesda MD	IBM360	Inst SADBMS alc irq sqf lsf bib Kattar <sup>76</sup> , también ORBIT
MICRO-SEED	1980	Microsoft Bellevue WA	Sistemas basados Z80,8080	Com DBMS hlc(FORTRAN) net sch basado en SEED
MODEL204	1972	Comp Corp of Am Cambridge MA	IBM360/370	Com DBMS hlc stq sch plf lxf(IFAM)
MORIS	1972	Polytechnico Milano Italy		Pro DBMS rel cal sch stq(COLARD) Bracchi en Klimble <sup>76</sup>
MUMPS	1966	Mass Gen Hospital Boston MA	DEC PDP15,11 8080 eo.	Com FMS alc hie trf Greene <sup>69</sup>
MRDS basado en RDMS *	1978	Honeywell Inf Sys Minneapolis MN	H6000 L68 (MULTICS)	Com DBMS hlp(PL/I) rel stq(LINUS) rpg vrf
NOMAD	1976	National CSS Norwalk CN	IBM370-CMS	Com DBMS alc, hlc hie stq sch rpg
NYTIS	1970	New York Times New York NY	IBM360	Inst IRS alc txf lxf Baker <sup>72</sup>
OASIS	1971	Stanford Adm DP Stanford CA	IBM360/370	Inst SATDBMS hlc(COBOL) hie irq lsf
ORACLE	1979	Relational Software Inc., Menlo Park CA	DEC VAX, IBM VM MVS DOS	DBMS hlc(COBOL, PL/I) cl rel stq lxf cpr pri vls
OSIRIS	1973	Survey Res.Ctr., Univ. of Michigan, Ann Arbor	IBM360/370 rpg Rattenbury <sup>74</sup>	Inst SATDBMS sch sqf lsf
PLUS/4	1979	Century Analysis Pacheco CA	NCR 101 etc.	Com FMS hlc



Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
POLYPHEME (SIRIUS)	1977	I. N. Polytechnique Grenoble France	CII & IBM	Exp DBMS pri LeBihan <sup>66</sup>
RAMIS	1967	Mathematics Princeton NJ	IBM360/370	Com DBMS hie stq rpg
RAP	1976	Univ of Toronto Toronto Canada		Exp DBCMP rel Oskarahan <sup>77</sup> Hawthorn <sup>88</sup>
RAPPORT	1978	Brit.Min.Def. & LOGICA, New York NY	Cualquier sistema basado en FORTRAN	Com DBMS hie stq rpg rec
RDF	1967	Rand Corp Santa Monica CA	IBM360	Exp IRS sic nlq rel Levien <sup>67</sup>
RDMS basado en MADAM	1971	Mass Inst of Tech Cambridge MA	H6000 (MULTICS PL/1)	Inst DBMS hlp rel stq rpg vrf Stewart en (Rustin:74)
REGIS nombre anterior	1972	General Motors Warren MI	IBM360-67	Dev DBMS hlp(PL/1) rel stq Joyce <sup>78</sup>
REL	1969	Calif Inst of Tech Pasadena CA	IBM360	Dev IRS nlq rel ampliable Thompson <sup>68</sup>
RELGRAF	1982	Adv.Rel.Techn.Inc. Menlo Park CA	PRIME OS o MPX	Com IRS stq rel cal vis txt gra sch ix f
RETRIEVE	1970	Tymshare Cupertino CA	XDS940	Com DBMS servicio sic stq 1-sqf; FMS(UMI) para > 3
RFMS	1971	Univ of Texas Austin TX	CDC6400	Dev DBMS hie stq sch Hardgrave <sup>69</sup>
RISS	1974	Forest Hosp. & MIT Des Plaines IL	DEC 11	DBMS 1-rel sch rpg sqf McLeod <sup>76</sup>
ROBOT	1973	Software Sciences Farnborough UK	ICL 1908 Univac9400	Com DBMS sic rpg tnf Palmer <sup>79</sup>

Nombre	Lugar de Año desarrollo	Computadora	Tipo y característica
RS/1, basado en PROPHET	1980 Bolt Beranek Newman Cambridge MA	DEC 11, VAX	SATDBMS slc rpg tnf tbq rpg grf sch
SAM base para RM, XRM	1968 IBM Scientific Center Cambridge MA	IBM360modificado 370	Exp FMS rel vrfSymonds <sup>64</sup> después RSS(SYSTEM R)
SAS	1972 Univ. N. Carolina & SAS Inst., Raleigh NC	IBM360/370	SATDBMS(estadística) slc stq rpg sqf
SBA usa QEB	1976 IBM Research Yorktown Heights NY		Dev DBMS rel tbq(por ejemplo) Zloof <sup>77</sup>
SCORE	1969 Programming Methods, New York NY	Cualquier sistema COBOL	Com FMS hlp(COBOL) hlc tbq sqf lsf
SDD-1	1978 Comp Corp of Am Cambridge, MA	DEC 10/20	Exp DDBMS slc rel sch DAPLEX Bernstein <sup>81</sup>
SEED	1978 Internat'l Data Base Syst., Philadelphia PA	DEC 11,10/20 IBM370 CDC6600	Com DBMS hlp(COBOL) hlc(FORTRAN) sch rpg net(1973) rnf
SEQUITUR	1981 Pacific Software Berkeley CA	DEC-11, VAX Sistemas Z-8000	Com DBMS slc, hlc(C) rel tbq lxf rpg txt
SESAM	1973 Siemens München FRG	S4004	Com FMS stq rpg sqf lsf
SHOEBOX	1970 MITRE Corp Bedford MA	IBM360	Exp SATDBMS slc txt plf stq Glantz <sup>70</sup>
SIBAS	1974 Shipping Res Svc Oslo Norway & Houston TX	IBM360/370 Univac1100 DEC PDP10	Com DBMS net hlp(COBOL) hlc sch Palmer <sup>76</sup>
SIR	1977 Scientific Information Retrieval, Evanston IL	IBM360/370 CDC 6000, Cyber	Com IRS stq stat.interfacc sch net prf
SOCRATE	1970 Univ of Grenoble, CII Louvenciennes, France	IBM360-370 CII Iris46,80	Dev, Com DBMS slc sch net stq vrf

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
SOLID	1967	Penn State Univ University Park CA	IBM360-40 -67	Exp DBMS alc epr rec DeMaine <sup>71</sup>
SOURCE	1979	Telecomputing Co McLean VA	múltiple	Com IRS <i>servicio alc negocios, información al consumidor, anuncios</i>
SPIRES	1967	Stanford Univ Stanford CA	IBM360-67 370-168	Inst IRS, DBMS alc hie lry bib trf lxf pri Schroeder in Kerr <sup>76</sup>
SQLDS	1981	IBM San José CA	IBM370 (DOS)	DBMS hlp(PL/1) rel stq lxf(vsam) pri vie
STAIRS	1972	IBM Stuttgart FRG	IBM360/370 con CICS	Com IRS alc stq txt sqf drf lxf pri
SWALLOW	1980	Mass. Inst. Tech. Cambridge MA		Exp DFMS obj
SYSIF	1970	CAP-SOGETI Paris 16 France	IBM360/370, CII Iris, Univac1100	Com IRS alc niq sqf lxf
SYSTEM 1022	1978	Software House Cambridge MA	DEC 10/20	Com DBMS alc, hie lxf <i>Servicio de tiempo compartido</i>
SYSTEM 2000 <sup>o</sup> S2000, S2K	1970	Intel-MRI Austin TX	IBM360/370 Univac1100 CDC6000	Com DBMS hlp(cosol. PL/1) hie stq lxf Kroenke <sup>76</sup>
SYSTEM C	1981	Software Clearing House, Cleveland OH	NCR, Criterion	Com DBMS hie net(1978) sch stq rnf rec pri
SYSTEM-R	1975	IBM Research San José CA	IBM370	Exp DBMS hlp(PL/1) rel stq ( <i>SEQUEL</i> ) vie Astrahan <sup>76</sup>
SYSTEM-R*	1981	IBM Research San José CA	IBM370 VMS	Exp DDBMS hlp rel sch stq vie

Nombre	Año	Lugar de desarrollo	Computadora	Tipo y característica
IDMS	1966	System Dev Corp Santa Monica CA	IBM360-50 (Adept)	Dev DBMS slc hie lrq sch Bleier <sup>65</sup> , CODASYL <sup>71A</sup>
TOD	1973	Stanford Univ. Stanford CA & ITTRI Chicago 16 IL	IBM360-50,67, 370 DEC VAX	Inst SATDBMS hlc(PL/I) sch lrq gra inf tnf cpr cip Wiederhold <sup>73</sup>
TOOL-IR	1974	Univ of Tokyo Tokyo Japan	HITAC8800	Inst IRS slc stq bib pri Yamamoto <sup>75</sup>
TOTAL	1971	Cincom Inc Cincinnati OH  <i>también en Siemens4004 Honeywell200 NCR Century</i>	IBM360/370 Univac 90,V70 CDC Cyber	Com DBMS hlc sch net (2 niveles:drf,rf) DDICT Cagan <sup>73</sup>
TRAMP	1987	Univ of Michigan Ann Arbor MI	IBM360-67	Exp DBMS nlq rel Ash <sup>68</sup>
UCC TEN	1976	University Comp. Dallas TX	IBM360/370 IMS	Com DDICT slc rpg sch IMS
UNIDATA	1970	United Computing Kansas City MO	CDC6400	Com DBMS hie sch rpg gra
VISIFILE	1982	Visicorp San José CA	IBM PC, Apple	Com FMS slc ixf stq
WOODSTOCK	1979	Xerox Research Lab Palo Alto CA	Xerox Altos	Dev DFMS Swinehart <sup>79</sup>
ZETA <i>también</i> TORUS	1974	Univ of Toronto Toronto Ontario	IBM360/370	Exp DBMS hlc(PL/I) rel stq vie drf Tsichritzis <sup>77</sup>

**Abreviaturas referentes al tipo y características de sistemas de bases de datos.**

alg álgebra de relaciones	DBCMP computadora de base de datos
bib datos bibliográficos	DBL lenguaje de base de datos
BOMP procesador de facturas de materiales	DBMS sistema de manejo de bases de datos
cal cálculo relacional	DDBMS DBMS distribuido
cip cifrado	DDICT sistema de diccionario de datos
Com comercial	Dev de desarrollo
cpr compresión	DFMS FMS distribuido

DPG generador de programas de bases de datos	plf organización de archivo de aplio
dif organización de archivo, directa a inmediata	pri protección de la privacidad
Exp experimental	Pro propuesto
FMS sistema de manejo de archivos	QUS sistema de consulta y actualización
FS sistema de archivo	rec apoyo a la recuperacion
gra apoyo gráfico de datos	rel organización relacional de base de datos
hle organización jerárquica de base de datos	rnf organización de archivo de anillo o cadena
hlc sistema de lenguaje de la computadora	rpg generador de informes
anfitrióna, al que se logra acceso mediante	SADBMS DBMS de aplicación única
CALL	SATDBMS DBMS del tipo de aplicación única
hlp sistema de lenguaje de la computadora	sch esquema
anfitrióna, con preprocesador	slc sistema autocontenido
Inat institucional	sqf organización secuencial de archivo
lrq procesador de consulta interrogativa	stq procesador de consulta orientado a
IRS sistema de recuperación de información	proposiciones
isf archivo secuencial indizado	tbq procesador tabular de consulta
ixf archivos indizados	tnf archivos transpuestos
net organización de redes de bases de datos	trf archivos con estructura de árbol
(el año indica el estándar CODASYL)	txt datos textuales
niq capacidad de consulta de lenguaje natural	via apoyo para visiones de usuarios múltiples
obj con base en lenguaje objeto	vrf apoyo a archivo virtuales

APENDICE B

Clasificación de los principales paquetes de soporte lógico durante el año de 1982.

Los paquetes de sistemas fueron instalados en las siguientes computadoras: IBM 76%; Burroughs 8%; Andhal 4%; NAS/Itel y Sperry Univac 3% cada una; DEC, Hewlitt Packard, Honeywell, y Magnuson 2% cada una; por último Control Data 1%.

-SISTEMAS MANEJADORES DE BASES DE DATOS-

Nombre:	Desarrollado por:
SAS	SAS Intitute, Inc.
QUIKJOB II	System Support Software, Inc.
CANDE	Burroughs Corp.
EASYTRIEVE	Pansophic Systems, Inc.
ADABAS	Software AG System Group.
IDMS	Cullinane Database Systems.
IMAGE/QUERY	Hewlett-Packard.
DMS II	Burroughs Corp.
DM-IV	Honeywell Information Systems.
DATACOM/DB	Applied Data Research.
RAMIS II	Mathematica Products Group.

-SISTEMAS GENERADORES DE REPORTES-

Nombre:	Desarrollado por:
DYL-260	Dylakor.
VIEW	Hewlett Packard.
DYL-280	Dylakor.
VISICALC	VisiCorp.
ODESY	Burroughs Corp.
WYLBUR	On-Line Business Systems, Inc.
CULPRIT/ AUDITOR	Cullinane Database Systems, Inc.
BRADS II/ BRADS III	IBM Corp./Information, Systems Group.
REPORT	Burroughs Corp.

Clasificación de los principales paquetes de soporte lógico en el año de 1983.

-SISTEMAS MANEJADORES DE BASES DE DATOS Y  
-PARA EL MANEJO DE DATOS-

Nombre:	Desarrollado por:
-----	-----
SAS	SAS Institute, Inc.
SISTEM 1022	Software House.
DMS-II	Burroughs Corp.
IDMS	Cullinet
IMAGE	Hewlett-Packard.
ADABAS	Software AG of North America.
FOCUS	Information Builders, Inc.
DATAKOM/DB	Applied Data Research.
TOTAL	Cincom Systems, Inc.
RAMIS II	Mathematica Products Group, Inc.
INFO	Henco.
DM-IV	Honeywell Information Systems, Inc.
DMS-1100	Sperry Corp.
DATA BASE	Tominy, Inc.
PLUS	

=====

-SISTEMAS DE CONSULTA-  
-AYUDAS PARA DBMS-

Nombre:	Desarrollado por:
-----	-----
DYL-280	Dylakor.
DYL-260	Dylakor.
EDP AUDITOR	Cullinet.
CULPRIT	
DATA	MSP, Inc.
MANAGER	
EASYTRIEVE	Pansophic Systems Inc.
KEYFAST	H & M Systems Software, Inc.
CA-EARL	Computer Associates International, Inc.
UCC-TEN	University Computing Company.
KEY/MASTER	TSI International.
DATA	Applied Data Research.
DICTIONARY	
DATA	TSI International.
ANALYZER	

-SISTEMAS MONITORES DE TELECOMUNICACIONES-

Nombre:	Desarrollado por:
WESTI	Westinghouse Electric Corp.
SHADOW	Altergo Products, Inc.
CICS	IBM Corp./Information, Systems Group.
DATACOM/DC	Applied Data Research
SWIFT	Mathematica Products Group.
COM-LETE	Software AG of North America.
INTERCOMM	SDA Products, Inc.
MINICOMM	SDA Products, Inc.
ENVIRON/1	Cincom Systems, Inc.
TASK/MASTER	TSI International.

-----  
-OTROS SISTEMAS DE COMUNICACIONES-

Nombre:	Desarrollado por:
HASP	IBM Corp./Information Systems Group.
VOLLIE	Applied Data Research.
JES 3	IBM Corp./Information Systems Group.
ABEND-AID	Compuware Corp.
NDL	Burroughs Corp.
TSO	IBM Corp./Information Systems Group.
ROSCOE	Applied Data Research.
GEMCOS	Burroughs Corp.
CMS	IBM Corp./Information Systems Group.
JES 2	IBM Corp./Information Systems Group.

-----  
-SISTEMAS OPERATIVOS / SISTEMAS DE SOPORTE-

Nombre:	Desarrollado por:
FAQS	Goal Systems International.
DOS/MUT	Software Pursuits, Inc.
WHITLOW-	Syncsort, Inc.
SYNCSORT	
FLEE/FLIM	Goal Systems International.
CA-SORT	Computer Associates, Inc.
FDR	Innovation Data Processing, Inc.
DOS DUMP/ RESTOR	Westinghouse Electric Corp.
PANVALET	Pansophic Systems Inc.



LIBRARY-SYS IMSL, Inc.  
SPACE/MANAGE Insac Software, Inc.  
UCC-ONE University Computing Company.  
UTILITY IBM Corp./Information Systems Group.  
ASM-2 Cambridge Systems Group.

=====

-AYUDAS PARA EL MANEJO DE SISTEMAS-

Nombre:           Desarrollado por:  
-----

ONEGAMON       Candle Corp.  
PLAN IV        Capex Corp.  
ACF-2          Cambridge Systems Group.  
LOOK          Applied Data Research.  
FORESIGHT     United Computing Systems, Inc.  
RESOLVE       Boole & Babbage, Inc.  
DITTO         IBM Corp./Information Systems Group.  
UCC-15        University Computing Company.  
JOB MONITOR   Westinghouse Electric Corp.  
JARS          Johnson Systems, Inc.  
JASPER        Computer Associates, Inc.  
KOMAND        Pace Applied Technology, Inc.

=====

-UTILERIAS DE PROGRAMACION-

Nombre:           Desarrollado por:  
-----

OPTIMIZER/     Capex Corp.  
OPTIMIZER II  
UCC-TWO        University Computing Company.  
OWL            NCI, Inc.  
MARK IV        Informatics, Inc.

=====

DATA TSI International.  
CATALOGUE 2  
INQUIRY/IV Informatics General, Corp.

=====

-AYUDAS PARA EL DESARROLLO DE PROGRAMAS Y  
-UTILERIAS DE PROGRAMACION.

Nombre:           Desarrollado por:  
-----

VOLLIE	Applied Data Research.
INTERTEST	On-Line Software International.
QUOTA II	Altergo Products, Inc.
MANTIS	Cincom Systems, Inc.
IMSL/ FORTRAN/ LIBRARIES	IMSL, Inc.
O-W-L	Pansophic Systems, Inc.
NATURAL	Software AG of North America.
CA- OPTIMIZER	Computer Associates International Inc.
UFO	Oxford Software Corp.
CPG	Altergo Products, Inc.
METACOBOL	Applied Data Research.
MARK IV	Informatics General Corp.
AUTOFLOW II	Applied Data Research.

=====

-SOPORTE LOGICO PARA COMUNICACIONES-

Nombre:           Desarrollado por:  
-----

WESTI	Westinghouse Electric Corp.
IMS/DC	IBM, National Accounts Division.
TONE 3	Tone Software Corp.
SHADOW II	Altergo Products, Inc
ROSCOE	Applied Data Research.
NDL	Burroughs Corp.
CICS/VS	IBM, National Accounts Division.
COM-LETE	Software AG of North America.
ENVIRON/1	Cincom Systems, Inc.
ICCF	IBM, National Accounts Division.
INTERCOMM	Polygon Systems.
DATACOM/DC	Applied Data Research.
TASK/MASTER	TSI International.

=====

-UTILERIAS Y SISTEMAS OPERATIVOS-

Nombre:	Desarrollado por:
FAST/DUMP/ RESTORE (FDR)	Innovation Data Processing, Inc.
SYSTEM/ MANAGER	Corodale, Inc.
LOGOUT/ MULTILOG	Macro 4, Inc.
SYNCSORT	Syncsort, Inc.
DISK/ UTILITIES	Westinghouse Electric Corp.
ACF-2	The Cambridge Systems Group.
DOS/MVT/VSE	Software Pursuits, Inc.
CA-SORT	Computer Associates International, Inc.
DMS/OS	Software Module Marketing.
ABEND-AID	Compuware.
SUPER-MSI	CGA Software Products Group.
CA-RAPS	Computer Associates International, Inc.
VCC-ONE	University Computing Company.
SHRINK	Informatics General Corp.
THE LIBRARIAN	Applied Data Research.

-SISTEMAS DE AYUDA Y EJECUCION-

Nombre:	Desarrollado por:
OMEGAMON	Candle Corporation.
TSO/MON	Morino Associates.
LOOK	Applied Data Research.
BEST/1	BGS Systems.
CAPTURE/MUS	BGS Systems.
CONTROL/IMSL & CONTROL/ REAL TIME	Boole & Babbage, Inc.
TSA/PPE	Boole & Babbage, Inc.
CMF/MONITOR	Boole & Babbage, Inc.
RESOLVE-	Boole & Babbage, Inc.
RESOLVE/CICS	
COMPUT-A- CHARGE	Value Computing, Inc.

Clasificación de los principales paquetes de soporte lógico durante el año de 1984.

-SISTEMAS DE COMUNICACIONES-

Nombre:	Desarrollado por:
SHADOWII	Altergo Products, Inc.
WESTI	Westinghouse Electric.
TONE 3/4	Tone Corporation.
ADR/ROSCOE	Applied Data Research, Inc.
DECNET	Digital Equipment Corporation.
CICS	IBM Corporation, National Accounts Division.
ENVIRON/1	Cincom Systems.
COM-PLETE	Software AG of North America.
TSO	IBM Corporation, National Accounts Division.
ICCF	IBM Corporation, National Accounts Division.

-SISTEMAS PARA EL MANEJO DE DATOS Y  
-AYUDAS PARA BASES DE DATOS-

Nombre:	Desarrollado por:
QUIKJOB III	System Support Software.
SAS	SAS Institute.
DYL-280	Dylakor.
BASIS	Battelle Labs.
TOOL KIT	DBMS, Inc.
DYL-260	Dylakor.
DATAMANAGER	Manager Software Products.
ASI-ST	Applications Software, Inc.
ADR/DATA	Applied Data Research, Inc.
DICTIONARY	
EXTRACTO	Oceanic Information Systems.

-SISTEMAS MANEJADORES DE BASES DE DATOS-

Nombre:	Desarrollado por:
SYSTEM 1022	Software House.
ADABAS	Software AG of North America.
IMAGE/3000	Hewlett-Packard.
IDMS	Cullinet Software.
DMS-1100	Sperry Corporation.
FOCUS	Information Builders, Inc.

RAMIS II	Mathematica Products Group.
MODEL 204	Computer Corporation of America.
DM-IV	Honeywell Information Systems.
INQUIRE	Infodata Systems.

=====

-SISTEMAS DE MONITOREO Y  
-AYUDAS DE EJECUCION-

Nombre:	Desarrollado por:
-----	-----
OMEGAMON	Candle Corporation.
MICS	Morino Associates.
DEXAN	Candle Corporation.
TSO/MON	Morino Associates.
CAPTURE/MVS	BGS Systems, Inc.
RESOLVE	Boole & Babbage, Inc.
ADR/LOOK	Applied Data Research, Inc.
BEST/1	BGS Systems, Inc.
CMF/MONITOR	Boole & Babbage, Inc.
IMF	Boole & Babbage, Inc.

=====

-AYUDAS PARA EL DESARROLLO DE PROGRAMAS

Nombre:	Desarrollado por:
-----	-----
CONDOR	Phoenix Computer Corporation.
MAPPER	Software AG of North America.
ADR/VOLLIE	Applied Data Research, Inc.
MANTIS	Cincom Systems.
INTERTEST	On-Line Software International.
NATURAL	Software AG of North America.
ISPF	IBM Corporation, National Accounts Division.
O-W-L	Pansophic Systems.
QUOTA II	Altergo Products.
CPG	Altergo Products.

=====

-SISTEMAS DE CONSULTA Y  
-GENERADORES DE REPORTES

Nombre:	Desarrollado por:
-----	-----
OMNI	Haverly Systems.
ADR/	Applied Data Research, Inc.
DATAQUERY	
EDP-AUDITOR	Cullinet Software.
EASYTRIEVE	Pansophic Systems.
CA-EARL	Computer Associates International.

AUDIT/	TSI International.
ANALYSER	
KEYFAST	H & M Systems Software.
DATATRIEVE	Digital Equipment Corporation.
CULPRIT	Cullinet Software.
QUERY/3000	Hewlett Packard.

=====

-SISTEMAS OPERATIVOS Y  
-UTILERIAS

Nombre:	Desarrollado por:
-----	-----
FLEE/XP	Goal Systems International.
FAVER/XP	Goal Systems International.
DISK/	Westinghouse Electric.
UTILITIES	
DOS/MVT/VSE	Software Purvits, Inc.
SYNCSORT-	Syncsort, Inc.
DOS	
LOGOUT/	Macro 4, Inc.
MULTILOG	
CA-SORT	Computer Associates International.
SYNCSORT-OS	Syncsort, Inc.
FDR	Innovation Data Processing.
INSTANT FBA	SDI, Inc.

=====

APENDICE C

DIRECTORIO DE LOS PRINCIPALES DISTRIBUIDORES  
DE PAQUETES DE SOPORTE LOGICO

ADPAC, Corp.  
120 Howard Street, San Francisco, CA 94105  
415-981-2710

Altergo Products, Inc.  
400 W. Cummings Park, Suite 6900, Woburn, MA 01801  
617-938-8811

Applications Software, Inc.  
21515 Hawthorne Boulevard, Terrance, CA 90503  
213-540-0111

Applied Data Research  
Route & Orchard Rd., Princeton, NJ 08540  
201-874-9000

Batelle Labs.  
505 King Avenue, Columbus, OH 43201  
614-424-5524

BGS Systems, One University Office Park  
29 Sawyer Road, Waltham, MA 02254  
617-891-0000

Boole & Babbage, Inc.  
510 Oakmead Parkway, Sunnyvale, CA 94086  
408-735-9550

Burroughs Corp.  
Burroughs Place, Detroit, MI 48232  
313-972-7000

Cambridge Systems Group  
24275 Elise Court, Los Altos Hills, CA 94022  
415-941-4558

Candle Corp.  
4676 Admiralty Way, Marina del Rey, CA 90291  
213-821-2902

Capex Corp.  
4125 North 14th Street, Phoenix, AZ 85014  
602-264-7241

CGA Software Products Group  
1370 Piccard Drive, Rockville, MD 20850  
301-948-9600

Cincom Systems, Inc.  
2300 Montana Avenue, Cincinnati, OH 45211  
513-662-2300

Computer Associates, Inc.  
125 Jericho Turnpike, Jericho, NY 11753  
516-333-6700

Computer Corporation of America  
4 Cambridge Center, Cambridge, MA 02142  
617-492-8860

Compuware Corp.  
29433 Southfield Rd., Southfield, MI 48076  
313-559-0890

Corodale, Inc.  
211 Congress Street, Boston, MA 02110  
617-426-8780

Cullinane Database Systems  
400 Blue Hill Drive, Westwood, MA 02090  
617-329-7700

Cullinet  
400 Blue Hill Drive, Westwood, MA 02090  
617-329-7700

DBMS, Inc.  
1801-A Mill Street, Naperville, IL 60540  
312-961-5700

Digital Equipment Corporation  
146 Main Street, Maynard, MA 01754  
617-897-5111

Dylakor  
17418 Chatsworth Street, Granada Hills, CA 91344  
213-366-1781

Goal Systems International  
P.O. Box 29481, Columbus, OH 43229  
614-268-1775

H & M Systems Software, Inc.  
351 Evelyn Street, Paramus, NJ 07652  
201-599-9111

Haverly Systems  
78 Broadway, P.O. Box 919, Denville, NJ 07834  
201-627-1424

Henco  
100 Fifth Avenue, Waltham, MA 02154  
617-890-8670



Hewlett-Packard, Data Systems Division  
11000 Wolfe Road, Cupertino, CA 95014  
408-257-7000

Honeywell Information Systems  
200 Smith Street, Waltham, MA 02145  
617-890-8400

IBM Corp./Information Systems Group, NAD  
1133 Westchester Ave., White Plains, NY 10604  
914-696-1900

IBM DPD  
1133 Westchester Ave., White Plains, NY 10604  
914-696-3043

IBM GSD  
5775 Glenridge Drive, Northeast, Atlanta, GA 30301  
914-696-3043

IBM National Accounts Division  
1133 Westchester Ave., White Plains, NY 10604  
914-696-1900

IMSL, Inc., NBC Bldg.  
7500 Bellaire Blvd., Houston, TX 77036  
713-772-1927

Infodata Systems  
5205 Leesburg Pike, Falls Church, VA 22041  
703-578-3430

Informatics General, Corp.  
9441 LBJ Freeway, Dallas, TX 75243  
214-231-1400

Informatics Inc., Software Products  
21050 Vanowen Street, Canoga Park, CA 91304  
213-887-9121

Information Builders, Inc.  
1250 Broadway, New York, NY 10001  
212-736-4433

Innovation Data Processing, Inc.  
970 Clifton Avenue, Clifton, NJ 07013  
201-777-1940

Insac Software, Inc.  
2300 Peachford Road, Suite 3235, Atlanta, GA 30338  
404-452-7676

Jason Data Services  
24871 Pylos Way, Mission Viejo, CA 92691  
714-770-7789

Johnson Systems, Inc.  
7923 Jones Branch Drive, McLean, VA 22102  
703-821-1700

Macro 4, Inc.  
1 West Hanover Avenue, Mount Freedom, NJ 07970  
201-895-4800

Manager Software Products (MSP)  
131 Hartwell Avenue, Lexington, MA 02173  
617-861-6130

Mathematica Products Group  
P.O. Box 2392, Princeton, NJ 08540  
609-799-2600

Morino Associates  
8133 Leesburg Pike, Suite 602, Vienna, VA 22180  
703-734-9494

MSP, Inc.  
131 Hartwell Avenue, Lexington, MA 02173  
617-861-6130

NCI, Inc.  
3720 Longview Drive, Atlanta, GA 30341  
404-451-7455

Nixdorf Computer Software Company  
6517 Everglades Drive, Richmond, VA 23225  
804-276-9200

Oceanic Information Systems (Optipro)  
2197 Leon-Harmel, Quebec City, POQ GIN 4N5  
418-681-7741

On-Line Business Systems, Inc.  
115 Sansome Street, San Francisco, CA 94104  
412-931-7600

On-Line Software International  
two Executive Drive, Fort Lee, NJ 07024  
201-592-0009

Oxford Software Corp.  
174 Boulevard, Hasbrouck Heights, NJ 07604  
201-288-1515

Pace Applied Technology, Inc.  
7900 Sudley Rd., Manassas, VA 22110  
703-369-3200

Pansophic Systems, Inc.  
709 Enterprise Drive, Oakbrook, IL 60521  
312-986-2260

Phoenix Computer Corporation  
11944 Jefferson Boulevard, Culver City, CA 90230  
213-827-4500

Polygon Systems  
71 Fifth Avenue, New York, NY 10003  
212-741-5200

Retida, Inc.  
25 Mohegan Trail, Saddle River, NJ 07458  
201-327-4223

SAS Institute, Inc.  
SAS Circle, Cary, NC 27511  
919-467-8000

SDA Products, Inc.  
475 Park Avenue, 26th Floor, New York, NY 10016  
212-481-6800

SDI, Inc.  
1700 South El Camino Real, P.O. Box 5801, San Mateo CA 94402  
405-572-1200

Software AG of North America  
11800 Sunrise Valley Drive, Reston, VA 22901  
703-860-5050

Software AG Systems Group  
11800 Sunrise Valley Drive, Reston, VA 22901  
703-860-5050

Software House  
1105 Massachusetts Avenue, Cambridge, MA 02138  
617-661-9440

Software Module Marketing  
1007 Seventh Street, Sacramento, CA 95814  
916-441-7234

Software Pursuits, Inc.  
444 Market Street, Suite 800, San Francisco, CA 94111  
415-392-7171

Sperry Univac Division, Sperry Corp.  
P.O. Box 500, Blue Bell, PA 19422  
215-542-4011

Sterling Software Marketing  
Crocker Bank Building, Suite 500,  
1007 Seventh Street, Sacramento, CA 95814  
916-441-7234

Syncsort, Inc.  
560 Sylvan Avenue, Englewood Cliffs, NJ 07632  
201-568-9700

System Support Software, Inc.  
5230 Springboro Pike, Dayton, OH 45439  
513-435-9514

Tominy, Inc.  
4221 Malsbary Road, Building One, Cincinnati, OH 45242  
513-984-6605

Tone Software Corporation  
1735 S. Brookhurst Anaheim, CA 92804  
714-991-9460

TSI International  
50 Washington Street, Norwalk, CT 06854  
203-853-2884

UCCEL Corporation  
UCCEL Tower, Exchange Park, Dallas, TX 75235  
214-353-7533

United Computing Systems, Inc.  
2525 Washington St., Kansas City, MO 64108  
816-221-9700

University Computing Company, UCC  
Tower/Exchange Park, Dallas, TX 75235  
214-353-7100

University of Waterloo  
Waterloo, Ontario, N2L 3G1, Canada  
519-885-1211

Value Computing, Inc.  
498 Kings Highway, Cherry Hill, NJ 08034  
609-482-2500

Visicorp  
2895 Zanker Rd., San Jose, CA 95134  
408-946-9000

Westinghouse Electric Corp.  
2040 Ardmore Blvd., Pittsburgh, PA 15221  
412-636-3100

APENDICE D

CATALOGOS Y DIRECTORIOS DE PRODUCTOS DE SOPORTE LOGICO

- 1) Auerbach Publishers, Inc.  
Computer Technology Reports  
Software Reports
- 2) Centre D'Experimentation de Progiciels (France)  
Guide European des Products Logiciels
- 3) CUYB Publications, Inc. (England)  
International Directory of Software
- 4) Datapro Research Corporation  
Directory of Software  
Annual User Rating of Proprietary Software
- 5) Honeywell Information Systems, Inc.  
Honeywell Software Catalog  
Application Reference Index
- 6) IBM  
Keyword Index and Program Information  
Service for Consultants Manual  
Systems and Products Guide  
Applications and Abstracts
- 7) Imprint Software  
International Microcomputer Software Directory
- 8) Infratest Information Services (Germany)  
ISIS Software Report
- 9) International Computer Programs, Inc.  
Interface Series  
The ICP Software Directory
- 10) Management Information Corporation  
Packaged Software Buyer's Guide

- 11) U.S. Government Department of Commerce, National Technical Information Service  
Directory of Computerized Data Files and Related Technical Reports  
Directory of Federal Statistical Data Files  
Directory of Computer Software and Related Reports  
Computer Control and Information Theory
  
- 12) U.S. Government, NASA, Technology Utilization Office  
Computer Software Management and Information Center  
(COSMIC), University of Georgia
  
- 13) UNIVAC  
Consultant's Handbook

## APENDICE E

Bibliografía sobre Metodologías para el Desarrollo de Sistemas de Soporte Lógico.

Administración de Proyectos de Software  
Curso de la División de Educación Continua de la  
Facultad de Ingeniería  
Marcial Portilla, Robertson

Administración de un proyecto de programación  
W. Metzger, Philip  
Ed. Trillas

Computer and information processing  
Gordon B., Davis  
Ed. McGraw-Hill  
pp. 55-186

Control del ciclo de desarrollo de sistemas de información  
I. Benjamin, Robert  
Ed. Limusa

Desarrollo y Administración de Programas  
de Computadora (Software)  
Gerez Greizer, Victor  
Mier Muth, Mauricio  
Nieva Gómez, Rolando  
Rodríguez Ortiz, Guillermo  
Ed. CECSA

Procesamiento de datos  
conceptos y sistemas  
J. Verzello, Robert  
Reutter III, John  
Mcgraw Hill  
pp. 319-351

Programmer Productivity Myths, Methods  
and Morphology  
Lowell Jay, Arthur  
Ed. John Wiley & Sons

Software engineering:  
a practitioner's approach  
S. Pressman, Roger  
Mc Graw Hill

The Software Development Project  
Planning & Management  
Bruce, Phillip and  
M. Pederson, Sam



**APENDICE F**

**FORMAS DE TRAMITES LEGALES**

FORMA DE INSCRIPCION AL REGISTRO DE  
LA TRANSFERENCIA DE TECNOLOGIA Y EL  
USO Y EXPLOTACION DE PATENTES Y MARCAS

C. DIRECTOR GENERAL DE TRANSFERENCIA DE TECNOLOGIA.  
SECRETARIA DE COMERCIO Y FOMENTO INDUSTRIAL,  
MEXICO, D.F.

Con fundamento en las disposiciones relativas de la Ley sobre el Control y Registro de la Transferencia de Tecnología y el Uso y Explotación de Patentes y Marcas, ruego a usted ordenar que previos los trámites respectivos, se lleve a cabo - - - - -

la inscripción en el Registro.

del acto, convenio o contratos, cuyas características adelante se expresan.

Los efectos de la presentación del acto, convenio o contrato para inscripción en el Registro, consisten en que esa Dirección deberá entrar el análisis legal, técnico y económico del mismo y dictar resolución sobre la procedencia de la inscripción, dentro de los 90 días hábiles siguientes a dicha presentación, según sea que hubiera sido celebrado antes o después de la iniciación de la vigencia de la Ley de la materia.

PARTE OTORGANTE: \_\_\_\_\_

PARTE RECEPTORA: \_\_\_\_\_

DOMICILIO PARA OIR NOTIFICACIONES: \_\_\_\_\_

TELEFONO: \_\_\_\_\_ CODIGO POSTAL \_\_\_\_\_

Por lo expuesto a esa Dirección, atentamente pido se de trámite a esta solicitud, para lo cual estoy anexando los documentos listados al calce.

Protesto a usted mi más atenta consideración.

México, D.F., a \_\_\_\_\_

\_\_\_\_\_  
NOMBRE DEL SOLICITANTE.

\_\_\_\_\_  
NOMBRE Y FIRMA DEL APODERADO LEGAL.

- ANEXOS: 1) Cinco tantos de esta solicitud.  
2) Cinco tantos del cuestionario elaborado por esa H. Dirección.

- 3) Copia autógrafa o copia simple certificada por fedatario público y cuatro copias simples o fotostáticas del acto, convenio o contrato, materia de la inscripción. En caso de que el acto, convenio o contrato haya sido redactado en idioma extranjero, deberá acompañarse un tanto de la traducción hecha por perito autorizado con cuatro copias simples o fotostáticas de la misma.
- 4) Recibo oficial del pago de derechos que señala la tarifa respectiva.
- 5) Poder Notarial de la persona que resuelva el cuestionario (con facultades para actos de administración); y
- 6) Carta poder otorgada por quien resuelva el cuestionario a quien presente esta solicitud, que satisfaga los requisitos de Ley.

FORMA DEL CUESTIONARIO DE INSCRIPCION  
AL REGISTRO DE LA TRANSFERENCIA DE  
TECNOLOGIA Y EL USO Y EXPLOTACION  
DE PATENTES Y MARCAS.

SECRETARIA DE COMERCIO Y FOMENTO INDUSTRIAL  
DIRECCION GENERAL DE TRANSFERENCIA DE TECNOLOGIA.

CUESTIONARIO PARA LA INSCRIPCION DE ACTOS, CONVENIOS O CONTRATOS PREVISTOS POR LA LEY SOBRE EL CONTROL Y REGISTRO DE LA TRANSFERENCIA DE TECNOLOGIA Y EL USO Y EXPLOTACION DE PATENTES Y MARCAS.

I DATOS DE LA EMPRESA RECEPTORA O LICENCIATARIA :

1.- NOMBRE, DENOMINACION O RAZON SOCIAL : \_\_\_\_\_  
\_\_\_\_\_

2.- DOMICILIO : \_\_\_\_\_  
OFICINAS : \_\_\_\_\_  
PLANTA : \_\_\_\_\_

3.- FECHA DE INICIO DE OPERACIONES : \_\_\_\_\_

4.- R.F.C. : \_\_\_\_\_

5.- CAPITAL CONTABLE : \_\_\_\_\_

6.- CAPITAL SOCIAL : \_\_\_\_\_

7.- PORCENTAJE DE PARTICIPACION DE CAPITAL EXTRANJERO EN EL CAPITAL SOCIAL : \_\_\_\_\_

8.- PRINCIPAL INVERSIONISTA EXTRANJERO, NACIONALIDAD Y MONTO DE PARTICIPACION : \_\_\_\_\_

9.- - PERTENECE LA EMPRESA RECEPTORA A ALGUN GRUPO INDUSTRIAL

SI ( ) NO ( )

NOTA: LAS CIFRAS DEBEN SER EXPRESADAS EN MILES DE PESOS.

EN CASO AFIRMATIVO MENCIONE EL NOMBRE DEL GRUPO : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

10.- PERSONAL OCUPADO (ULTIMO EJERCICIO)  
 OBREROS \_\_\_\_\_ TECNICOS \_\_\_\_\_ ADMINISTRATIVOS \_\_\_\_\_ TOTAL \_\_\_\_\_

11.- PRINCIPALES PRODUCTOS DE LA EMPRESA EN EL ULTIMO EJERCICIO EN  
 ORDEN DE IMPORTANCIA Y PORCENTAJE QUE REPRESENTARON DE LAS --  
 VENTAS NETAS TOTALES :

DESCRIPCION	%
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

12.- VENTAS NETAS TOTALES DE LOS ULTIMOS 5 AÑOS (EN CASO DE QUE SE  
 ESTE INICIANDO OPERACIONES, PROPORCIONAR ESTIMACIONES) :

AÑO	MONTO
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____

13.- UTILIDADES NETAS TOTALES DE LOS ULTIMOS 5 AÑOS (EN CASO DE QUE ESTE INICIANDO OPERACIONES, PROPORCIONAR ESTIMACIONES) :

AÑO	MONTOS
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____

14.- PRESUPUESTO DE DIVIDAS 3 ULTIMOS EJERCICIOS (En caso de que esté iniciando operaciones, proporcionar estimaciones).

CONCEPTO	AÑOS		
	19 __	19 __	19 __
EXPORTACION DE MERCANCIAS			
EXPORTACION DE SERVICIOS			
OTROS INGRESOS DEL EXTERIOR			
<b>T O T A L :</b>			
IMPORTACION DE MERCANCIAS			
IMPORTACION MATERIAS PRIMAS			
IMPORTACION DE MAQUINARIA Y EQUIPO			
IMPORTACION DE REPARACIONES, PARTES Y HERRAMIENTAL			
PAGO DE DIVIDENDOS			
PAGOS POR TECNOLOGIA			
PAGO DE INTERESES AL EXTERIOR			
OTROS PAGOS AL EXTERIOR			
<b>T O T A L :</b>			
<b>S A L D O :</b>			



15.- INVERSION NETA AL ULTIMO EJERCICIO : AÑO \_\_\_\_\_  
 T O T A L \_\_\_\_\_ MAQUINARIA Y EQUIPO \_\_\_\_\_

16.- DEPRECIACION ACUMULADA \_\_\_\_\_  
 \_\_\_\_\_

17.- INDICAR SI LA EMPRESA TIENE COMPROMISOS CON ESTA DIRECCION RE-  
 LATIVOS A :

INVESTIGACION Y DESARROLLO TECNOLOGICO	( )	ASIMILACION DE TECNCLOGIA	( )
CAPACITACION DE PERSONAL	( )	EXPORTACIONES	( )
COMPENSACION DE DIVISAS	( )	OTROS *	( )

\* ESPECIFIQUE : \_\_\_\_\_  
 \_\_\_\_\_

18.- GASTOS EN INVESTIGACION Y DESARROLLO TECNOLOGICO AL ULTIMO - -  
 EJERCICIO : \_\_\_\_\_

19.- INDICAR SI LA EMPRESA CUENTA CON LABORATORIO DE CONTROL DE CA-  
 LIDAD Y EL NUMERO DE PERSONAL TECNICO QUE LABORA EN EL NORMAL-  
 MENTE : SI \_\_\_\_\_ NO \_\_\_\_\_  
 NUMERO DE PERSONAL \_\_\_\_\_ TIEMPO COMPLETO \_\_\_\_\_  
 MEDIO TIEMPO \_\_\_\_\_

20.- INDICAR SI LA EMPRESA CUENTA CON DEPARTAMENTO DE INGENIERIA :  
 SI \_\_\_\_\_ NO \_\_\_\_\_  
 NUMERO DE PERSONAL \_\_\_\_\_

21.- INDICAR SI LA EMPRESA CUENTA CON LABORATORIO DE INVESTIGACION TECNOLÓGICA :

SI \_\_\_\_\_ NO \_\_\_\_\_

NUMERO DE PERSONAL \_\_\_\_\_

22.- INDICAR EL NUMERO DE CONTRATOS VIGENTES CELEBRADOS CON LA - - EMPRESA OTORGANTE Y SUS DECIMALES CORRESPONDIENTES CON OTRAS LICENCIANTES : \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

II DATOS DE LA EMPRESA OTORGANTE O LICENCIANTE :

1.- NOMBRE, DENOMINACION O RAZON SOCIAL : \_\_\_\_\_  
\_\_\_\_\_

2.- NACIONALIDAD : \_\_\_\_\_

3.- FECHA DE INICIO DE OPERACIONES : \_\_\_\_\_

4.- ACTIVIDAD : \_\_\_\_\_  
\_\_\_\_\_

5.- PRINCIPALES PRODUCTOS QUE FABRICA Y/O SERVICIOS QUE PRESTA : \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

6.- CAPITAL SOCIAL DE LA OTORGANTE : \_\_\_\_\_

7.- PORCENTAJE DE CAPITAL EXTRANJERO EN EL CAPITAL SOCIAL DE LA - - OTORGANTE : \_\_\_\_\_

8.- ORIGEN DEL CAPITAL EXTRANJERO : \_\_\_\_\_

9.- PARTICIPACION DE LA OTORGANTE EN EL CAPITAL SOCIAL DE LA EMPRESA RECEPTORA O LICENCIATARIA : (%) \_\_\_\_\_

10.- EN QUE OTROS PAISES HA EXPEDIDO LICENCIAS SIMILARES :  
\_\_\_\_\_  
\_\_\_\_\_

11.- GASTOS EN INVESTIGACION Y DESARROLLO TECNOLOGICO A NIVEL CORPORATIVO Y POR DIVISION DE PRODUCTOS :  
CORPORATIVO : \_\_\_\_\_  
DIVISIONAL : \_\_\_\_\_

12.- PRINCIPALES COMPETIDORES EN EL PAIS DE LA OTORGANTE :  
\_\_\_\_\_  
\_\_\_\_\_

III DATOS DEL ACTO, CONVENIO Y CONTRATO :

1.- DESCRIPCION DE LOS PRODUCTOS Y/O SERVICIOS OBJETO DEL CONTRATO :  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

2.- FECHA DE INICIACION DE LA FABRICACION DE LOS PRODUCTOS OBJETO DEL ACTO, CONVENIO O CONTRATO :  
\_\_\_\_\_

3.- FECHA DE CELEBRACION DEL PRIMER ACTO, CONVENIO O CONTRATO ENTRE LAS PARTES, PARA LA FABRICACION DE LOS PRODUCTOS OBJETO DEL MISMO :  
\_\_\_\_\_

4.- PLAZO EN EL CUAL LA EMPRESA RECEPTORA NO REQUERIRA MAS DE LA ASIS-  
TENCIA TECNICA Y/O DE LOS DERECHOS OTORGADOS POR LA LICENCIANTE :

---

5.- EN CASO DE QUE SE CONCEDAN LICENCIAS DE MARCAS, PATENTES Y NO --  
APAREZCAN EN EL CONTRATO, ACOMPAÑAR COMO ANEXO UNA LISTA DE LAS --  
MARCAS Y/O PATENTES, SEÑALANDO SU NUMERO Y FECHA DE REGISTRO EN ME  
XICO, ASI COMO LA FECHA DE REGISTRO DE LAS PATENTES EN EL PAÍS DE  
ORIGEN :

---



---



---

6.- EN CASO DE QUE SE CONCEDAN LICENCIAS DE PROGRAMAS DE COMPUTO, Y --  
NO APAREZCAN EN EL CONTRATO, ACOMPAÑAR COMO ANEXO EL NOMBRE DEL --  
PROGRAMA Y SU VERSION, LA DESCRIPCION DE EQUIPO DONDE SE VA A UTI  
LIZAR Y LA DESCRIPCION DE SUS FUNCIONES :

---



---

7.- PRINCIPALES MATERIAS PRIMAS UTILIZADAS EN LOS PRODUCTOS OBJETO DEL  
CONTRATO :

P R O D U C T O S	M A T E R I A S P R I M A S
-------------------	-----------------------------


8.- PORCENTAJE DE INTEGRACION NACIONAL Y ORIGEN DE LOS INSUMOS IMPOR-  
TADOS DE LOS PRODUCTOS OBJETO DEL CONTRATO :

P R O D U C T O	P R I N C I P A L E S I N S U M O S	% DE I N T E G R A C I O N N A C I O N A L	O R I G E N D E L O S I N S U M O S



11.- VENTAS O INGRESOS NETOS DE LOS PRODUCTOS Y/O SERVICIOS OBJETO DEL CONTRATO DURANTE LOS ULTIMOS 5 AÑOS (EN SU CASO, PROPORCIONAR ESTIMACIONES) :

AÑOS	MONTOS
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____

12.- UTILIDADES NETAS DE LOS PRODUCTOS Y/O LOS SERVICIOS OBJETO DEL CONTRATO, DURANTE LOS ULTIMOS 5 AÑOS (EN SU CASO, PROPORCIONAR ESTIMACIONES) :

AÑOS	MONTOS
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____

13.- SEÑALAR EN PORCENTAJE, EL GRADO DE PENETRACION EN EL MERCADO NACIONAL CON LOS PRODUCTOS OBJETO DEL CONTRATO : \_\_\_\_\_

14.- MONTO TOTAL DE LA CONTRAPRESTACION PAGADA EN LOS ULTIMOS 5 AÑOS.

AÑOS	MONTOS
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____

15.- MONTO TOTAL DE LA CONTRAPRESTACION A PAGAR EN LOS PROXIMOS 5 AÑOS POR CONCEPTO DE LOS PRODUCTOS Y/O SERVICIOS OBJETO DEL CONTRATO, - (EN SU CASO, ESTIMACIONES) :

A Ñ O S	M O N T O
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____

16.- MONTO ANUAL DE PAGOS ADICIONALES POR CONCEPTO DE HONORARIOS DE TECNICOS DE LA LICENCIANTE, REEMBOLSOS DE GASTOS Y SERVICIOS TECNICOS ESPECIALES SOLICITADOS POR LA LICENCIATARIA Y NO INCLUIDOS EN LA CONTRAPRESTACION GLOBAL, DURANTE LOS PROXIMOS 5 AÑOS.

A Ñ O S	M O N T O
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____
19 _____	_____

17.- MONTO DE LAS EXPORTACIONES DE LOS PRODUCTOS Y/O SERVICIOS OBJETO DEL CONTRATO DURANTE LOS ULTIMOS 5 AÑOS, ESPECIFICANDO LOS PRINCIPALES PRODUCTOS Y PAISES INVOLUCRADOS (EN SU CASO, PROPORCIONAR ESTIMACIONES) :

A Ñ O	P R O D U C T O S	M O N T O	P A I S
19 _____	_____	_____	_____
19 _____	_____	_____	_____
19 _____	_____	_____	_____
19 _____	_____	_____	_____
19 _____	_____	_____	_____

16.- EN CASO DE QUE EL CONTRATO INVOLUCRE LA CONSTRUCCION, INSTALACION Y/O PUESTA EN MARCHA DE UNA PLANTA, INDICAR :

FECHA DE INICIO DE LA CONSTRUCCION DE LA PLANTA : \_\_\_\_\_

FECHA PROBABLE DE LA PUESTA EN MARCHA DE LA PLANTA : \_\_\_\_\_

MONTO TOTAL DE LA INVERSION EN MAQUINARIA Y EQUIPO EN LA PLANTA : \_\_\_\_\_

MONTO TOTAL DE LA INVERSION EN LA PLANTA : \_\_\_\_\_

19.- SEÑALAR CUALES SON LOS PRINCIPALES COMPETIDORES NACIONALES PARA -  
LOS PRODUCTOS OBJETO DEL CONTRATO : \_\_\_\_\_

20.- FECHA DE INICIO DE EXPLOTACION POR PARTE DE LA OTORGANTE, DE LA -  
TECNOLOGIA LICENCIADA : \_\_\_\_\_

21.- DESGLOSE, POR RUBROS, LA ESTRUCTURA DEL COSTO TOTAL DE PRODUCCION:

<u>RUBROS</u>	<u>M C N T O S</u>
COSTOS DIRECTOS :	
A) DIFERENCIA DE INVENTARIOS	_____
B) MATERIAS PRIMAS	_____
C) MANO DE OERA.-	_____
C.1 SUELDOS	_____
C.2 SALARIOS	_____
COSTOS INDIRECTOS :	
A) INSUMOS.-	_____
a.1 COMBUSTIBLES	_____
a.2 ENERGIA ELECTRICA	_____
a.3 PAGOS POR MAQUILA	_____
B) OTROS COSTOS :	_____
b.1 DEPRECIACION	_____
b.2 PUBLICIDAD	_____



22.- SEÑALE QUE OTRAS ALTERNATIVAS TECNOLOGICAS SE CONSIDERARON PARA LA FABRICACION DE LOS PRODUCTOS DEL CCNTRATO : \_\_\_\_\_

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

23.- INDIQUE LAS PRINCIPALES VENTAJAS DE LA TECNOLOGIA ADQUIRIDA EN - RELACION CON LAS ALTERNATIVAS TECNOLOGICAS CONSIDERADAS :

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

24.- DESCRIBA BREVEMENTE EL PROCESO DE FABRICACION PARA LOS PRODUCTOS OBJETO DEL CCNTRATO : \_\_\_\_\_

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

25.- EN RELACION A LA VISITA DE TECNICOS DE LA LICENCIANTE, INDICAR -- CUANTOS DIAS HOMBRE VAN A REPRESENTAR DURANTE LOS PROXIMOS 5 AÑOS (EN SU CASO ESTIMACIONES).

AÑOS	No. DE PERSONAS	PROFESION	AREA DE APLICACION	HONORARIO POR DIA	DIAS DE ESTANCIA

ANEJOS: DIAGRAMA DE FLUJO (BLOQUES)  
 FOLLETOS, PUBLICACIONES Y/O CATALOGOS DE LOS PRODUCTOS OBJETO DEL CONTRATO.

26.- EN RELACION A LA VISITA DE TECNICOS DE LA LICENCIATARIA A LA -  
PLANTA DE LA LICENCIANTE, INDICAR CUANTOS DIAS-HOMBRE VAN A RE-  
PRESENTAR DURANTE LOS PROXIMOS 5 AÑOS (EN SU CASO ESTIMACIONES).

AÑOS	NO. DE PERSONAS	PROFESION	AREA DE APLICACION	HONORARIO POR DIA	DIAS DE ESTANCIA

C A R A C T E R I S T I C A S Y A T R I B U T O S D E L A  
T E C N O L O G I A C O N V E N I D A E N L A P L A N T A

D E :

IV.- LA RECEPTORA :

IV.A.- DESCRIPCION DE LA TECNOLOGIA TRANSMITIDA : \_\_\_\_\_

---



---



---



---



---

IV.B.- TIPOLOGIA DE LA TECNOLOGIA :

- TECNOLOGIA DE PRODUCTO \_\_\_\_\_
- TECNOLOGIA DE PROCESO \_\_\_\_\_
- TECNOLOGIA DE DISEÑO \_\_\_\_\_
- TECNOLOGIA DE OPERACION \_\_\_\_\_
- INVOLUCRADA CON HERRAMIENTAS \_\_\_\_\_
- INVOLUCRADA CON MAQUINARIA Y EQUIPO \_\_\_\_\_

PARA LA ELABORACION DE : \_\_\_\_\_

---



---

IV.C.- ALCANCE DE LA TECNOLOGIA CONVENIDA :

- a) REDUCE DESPERDICIO INDUSTRIAL \_\_\_\_\_ %
- b) OPTIMIZA FACTORES PRODUCTIVOS \_\_\_\_\_ %
- c) REDUCE CONSUMO DE ENERGIA \_\_\_\_\_ %
- d) INCREMENTA LA PRODUCTIVIDAD \_\_\_\_\_ %
- e) EFICIENTA LA PRODUCCION \_\_\_\_\_ %
- f) SUSTITUYE IMPORTACIONES \_\_\_\_\_ %
- g) CONSUME MATERIAS PRIMAS NACIONALES \_\_\_\_\_ %
- h) PRODUCCION EXPORTABLE \_\_\_\_\_ %
- i) DIVERSIFICA FUNCIONES PRODUCTIVAS \_\_\_\_\_ %
- j) ABATE COSTOS \_\_\_\_\_ %
- k) FAVORECE MAYOR INTEGRACION NACIONAL \_\_\_\_\_ %
- l) PUEDE SUBCONTARSE \_\_\_\_\_ %
- (OPERACIONES) \_\_\_\_\_ %

## IV.D.- IMPACTO DE LA TECNOLOGIA :

LA TECNOLOGIA CONTRATADA SE DESTINARA A:

- NUEVA LINEA DE PRODUCTOS \_\_\_\_\_
- SUSTITUCION DE PRODUCTOS \_\_\_\_\_
- APLICACION DE NUEVOS PROCESOS \_\_\_\_\_
- MAYOR INTEGRACION EN PLANTA \_\_\_\_\_
- MEJORAR LA CALIDAD DEL PRODUCTO \_\_\_\_\_
- ARRANQUE DE LA PRODUCCION \_\_\_\_\_
- CAPACITACION DE PERSONAL \_\_\_\_\_
- APROVECHAMIENTO DE CAPACIDAD  
INSTALADA. \_\_\_\_\_

V. LA OTORGANTE :

## V.A.- CARACTERISTICAS:

- LA TECNOLOGIA INVOLUCRADA ES:

PROPIA \_\_\_\_\_ ADQUIRIDA \_\_\_\_\_  
 DEL GRUPO \_\_\_\_\_ INDEPENDIENTE \_\_\_\_\_

- INTENSA EN MANO DE OBRA \_\_\_\_\_
- INTENSA EN CAPITAL \_\_\_\_\_
- REQUIERE RECURSOS HUMANOS ESPECIALIZADOS \_\_\_\_\_

- 
- GRADO DE COMPLEJIDAD:

BAJO \_\_\_\_\_ MEDIO \_\_\_\_\_ ALTO \_\_\_\_\_

- 
- LA TASA DE CAMBIO TECNOLOGICO ES:

DINAMICA \_\_\_\_\_ ESTATICA \_\_\_\_\_  
 DE INNOVACIONES TECNOLOGICAS GRADUALES \_\_\_\_\_

- 
- EL CICLO DE VIDA DE LA TECNOLOGIA ES:

EMBRIONICA \_\_\_\_\_

a) DEPENDIENTE \_\_\_\_\_ b) TRADICIONAL \_\_\_\_\_  
EN CRECIMIENTO (DE VARIACIONES TECNICAS FRECUENTES)  
 \_\_\_\_\_

MADURA (DE INNOVACIONES SIN IMPORTANCIA)

a) DEFENSIVA \_\_\_\_\_ b) OFENSIVA \_\_\_\_\_

**V.B.- RESULTADOS:**

- GRADO DE INTEGRACION DE LA TECNOLOGIA EN EL PAIS DE LA OTORGANTE:

MATERIAS PRIMAS \_\_\_\_\_ %

OTROS INSUMOS \_\_\_\_\_ %

MANO DE OBRA \_\_\_\_\_ %

CAPACIDAD INSTALADA:

UNIDADES \_\_\_\_\_ KGS. \_\_\_\_\_ LITROS \_\_\_\_\_

OTROS (especifique) \_\_\_\_\_

EL DATO DE CAPACIDAD INSTALADA SE REFIERE A:

TURNO \_\_\_\_\_ DIA \_\_\_\_\_ OTROS (especifique) \_\_\_\_\_

CAPACIDAD UTILIZADA:

UNIDADES \_\_\_\_\_ KGS. \_\_\_\_\_ LITROS \_\_\_\_\_

OTROS (especifique) \_\_\_\_\_

EL DATO DE CAPACIDAD UTILIZADA SE REFIERE A:

TURNO \_\_\_\_\_ DIA \_\_\_\_\_ OTROS (especifique) \_\_\_\_\_

**V.C.- VENTAJAS:**

- REDUCE ENERGETICOS \_\_\_\_\_ %

- LA COMBINACION DE FACTORES PRODUCTIVOS ES:

OPTIMA \_\_\_\_\_ MEDIANA \_\_\_\_\_ BAJA \_\_\_\_\_

EN RELACION COMPARATIVA A OTRAS ALTERNATIVAS TECNOLOGICAS SEME-  
JANTES, ESTA:

- a) REDUCE COSTOS INDUSTRIALES \_\_\_\_\_ %  
b) REDUCE COSTOS FINANCIEROS \_\_\_\_\_ %  
c) REDUCE DESPERDICIOS INDUSTRIALES \_\_\_\_\_ %  
d) DIVERSIFICA USOS PRODUCTIVOS  
EN \_\_\_\_\_ Nº DE ALTERNATIVAS  
e) EL NIVEL DE PRODUCTIVIDAD TIENE UN ALCANCE DE \_\_\_\_\_ %

DEPARTAMENTO DE ESTUDIOS Y ESTADISTICA  
SUBDIRECCION DE VERIFICACION Y APOYO

ESTE CUESTIONARIO HA SIDO RESUELTO POR: \_\_\_\_\_

PERSONA AUTORIZADA PARA PROPORCIONAR LOS DATOS ASENTADOS EN EL -  
MISMO, SEGUN FACULTADES QUE LE FUERON OTORGADAS EN LOS TERMINOS  
DEL DOCUMENTO QUE SE ANEXA Y QUE ACREFITA SU CALIDAD DE: \_\_\_\_\_

DE LA EMPRESA QUE SOMETE A REGISTRO EL ACTO, CONVENIO O CONTRATO  
QUE TAMBIEN SE ADJUNTA A ESTE CUESTIONARIO Y QUE HA SIDO RESUELTO  
BAJO FORMAL PROTESTA DE DECIR VERDAD.

DOMICLIO PARA OIR NOTIFICACIONES: \_\_\_\_\_

TELEFONO: \_\_\_\_\_

LUGAR Y FECHA: \_\_\_\_\_

\_\_\_\_\_  
F I R M A



**FORMA DE SOLICITUD AL REGISTRO DEL  
DERECHO DEL AUTOR**

SOLICITUD DE REGISTRO

SECRETARIA DE EDUCACION PUBLICA  
DIRECCION GENERAL DEL DERECHO DE AUTOR  
DEPARTAMENTO DE REGISTRO PUBLICO  
MARIANO ESCOBEDO No. 438, 3er. PISO  
MEXICO, D. F., C.P. 11590

CONTROL \_\_\_\_\_

\_\_\_\_\_  
(Nombre y apellidos completos o razón social del solicitante)

Representado por \_\_\_\_\_  
(Llénesse en caso de que promueva el representante legal)

Con domicilio para recibir notificaciones y toda clase de documentos:

\_\_\_\_\_  
(Anótese la calle, número, colonia, entidad federativa y código postal)

\_\_\_\_\_  
Teléfono: \_\_\_\_\_

Solicita el registro a favor de: \_\_\_\_\_

De la obra titulada: \_\_\_\_\_  
\_\_\_\_\_

Para lo cual acompaño \_\_\_\_\_ ejemplares debidamente firmados.

Al efecto manifiesto (an) que dicha obra \_\_\_\_\_  
(Indicar si es inédita o la fe-

\_\_\_\_\_ cha en que se dió a conocer y en el caso, el número de la edición)

Así como el contrato de: EDICION ( ) AUTORIZACION ( ) CESION DE  
DERECHOS ( ) OTROS ( )

Colobrado entre \_\_\_\_\_  
(Menciónese en primer término el nombre de quien (es) da (n) los  
(sus) derechos y a continuación, el nombre del cesionario)

Nombre de (de los) autor (es) \_\_\_\_\_

Domicilio \_\_\_\_\_ Teléfono: \_\_\_\_\_

Nacionalidad \_\_\_\_\_

Nombre (s) del (los) coautor (es) colaborador (es) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

En caso de ser colaborador, especificar si la colaboración se hizo en forma gratuita  
o remunerada. Si no ha sujeción al espacio, anexar copia con los datos de los colabo-  
radores.

Señale con (X) la clase de obra de conformidad con las características de las ramas  
siguientes:

LITERARIA ( )                      ARTISTICA ( )                      MUSICAL ( )

EN CASO DE QUE LA OBRA CUYO REGISTRO SE SOLICITA SEA ALGUNA VERSION DE LAS QUE SE INDICAN A CONTINUACION SEÑALESE CON UNA (X).

ARREGLO ( ) AMPLIACION ( ) ADAPTACION ( ) COMPENDIO ( )  
TRANSFORMACION ( ) TRADUCCION ( ) COMPILACION ( ) O T R O S ( )

UNICAMENTE EN EL CASO DE QUE LA OBRA CUYO REGISTRO SE SOLICITA DEL SUPUESTO ANTERIOR DEBERA PROPORCIONAR LOS DATOS SIGUIENTES:

NOMBRE DEL AUTOR PRIMIGENIO \_\_\_\_\_

TITULO ORIGINAL DE LA OBRA EN QUE SE BASO \_\_\_\_\_

IDIOMA ORIGINAL \_\_\_\_\_

MEXICO, D. F., DE \_\_\_\_\_ DE 1984

A T E N T A M E N T E

\_\_\_\_\_  
FIRMA DEL SOLICITANTE O REPRESENTANTE LEGAL.

NOTA IMPORTANTE: Despues de realizar el pago correspondiente: debera presentar los comprobantes respectivos ante esta Direccion General, en la inteligencia que de no hacerlo asi se declarara abandonado el presente tramite.

NO ESCRIBA EN ESTOS ESPACIOS

Expidase orden de cobro por la cantidad de \$ \_\_\_\_\_

por concepto de:

Recepcion, Analisis y Estudio ( ) Inscripcion ( ) Cotejo de Documentos ( )

para lo cual se recibieron:

Ejemplares \_\_\_\_\_ Comp. Musical \_\_\_\_\_ Fonogramas \_\_\_\_\_ Contratos \_\_\_\_\_

Programa Fuente ( )

Programa Objeto ( )

Combinacion en  
ambos Programas ( )

Tipo de soporte material \_\_\_\_\_

Anexos \_\_\_\_\_

Utilizando la forma \_\_\_\_\_ Articulos \_\_\_\_\_

\_\_\_\_\_  
Dictaminador.

EJEMPLO DE UNA LICENCIA DE USO DE LA EMPRESA

DIGITAL EQUIPMENT CORPORATION

# DIGITAL'S Software Binary License Agreements

---

## Principles

Software is treated as proprietary information. Customers do not own it, but are licensed to use it under the terms and conditions of software license agreements. Key points of DIGITAL's software binary license agreements are:

- Customers must have a binary license to use any of DIGITAL's binary software products.
- This license allows **one** customer to run **one** software product on the CPU it is first installed on.
- DIGITAL retains title and ownership.
- DIGITAL's licensing agreement does not allow the transfer of software from one end user to another or from one CPU to another without prior permission from DIGITAL. Software may only be transferred to another party with written permission from DIGITAL.
- A customer may reproduce the software, if necessary, but only for use on the specific CPU licensed to use it.
- The use of an update version of the software on the licensed CPU requires that the customer purchase a software update option if out of warranty or not covered by a software service contract.
- The software may be used on another single CPU on a temporary basis during a malfunction of the original CPU which causes the software to be inoperable.
- Any modification to DIGITAL-licensed software doesn't exempt the software product from DIGITAL licensing or sublicensing terms, conditions, or fees. Only those modifications that are not part of the original software are the customer's property.

## Software Ordering Options

### DIGITAL-SUPPORTED Binary License Option

This is a standard binary license that includes media, manuals, documentation, and warranty packaged together. A 90-day warranty, as specified in the Software Product Description (SPD) Addendum, is the support received (unless different warranty conditions are specified in the SPD.) Main features of the warranty include: product updates, technical information, telephone support, and on-site remedial support. Depending on complexity, DIGITAL-supported products are designed as either DIGITAL-installed or Customer-installed.

#### DIGITAL-INSTALLED

DIGITAL provides installation services for products that are complex to install. Installation services include verification of complete product delivery and standard installation of the product. Hardware and the operating system are installed together.

#### CUSTOMER-INSTALLED

Since many DIGITAL-supported software products require no special skills to install, the customer can install these using the comprehensive, step-by-step documentation sets provided with them. The documentation sets detail all procedures necessary for proper installation. Once the software products have been installed, they too qualify for warranty service.

#### ADD-ON INSTALLATION FEE

With the exception of operating systems, customers who buy DIGITAL-Supported/DIGITAL-Installed products as add-ons after the original system is installed will be quoted an installation fee.

### CUSTOMER-SUPPORTED Binary License Option

This is a standard binary license which includes media, manuals, documentation but no warranty support. It is only offered when a DIGITAL-supported license option is not offered.

### LICENSE-ONLY Option

A license-only option is a standard binary license, but has no media, manuals, documentation or support. Software products can be ordered at considerable cost reduction, but the customer must first purchase a license with media for that particular software product. The license-only option is a one time right to copy. It is a license to run a single software product on one additional CPU using a copy of the software the customer made from the original licensed product. Customers may order additional copies of the documentation.

### OUT-OF-WARRANTY Update Option

A customer with a binary license may order a product update for each licensed CPU. An additional license fee is charged for each product update and for each one-time right to copy the update for each previously licensed CPU.

**EJEMPLO DE UN CONTRATO DE LA EMPRESA**

**SISTEMAS LOGICOS**

CONTRATO QUE CELEBRAN, POR UNA PARTE -----  
-----, DEBIDAMENTE REPRESENTADA EN ESTE ACTO POR  
-----, EN SU CARACTER DE -----,  
-----, A LA QUE EN ADELANTE SE DENOMINARA "-----", Y POR  
LA OTRA. "SISTEMAS LOGICOS", REPRESENTADA POR EL ING. ALFREDO  
BRONSOILER FRID, EN SU CARACTER DE DIRECTOR DE DESARROLLO, AL  
TENER DE LAS SIGUIENTES DECLARACIONES Y CLAUSULAS.

## DECLARACIONES

### I. DECLARA "-----"

a) Que tiene su domicilio en -----  
-----.

b) Que cuenta con la capacidad y la disposición de adquirir  
la licencia para utilizar el paquete LogiCat y los mate-  
riales auxiliares.

### II. DECLARA "SISTEMAS LOGICOS"

a) Que es una empresa formada como causante menor bajo el  
nombre de Alfredo Bronsoiler Frid y establecida conforme a  
las leyes mexicanas, tal como lo acredita con su cédula de  
empadronamiento 1218494 y R.F.C. BOFA-570911.

b) Que se dedica al desarrollo de programas de microcomputa-  
dores, enfocados a automatizar todos los procesos y servi-  
cios bibliotecarios.

c) Que son los autores y propietarios del paquete LogiCat  
para automatización de catálogos de material bibliográfico,  
por lo que está en capacidad y disposición de autorizar y  
facilitar el uso de dicho paquete así como de los materiales  
auxiliares.

d) Que el Ing. Alfredo Bronsoiler, es su Director de Desa-  
rrollo y que tiene facultades para celebrar esta clase de  
actos jurídicos.

e) Que tiene su domicilio en la Calle de Goethe No. 107,  
México 11590, D.F.

## CL A U S U L A S

PRIMERA.- El presente contrato tiene por objeto que "SISTEMAS LOGICOS" se obligue a entregar los bienes y servicios que mas adelante se especifican y "-----" se obligue a pagar el precio pactado y a ceñirse a las condiciones establecidas en el mismo.

SEGUNDA.- "SISTEMAS LOGICOS" otorga a "-----" la licencia para el uso del paquete LogiCat y del Manual de Operaciones de dicho sistema. Todo esto será entregado a "-----" para que lo utilice únicamente en sus instalaciones de la Ciudad de México.

TERCERA.- "-----" pagará a "SISTEMAS LOGICOS" la cantidad de \$300,000.00 M.N. (TRESCIENTOS MIL PESOS 00/100 M.N.), al contado, más \$45,000.00 M.N. (CUARENTA Y CINCO MIL PESOS 00/100 M.N.), por concepto de I.V.A. por el uso de la licencia del paquete LogiCat, la capacitación de personal y el manual de operación del sistema.

CUARTA.- "SISTEMAS LOGICOS" no se hace responsable de los daños que pudiera ocasionar el uso del paquete LogiCat, sin embargo, en caso de que el sistema o los materiales impresos suvieran defectos, éstos serán reemplazados por "SISTEMAS LOGICOS", sin costo alguno durante los primeros seis meses después de firmado el contrato.

QUINTA.- Será responsabilidad de "-----" contar con materiales de soporte adecuados para la correcta operación de LogiCat, tales como, un computador compatible, diskettes de proyección, papelería y tarjetas catalográficas.

SEXTA.- "SISTEMAS LOGICOS" capacitará al personal que designe "-----" en el uso del sistema LogiCat y de los materiales auxiliares durante un periodo de 20 horas en su domicilio de la Ciudad de México sin costo alguno.

SEPTIMA.- "-----" podrá hacer uso de un contrato anual de mantenimiento del paquete LogiCat que abarca el acceso a todas las versiones actualizadas del sistema y 20 (treinta) horas de capacitación y asesoría.

OCTAVA.- Cualquier modificación o asesoría posterior a la instalación del sistema fuera del contrato de mantenimiento, causará honorarios que serán negociados por ambas partes.



NOVENA.- "-----" reconoce no ser propietaria del paquete LogiCat, a pesar de que podrá utilizarlo como mejor convenga a sus fines, comprometiéndose a no ceder los derechos o a transferir los materiales a otras personas, instituciones o empresas sin el consentimiento de "SISTEMAS LOGICOS", así como a hacer todo lo posible por protegerlos contra reproducción, publicación o distribución no autorizadas.

DECIMA.- Para todo lo no expresamente convenido, así como para la interpretación o cumplimiento de este contrato, las partes se someten a la jurisdicción y competencia de los tribunales de la Ciudad de México, por lo que renuncian a invocar todo fuero que pudiera corresponderles en razón de su domicilio presente o futuro.

Leído el presente y conformes ambos contratantes, lo ratifican y firman en la Ciudad de México, D.F. el 8 de Junio de 1984.

Por "SISTEMAS LOGICOS"

Por "-----"

Ing. Alfredo Broncoiler Frid  
Dir. de Desarrollo

-----



**Sistemas  
Lógicos**

Goethe 107 México, D. F. 11580  
545-9611 562-0293 576-8438

Condiciones de Pago:

50% al firmar contrato y 50% contra entrega del equipo (cotizados de acuerdo al tipo de cambio).

Tiempo de Entrega del Equipo de Computo:

Seis semanas después de fincado su pedido correspondiente.

Garantía del Equipo:

Tres meses.

Origen del Equipo:

El equipo de cómputo es de integración nacional.

Mantenimiento:

Por póliza, el 1% mensual sobre el valor del equipo de cómputo.

**Nota Importante**

Para seguridad del PUC y de los usuarios de la UNAM, por medio de la presente nos comprometemos a que en el caso de que la UNAM utilice alguna de las opciones anteriores y nuestra empresa desaparezca o quiebre, los programas fuente serán entregados a la misma por un costo igual a tres veces el precio en vigencia del sistema derazado.

Sin más por el momento, quedamos a su disposición para aclarar cualquier duda que tenga sobre la presente cotización.

Atentamente

Ing. Alfredo Bronsoller  
Director de Ventas y Desarrollo  
Sistemas Lógicos

C.c.p. Dr. Leopoldo Zea, Director, CCYDEL  
C.c.p. Lic. Alicia Gajman, Coordinadora, ENEP Acatlán  
C.c.p. Ing. Carlos Strasburger, PUC

Se anexa:

1. Contrato de licencia del sistema LogiCat
2. Contrato de mantenimiento de sistemas



Sistemas  
Lógicos

Goethe 107 México, D. F. 11590  
545-9611 562-0293 576-8438

<u>Cant.</u>	<u>D e s c r i p c i ó n</u>	<u>P. Total</u>
1	Computadora Micron PC/X de 16 bits (99% compatible con la IBM PC), con: 128 K de memoria, monitor y teclado importados, una unidad de disco flexible con capacidad de 360 K, un disco duro de 20 MB, dos puertos seriales RS-232, un puerto paralelo tipo Centronix, graficación, sistema operativo MS-DOS y Basic.	\$ 8,700.00 U.S.
1	Impresora ATI-II de 165 cps.	\$ 2,024.00 U.S.
	<b>S U B T O T A L</b>	<b>\$10,724.00 U.S.</b>
	<b>15% DE DESCUENTO</b>	<b>\$ 1,609.00 U.S.</b>
	<b>T O T A L</b>	<b>\$ 9,115.00 U.S.</b>

#### Micron PC:

El costo del computador Micron PC con dos unidades de diskette de 360 K cada una y las mismas características que la PC/X (sin el disco duro) es de \$5,300.00 U.S., a lo que también se le aplica el 15% de descuento.

#### Otros Descuentos:

Existen otros descuentos para volúmenes que se logren durante un mismo mes, de la siguiente manera:

1-4	15%
5-10	15+5+5%
11-	15+5+5+5%

#### Impuestos:

Todos los rubros causan I.V.A.

#### Vigencia de la Cotización:

Treinta días.

#### Forma de Pago:

Los dólares para el pago de los equipos Micron se cotizan 10 pesos arriba del tipo de cambio preferencial.



**Sistemas  
Lógicos**

Goethe 107 México, D. F. 11590

545-9611 562-0295 578-8438

En resumen, los descuentos en sistemas por volumen de compra, quedan de la siguiente manera:

	1	2-5	6-10	11-
Solo sistema con capacitación	100%	90%	80%	70%
Solo sistema sin capacitación	100%	70%	60%	50%
Sistema/equipo con capacitación	100%	85%	70%	60%
Sistema/equipo sin capacitación	100%	65%	50%	30%

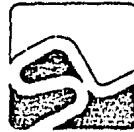
#### **Economía de Escala en Equipos**

El equipo de cómputo que recomendamos para automatizar los procesos de su Biblioteca, es compatible con la microcomputadora más popular en los Estados Unidos (IBM-PC). La Micron PC/X, es una computadora de integración nacional y cuenta con centros de servicio y partes dentro del país.

Este equipo puede ser utilizado además para investigación, desarrollos de software locales, y enseñanza. Las principales ventajas que tiene el equipo son:

1. Compatibilidad con IBM
2. Graficación en 16 colores con resolución de 640 X 200
3. Amplio margen de crecimiento en memoria y disco
4. Variedad en lenguajes de cómputo y paquetes de aplicación, entre otros:

- APL
- Lotus 1-2-3
- Logo
- Modula-2
- C
- Context MBA
- Pascal
- Cobol
- WordStar
- dBASE II



**Sistemas  
Lógicos**

Goethe 107 México, D. F. 11590  
545-9611 562-0295 578-8438

Con el fin de reducir los costos por sistema, se ofrece la siguiente alternativa donde se ofrece la asesoría una sola vez al PUC y esta dependencia se encarga de asesorar a los usuarios de nuestros paquetes; la alternativa incluye:

1. Sistema personalizado
2. Manual de Operación
3. Versión más actualizada

Primer sistema	100% precio vigente
2-5	70 %
6-10	60 %
11-	50 %

#### Combinaciones Sistemas/Computadora

Como distribuidores de las microcomputadoras Micron, nuestra empresa ofrece la siguiente combinación:

- Sistema de cómputo
- Computadora Micron PC/X o PC \*
- Impresora ATI-II

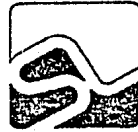
Con la adquisición de sistemas con asesoría y equipo de cómputo, se ofrece el 15% de descuento en la computadora y los siguientes descuentos en los sistemas:

Primer sistema	100% precio vigente
2-5	85 %
6-10	70 %
11-	60 %

En caso de no desear la asesoría y capacitación con cada sistema, se ofrece el paquete sistema/equipo con un descuento del 15% en el equipo y los siguientes descuentos en sistemas:

Primer sistema	100% precio vigente
2-5	85 %
6-10	50 %
11-	30 %

\* Ver las características en la siguiente página.



**Sistemas  
Lógicos**

Goethe 107 México, D. F. 11590  
545-9811 562-0293 576-8438

La actualización fuera de contrato se cotiza al 20% del precio al público del sistema.

**Notas:**

Vigencias: 30 días. Precio sujeto a cambios sin previo aviso.

### Derechos de Reproducción

Cada sistema es personalizado a la Institución que lo adquiere y es intransferible por contrato.

En el caso particular de la UNAM, podemos ofrecerle los derechos de reproducción ilimitada para sus dependencias únicamente, por un costo de \$2,000,000.00 M.N. por sistema.

Esta alternativa incluye:

1. Sistema personalizado como Universidad Nacional Autónoma de México con derechos de reproducción
2. Un curso de 30 horas de capacitación
3. Un manual de operación con derechos de reproducción

Si la UNAM desea capacitación para alguna dependencia en particular, se cotiza en un 40% del precio al público vigente.

**Nota**

La UNAM recibiría únicamente el programa objeto.

Se aplican las mismas notas que se mencionaron arriba.

### Economía de Escala en Sistemas

Otra alternativa es la de ofrecer descuentos por cada sistema que se adquiriera en forma independiente y a través del tiempo, de la siguiente manera:

1. Cada sistema se entrega con capacitación (20 horas) y manual de operación
2. El sistema se entrega personalizado
3. Se entrega la versión más actualizada

Primer sistema	100% precio vigente
2-5	90 %
6-10	80 %
11-	70 %



**Sistemas  
Lógicos**

Goethe 107 México, D. F. 11590  
545-9611 562-0295 576-8438

DocuSis ha sido adquirido por la Universidad Iberoamericana para utilizarse como apoyo en la catalogación de la Colección Porfirio Díaz.

Nuestros sistemas han sido implementados en Instituciones que no cuentan con personal técnico en computación y han obtenido resultados inmediatos por las razones siguientes:

1. Los sistemas están bien documentados
2. Son fáciles de operar
3. Se ofrecen en forma gratuita 20 (veinte) horas de capacitación y asesoría especializada por cada sistema
4. Los sistemas se adaptan a las necesidades de la Biblioteca y no viceversa

El aspecto de cooperación entre Bibliotecas, cada vez atrae a nuevos usuarios, ya que al contar con bases de datos compatibles, podrán compartir información y recursos materiales para aprovechar óptimamente los presupuestos asignados.

Además, puede ser comprobado los usuarios actuales de los sistemas, que nuestra empresa está siempre dispuesta a efectuar adaptaciones, modificaciones y los apoya con capacitación y asesoría en forma rápida y seria.

#### Cotización:

El precio total de cada sistema es de \$300,000.00 M.N. e incluye:

- Programa de cómputo
- Manual del usuario
- Veinte horas de capacitación y asesoría

#### Inuestos:

Todos los rubros causan I.V.A.

#### Garantía de los Sistemas:

Un año.

#### Mantenimiento:

Por contrato anual incluye: todas las versiones actualizadas y 20 (veinte) horas adicionales de capacitación y asciende a 30% del precio al público en vigencia del sistema.

CONTRATO DE MANTENIMIENTO DE SISTEMAS DE COMPUTACION

Que celebran por una parte **Sistemas Lógicos**, representada por **Alfredo Bronsoiler Frio** (P.F.C. BOPA-570911 y cédula de empadronamiento 1218494), como empresa capaz y que proporcionará los servicios de actualización y mantenimiento, y

-----  
como usuario de sistemas que desea ser cubierto por este contrato, quienes lo sujetan a las siguientes cláusulas:

PRIMERA.- La duración del contrato será por un periodo de 12 meses contados a partir del

SEGUNDA.- Los sistemas objeto de este contrato son:

Nombre del Sistema	Número de Serie
-----	-----
-----	-----
-----	-----

TERCERA.- El presente contrato es por la cantidad de -----  
----- más impuestos, pagaderos a la firma del contrato.

CUARTA.- **Sistemas Lógicos** se compromete a reparar o reemplazar, sin cargo alguno, cualquier falla de funcionamiento de los sistemas durante la vigencia del contrato.

QUINTA.- Durante la duración del presente contrato, **Sistemas Lógicos** proveerá al usuario, sin cargo alguno, todas las versiones actualizadas de los sistemas.

SEXTA.- **Sistemas Lógicos** proveerá al usuario veinte horas (20) de asesoría por cada sistema para cuestiones técnicas en relación con la instalación y utilización de los sistemas. La asesoría adicional se cotizará a

----- por hora.



SEPTIMA.- En caso de cancelación del presente contrato por la parte contratante, Sistemas Lógicos no tendrá obligación de regresar los pagos realizados.

OCTAVA.- Sistemas Lógicos no será responsable por daños, perjuicios, pérdidas de utilidades u otros daños consecuenciales que pudieran resultar por fallas del sistema ni por retrasos ocasionados por la reparación del mismo. Sistemas Lógicos hará su mejor esfuerzo para resolver las fallas en los sistemas a la brevedad posible.

NOVENA.- Para el cumplimiento de este contrato, ambas partes se someten expresamente a la jurisdicción de los tribunales del Fuero Común de la Ciudad de México; renunciando expresamente a los beneficios del fuero, que por domicilio posterior, pudiera corresponderles.

Leído el presente y conforme ambos contratantes, lo ratifican y firman en la Ciudad de México, D.F., el \_\_\_\_\_

-----  
-----  
-----  
-----  
-----

-----  
Ing. Alfredo Bronsoiler  
Director de Desarrollo  
Sistemas Lógicos



**Sistemas  
Lógicos**

Goethe 107 México, D. F. 11590  
545-9611 562-0295 576-8438

México, D.F. a 24 de Junio de 1984

Ing. Jorge Gil  
Director  
Programa Universitario de Computo  
Ciudad Universitaria, D.F.

Estimado Ing. Gil:

De acuerdo a las pláticas que sostuvimos el día 22 del presente, y en virtud de la demanda que han tenido nuestros sistemas de cómputo entre algunas dependencias de la UNAM, nos permitimos presentarle una nueva proposición.

La propuesta está dividida en las siguientes partes:

1. Información general
2. Derechos de reproducción
3. Economía de escala en sistemas
4. Combinaciones sistemas/computadora
5. Economía de escala en equipos

### Información General

Nuestra empresa se dedica principalmente a la automatización de Bibliotecas y Centros de Información, para lo cual contamos con diversos programas terminados e implementados, otros en desarrollo y algunos más en diseño, todos ellos para sistemas CP/M y MS-DOS:

- LogiCat - Catalogación
- LogiPres - Préstamo
- DocuSis - Catálogos de documentos y correspondencia
- Periódicas - Control de publicaciones periódicas
- LogiCom - Sistema de adquisiciones

En forma breve, a continuación se explican las características de los tres primeros:

#### **1. LogiCat**

Sistema de apoyo a procesos técnicos y servicios de información. Abarca los siguientes aspectos:



**Sistemas  
Lógicos**

Goethe 107 México, D. F. 11590  
545-9611 562-0295 576-8438

- Captura de fichas catalográficas en formato MARC
- Modificación de fichas en la Base de Datos
- Recuperación con diversas llaves y operaciones booleanas
- Impresión de reportes alfabéticos por autor, título, temas, clasificación, etc. (se ofrece la opción de grabar en disco para producir microfichas y galerías de impresión)
- Impresión de tarjetas catalográficas bajo las normas angloamericanas de catalogación; incluye desarrollo de fichas.

LogiCat ha sido adquirido por las siguientes Instituciones:

- Universidad Iberoamericana - Programa Universitario Justo Sierra
- Centro Universitario de Investigaciones Bibliotecológicas
- Instituto Nacional de Administración Pública
- Universidad de Guanajuato
- Universidad Autónoma de San Luis Potosí

## 2. LogiPres

Sistema automatizado de circulación. Abarca:

- Alta, baja y modificación de copias
- Informes sobre inventarios por título
- Alta, baja y modificación de usuarios
- Informes sobre datos personales y administrativos de los usuarios
- Préstamos internos, a domicilio, interbibliotecarios, devoluciones y extensión de plazos.
- Impresión de multas, cartas de no adeudo, recordatorios periódicos, credenciales y tarjetas de préstamo
- Plazos de préstamo variables
- Estadísticas

## 3. DocuSis

DocuSis es un sistema de apoyo a procesos técnicos y servicios de información. Abarca los siguientes aspectos:

- Captura de fichas catalográficas
- Modificación de fichas en la Base de Datos
- Recuperación con diversas llaves y operaciones booleanas
- Impresión de reportes alfabéticos diversas entradas a la ficha (se ofrece la opción de grabar en disco para producir microfichas y galerías de impresión)
- Impresión de tarjetas catalográficas en formato apto para publicación de catálogo
- Desarrollo de índices analíticos geográficos, onomásticos y temáticos para consulta del catálogo en forma manual.

## BIBLIOGRAFIA

A cooperative industry study: software development productivity  
Kunkler, J. E.  
Xerox Corporation, May. 1983

Application development without programmers  
Martin, J.  
Prentice Hall

Buying or making the software package  
that is the best for you  
L. Golland, Marvin  
Journal of System Management

Computación y cambio social  
Ciencia y Desarrollo  
Dr. Enrique Calderón Alzate  
pp. 22-29

CRITICAL ISSUES IN SOFTWARE  
A guide to software economics,  
strategy, and profitability  
Werner L., Frank  
John Wiley & Sons (1983)

Database Design  
Wiederhold, Gio  
Mc Graw Hill

Exploratory experimental studies comparing online and  
offline programming performance  
Sackman, H. y Col.  
CACM. Vol. 11, No. 1 (1968)  
pp. 3-11

Language barrier to productivity  
DATAMATION. FEB. 1983  
pp. 209-212

Lenguajes de Cuarta Generación y productividad en el  
desarrollo de sistemas  
Burroughs Corp.

Los aspectos legales de la informática  
ComputerWORLD/México  
Lic. Luis Vera Vallejo  
Año 5, Núms. del 101 al 110  
pp. 11

Productivity in computer application development  
Rudolph, F.  
Depto. of Management Studies Working paper No. 9  
Univ. of Auckland, Mar. 1983

Structured Design  
Fundamentals of a Discipline of Computer  
Program and Systems Design  
Yourdon, Edward and  
L. Constantine, Larry  
Prentice Hall

System software survey  
Data Decisions  
DATAMATION. December 1982  
Volumen 28, Number 13  
pp. 96-138

System software survey  
Data Decisions  
DATAMATION. December 1983  
Volumen 29, Number 12  
pp. 104-150

System software survey  
Data Decisions  
DATAMATION. December 1984  
Volumen 30, Number 20  
PP. 85-138

The golden age of packaged software  
G. Withington, Federic  
DATAMATION. DECEMBER 1980

The New tower of Babel  
Fhrman, J. R.  
DATAMATION. March 1980  
pp. 157-160