

24.20



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

ANALISIS, DISEÑO Y CONSTRUCCION DEL SISTEMA
DE REINSCRIPCIONES EN LA U. N. A. M.,
UTILIZANDO LAS TECNICAS DE LA
INGENIERIA DE SOFTWARE

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A
JOSE MARTINEZ COLIN

DIR. M. EN C. CARLOS RAMOS LARIOS



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

Página

Introducción	I
<u>PRIMERA PARTE</u>	
Capítulo I : Descripción del Sistema de Control Escolar en la U.N.A.M.	
I.1 Antecedentes	
I.1.A El desarrollo de la computación en la U.N.A.M.	1
I.1.B El desarrollo del sistema administrativo en la U.N.A.M.	3
I.2 Necesidad del sistema computarizado	9
I.3 La reinscripción dentro del Sistema de Control Escolar	10
Capítulo II: Introducción a la Ingeniería de Software	
II.1 Fase de planeación de un sistema	17
II.1.1 Definición del sistema	18
II.1.2 Planeación del software	23
II.1.3 Análisis de los requerimientos	32
II.2 Fase de desarrollo	44
II.2.1 Proceso del diseño	44
II.2.2 Codificación	47
II.2.3 Integración	51
II.2.4 Pruebas del software	53
II.3 Fase de mantenimiento	57
II.3.1 Características del mantenimiento	59
II.3.2 Efectos del mantenimiento	60
II.3.3 Manual del usuario	60
II.4 Resumen	61
II.5 Técnicas y herramientas de la Ingeniería de Software	63
II.5.1 Su concepto	63
II.5.2 Programación estructurada	63
II.5.3 Pseudocódigo	63
II.5.4 Técnica de "Arriba-hacia-Abajo"	69
II.5.4 Diagramas HIPO	71
II.5.5 Modularidad	72

SEGUNDA PARTE

Página

Capítulo III: Planeación del sistema de reinscripciones

Reporte para la especificación de los requerimientos de software	75
1.0 Introducción	76
2.0 Descripción de la información	78
3.0 Descripción de las funciones	31
4.0 Criterio de validación	86
5.0 Bibliografía	87

Capítulo IV: Desarrollo del sistema de reinscripciones

IV.1 Especificación del diseño para el sistema de reinscripciones en la U.R.A.A.	89
1.0 Alcance del diseño	90
2.0 Descripción del entorno	95
3.0 Descripción del diseño	98
4.0 Descripción de los módulos	103
5.0 Interfase externa	122
6.0 Equipo y lenguaje	126
7.0 Recuperación de fallas	126
8.0 Certificación de calidad	127
9.0 Apéndices	
a) Estructura física de los archivos	128
b) Diferentes tipos de "celdas"	132
IV.2 Codificación	
IV.2.1 Estilo de codificación	134
IV.2.2 Integración	139
IV.3 Pruebas	
IV.3.1 Análisis de entrada-salida	148
IV.3.2 Por cobertura de lógica	150

Capítulo V: Mantenimiento del sistema

V.1 Manual del usuario	153
I) Guía de instalación	154
II) Guía de operación	157
III) Guía de mantenimiento	163
Apéndices	
a) Campos para la captura de datos	165
b) Diagnósticos del sistema	166
c) Relación de los planteles	170
d) Formas de modificación	172

Conclusiones 173

Bibliografía 174

INTRODUCCION

Es sabido que la U.N.A.M. fue una de las primeras instituciones en comenzar la era de la computación en México al instalar en 1950 en el Centro de Cálculo electrónico una computadora IBM-650. De igual manera, fue en la U.N.A.M. donde se hicieron los primeros esfuerzos por desarrollar sistemas automatizados apoyados en la computación, entre los que cabe mencionar el sistema de registro y control escolar.

Debido a este inicio tan temprano, tanto el hardware como el software con que se contaba era, aunque muy útil y conveniente en su momento, muy limitado si se le observa con los ojos de las posibilidades de hoy en día. Es por este motivo que la U.N.A.M. ha venido actualizándose en el hardware del equipo, y así, mientras que en el año de 1972 se adquiría un computador Burroughs 5500 para atender los requerimientos de la administración escolar, administración central, docencia e investigación, hoy en día en cambio, se cuenta con una computadora Burroughs 6800 específicamente para la administración escolar, y otras tres para las áreas de trabajo mencionadas anteriormente.

En contraste con las mejoras en el hardware, durante muchos años el software se mantuvo en todo el mundo un paso atrás, y el sistema de registro y control escolar no fue la excepción. El principal motivo de este retraso fue siempre la falta de una disciplina completa para la planeación, desarrollo y mantenimiento de sistemas en información. Y no fue sino hasta finales de la década de los 70 e inicios de los 80 cuando surge la disciplina conocida con el nombre de "Ingeniería de Software", en la cual se reconoce la necesidad de su existencia, y lo más importante desde el punto de vista práctico: se proponen un conjunto de técnicas para resolver el problema, y así surgen entonces las llamadas "técnicas estructuradas" para el desarrollo de sistemas.

Aunque cada vez más difundidas en otros países en el ámbito práctico, en México estas técnicas no han tenido la difusión que se considera que debieran tener, y se han limitado muchas veces solamente a la exposición dentro de clases de la Facultad de Ingeniería, siendo uno de los motivos más determinantes el que quizá falte en el medio académico un mayor estudio, evaluación y aplicación de estas técnicas a sistemas prácticos, considerando las condiciones de nuestro medio, para poder realizar una verdadera transferencia de tecnología.

El presente trabajo de tesis: "Análisis, diseño y construcción del sistema de reinscripciones en la U.N.A.M., utilizando las técnicas de Ingeniería de Software" obedece pues, a la necesidad mencionada

anteriormente, y aprovechando la magnífica oportunidad que se presenta en la que los requerimientos actuales del sistema de reinscripciones de la U.N.A.M. no son satisfechos plenamente por el sistema actual, por lo que se requiere retomar el problema y replantearlo considerando que ahora existen nuevas necesidades, nuevo hardware y nuevas técnicas para el desarrollo del software.

Para ello se divide el trabajo en dos grandes partes: en la primera, se presentan brevemente tanto los antecedentes computacionales como del sistema administrativo en la U.N.A.M. (capítulo I) y la teoría sobre la Ingeniería de Software con sus principales técnicas (capítulo II). En la segunda parte se aplica la teoría analizada al sistema específico de la reinscripción en la U.N.A.M. y por tanto, se presentan las tres fases de que consta todo sistema computacional: la planeación (capítulo III), su desarrollo (capítulo IV) y su mantenimiento (capítulo V).

Finalmente, conviene resaltar el hecho de que el sistema de reinscripciones es un sistema tal que, aunque corresponde a una institución académica por autonomía, su carácter es más bien administrativo, por lo que las conclusiones que se obtienen de este trabajo son directamente aplicables a muchos sistemas de este tipo que se están desarrollando y que deberán desarrollarlo los ingenieros en computación de nuestro país en los próximos años.

Por último, antes de entrar en materia, deseo agradecer de modo especial al Dr. Carlos Ramos Larios que amablemente aceptó dirigirme la tesis y cuyos valiosos consejos e indicaciones hicieron posible la realización de este trabajo; a la Coordinación de la Administración Escolar y muy en especial al Ing. Luis Carrasco, subdirector, por darme la oportunidad de desarrollar el presente trabajo y a cada uno de los compañeros de la Unidad de Registro y Control Escolar que con su ayuda y amistad facilitaron su realización.

DESCRIPCION DEL SISTEMA DE CONTROL ESCOLAR EN LA U.N.A.M.

I.1 ANTECEDENTES

A continuación se presentan dos aspectos respecto al control escolar: primero, el desarrollo de los sistemas automatizados de control y registro escolar a lo largo de la vida de la U.N.A.M.; y segundo, la descripción de los antecedentes relacionados con la introducción de los primeros computadores en la U.N.A.M., el uso que se les asignó y su relación con los sistemas automatizados de la administración escolar.

I.1.A) El desarrollo de la computación en la U.N.A.M.

La Universidad Nacional Autónoma de México fue la primera institución de enseñanza superior, que en función del elevado número de alumnos inscritos, desarrolló e implementó sistemas de control y registro escolar automatizados.

La necesidad de contar con un sistema confiable, capaz de registrar de manera oportuna y veraz los movimientos que se generan para el registro de cada alumno inscrito, durante toda su vida académica, ha motivado una continua inquietud de las autoridades universitarias para implantar sistemas automatizados que cumplan con las características mencionadas.

De hecho, se puede pensar que la adquisición de los primeros computadores electrónicos en la U.N.A.M. es motivada por la inquietud en desarrollar aplicaciones de tipo administrativo; tal es el caso de los sistemas de la nómina de los empleados universitarios, la contabilidad, y posteriormente, los sistemas para el control escolar.

El inicio del procesamiento de datos en la U.N.A.M. es en 1958, fecha en que se adquiere una computadora IBM-650, siendo la primera instalada en una universidad en América Latina. Su finalidad se asignó al desarrollo de proyectos de investigación en las ramas de Ingeniería, Matemáticas, Física, Economía y Administración de empresas.

Paralelamente a este evento se creaba en ese entonces el Centro de Cálculo Electrónico, cuyo principal objetivo era implantar y promover el uso de la computadora en apoyo a la investigación y la docencia.

En el año de 1961 se implanta el examen de admisión para aspirantes a primer ingreso al nivel profesional en la U.N.A.M., contando con el apoyo de sistemas

automatizados. También se aplica el mismo tipo de exámenes para el ingreso a la Escuela Nacional Preparatoria en 1963.

Fue hasta 1965 cuando se instala la primera computadora con la finalidad de atender los requerimientos de los sistemas de control escolar y los administrativos. Un año después, se pone en funcionamiento un segundo equipo de cómputo en el Departamento de Sistemas del Patronato Universitario con la idea de desarrollar los sistemas de nómina y contabilidad en la U.N.A.M.

En 1967 se crea la Dirección General de Sistematización de Datos, que entre sus funciones básicas está la de integrar todos los sistemas computarizados de apoyo a la administración universitaria, y entre ellos el de control escolar. Dos años después, se integra en el Centro de Cálculo Electrónico una infraestructura de personal dedicado al estudio de aplicaciones relacionadas con la computación; de ahí se derivan proyectos para la reestructuración del uso de la computación en aplicaciones de tipo administrativo que motivaron la creación del Centro de Investigación en Matemáticas Aplicadas, Sistemas y Servicios (CIMASS). Dentro de sus funciones esta la investigación en las disciplinas mencionadas, la del procesamiento de datos para toda la universidad y la prestación de servicios de desarrollo e implantación de sistemas a las dependencias de la U.N.A.M.

En 1971, el CIMASS se encarga de diseñar y desarrollar el "Sistema Automatizado de Registro y Control Escolar" (SARCE). En 1976, como resultado del continuo crecimiento de las aplicaciones académico-administrativas, y en particular las de la administración escolar, el Centro de Servicios de Cómputo (CSC) y la Coordinación de la Administración Escolar (CAE) acuerdan desconcentrar la responsabilidad de la operación de los sistemas escolares, evitando así triangulaciones innecesarias que hacían menos expedito el manejo de la información. Quedando el CSC como responsable únicamente de proporcionar el tiempo de máquina necesario y los servicios asociados a él, tales como asesoría, operación, etc., garantizando el buen funcionamiento de los equipos; la CAE, a su vez, queda como la única responsable del manejo y control de todo el proceso administrativo escolar de la U.N.A.M.

Una de las primeras tareas a la que se abocó la Subdirección Técnica de la CAE fue la de reconstruir los archivos de historias académicas del Colegio de Ciencias y Humanidades y de crear el Departamento de Kardex Electrónico.

En 1973, se inicia una serie de cambios estructurales en los sistemas de registro y control escolar, que consisten en el desarrollo de nuevos programas y

procedimientos administrativos-escolares que permitan capturar la información con mayor seguridad, confiabilidad y rapidez. Es cuando surge la utilización de la lectura óptica aplicándola en las actas de examen, en las solicitudes de reinscripción y en las solicitudes de examen extraordinario.

En 1981 se inicia el diseño del "Sistema Computarizado de Administración Escolar" (SISCAE), que aparte de contemplar los requerimientos vigentes, debe prever las necesidades futuras a través de un sistema estructurado que permita flexibilidad, confiabilidad y control.

Es así que en 1982 se concluye la programación del nuevo sistema de las historias académicas, y al año siguiente se reemplaza el subsistema de profesores por uno nuevo y se implementa el subsistema de exalumnos.

A partir de entonces, se ha venido cambiando cada uno de los programas de tal manera que no se resientan los servicios que se prestan a las diversos plantales. Para ello, se realiza en paralelo el nuevo sistema, sustituyendo al anterior una vez que se cumplen todos los requerimientos. Este fue el caso del subsistema de reinscripciones que se muestra en el presente trabajo.

I.1.B) El desarrollo del sistema administrativo escolar en la U.N.A.M.

Paralelamente al desarrollo computacional, dentro de la universidad se fueron modificando las estructuras administrativas en áreas como la del registro y control de alumnos, la contabilidad, la emisión de los cheques para el pago de profesores y trabajadores universitarios, etc.

En el año de 1933 fue creado el Departamento Escolar con el objeto de atender los asuntos relacionados entre los alumnos y la administración, tales como su inscripción, el registro de calificaciones, la expedición de certificados, etc.

Años más tarde, en 1962, las funciones principales de la Administración Escolar se podían resumir en las siguientes:

- * Observar y aplicar las normas y políticas aprobadas por el H. Consejo Universitario, así como los acuerdos dictados por el Rector y demás autoridades con relación a la inscripción de los alumnos y el control de sus estudios.
- * Realizar la inscripción misma de los alumnos de

primer ingreso y efectuar la reinscripción de los alumnos ya inscritos.

- * Expedir las actas y boletas de examen, así como el historial académico.
- * Realizar la revalidación de estudios de los alumnos provenientes de otras instituciones educativas.
- * Procesar en el Departamento Central de Máquinas las estadísticas escolares, listas de alumnos, boletas de calificaciones, etc.
- * Expedir documentos certificados relacionados con la revisión de estudios realizados por el alumno, su certificado de estudio, los títulos profesionales, etc.

Para llevar a cabo las funciones mencionadas, la Dirección General de Servicios Escolares contaba con una estructura organizativa compuesta por secciones escolares en cada plantel y varios departamentos: de control escolar, de coordinación de estudios, de admisión y credenciales

En el año de 1965 se creó la Oficina de Selección de Alumnos, teniendo a su cargo la elaboración de los exámenes de admisión a nivel profesional y preparatoria. Dos años más tarde, en 1967, debido a la naturaleza misma de las funciones que tenía a su cargo se efectuaron algunos cambios estructurales dividiéndola en departamentos y oficinas.

Posteriormente, durante 1969-1970, ante la necesidad de crear nuevas unidades, se hacen modificaciones en la organización, agrupándose las principales funciones de la Dirección General de Servicios Escolares de la siguiente manera:

- * Vigilar el cumplimiento de los estatutos, reglamentos, acuerdos y procedimientos relacionados con los trámites escolares.
- * Integrar y mantener actualizado el archivo general del registro escolar de los alumnos.
- * Recibir, transmitir y registrar en los archivos la documentación proveniente de las Escuelas y Facultades.
- * Coordinar con las Divisiones de Estudios Superiores de las Facultades y Escuelas, la determinación de criterios de admisión y de control escolar a los estudiantes.

- * Registrar y controlar los planes y programas de estudio presentados por el H. Consejo Universitario, así como crear claves a las asignaturas de nueva creación.

Para el cumplimiento de sus funciones, la Dirección General de Servicios Escolares se estructuró de la siguiente forma:

Consejo de la Administración Escolar

- Subdirección Ciclo Preparatoria

Departamento de Control Escolar
Oficina de Revisión de Estudios
Oficina de Secciones Escolares

- Subdirección Técnica

Departamento de:
Control de Información
Inscripciones
Registro
Exámenes Profesionales

Oficinas de:

Registro y Planes de Estudio
Control de Programas
Revisiones
Registro de Incripciones
Exámenes
Reinscripción y cambios
Exámenes Profesionales
Expedición de títulos

- Subdirección de Estudios Superiores

Oficina de Admisión
Oficina de Control Escolar

Durante el año de 1971, mientras se desarrollaba SARCE, la estructura administrativa de la Dirección General de Servicios Escolares se transformó, por acuerdo del Rector, en la Coordinación de la Administración Escolar el 10 de diciembre de 1971, y que hasta la fecha, año de 1987, sigue funcionando como tal.

La Coordinación dividió su esquema laboral en cinco unidades de trabajo:

- Unidad de Actualización de Registro
- Unidad de Registro escolar
- Unidad de Recepción y Expedición de Documentos
- Unidad de Registros de Planes de Estudio
- Unidad de Estudios Superiores

También se crearon las Secretarías de Servicios Escolares que sirven de enlace con los planteles, estableciendo de esta manera una comunicación permanente entre los planteles, la CAE y el CINASS.

Coordinación de la Administración Escolar

En su estructura actual, la Coordinación de la Administración Escolar depende de la Secretaría General (Fig. I.1), y está constituida por :

- Subdirección Técnica
- Subdirección de Diseño y Desarrollo de Nuevos Proyectos (SDDNP).
- Subdirección de Certificación de Estudios
- La Unidad de Registro de Planes de estudio

La Coordinación tiene como objetivo fundamental proporcionar atención e información a la comunidad universitaria y a las dependencias estatales y federales en materia de primer ingreso y registro y control escolar de la U.N.A.M., basándose para ello, en el marco jurídico de la propia institución.

Sus principales funciones son:

- Coordinar las actividades de administración escolar
- Mantener comunicación permanente con las autoridades de los planteles a fin de mejorar los servicios por medio de un asesoramiento sobre los trámites escolares.
- Llevar el registro de los planes y programas de estudio aprobados por el M. Consejo Universitario.
- Conferir la inscripción y reinscripción a los alumnos de acuerdo a lo establecido en el Regl
- Coordinar y participar en las actividades relacionadas con el ingreso en la U.N.A.M.
- Tramitar los exámenes profesionales.
- Expedir los títulos, certificados y constancias que otorga la institución.
- Registrar ante la Dirección General de Profesiones los títulos que sean expedidos por la U.N.A.M.

Para llevar a cabo estas funciones, la CAE utiliza diferentes medios, tales como: la computación, la estadística, la administración de recursos humanos,

materiales y tecnológicos, etc. Estos medios organizados metódicamente coadyuvan a una adecuada planeación y desarrollo de la institución.

SUBDIRECCION TECNICA

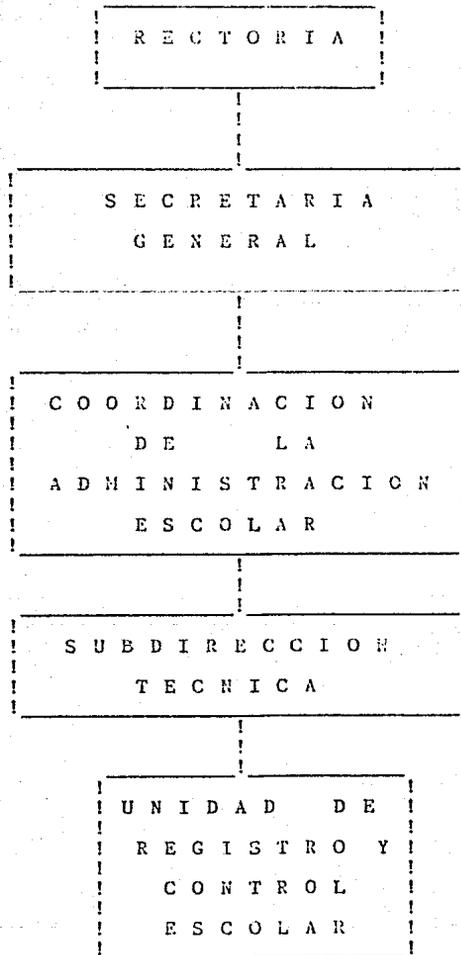
La Subdirección Técnica tiene como objetivo fundamental el proporcionar el servicio de procesamiento de información administrativo-escolar a los planteles que así lo requieran.

Sus funciones principales son:

- Planear, organizar, dirigir, evaluar, supervisar y controlar las actividades de la Unidad de Registro y Control Escolar, y de la Unidad de Historias Académicas y Microfilmación.
- Programar las actividades de la Subdirección Técnica e informar sobre las mismas al coordinador de la administración escolar.
- Programar en coordinación con los planteles los servicios relacionados con las actividades administrativo-escolar en cada período lectivo.
- Recibir, registrar, procesar y emitir los documentos y la información sobre la reinscripción, inscripción, exámenes y calificaciones procedentes de las dependencias docentes de la U.N.A.M.
- Coadyuvar en la elaboración de manuales administrativos para el manejo y control de la documentación e información del área de su competencia.
- Planear, analizar, evaluar y dar mantenimiento a los sistemas computarizados del Registro y Control Escolar e Historias Académicas.
- Resguardar la información de los archivos magnéticos relativos al registro escolar y al historial académico.

Actualmente la Subdirección Técnica está integrada por dos unidades:

- Unidad de Registro y Control Escolar: es la responsable de la reinscripción de los alumnos en el sistema computarizado, tema que aborda el presente trabajo.
- Unidad de Historias Académicas y Microfilmación



La Unidad de Registro y Control Escolar en la U.N.A.M.

Fig. I.1

I.2 Necesidad del sistema computarizado

Aunque el procesamiento de datos en la U.N.A.M. se inicia en 1958, no fue sino hasta el año de 1972 cuando se automatizan algunos procedimientos de la administración escolar, y a partir de entonces los sistemas computarizados han sido dinámicamente transformados para atender las necesidades tanto de demanda como de adecuación a las políticas rectoras y de la legislación vigente.

Actualmente la U.N.A.M. cuenta con 85 Escuelas, Facultades y Colegios distribuidos en tres niveles:

-Nivel Bachillerato:	5	Colegios de Ciencias y Humanidades
	9	Escuelas Preparatorias
-Nivel Licenciatura:	30	Escuelas
	18	Facultades
-Nivel Posgrado:	4	Escuelas
	18	Facultades

Formando todo lo anterior un volumen aproximado para el año de 1986 de :

Alumnos registrados	382,680
Alumnos activos	288,300
Alumnos-Materia al año	3'200,000
Calificaciones registradas	27'500,000
Asignaturas	16,467
Grupos por semestre	45,035

Además, anualmente la población estudiantil se va incrementando aproximadamente en 30,000 alumnos.

Teniendo en cuenta los volúmenes de información que se requieren manejar, así como las necesidades en cuanto rapidez y confiabilidad, se hace imprescindible el uso de la computadora.

En los procesos administrativos, los sistemas computarizados juegan un papel muy relevante por su eficacia y eficiencia en el manejo de información, teniendo en cuenta los constantes cambios a que es sometido un sistema para actualizarlo. El Control Escolar es de los sistemas menos pasivos, pues nunca hay períodos de inactividad; en la universidad siempre hay actividad, y entre los períodos escolares se prepara la estructura administrativa para ellos.

Para llevar a cabo sus actividades computarizadas, la Coordinación de la Administración Escolar cuenta con el servicio proporcionado por la Dirección de Cómputo para la Administración Académica.

La Dirección le facilita el uso del equipo Burroughs 6800 en una de sus instalaciones. La computadora cuenta con un procesador central, manteniendo en línea siete unidades de disco removible con una capacidad de más de mil millones de caracteres. Tiene, además, seis unidades para utilizar cintas magnéticas. La CAE utiliza doce terminales de las 114 posibles de asignar al sistema. Cuatro de estas terminales son microcomputadoras B-20, las cuales pueden trabajar independientes de la B-6800.

Se cuenta además con la posibilidad de utilizar dos minicomputadoras HP-1000, asociadas a dos lectoras ópticas de marcas reflectivas. Ocupa, por último, los servicios de perforación y microfilmación (el capítulo III muestra detalladamente los recursos con que se cuenta).

I.3 La reinscripción dentro del Sistema de Control Escolar

La Coordinación de la Administración Escolar, al ser la encargada de aplicar las políticas de registro y control escolar, necesita siempre tomar en cuenta las disposiciones del H. Consejo Universitario para mantener en sus actividades todo el marco legal.

La CAE, mediante la Subdirección Técnica, lleva a cabo diversas actividades computarizadas, entre las que cabe destacar el "Sistema de Control Escolar", así como el "Sistema de Historias Académicas". A continuación se presenta una breve descripción de la información con que cuenta el "Sistema de Registro y Control Escolar", dejando algunas precisiones para los capítulos III y IV.

El control escolar se realiza a partir de que los alumnos ya han sido inscritos en la U.N.A.M. por primera vez, se realiza durante todo el tiempo en que el alumno cursa su formación académica. Y termina una vez que el alumno concluye sus estudios, pasando a formar parte de exalumnos.

En el tiempo en que el alumno cursa sus estudios, hay una serie de movimientos académicos que realiza y que es necesario registrarlos, tales como el darse de alta o de baja en alguna asignatura, calificarlo, modificarle o actualizarle alguna información, etc. Para ello, es necesario contar con toda la información sobre alumnos, grupos, asignaturas, carreras, planteles, etc. Según los tipos de información, se tienen diversos subsistemas:

- Alumnos:

Este subsistema contiene la información sobre los alumnos inscritos en la U.N.A.M. Como se indica anteriormente, la población estudiantil activa es superior a los 280,000 alumnos, siendo éstos los que realizan reinscripciones normalmente. Todos ellos se encuentran registrados con sus datos generales. Adicionalmente se encuentran registrados en los sistemas computarizados una población inactiva de aproximadamente de 400,000 y más de 300,000 exalumnos. Con ello se mantienen registrados más de un millón de personas.

- Profesores:

Al igual que los alumnos, se tienen registrados todos los profesores que imparten asignaturas en la universidad, con sus datos generales. Se manejan con el mismo criterio que con los alumnos, pero en vez de considerar que tienen asignaturas cursando, son asignaturas impartidas. A su vez, cada profesor tiene asignados diferentes grupos de una asignaturas o varias.

- Planteles:

Cada plantel se encuentra registrado con sus datos generales, tales como el número de asignaturas que se imparte, la cantidad de alumnos inscritos y si están activos o no, la cantidad de alumnos de otro plantel que están reinscritos en él, etc. Se cuenta con 49 planteles para profesional, nueve preparatorias y 5 colegios de ciencias y humanidades (Cfr. cap.V, anexo D.1).

- Carreras:

Para los estudios correspondientes al nivel profesional, se tienen registradas las carreras que imparte, totalizando la cantidad de 155 carreras. Estas contienen diversas características como el número de créditos necesarios para cumplir con ella, y cuantos de ellos son optativos y cuantos obligatorios. Para los estudios correspondientes al nivel medio superior, se tienen en lugar de las carreras el tipo de plan de estudios: tres años para la preparatoria y seis semestres para los colegios de ciencias y humanidades.

-Asignaturas:

Se cuenta con más de 12,300 asignaturas registradas (año de 1986). Entre ellas hay de todo tipo: teóricas, prácticas, clínicas o experimentales; en cuanto a su duración pueden ser mensuales, bimestrales,

semestrales o anuales, etc. A su vez, cada asignatura tiene registrada el número de grupos asignados.

- Grupos:

Semestralmente se ofrecen clases en aproximadamente 45,000 grupos en los niveles medio-superior y superior. A cada grupo puede estar asignado uno o dos profesores. Los grupos pueden ser de ordinario o de extraordinario, pudiendo contar con un límite de alumnos por aceptar; también se contabiliza el número de alumnos reinscritos que son de otro plantel.

El proceso de reinscripción:

Antes de iniciar el período de clases en la U.N.A.M., los planteles reportan a la C.A.E. los diferentes grupos a fin de que sean registrados para el curso por iniciar. Las asignaturas que se imparten no son reportadas a la URCE por los planteles sino por la Unidad de Registro de Planes de Estudio (también dependiente de la C.A.E.). El plantel reporta por su parte a los profesores asignados a impartir las diversas asignaturas-grupo.

Existen varios mecanismos mediante los cuales los alumnos pueden realizar el trámite de su reinscripción, siendo el más común utilizar hojas para lectura óptica. En dichas formas los alumnos anotan las claves de las asignaturas y grupos en que desean estar reinscritos. Con ello se responsabiliza al alumno de la información solicitada, obteniendo además una gran agilidad y confiabilidad en el sistema.

Ya obtenida la información, se procede a ejecutar el Sistema de Reinscripciones, produciéndose después algunos diagnósticos que permiten contemplar los resultados y remitirlos a los diversos planteles. Este es el proceso del que se trata en el presente trabajo.

Posteriormente existe un período en que se puede modificar la reinscripción, ajustando las disconformidades que se encuentran.

Una vez que se tiene registrada la reinscripción definitiva, la URCE genera una serie de reportes, entre los que se cuenta:

- Listas de asistencia: donde se muestra cada asignatura-grupo con la relación de alumnos ordenada alfabéticamente, siendo de gran utilidad al profesor de la asignatura.
- Registro de Asignaturas: es una forma que presenta

por cada alumno las asignaturas que cursa en el período escolar actual. De utilidad para el alumno a fin de comprobante de haber realizado con corrección su reinscripción.

- Actas de examen: Son similares a las listas de asistencia, pero con posiciones específicas para asentar una calificación. Dicha hoja se entrega al profesor para que califique al alumno, y la devuelva, a fin de ser leída ópticamente por la computadora y registrar, así, la información. Este documento tiene validez oficial.

Paralelamente se cuenta con dos documentos adicionales que ayudan a oficializar la reinscripción: las "actas adicionales" emitidas para aquellos alumnos que por alguna razón no aparecieron en las actas normales y que deben estar registrados; y las "actas de rectificación" que permiten al profesor cambiar la calificación de un alumno en caso de algún error.

Cabe hacer notar que muchos planteles de la U.N.A.M. utilizan diferentes mecanismos de reinscripción, por ejemplo, la "reinscripción por bloque" en que un mismo grupo de alumnos son reinscritos en el siguiente ciclo escolar sin que la soliciten; o bien la reinscripción basada en dar prioridad a los alumnos con mejor rendimiento; o el aceptar la información por hojas ópticas, tarjetas perforadas o en disco magnético, etc. La URCE tiene que hacer frente a estas modalidades para aceptar esta diversidad de métodos, así como mantener un sistema uniforme para todos los servicios sin caer en una multiplicidad de sistemas para cada necesidad.

INTRODUCCION A LA INGENIERIA DE SOFTWARE

Evolución en los sistemas de cómputo

En la breve historia de las computadoras electrónicas digitales, se podría decir que las décadas de 1950 y 1960 fueron dedicadas al hardware. La década de 1970 fue un período de transición y de reconocimiento del software. Ahora, la década de 1980 corresponde al software.

Durante la década pasada se vivió una serie de problemas que en su conjunto se les denominó "la crisis del software". El costo del software llegó a ser el elemento más caro en un sistema computarizado. Los proyectos eran formulados pero raramente llevados a su término en la forma prevista, pues o eran suspendidos o e terminados de prisa, lo que acarrearía dificultades mayores al momento de corregirlos. El responsable del desarrollo del proyecto de software utilizaba como guía y control únicamente los procesos y datos históricos que iba acumulando.

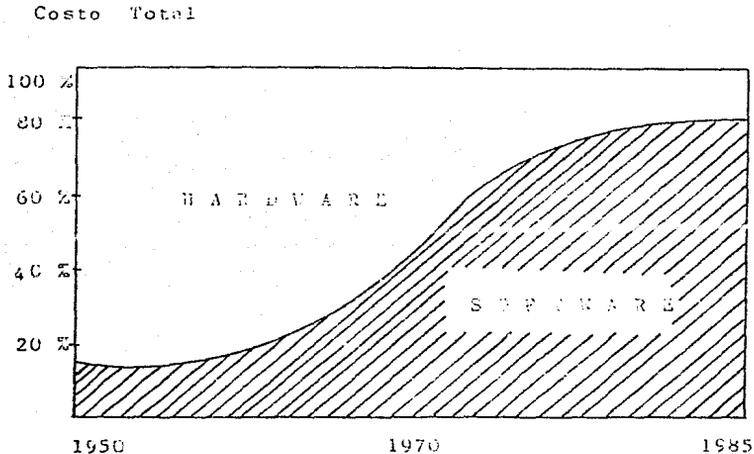
Ante esta crisis, un conjunto de técnicas llamadas en su totalidad "Ingeniería de Software" (Software Engineering), fueron desarrolladas como respuesta. Estas técnicas tratan al software como un producto ingenieril el cual requiere de planeación, análisis, diseño, implementación, pruebas y mantenimiento.

A un nivel jerárquico mayor aún, surge la ingeniería en sistemas computacionales la cual está compuesta de dos grandes ramas: la ingeniería de software y la ingeniería de hardware. Cada una de estas disciplinas representan un intento de poner orden y perfeccionamiento al desarrollo de sistemas computacionales.

Por un lado, las técnicas de ingeniería para el hardware de las computadoras ha sido desarrollada mediante diseños electrónicos y ha alcanzado, en poco más de tres décadas, un estado de madurez tal que su confiabilidad es muy alta. Sus métodos están bien establecidos y va logrando un desarrollo ascendente satisfactorio.

Pero por otro lado, desafortunadamente, respecto al software el camino ha sido difícil. En un sistema de cómputo, el software se ha constituido en el elemento más difícil de planear, siendo complicada la estimación que se tiene respecto al tiempo y costo, así como de su plena confiabilidad. Sin embargo, la demanda del software no ha disminuído, sino todo lo contrario, sigue creciendo en número, complejidad y aplicaciones. Martin L. Shooman (1)

presenta en la figura II.1 la relación del costo respecto al hardware y al software, observándose un incremento considerable de éste último respecto al primero.



Relación del costo para el hardware-software
Figura II.1

Sólo últimamente han ganado aceptación las técnicas de la ingeniería de software, pues anteriormente la programación era vista como un arte, no existían ni era aplicado ningún precedente ingenieril. Y ello ha favorecido a que dichas técnicas se desarrollen y perfeccionen con mayor rapidez.

CONSIDERACIONES SOBRE EL SOFTWARE

El software es el elemento lógico en un sistema computarizado. Es por ello que está íntimamente relacionado con el pensar y la lógica del hombre. La diferencia con el hardware, que es el elemento físico del sistema, es radical aunque están sumamente relacionadas.

Mientras que al fabricarse el hardware de una máquina, ésta se ajusta a las propiedades físicas de los elementos, y una vez conformada, los problemas que se presentan son de desgaste. Para el software, en cambio, no hay fase de fabricación. Todos los costos están concentrados en su planeación, desarrollo y mantenimiento.

Una ventaja del software es que no sufre "desgaste" como el hardware, ni pierde efectividad; un algoritmo puede funcionar para siempre. No obstante, el software requerirá de un continuo mantenimiento y actualización según los requerimientos propios del sistema.

El software es posible de ser aplicado a las más diversas áreas:

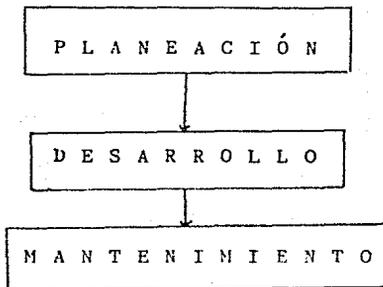
- Sistemas: telecomunicaciones, compiladores, editores, sistemas operativos, etc.
- Control : Sistemas de medición, monitoreo para control en tiempo real, etc.
- Científicas : Astronomía, física nuclear, biología, medicina, química, matemáticas, etc.
- Varios : Juegos, patrón de reconocimiento de voz o imágenes, cajas registradoras, etc.

Estas aplicaciones demandan diversas habilidades para reconocer y solucionar los problemas. La implementación de la solución a tales problemas puede ser abordada usando un conjunto de técnicas que son independientes de su aplicación. Estas técnicas forman la base de toda metodología en la ingeniería de software.

Los objetivos clave para la ingeniería de software son tres:

- 1) Desarrollar una metodología bien definida que guíe el desarrollo del software en sus fases de planeación, desarrollo y mantenimiento.
- 2) Establecer el conjunto de elementos que componen el software y que documentan y muestran cada paso en el desarrollo del sistema.
- 3) Un conjunto de tópicos predecibles que pueden ser revisados a intervalos regulares durante el desarrollo del software.

A continuación se presentarán las tres fases que intervienen en la metodología seguida por la Ingeniería de Software a fin de alcanzar los objetivos especificados. Estas tres fases (fig. II.2) son tratadas por Roger S. Pressman (2) al tratar sobre la Ingeniería de Software y coincide con la mayoría de los autores que tratan sobre ello, difiriendo sólo en cuanto a la nomenclatura o en pequeñas características, pero conservando esencialmente los mismos objetivos.



Fases en todo proceso de la Ingeniería de Software
Figura II.2

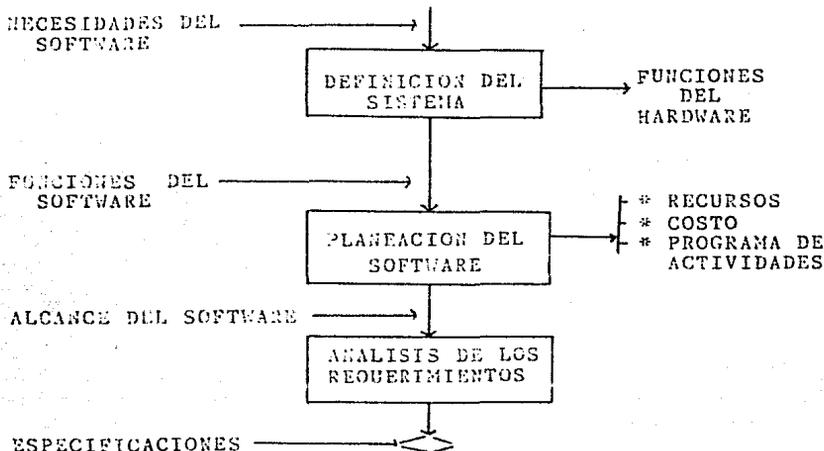
II.1 FASE DE PLANEACION DE UN SISTEMA.

A través de la historia de la humanidad el éxito de los proyectos, tanto grandes como pequeños, se ha debido a la claridad con que se definieron las metas.

Aunque hoy en día se reconoce la importancia de la planeación, sin embargo se comprende poco acerca del proceso de planeación en sí mismo. En algunas ocasiones se pierde mucho tiempo en la planeación, y en cambio, la implementación de la solución se hace en muy poco tiempo y con menos recursos. Si se "sobrepiana" un sistema, se puede dejar ir el tiempo oportuno para su implementación, y por tanto, perder la posibilidad de éxito. Sin embargo, es más peligroso el subestimar la planeación, siendo ésta la causa principal de que un proyecto no llegue a su término; por ejemplo, una inexactitud en el costo o en el programa de actividades significarán graves problemas durante la implementación.

La fase de planeación de un sistema es un proceso que consta de tres etapas: definición, planeación del software y análisis de los requerimientos (fig. II.3).

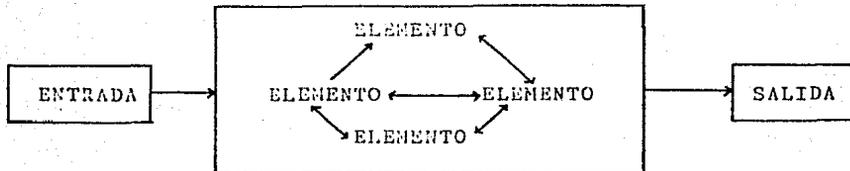
Las etapas asociadas a esta fase requieren de una combinación, intuición, experiencia y tenacidad. Estos rasgos, sin embargo, se deben acoplar a un conjunto de tareas sistemáticas que permita a la planeación ser conducida de una manera controlada.



Etapas en la fase de planeación de un sistema utilizando ingeniería de software
 Figura II.3

II.1.1 DEFINICION DEL SISTEMA.

El primer paso en la fase de planeación de un sistema es su definición. En este punto la atención es enfocada sobre el sistema como a un todo. Para ello, y bajo el contexto del presente trabajo, se puede definir un sistema como una colección de elementos que se relacionan de tal manera que permitan el cumplimiento de un objetivo concreto (fig. II.4).



Un sistema es un conjunto de elementos relacionados de tal manera que cumplan con un objetivo determinado.
 figura II.4

La definición del sistema precede a la ingeniería de software y de hardware, pues es a partir de ella que surgen las decisiones sobre las opciones a tomar en estos dos campos.

Entre los principales objetivos que persigue la definición están los siguientes:

- Describir las funciones, interfaces y tareas del sistema.
- Evaluar el concepto definido para el sistema concreto y así poder estimar su factibilidad, su factor costo-beneficio, las necesidades requeridas, etc.
- Colocar cada función en su respectivo lugar, ya sea en hardware, software u otro elemento suplementario del sistema.
- Establecer el costo y las restricciones del sistema.

La definición del sistema constituye el fundamento para todas las subsiguientes tareas del desarrollo. En esta fase se requiere tanto de expertos en software como en hardware para vislumbrar el alcance de la computadora con los objetivos propuestos. La definición de un sistema se caracteriza por cuatro pasos:

- a) Análisis.
- b) Estudio de factibilidad.
- c) Selección y asignación de funciones.
- d) Especificación del sistema mediante un reporte.

a) Análisis.

Este primer paso está enfocado a la comprensión y justificación de la solución propuesta. El análisis está compuesto de varias tareas, en donde se definen:

- a.1) El objetivo a llevar a cabo
- a.2) El tiempo en que es posible llevarlo a cabo
- a.3) La relación costo-beneficio del sistema

Para llevar a cabo la comprensión del problema es necesario tener en cuenta diversos aspectos tales como los que se mencionan a continuación, en donde cada uno de ellos puede ir acompañado del respectivo análisis debidamente evaluado y documentado.

- Un contacto directo y continuo con el usuario a fin de contar con la información a partir de su fuente original.

- Fijar los objetivos concretamente, no vagamente, del sistema a desarrollar. Así como las razones que justifiquen su implementación.
- Hacer un estudio sobre el sistema vigente para tomar de él sus conveniencias y mejorar las soluciones a sus problemas.
- Definir la estructura en que se encuentra la información de entrada al sistema, así como los requerimientos en su salida.
- Realizar una investigación de campo, consistente en buscar de organizaciones similares la solución encontrada ante problemas similares.
- Desarrollar diversas alternativas para la solución del sistema a fin de seleccionar de entre ellas la óptima.
- Tener planeada la siguiente fase, con sus requerimientos, para cumplir ordenadamente con ella.
- Presentar el estudio a niveles gerenciales y de usuario para poder establecer que el sistema ha sido comprendido adecuadamente.

b) Estudio de factibilidad.

Cualquier proyecto, siempre y cuando no fuera ficticio, podría ser llevado a cabo si se dispusiera de recursos y de tiempo ilimitados. Sin embargo, desafortunadamente un sistema computarizado normalmente no cuenta con grandes recursos y se tiene el tiempo encima.

Para saber la posibilidad que tiene el proyecto se procede a una estudio sobre su factibilidad, abarcando tres áreas principales:

b.1) Factibilidad económica:

Consiste en una evaluación sobre el costo del sistema (dinero, esfuerzo, tiempo, etc) frente al beneficio que rinde. Su estudio ofrece la información más importante para visualizar la factibilidad. El análisis costo-beneficio involucra criterios que variarán de acuerdo al sistema, sin embargo, la mayoría de los sistemas, en procesamiento de datos, tienen como objetivo principal mejorar la información en cuanto a calidad, cantidad, confiabilidad, optimización de tiempo y en organización.

b.2) Factibilidad técnica:

Esta área efectúa el estudio sobre las restricciones, funcionabilidad y medios técnicos que afectan a la consecución del sistema. Normalmente a esta área se le asocian dos restricciones principales: los recursos disponibles (sean de hardware o software) y la tecnología desarrollada hasta el momento (que sea lo suficientemente avanzada para que pueda cumplir con las necesidades requeridas).

b.3) Factibilidad legal:

Se determina el sistema de acuerdo con las normas legales que rijan el área a la que se atañe. Aparte de consultar los códigos específicos, existen varias lecturas al respecto convenientes de consultar en caso de necesitarlo (4).

Hay ocasiones en que la justificación económica es obvia, o las alternativas técnicas son mínimas, o no hay una relación que involucre aspectos legales, por lo que en este estudio se pueden omitir o reducir en las restricciones.

c) Selección y asignación de funciones.

Una vez que han sido cumplidos los objetivos asociados a la tarea del análisis, se consideran las diversas alternativas, seleccionando las que cumplan de la mejor manera con los requisitos solicitados.

Para dicha selección, una de las diferentes alternativas es tomar en cuenta los criterios de la organización en que se lleva a cabo el proyecto, así como el balance entre las ventajas y desventajas que ofrece cada una de las alternativas. En la mayoría de los sistemas el costo es uno de los factores decisivos, sin embargo, también son importantes otros factores como el número de sistemas que se van a producir, el grado de decisión humana que sea necesaria tomar ante ciertos casos, la estandarización de las interfaces en los subsistemas, la adaptabilidad del sistema al ambiente en que se ejecute, etc. Cabe decir que ninguno de los criterios señalados ha de ser considerado aisladamente, pues será a partir de una consideración en conjunto como se llegue a la mejor decisión.

Después, cada función del sistema con las características de sus interfaces se asignan a uno o varios elementos del sistema.

d) Especificación del sistema.

El último paso en la definición de un sistema consiste en documentar los estudios hechos hasta el momento. Será el primer diagnóstico que se entregue y para ello se especificará cada paso. Se puede tomar como marco de referencia el formato recomendado por Pressman (2,p.53) y que a continuación se presenta:

ESPECIFICACION DEL SISTEMA

1.0 Introducción.

La sección introductoria describe los objetivos del sistema y las condiciones en que opera el sistema. Esta sección también contiene de forma breve un resumen que especifica el alcance del proyecto, su factibilidad, su justificación, los recursos requeridos y un panorama general sobre el costo y el programa de actividades.

2.0 Descripción funcional.

Se describe cada una de las funciones del sistema, incluyendo: la información de entrada y salida, las tareas a ser ejecutadas. Es conveniente mostrar un diagrama de su descripción.

3.0 Asignación de funciones.

Cada función descrita en la sección 2.0 es asignada como una parte en el sistema total. Los elementos de hardware y de software son descritos por separado, así como la información de la base de datos o paquetes utilizados.

4.0 Restricciones.

Las restricciones técnicas y administrativas que pueden afectar al sistema se describen en esta sección, la cual debe ser revisada cuidadosamente para asegurar su correcta implementación.

5.0 Costo.

Aunque en esta etapa todavía no es posible la determinación precisa del costo del sistema, es necesaria una estimación previa para considerar su factibilidad.

6.0 Plan de actividades.

Al igual que con el costo, el plan de actividades no es posible determinarlo con precisión, sin embargo, ya se puede tener como información precisa el inicio y el fin de éstas.

La definición del sistema es el resultado de la comunicación que han tenido el usuario y el ingeniero de software; éste deberá saber qué preguntas formular, a fin de saber las necesidades esenciales por implementar.

II.1.2 PLANEACION DEL SOFTWARE.

La segunda etapa en la fase de la planeación del sistema es la planeación del software (software planning), como se indicó en II.1 (fig. II.3.). El objetivo de la planeación del software es proveer un marco de referencia que sirva de guía para poder hacer una estimación razonable de recursos, costos y plan de actividades. Para cumplir con dicho objetivo se debe tener comprendido completamente el alcance del software, así como tener definidos los recursos requeridos para satisfacer este alcance. Para esta fase deben de quedar resueltas tres cuestiones:

1a) La cantidad de tiempo y esfuerzo que es conveniente dedicar a la planeación.

Aunque es difícil establecer una guía común, si se contempla en su conjunto toda la ingeniería de software, se puede decir que en términos generales representa de un 10 % a un 20 % del total de tiempo y trabajo dedicado al proyecto.

El tamaño del software y su complejidad, el área de aplicación, sus resultados finales, las obligaciones contraídas respecto a su inicio y terminación, son sólo algunas de las muchas variables que afectan al porcentaje dedicado a esta etapa.

2a) Persona que le corresponde hacer la planeación.

La persona dedicada a su desarrollo debe ser un analista experimentado que tenga cualidades para dirigir a un grupo de personas, así como saber distribuir y controlar las diferentes tareas asociadas a la planeación.

El analista trabaja en conjunto con el gerente o director, con el departamento técnico y con el usuario. Para grandes proyectos se puede formar un equipo de analistas que lleven a cabo cada uno de los pasos de la planeación.

3a) Tener presente la dificultad en llevar a cabo esta etapa.

En ocasiones la dificultad estriba en que una idea vaga debe ser concretada en un conjunto de elementos tangibles. En esta fase hay un diálogo en que se solicitan todas las necesidades del sistema, lo que puede provocar que aumente la posibilidad de transmitir la información erróneamente, de malinterpretarla o de omitirla, pudiéndose tener métodos de verificarla,

La planeación del software combina dos herramientas: la investigación y la estimación.

- La Investigación.

Permite definir el alcance de los elementos del software que serán utilizados en el sistema computarizado. La especificación del sistema de la que se habló anteriormente, sirve como guía, pues es donde se describe el límite de cada función.

- La Estimación.

La estimación de recursos, costo y programas para el desarrollo del software requiere de experiencia, buena información y el valor de comprometerse a lo medido cuantitativamente cuando con lo único con que se cuenta, en ocasiones, son datos puramente cualitativos. La estimación lleva un inherente riesgo, que está en función la complejidad y tamaño del proyecto y del grado que se ha alcanzado en la definición, variabilidad y estructuración del sistema.

Ayudándose de estas dos herramientas, la planeación del software se puede dividir en cinco tareas principales:

- A) La determinación del alcance del software.
- B) El establecimiento de los recursos necesarios.
- C) La determinación del costo del software.
- D) La planeación del programa de actividades (Schedule).
- E) La realización de la documentación sobre las especificaciones del software.

A) La determinación del alcance del software.

Esta primera tarea describe el alcance del software de una manera entendible, nunca ambiguamente, dirigido tanto a un nivel técnico como al administrativo. Este alcance debe ser descrito limitativamente, es decir, considerando los datos cuantitativamente (por ejemplo, el número de usuarios simultáneos) y explicitando las restricciones y limitaciones (por ejemplo, tamaño de la memoria con que se cuenta, número de impresoras, etc).

El software interactúa con otros elementos del sistema computarizado y por ello debe considerar la naturaleza y complejidad de cada interface para determinar cualquier efecto sobre los recursos, el costo y el programa de

actividades desarrollado.

Generalizando, el concepto de interface abarca cuatro ámbitos (6, pag. 51):

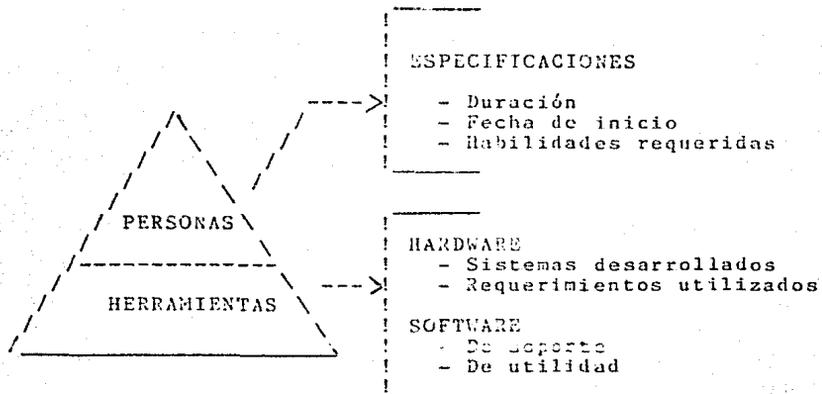
- (1) El hardware que ejecuta el software así como los instrumentos que son controlados indirectamente por el software. Por ejemplo, los periféricos, los controladores en tiempo real, etc.
- (2) El software vigente que ya existe y debe ser incorporado al "nuevo" software desarrollado. Por ejemplo, rutinas de base de datos, sistema operativo o paquetes.
- (3) Las personas que hacen uso del software a través de terminales y/u otro instrumento.
- (4) Las operaciones o trabajos que preceden o le siguen a las tareas del software.

Es en esta tarea donde se contempla la confiabilidad del sistema. Varios autores han propuesto diferentes métodos para medir la confiabilidad del software; por ejemplo, Sukert y Goel (4), Mills (5) o Musa (6). Sin embargo, se puede decir que aún están en desarrollo y no resultan tan precisas como las que hay para el hardware. No obstante la naturaleza general del software puede asumir consideraciones especiales para asegurar la confiabilidad. Por ejemplo, el software para el sistema de control de tráfico aéreo debe ser altamente confiable, pues de ello dependen muchas vidas, en cambio, el software de un sistema de control de inventarios, aunque no debe fallar, en caso de presentarse una falla no es tan grave como el primer caso.

Aunque no se puede cuantificar precisamente la confiabilidad del software, se puede usar como criterio la misma naturaleza del proyecto a fin de ayudar a una mejor estimación.

B) Establecimiento de Recursos.

La segunda tarea de la planeación del software es la estimación de recursos requeridos para completar el desarrollo del proyecto. La figura II.5 ilustra a los recursos como una pirámide: en su base, las herramientas -hardware y software- son necesarias para poder soportar cualquier sistema computarizado, y en su nivel superior se encuentra el recurso primario, las personas, indispensables en todo sistema. Cada recurso se caracteriza por su descripción, la fecha de su inicio y término en su utilización.



RECURSOS
figura II.5

B.1) Recursos Humanos.

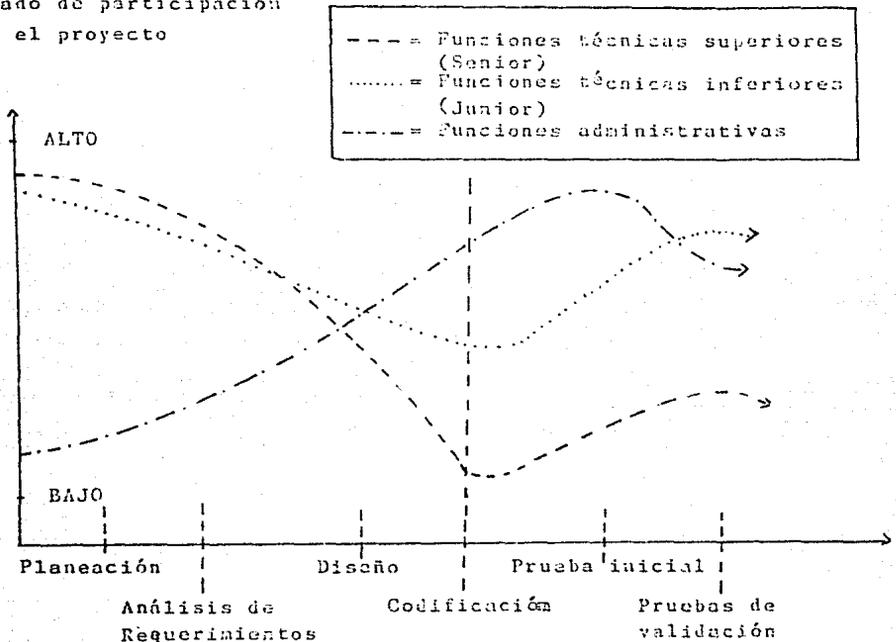
Las personas humanas son el fundamento principal en cualquier desarrollo de software. Entre los problemas que tuvo la "crisis del software", ninguno fue más grave que la escasez de recursos humanos capacitados para desarrollar el software.

La planeación empieza evaluando y seleccionando las aptitudes necesarias para llevar a cabo el proyecto. En esta etapa se especifican tanto los puestos organizativos (administrador, ingeniero de software, analistas, etc) así como los de nivel especializado (en telecomunicaciones, en base de datos o en microprocesadores, etc).

Los proyectos se pueden clasificar de acuerdo a su tamaño. Los llamados pequeños son aquellos que comprenden el trabajo que desarrolla una persona durante un año, y se encomiendan a una sola persona si ésta es capaz de realizar todos los pasos del software, teniendo un respaldo de asesores para consultar.

Si se trata de proyectos grandes, la participación variará a través del ciclo de vida del proyecto, y se hará necesario contar con un equipo para su desarrollo. La figura II.6, proporcionada por Pressman [2, pag.634], ilustra la participación que tienen en un proyecto los diferentes miembros del departamento de desarrollo del software.

Grado de participación
en el proyecto



Diversas participaciones en el software
de un sistema computarizado
Figura 11.6

B.2) Hardware.

El hardware es también una herramienta muy importante en el desarrollo del software del sistema. Durante la planeación se podrán considerar dos categorías respecto del hardware:

(1a) Equipo para el desarrollo del sistema.

Este recurso es la computadora y sus periféricos anexos utilizados durante la fase en que se desarrolla el sistema. Por ejemplo, una computadora VAX 11/780 puede servir en el desarrollo de un sistema para un microprocesador de 8 bits en el cual se ejecute finalmente el proyecto. Con excepción de grandes proyectos, esta computadora no se compra ex-profeso, sino que se utiliza normalmente una ya existente. También es posible ir

ejecutando el software en un procesador aparte para después simplemente cambiarlo.

(2a) Elementos complementarios de hardware.

También pueden ser especificados como recursos para el desarrollo del software equipo extra que sirve de interface con otros sistemas, que no siendo propios del sistema a desarrollar, son necesarios para su completa especificación. Por ejemplo, en un sistema que se necesite validar una información captada por lectura óptica, necesitará de hardware y software especiales.

B.3) Software.

Así como se utiliza hardware para construir nuevo hardware, también es utilizado software para desarrollar nuevo software. La primera aplicación de software fué para desarrollar un "bootstrapping", el cual consiste en un primitivo traductor de lenguaje ensamblador escrito en lenguaje de máquina. Se utilizó para desarrollar un ensamblador más sofisticado. A partir de la nueva versión del compilador se elaboraron otros de mayor nivel, y así sucesivamente. Hoy en día hay disponible un vasto conjunto de herramientas en software, entre las que destacan los compiladores, editores y traductores.

Durante la planeación se consideran dos categorías de software: el de soporte y el de utilidad.

(1) Software de soporte:

Este soporte ofrece una amplia gama de herramientas. La más conocida es el compilador del lenguaje de programación, el cual ha venido a ser un soporte "sine qua non" en el desarrollo del sistema.

Existen varios "software de soporte" para cada paso en el proceso de la ingeniería de software, por ejemplo:

- En el análisis:
Herramientas para la especificación automática de los requerimientos del sistema, ya sea en el costo o en el plan de actividades.
- En el diseño :
Simuladores, generador de diagrama de flujo o de bloques.
- Para la codificación y pruebas:
El cruce de variables ("cross-assemblers"), los compiladores, macroprocesadores, etc.

La productividad en el desarrollo puede ser mejorada sustancialmente con la utilización de estos recursos. En la planeación se reconoce la importancia de dedicarle un costo y un esfuerzo sustancial para llevar a cabo el proyecto. La relación costo-beneficio puede servir como criterio en la consideración para elegir el software de soporte.

(2) Software de utilidad.

Este tipo de software se consigue de la biblioteca del sistema utilizado o comprándolo. El software de utilidad (utility software) ha tenido un considerable aumento en su producción después de la "crisis del software", lo que ha ocasionado que actualmente se le ponga más atención y se le dediquen más recursos a su desarrollo.

Por último, cabe señalar dos reglas prácticas durante esta etapa:

Primera: Si existe el "software de utilidad" necesario para el sistema, es mejor conseguirlo en vez de realizarlo. El costo de adquisición del software existente, es en la mayoría de los casos, menor que el costo para desarrollarlo.

Segunda: Si el "software de utilidad" existe, pero requiere de algunas modificaciones antes de poder ser integrado al sistema, hay que proceder con mucho cuidado. El costo y esfuerzo dedicado al modificar el software existente puede ser mayor que el utilizado en desarrollar el sistema equivalente.

C) Costo del software.

En los primeros días de la computación, el costo del software comprendía sólo un pequeño porcentaje del costo total del sistema, un error en el costo de su estimación representaba sólo un pequeño impacto. Hoy en día, hay sistemas en que el software es el recurso más caro, y su estimación es factor decisivo para poder llevar a cabo el proyecto.

Sin embargo, es muy difícil tener el costo del software como un dato preciso debido a que están involucradas demasiadas variables (técnicas, humanas, ambientales, políticas, etc.) que pueden afectar durante su desarrollo al costo último y total. A pesar de ello, es posible tener una estimación bastante aproximada si se toman en cuenta los diversos factores que intervienen en la "producción" del software. Dentro de estos factores se pueden considerar los siguientes:

- El número total de funciones que ejecutará el proyecto.
- El número aproximado de líneas de código.
- La complejidad del proyecto.
- El esfuerzo desarrollado (medible en el trabajo de hombre/mes).
- Tiempo empleado.
- Número de personas que intervienen en el proyecto.
- La documentación del proyecto.
- El tipo de herramientas utilizadas.
- La categoría y experiencia del personal empleado.
- La magnitud del proyecto.
- Las restricciones observadas.
- Los recursos con que se cuentan.
- Factores externos al sistema (la economía característica del lugar donde se desarrolla el proyecto, legislación sobre factores que intervienen en el proyecto, etc.).

En algunas organizaciones se han desarrollado sistemas para calcular el costo del software en base a modelos de estimación. Entre los principales están los modelos desarrollados por Putnam (7), Esterling (8) y el llamado "SLIM" (9) el cual es un sistema automatizado para costos basado en el modelo de Putnam, en la programación lineal y en la simulación estadística.

Sin embargo, aún falta un mayor perfeccionamiento en las técnicas, y las que hay actualmente han de ser usados con mucha prudencia, pues a veces no se tiene en cuenta algunos de los factores mencionados anteriormente llegando a una estimación irreal.

Otro criterio para evaluar el costo del software son las técnicas que toman en cuenta las líneas de código generadas (lines-of-code "LOC") (10), (11). Para ello, lo primero que se hace es especificar todas las funciones que van a ser implementadas en el código. Después se descomponen dichas funciones en sus elementos básicos hasta alcanzar una estimación confiable del número de líneas requeridas para implementar cada función. Una vez hecho ésto, se procede a fijar unos márgenes que limiten el número de líneas, un máximo (siendo pesimistas), y un mínimo (siendo optimistas). Luego se calcula el número esperado y se asigna el costo por línea de código. Todo ello regido por cálculos estadísticos.

Una técnica más, es la que se genera al evaluar en base a la complejidad que supone resolver el proyecto. Se toman en cuenta dos factores: el número de personas y el tiempo; Se divide el trabajo por tareas (planeación, análisis, diseño, etc) y se tiene en cuenta la tasa de salarios y la capacidad del personal para estimar el costo total del software.

D) Programación de actividades (Schedule)

El programa de actividades se puede considerar desde dos perspectivas:

(1a) Si la fecha para la terminación del sistema computarizado ya ha sido establecida irrevocablemente. En este caso el departamento de software está restringido a distribuir su esfuerzo dentro del período establecido y hace sus requerimientos en base al tiempo señalado.

(2a) Si el programa de actividades ha sido discutido y la fecha de terminación fue establecida por el departamento de software. En este caso, el esfuerzo que conlleva el proyecto es distribuido en base al mejor uso de los recursos disponibles. La fecha de terminación fue fijada después de un cuidadoso análisis de los elementos del software. Desafortunadamente, esta situación prevalece sobre la primera en el medio computarizado.

La exactitud en el programa de actividades suele ser más importante incluso que en el costo del proyecto. Cuando se sobrepasa del costo, éste puede ser absorbido por una revalorización en el costo del producto, por una amortización en las ventas, etc. En cambio, un retraso en el programa de actividades puede reducir el impacto del producto en el mercado o puede provocar tal disgusto en el cliente que se venga abajo todo el proyecto junto con el prestigio del departamento.

Por otro lado, es falsa la creencia común de que si se empieza a retrasar en tiempo el proyecto, se resuelve incrementando el número de personas; muchas veces se pierde más tiempo en enseñar y actualizar a las nuevas personas, que incluso puede resultar más retrasado y costoso que originalmente.

La programación de actividades es la tarea más difícil en la planeación del software. Aquí se trata de coordinar los recursos disponibles con el esfuerzo y tiempo dedicados al proyecto. Existe una diversidad en la manera de representar dicho programa, pero normalmente cada organización tiene su política a seguir. Este programa es semejante al utilizado en diversos proyectos que no son necesariamente computacionales; así, las técnicas y herramientas utilizadas en programar otras actividades pueden ser aplicadas a la planeación del software. Entre los métodos más importantes están los desarrollados por Riggs (12) y por Wiest (13), quienes tratan el método de ruta crítica y el de "evaluación de programas y técnicas de revisión" (Program Evaluation and Review Technique "PERT").

E) Documentación del software

Cada paso que se da en el proceso de ingeniería de software es señalado en un reporte, que viene a ser la culminación de la planeación, y que sirve como guía para las siguientes fases. Normalmente es un breve documento dirigido a diversas áreas y cumple con tres fines:

- 1) Comunicar el alcance y los recursos del software al gerente, al departamento técnico y al usuario.
- 2) Definir el costo y el programa de actividades para que sean revisados.
- 3) Proveer de una visión global del desarrollo del software a todas las personas involucradas de algún modo en el proyecto.

Para cumplir con dichos fines, se puede utilizar como guía el índice que se muestra a continuación (2, pág. 82):

- 1.0 Alcance del software
 - 1.1 Objetivos del proyecto
 - 1.2 Funciones principales
 - 1.3 Otras características
- 2.0 Recursos
 - 2.1 Recursos Humanos
 - 2.2 Recursos de Hardware
 - 2.3 Recursos de Software
- 3.0 Costo
- 4.0 Programa de actividades

La forma de presentar el costo y el programa de actividades varía dependiendo a quien va dirigido. Este documento no debe ser complicado y extenso, ni detenerse en decir "cómo" se lleva a cabo el sistema, sino apuntar únicamente "qué" es lo que hace.

II.1.3 ANALISIS DEL LOS REQUERIMIENTOS

Una especificación completa de los requerimientos del software es esencial para cumplir satisfactoriamente con el desarrollo del software, que es la siguiente fase. A pesar de que el sistema esté mal diseñado o mal codificado, no importa tanto como si estuvo mal analizado, pues en caso este caso no se obtiene el resultado deseado.

El alcance del software, que inicialmente se estableció durante la planeación, ahora es definido en detalle. Las diferentes alternativas son analizadas, y es escogida la más adecuada. También se asignan las diferentes tareas del proyecto. Tanto el responsable del proyecto como el usuario tienen una participación muy activa en esta etapa. Por su parte, el usuario intenta reformular aquellos conceptos que hayan quedado un tanto ambiguos, mientras que el ingeniero en software actúa interrogando, consultando y tratando resolver las dudas que encuentre sobre las restricciones requeridas.

El análisis y la especificación de los requerimientos puede parecer una tarea fácil, sin embargo, las apariencias engañan. Cuando se maneja mucha información, es muy alta la probabilidad de malinterpretar el contenido; una de las causas se debe al lenguaje, pareciera que el usuario y el ingeniero hablan en dos idiomas diferentes. El usuario puede tener cualquier profesión (médico, administrador, economista, etc) y por tanto, hacer continuas referencias a terminologías específicas. El analista tratará de comprender lo que se le solicita, haciendo un esfuerzo por "traducir" todas las necesidades a tareas específicas, y en su caso, hacer ver la factibilidad de esos requerimientos.

A) Objetivos del Análisis.

El análisis de los requerimientos es el último paso en la fase de la planeación (ver fig. 11.3). Utilizando el alcance del software especificado durante la planeación, el análisis debe de satisfacer los siguientes objetivos:

- Proveer de un fundamento para el desarrollo del software, estableciendo el flujo y la estructura de la información.
- Describir el software identificando los detalles en las interfaces, es decir, señalando las entradas y salidas para cada tarea.
- Determinar las restricciones del diseño y definir los requerimientos de software.
- Establecer y mantener una comunicación estrecha con el usuario de tal manera que se cumplan los objetivos establecidos.

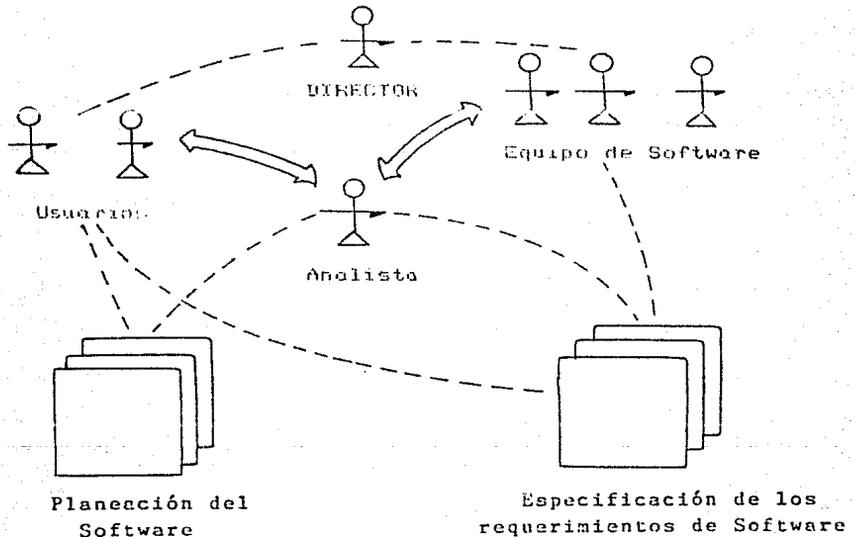
El análisis en el software puede ser dividido en cuatro tareas {2, pág.96}:

(1a) Reconocimiento del problema.

Es necesario que el analista tenga contacto con

diversas personas: el director del proyecto, el departamento técnico, el usuario y el departamento encargado de desarrollar el software (fig. II.7). El director del proyecto puede servir como coordinador para facilitar la comunicación entre los diferentes ámbitos.

La meta que debe perseguir el analista es reconocer el problema fundamental del proyecto, la cual se facilita con dicha comunicación.



Comunicación del analista
Figura II.7

(2a) Evaluación y síntesis.

El analista evalúa el flujo y la estructura a que es sometida la información. Y también especifica con detalle las diferentes funciones, estableciendo las características y cubriendo las restricciones señaladas durante la planeación.

Para una comprensión del problema, el analista divide el problema en diversas funciones o tareas -esta técnica es tratada más adelante- y cada una de ellas se evalúa y resuelve. Una vez resueltas todas las funciones, se tiene solucionado el problema completo.

(3a) Especificación.

Esta tarea consiste en mostrar una representación del software, de tal manera que sirva para una revisión por parte del usuario. Lo ideal es que el mismo usuario desarrolle una documentación de la especificación de sus requerimientos, sin embargo, es muy difícil que se presente el caso. Siendo optimistas, estas especificaciones se desarrollan en conjunto, el analista y el usuario.

(4a) Revisión.

Una vez que se han desarrollado las especificaciones y el manual preliminar, se hace necesaria una revisión tanto por parte del usuario como del director del proyecto. De su revisión casi siempre surgen modificaciones al tener ya más clarificados los problemas. Tanto el costo como el programa de actividades se revisan, y en su caso, se modifica lo necesario.

B) El analista.

Ingeniero de sistemas, analista-programador, analista de sistemas, o simplemente analista son algunos de los nombres que recibe quien realiza la función de analizar el sistema, y que cumple con las siguientes características:

- La habilidad de comprender en conceptos abstractos los problemas concretos, así como saber reorganizarlos en divisiones lógicas, sintetizando soluciones para cada división.
- El saber distinguir hechos pertinentes de fuentes de información confusas o conflictivas.
- La capacidad de entender y "traducir" las circunstancias en las que se encuentran las peticiones del usuario.
- La habilidad de aplicar sus conocimientos de software y hardware a los requerimientos del usuario.
- La habilidad para comunicar de manera clara y precisa, de forma oral y escrita, las soluciones a los problemas que se le expongan.
- La habilidad de poder "ver el bosque sobre los árboles", es decir, de sintetizar el problema en su conjunto sin perderse en detalles.

El analista coordina y/o ejecuta cada tarea asociada con el análisis de los requerimientos. Durante el reconocimiento de las tareas se comunica con su departamento y con el usuario para asegurar que se cumplan las necesidades requeridas en la implementación del sistema. Para ello puede contar con un departamento de desarrollo el cual evalúa y sintetiza las diversas tareas. Generalmente, el analista es el responsable de desarrollar el documento llamado "Especificación de los requerimientos de Software" y participa, por tanto, de todas las revisiones que se le hicieran al sistema.

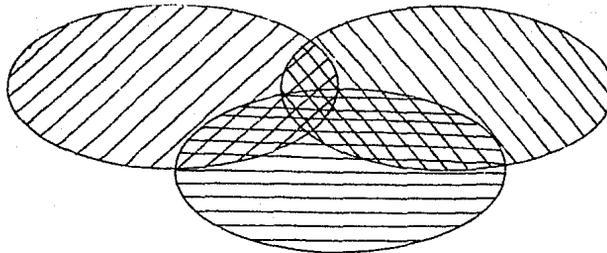
Es importante hacer notar que el analista debe entender y tener presente todas y cada una de las fases de la ingeniería de software -planeación, desarrollo y mantenimiento-, pues cualquier cambio que se introduzca requiere de una concepción total del sistema.

C) El análisis.

Como en todos los pasos del proceso de ingeniería de software, no hay una fronteras precisas que separen la planeación del software, el análisis de los requerimientos y el diseño. La figura II.8 ilustra la intersección que hay entre las diversas etapas de la ingeniería de software.

PLANEACION DEL
SOFTWARE

DISEÑO

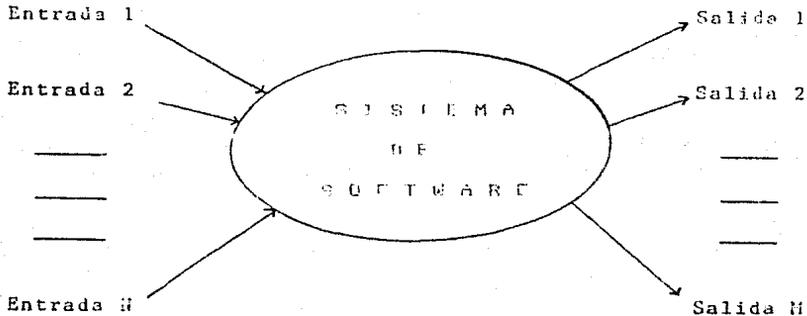


ANALISIS DE LOS
REQUERIMIENTOS

Intersección entre diferentes pasos
Figura II.8

C.1) Modelo fundamental.

Un sistema computarizado transforma la información. El sistema acepta como entrada una variedad de datos y le aplica elementos del sistema para convertir esas entradas en salidas ya modificadas. Una manera de representar este modelo fundamental de un sistema se ilustra en la figura II.9 (2, pág.98). Es de notar que el modelo se puede aplicar para el sistema completo o para cada uno de sus elementos.



Modelo de un sistema fundamental
Figura II.9

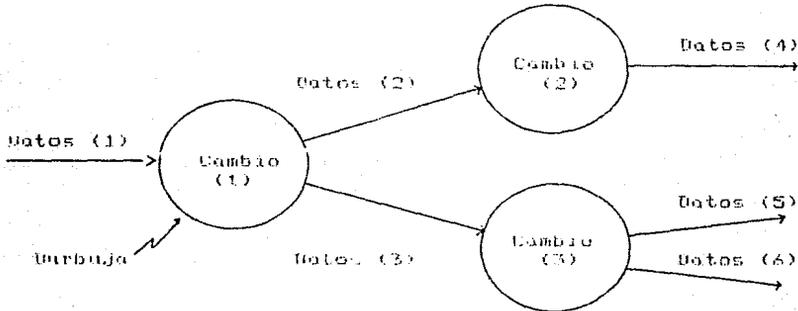
C.2) Flujo de información.

En tanto que la información fluye a través del software, es modificada por una serie de transformaciones. Un "diagrama de flujo de datos" es una técnica gráfica que describe tanto el flujo de información y las transformaciones que se le aplican, como la entrada y salida de datos a cada elemento del sistema. La forma básica de presentar el diagrama es como se indica en la fig. II.10 (2, pág.100).

El diagrama es similar a otros diagramas de flujo - por ejemplo, en un proceso de producción- y a sido incorporado dentro de las técnicas de análisis y diseño por Yourdon (15), (16), Constantino (17) y DeMarco (18). Se conocen también como "gráfica de flujo de datos".

Un modelo de un sistema fundamental puede ser representado como un diagrama de flujo de datos. Todos los elementos del software están comprendidos en una sola

"burbuja" con entradas y salidas descritas por flechas que entran o salen de ella. Cada burbuja puede ser a su vez redefinida con mayor detalle, utilizando la misma técnica, es decir, el sistema se puede expandir cuanto sea necesario. Es importante notar que no hay una indicación sobre la secuencia de eventos, aunque puede estar implícita en el diagrama; sin embargo, la secuencia es propia de ser señalada hasta el diseño del software.



Flujo de información
Figura 11.10

C.3) Diagrama de flujo de datos.

El diagrama de flujo de datos tiene tres atributos que son importantes de notar (2, pág.101) :

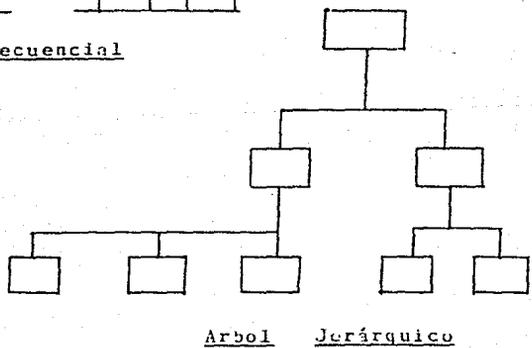
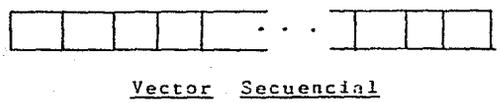
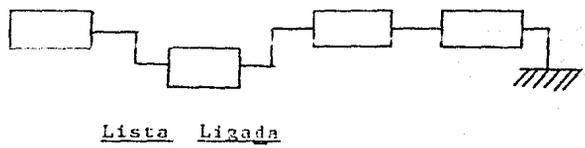
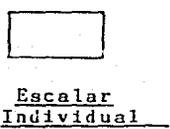
- 1) El flujo de información en cualquier sistema, ya sea manual, automático o híbrido, puede ser representado.
- 2) Cada "burbuja" puede ser a su vez subdividida hasta quedar totalmente definido el sistema.
- 3) Normalmente se enfatiza más un flujo de datos que un flujo de control.

Adoptando las herramientas utilizadas por Pressman (2, pág.105) para describir un diagrama de flujo de datos se tienen las siguientes convenciones:

- La información es representada por líneas con flechas.
- Los procesos o transformaciones son representadas por "burbujas" con etiquetas apropiadas.

- Las fuentes de información y los destinos se muestran en cajas etiquetadas. Una fuente de información es la localidad de donde provienen los datos (por ejemplo, un traductor, un hombre o una máquina), y una información destino es el lugar a donde va a parar el movimiento de los datos.
- La información almacenada (por ejemplo un archivo de datos) se representa con una doble línea horizontal.

La organización y complejidad de una estructura de datos sólo está limitada por el ingenio del diseñador. Hay sin embargo, un número limitado de estructuras de datos clásica que forman los bloques básicos que sirven para estructuras más sofisticadas. Estas estructuras clásicas se ilustran en la figura II.11 (2,pág.107).



Clásicas estructuras de datos
Figura II.11

Un "escalara individual" es la más simple de todas las estructuras de datos. Como su nombre indica, representa un elemento individual de información que puede ser direccionado por un identificador. Su acceso puede ser conseguido especificando una sola dirección. El tamaño y el formato de este elemento puede ser una variable lógica de un bit de longitud, o un número entero o real de 8 a 64 bits de longitud o una cadena de caracteres de cientos o miles de bits de longitud. Sin embargo, siempre se considera como una sola unidad.

Cuando se unen varios escalares individuales en un grupo o lista continua se forma un "vector secuencial". Los vectores son los más comunes de todas las estructuras de datos y abren la puerta a la información indexada. Cuando el vector secuencial se extiende a dos, tres o un número arbitrario de dimensiones, se crea un espacio de n-dimensiones. El espacio más común es la matriz de dos dimensiones, y que en lenguaje de programación ese espacio es denominado como "arreglo".

Los escalares individuales, los vectores y los espacios pueden ser organizados de varias formas. Una lista ligada es una estructura de datos organizada de manera que trata a la información como nodos unidos por apuntadores. Cada nodo contiene datos organizados, así como los apuntadores que indican la dirección de la información almacenada del siguiente nodo de la lista. Una gran ventaja es que a la lista se le pueden añadir nodos en cualquier parte necesitando hacer sólo pequeñas variaciones en los apuntadores.

Otra estructura de datos que utiliza las características mencionadas es la "estructura de datos jerarquizada", compuesta de múltiples listas ligadas. Este tipo de estructura se utiliza normalmente en aplicaciones que requieren información categorizada, es decir, dividida en géneros de una misma especie.

D) Especificación de los requerimientos del software

En esta parte del análisis de requerimientos se elabora un diagnóstico llamado "Especificación de los Requerimientos en Software". Dicho reporte extiende el alcance ya prescrito en la planeación, estableciendo una descripción completa y detallada de la información y de las funciones. También se dictan los criterios apropiados de validación, así como otros datos pertinentes para los requerimientos. A continuación se presenta un posible marco de referencia para elaborar dicho diagnóstico:

REPORTE PARA LA ESPECIFICACION DE LOS REQUERIMIENTOS DE SOFTWARE

1.0 Introducción.

2.0 Descripción de la información.

- a) Diagrama del flujo de datos.
- b) Representación de la estructura de datos.
- c) Descripción de las interfaces internas del sistema.
- d) Interfaces internas.

3.0 Descripción de las funciones

- a) Funciones.
- b) Narración del proceso.
- c) Restricciones del diseño.

4.0 Criterio de validación.

- a) Fronteras o límites del sistema.
- b) Clases de pruebas.
- c) Consideraciones especiales.

5.0 Bibliografía.

6.0 Apéndice.

En la introducción se indican los objetivos y metas del software, describiéndolo en el contexto del sistema computarizado. Normalmente contiene el alcance del software.

La "Descripción de la información" provee al software de una valiosa ayuda para resolver los problemas presentados. Tanto el flujo como la estructura de la información se documentan. Las interfaces de hardware, software o humanas se describen como elementos externos del sistema, pero se toman en cuenta dentro de las funciones internas del software.

En la "Descripción de las funciones" se especifican los procedimientos que se requieren en cada función para resolver el problema correspondiente. Se añade una narración del proceso, así como las restricciones del diseño. Se puede ilustrar con uno o más diagramas de bloques para visualizar de manera fácil y rápida dichas funciones.

La cuarta sección de la "Especificación de requerimientos" es probablemente la más importante pero, irónicamente, la más olvidada. Aquí se trata sobre los criterios para asegurar que la implementación sea adecuada, y las pruebas necesarias que conduzcan a la certeza de haber tomado en cuenta las restricciones requeridas. Estos puntos se tratan con más detalle en la siguiente fase, en la correspondiente a la del desarrollo.

Finalmente se incluye una bibliografía y un apéndice. La bibliografía contiene referencias a todos los documentos relacionados con el software: la documentación de otras fases de planeación, referencias técnicas, literatura de sistemas similares y estándares del sistema utilizado. El apéndice, por su parte, contiene la información que complementa la especificación: una descripción detallada de los algoritmos utilizados, gráficas, diagramas de flujo, etc.

E) REVISION DE LAS ESPECIFICACIONES

El último paso en el análisis de los requerimientos, es una revisión exhaustiva del documento "Especificación de los requerimientos de software"; es llevada a cabo en conjunto por el analista y el usuario. La especificación constituye el fundamento de la fase de desarrollo, por lo que es preciso tener mucho cuidado durante su revisión.

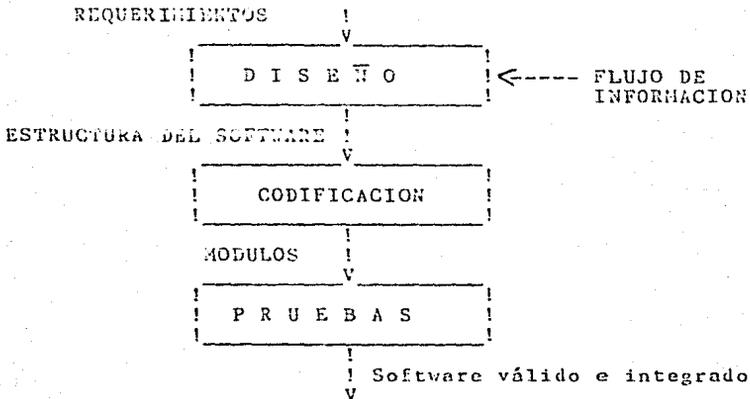
Una manera de revisar los requerimientos es considerando los planteamientos de los siguientes objetivos:

- + La coincidencia de las metas y objetivos del software con las metas y objetivos del sistema en general
- + Que hayan sido descritas las interfaces más importantes que se tienen con el sistema en general
- + Que el flujo y la estructura de la información estén adecuadamente definidos para solucionar los problemas
- + Que los diagramas estén claros
- + Que hayan sido descritas las principales funciones
- + Que sean realistas las restricciones del diseño
- + Que hayan sido consideradas varias alternativas en los requerimientos de software
- + Que hayan sido detallados los criterios de validación
- + Que los criterios de validación sean los adecuados para alcanzar el cumplimiento del sistema
- + Que no haya inconsistencias, omisiones o información redundante
- + Que haya habido una comunicación completa entre el usuario y el responsable del sistema

Una vez que se han aprobado las especificaciones, se procede a la fase del desarrollo. Aún con una muy buena revisión es posible que se presenten diversos problemas propios del desarrollo, pues en la especificación es difícil cubrir las pruebas de una manera exhaustiva; omisiones e inconsistencias pueden pasar desapercibidas, pero son mínimas si se hizo con cuidado dicha revisión.

II.2 FASE DE DESARROLLO

La fase central en el ciclo de vida del software es su desarrollo, que comprende tres etapas: diseño, codificación y pruebas. El flujo de la información durante la fase de desarrollo se ilustra en la figura II.12 (2, pág.129).



Fase de Desarrollo en la Ingeniería de Software
Figura II.12

II.2.1 Proceso del diseño

El diseño es el primer paso en la fase de desarrollo para cualquier producto ingenieril o sistema. Puede ser definido como "el proceso de aplicar varias técnicas y principios con el propósito de definir un dispositivo, un proceso, o un sistema con el suficiente detalle, de manera que permita su realización física" (19).

La meta del diseño es producir un modelo o representación de una entidad que después será "construida". El proceso por el que se desarrolla el modelo combina varias características:

- * Intuición y juicio basado en la experiencia de construir entidades similares.
- * Un conjunto de criterios que proporcionan facilidad para la evaluación del modelo.
- * Un proceso iterativo que permite una representación final del diseño.

* Un conjunto de principios que guían dicho desarrollo.

La etapa del diseño incluye dos pasos:

a) Diseño preliminar.

Las entradas al diseño preliminar son los requerimientos del software, el flujo de información y la estructura. La estructura del software, también llamada arquitectura del software, define la relación entre los principales elementos del programa. Este diseño es un intento de desarrollar el software con la técnica que recorre todo el sistema de arriba a abajo (top-down).

b) Diseño detallado.

El diseño detallado muestra la transformación de los elementos estructurados en procedimientos de software. El flujo de la información o su estructura, determinados en los requerimientos, se vuelven una útil herramienta que facilita una representación global del software.

El proceso del diseño es aquel en que los requerimientos son trasladados en una representación de software. Inicialmente la representación describe un panorama general del software. Un paso siguiente permite refinar el diseño de tal manera que se esté muy cerca del código.

Para evaluar la representación del diseño se pueden tomar como parámetros su organización jerarquizada, su modularidad, estructuración, cohesión, etc. Términos que se definen en la sección II.4 del presente trabajo.

Las anteriores características no son conseguidas por casualidad, sino que el proceso de diseño en la ingeniería de software obtiene la mejor optimización a través de una metodología sistemática, que se sigue estudiando con profundidad.

Documentación del diseño.

La "Especificación del diseño" tiene un doble propósito: Primero, proveer de una guía para la implementación del software (codificación), y segundo, guiar las pruebas y consultas durante el mantenimiento una vez que el proyecto es ejecutado. La especificación puede tener cambios a lo largo del proceso, por tanto, es importante el control y la revisión de la documentación en cada paso de la fase del desarrollo. El formato que sigue a continuación puede servir como modelo para la especificación del diseño.

Especificación del diseño del software

- 1.0 Alcance del diseño
 - 1.1. Objetivos del sistema y papel del software como un elemento del sistema
 - 1.2. Interfaces
 - a) Interface humana
 - b) Interface con el hardware
 - c) Interface con el software
 - 1.3. Principales funciones del software
 - 1.4. Definición de la base de datos externa
 - 1.5. Limitaciones y restricciones
- 2.0 Descripción del entorno al sistema
 - 2.1 Descripción de los archivos externos al sistema
Para cada archivo:
 - a) Contenido
 - b) Descripción física
 - 2.2 Diagrama de estructura de datos
- 3.0 Descripción del diseño
 - 3.1 Descripción del flujo de datos
 - a) Datos de entrada
 - b) Especificaciones para el proceso
 - c) Datos de salida
 - 3.2 Diagrama de la estructura del proceso
- 4.0 Descripción de los módulos
Para cada módulo:
 - a) Reseña del proceso
 - b) Descripción de sus interfaces
 - c) Detalle del módulo
- 5.0 Interfase externa
 - 5.1. Diagrama de la interfase con la base de datos
 - 5.2. Mensajes
 - 5.3. Paquetería
- 6.0 Equipo y lenguaje
- 7.0 Recuperación de fallas
- 8.0 Certificación de calidad
- 9.0 Apéndices

II.2.2 Codificación

Todos los pasos en la ingeniería de software que se han presentado hasta este punto son dirigidos hacia un objetivo final: Trasladar las representaciones del software de manera que sea "entendido" por una computadora. Para ello se utiliza un lenguaje artificial tales como FORTRAN, COBOL, PASCAL, o lenguaje ensamblador.

Quando se considera la codificación como un paso más en el proceso de la ingeniería de software, parece que es un resultado natural del diseño. Sin embargo, las características del lenguaje de programación y el estilo de la codificación puede afectar profundamente la calidad del software y de su mantenimiento.

La etapa de "Codificación de programas de computadora" tiene como objetivo traducir las especificaciones del proceso de cada módulo descritas en el paso anterior, en instrucciones ejecutables por un lenguaje de programación específico.

La codificación de programas de computadora había sido considerada un arte hasta los años cincuenta debido a la falta de métodos para el diseño de programas. Actualmente, gracias al trabajo de personas como Dijkstra (20), Wirth (21) y Weinberg (22), se han empezado a desarrollar tecnologías de programación basadas en el método de análisis y resolución de problemas.

Quando el problema es complejo no es fácil de convertirlo inmediatamente en instrucciones de máquina o código. Es así que se hace necesario traducir el programa a un "pseudocódigo", a fin de codificar posteriormente las operaciones descritas en un lenguaje que sea interpretable por la computadora. Para ello se utilizan varias técnicas, entre ellas la "Programación estructurada", la cual se examina más adelante.

Esta etapa de codificación se lleva muchas veces a cabo en paralelo con la del diseño, dependiendo del enfoque que se tome. Se han propuesto dos enfoques diferentes (23,pág.66): el radical y el conservador. El radical se basa en el diseño de uno de los programas de la arquitectura, procediendo a su codificación y prueba inmediatamente después. El enfoque conservador, por su parte, sugiere terminar primero el diseño de todo el sistema, y pasar posteriormente a las etapas de codificación y pruebas. El enfoque apropiado depende de las circunstancias, entre las que hay que considerar las siguientes:

- 1) Conocimiento del usuario: Si el usuario no tiene bien clara la idea de lo que quiere y cambia continuamente las especificaciones, es conveniente

adoptar el enfoque radical; en cambio, si el usuario sabe con precisión lo que quiere, es más conveniente utilizar el enfoque conservador.

- 2) Limitación de tiempo: Si se está sujeto a fuertes presiones de tiempo para producir rápidamente resultados tangibles es conveniente el enfoque radical.
- 3) Programa de actividades preciso: si se requiere cumplir con calendarios precisos y detallados para la utilización de recursos, es más conveniente el enfoque conservador, pues de adoptarse el radical, es casi imposible determinar con precisión la duración y los costos de desarrollo del sistema, ya que no se conocen ni el número, ni la complejidad de los módulos del programa.

Independientemente del enfoque adoptado, los módulos del programa deben ser codificados. Para ello se puede recurrir a varias estrategias, entre ellas, por ejemplo, la codificación descendente (Top-Down Programming) que posteriormente se trata con mayor detalle (Cap.II.4).

Estilo de codificación

Después de que el código fuente es generado, la función de un módulo es no hacer referencia a la especificación del diseño, en otras palabras, el código debe ser entendible por sí mismo. El estilo de codificación acompaña una filosofía que persigue simplicidad y claridad. En un texto al respecto, Kernighan and Plauser (24) señalan que: "De cómo se exprese cada una de las instrucciones dependerá en gran medida la inteligibilidad del todo...".

Los elementos del estilo incluyen una documentación interna (a nivel código fuente), métodos para la declaración de datos, un acercamiento a la construcción de proposiciones, y técnicas de entrada y salida.

a) Documentación del código.

La documentación interna del código fuente comienza con la selección de los nombres de identificadores (variables o etiquetas), continúa con la colocación y composición de comentarios, y concluye con la organización visual del programa.

La selección de nombres significativos para los identificadores es crucial para su comprensión. Los lenguajes que limitan los nombres de las variables a sólo unos caracteres oscurecen su significado.

Se puede debatir que las expresiones muy largas oscurecen el flujo lógico y dificultan las modificaciones, sin embargo, estudios elaborados (25) han indicado que aún para programas "pequeños" los identificadores significativos mejoran la comprensión. Todos los lenguajes de programación de propósito general proveen la facilidad para expresar comentarios en lenguaje natural. Aunque se dude sobre la cantidad suficiente de comentarios o su lugar dentro del código, lo cierto es que son necesarios, pues facilita tanto la comprensión para otros lectores del código, así como sirven de guía durante la última fase del ciclo de vida del software: su mantenimiento.

b) Declaración de los datos.

La complejidad y organización de la estructura de los datos son definidas durante el diseño. El estilo de la declaración de datos se establece cuando el código es generado.

El orden de la declaración de los datos debe de estandarizarse aún si el lenguaje de programación no lo requiere con necesidad. Cuando hay una multiplicidad de nombres de variable vale la pena declarar cada una en una línea y en orden alfabético. Igualmente respecto a las etiquetas (numéricas o alfabéticas) deben llevar un orden.

c) Construcción de las instrucciones.

La construcción del flujo lógico del software se establece durante el paso del diseño. La construcción de instrucciones individuales, sin embargo, es parte de la etapa de codificación. Dicha construcción debe cumplir con una regla permanente: Cada instrucción debe ser simple y directa; el código no debe ser confuso aunque persiga eficiencia. Muchos programadores permiten múltiples instrucciones por línea, sin embargo, el espacio que se ahorra no lo justifica debido a la poca inteligibilidad que resulta. Hay ciertas recomendaciones que ayudan a una simplificación en el código:

- * Evitar el uso de condicionales complicados.
- * Evitar la utilización de condicionales "negativos".
- * Evitar un acumulamiento de iteraciones (loops) o condicionales.
- * Usar paréntesis para clarificar las expresiones lógicas o aritméticas.
- * Utilizar espacios y símbolos legibles para clarificar el contenido de la instrucción.

- * Pensar si el código sería entendible por las demás personas que lo leyeran.

d) Sobre la entrada-salida

Las entradas y salidas se establecen durante el análisis de los requerimientos y en el diseño, no en la codificación. Sin embargo, la manera en que la entrada-salida es implementada puede ser la característica que determine la aceptación del sistema por parte del usuario.

El estilo de la entrada-salida (E/S) varía dependiendo del grado de interacción humana con el sistema. Para sistemas orientados a turnos por colas ("batch") se requiere normalmente de características tales como una organización lógica para las entradas, unas rigurosas pruebas de verificar la E/S, disponer de los errores encontrados en la E/S, un reporte con formatos legibles para la salida, etc.

Quando el sistema está orientado a procesos interactivos de E/S se necesitan características tales como un esquema que guíe la entrada, una salida en lenguaje natural, una consistencia en los formatos de E/S, capaz de detectar errores y marcarlos inmediatamente sin que por ello se termine de ejecutar el proceso, etc.

Para cualquier tipo de sistema se pueden tomar en cuenta las siguientes características en la construcción de la entrada-salida:

- * Validación de todos los datos de entrada.
- * Procurar que las entradas sean de un formato simple.
- * Usar un indicador de fin-de-datos en lugar de requerir que un usuario especifique el contador de iteraciones.
- * Solicitar la entrada de etiquetas o variables interactivas que especifiquen cambios o valores límites.
- * Catalogar todas las salidas y diseñar todos los reportes.

El estilo de las entradas-salidas son afectadas por otras características, tales como los dispositivos de entrada-salida (p.ej. el tipo de terminal, la graficadora, tipo de impresora, etc), la sofisticación del usuario o del sistema, la papelería especial, etc. Wasserman (26) señala un conjunto de recomendaciones para los sistemas interactivos, que pueden ser utilizados a lo largo de la Ingeniería de Software, y entre éstas se encuentran:

- * Hacer que los aspectos técnicos y propios de la computadora sean invisibles al usuario.
- * Hacer el programa "a prueba de balas", es decir, hacer virtualmente imposible que el usuario cause una terminación anormal del programa.
- * Preguntar al usuario si cualquier requerimiento puede ser mejorado.
- * Proveer de una ayuda "en línea" durante el uso del programa (menús, ayudas, etc).
- * Hacer los requerimientos de las entradas de acuerdo a las habilidades del usuario.
- * Coordinar los mensajes de salida de acuerdo a la velocidad del dispositivo de salida.
- * Mantener consistente el tiempo de respuesta.
- * Minimizar el trabajo extra para el usuario en caso de que ocurra un error.

II.2.3 Integración

El objetivo principal de esta etapa es "la integración funcional de los diversos módulos de un programa, ajustándolos a las particularidades del sistema de cómputo utilizado" (23,pág.67).

En esta etapa de integración se toman en cuenta dos aspectos importantes: la manera como se combinan los módulos para formar programas y el diseño de las pruebas que permiten identificar errores de codificación. Para ello pueden seguirse dos estrategias:

- a) Integración incremental. Validación de nuevos módulos (no probados) agregándolos a módulos ya probados e integrados.
- b) Integración no incremental, que consiste en la validación de programas de computadora a partir de pruebas modulares independientes.

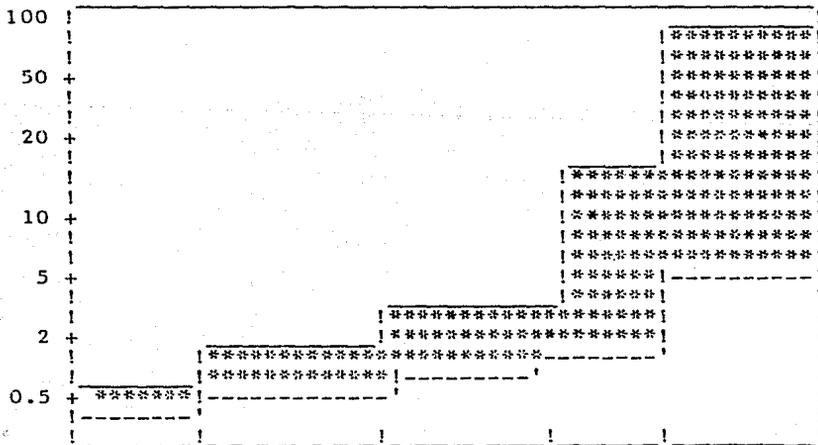
El proceso de integración no incremental requiere de un mayor esfuerzo, pues para probar un módulo es necesario crear unos módulos ficticios que simulen los verdaderos. En cambio, la integración incremental se puede probar sin dichos módulos. Este tipo de integración permite detectar

con relativa facilidad los errores de programación relacionados con la intercomunicación entre los módulos.

La integración incremental permite identificar errores en los módulos del programa que se adicionaron más recientemente. La integración no incremental no propicia esta identificación, ya que los errores emergen, en este caso, hasta que todos los módulos se ensamblen. Estos errores pueden estar igualmente en cualquier módulo del programa complicando gravemente la identificación de los componentes incorrectos.

Los errores de programación contribuyen a engrosar los costos de ésta; el detectarlos oportunamente ayuda a disminuir el precio de la programación. La variante incremental favorece precisamente esta detección oportuna de errores, por lo que se sugiere preferentemente su uso. Un interesante estudio sobre el costo que resulta de detectar un error en las diversas etapas del desarrollo (fig.II.14), nos muestra cómo se incrementa conforme se avanza en el desarrollo (1,pág.15).

Costo relativo para la corrección de un error



Diseño Codificación Integración Pruebas Operación

Relación del costo en la detección del error durante la fase de desarrollo
Figura II.13

II.2.4 Pruebas del software

La prueba del software es un elemento crítico para asegurar la calidad del software y representa la última revisión de las especificaciones, del diseño y de la codificación.

La primera etapa de las pruebas se refiere a la de los módulos, y ésta puede ser seguida por la prueba de integración. Un aliado del proceso es la localización y la determinación de la posibles causas de error. Si se observa toda la estructura del desarrollo del software (fig. II.6) se ve que las pruebas constituyen aproximadamente del 30 % al 50 % del total de la fase.

Es importante recordar que los propósitos, métodos y herramientas para las pruebas cambian a lo largo de las diferentes fases del desarrollo del software. Por ejemplo, si tomamos un módulo individual para probarlo, la prueba inicial es ejecutada sobre el software al checar la sintaxis del programa por el compilador (o ensamblador) utilizado. Por supuesto, el código no puede estar correcto si la sintaxis está incorrecta. Después, el programador alimenta con un conjunto de datos al sistema y examina los resultados de salida para detectar los errores de ejecución. A continuación se observan también las diversas funciones que están involucradas con dicho módulo. Al final de la etapa de pruebas, se extreman los esfuerzos para detectar posibles fallas futuras, poniendo las variables o parámetros en los límites de sus rangos o probando situaciones críticas que puedan presentarse en alguna rara ocasión. Y así como se hace con cada módulo, se hace con todo el sistema durante la integración final.

Para la aceptación de las pruebas, dos criterios prevalecen: Primero, el encargado de desarrollar el software trata de demostrar todas las características que contiene el sistema y procesa varias pruebas con su trabajo, de tal manera que se ajuste a los requerimientos solicitados. Y segundo, quien acepta dichas pruebas trata de objetar con el máximo número de casos que se puedan presentar a fin de que queden contemplados.

Glen Myers (27), en un excelente libro que señala pruebas sobre el software, asienta un número de reglas que pueden servir como objetivos en esta etapa:

- * Las pruebas son el proceso por el cual se ejecuta un programa con el intento de encontrar un error.
- * Una buena prueba es aquella que tiene una alta probabilidad de encontrar un error no descubierto.
- * Una exitosa prueba es aquella que descubre los errores ocultos.

Al terminar la etapa de integración, en la que se han identificado los errores de codificación, quedan por detectarse los de análisis, de especificación y de diseño:

- a) Errores de análisis. Son aquellas fallas no detectadas en la etapa del "Análisis de los requerimientos", y que impiden satisfacer las necesidades del usuario.
- b) Errores de especificación. Son los causados por especificaciones incompletas o imprecisas, y que dan lugar a una interpretación equivocada de las mismas.
- c) Errores de diseño. Son debido a que el diseño resulta imposible de implantar por el equipo de cómputo que se pretende utilizar o porque es incapaz de satisfacer los requerimientos del programa.

Para detectar estos tipos de errores se realizan varios tipos de pruebas denominadas "Pruebas de alto nivel" (23, pág.70), y entre las más importantes están las cuatro siguientes:

1. Pruebas funcionales

El objetivo de estas pruebas es encontrar errores de análisis, de especificación y de diseño. En general, los casos para este tipo de prueba son generados mediante técnicas de entrada y salida, en donde la lógica interna del programa es irrelevante, pues simplemente se visualizan casos de pruebas desde un punto de vista externo.

La búsqueda de errores conceptuales y de especificación debe ser organizada. Para ello se diseñan las pruebas funcionales jerarquizando los componentes a probar en distintos niveles de abstracción.

2. Pruebas de implantación

Por lo general, los programas de cómputo son sometidos a pruebas modulares de construcción y funcionales en un entorno distinto al que serán finalmente integrados.

Las pruebas de implantación tienen como objetivo encontrar errores de especificación, concepto y diseño en el ambiente real (equipo, sistema operativo, etc), en donde los programas son finalmente instalados.

3. Pruebas de sistema

Estas pruebas consisten en poner en tela de juicio la compatibilidad del sistema con la documentación del usuario. La revisión exhaustiva del "Manual del usuario" es una parte integral del proceso de prueba del sistema. El beneficio más importante de esta prueba es la revisión de la documentación de usuario a fin de corregirla en caso necesario. Este manual, elaborado en paralelo con la realización de las pruebas de sistema, constituye la documentación de carácter operativo, que permite al usuario del sistema su correcta explotación.

La prueba de sistema no sólo es relevante para sistemas de programa de cómputo, sino que también es aplicable al caso de programas aislados. El primer contacto entre muchos usuarios y los programas es el manual. Un buen manual, fácilmente entendible y sin errores, facilita no sólo el empleo del programa por parte del usuario, sino inclusive su aceptación.

4. Pruebas de aceptación

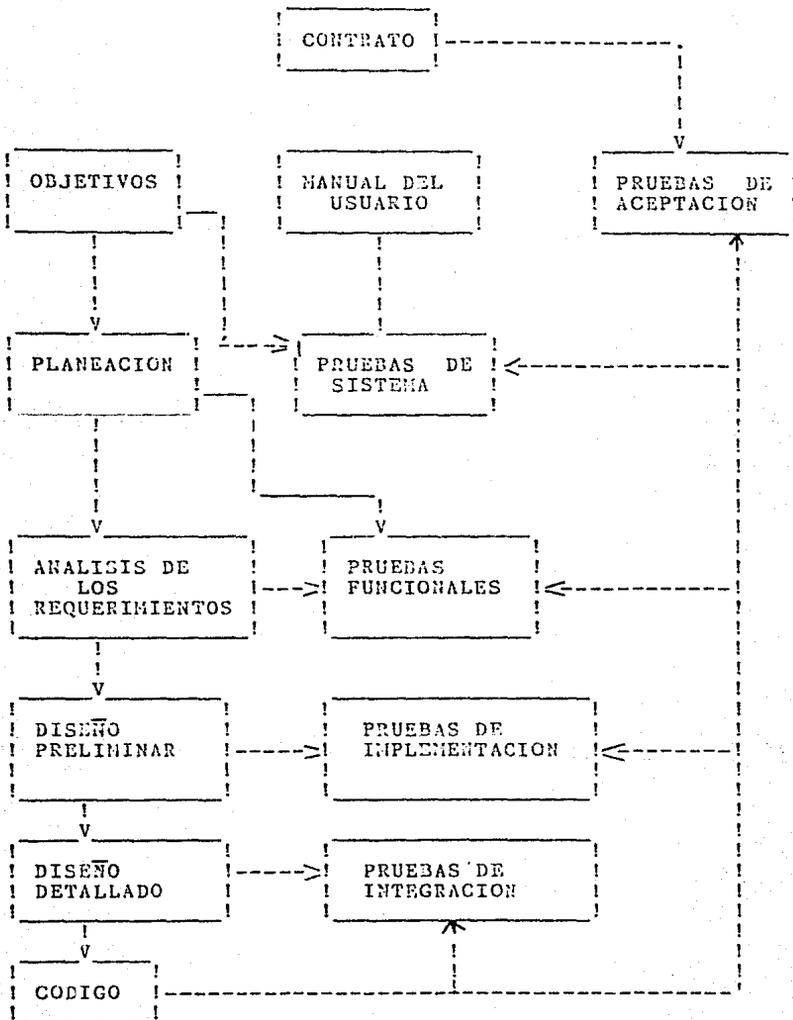
Las pruebas de aceptación tienen como objetivo comparar los resultados con el contrato original. Por lo general, el usuario es el responsable de esta prueba. En el caso de sistemas grandes y complejos, es práctica recomendada delegar la responsabilidad de esta prueba a una tercera institución, distinta del usuario y de la organización que desarrolla el producto. Puede decirse que las pruebas de aceptación son un subconjunto de las pruebas de implantación. La figura II.15 ilustra el alcance de los distintos tipos de prueba.

Técnicas para el diseño de casos de prueba

Existen dos técnicas para el diseño de los casos de prueba, y aunque ninguno puede considerarse el ideal o completo, ambos tienen sus propias ventajas y deficiencias:

- a) Métodos de análisis de entrada-salida: el cual consiste en diseñar casos de prueba a partir de la especificación del programa, sin requerir del conocimiento de la forma en que el programa ha sido realizado.
- b) Métodos de cobertura de lógica: son los que permiten diseñar casos de prueba a partir del diseño y código del programa.

La utilización correcta de ambos métodos permite obtener un número reducido de casos de prueba, con alta probabilidad de identificar los errores.



Alcance de los diferentes tipos de prueba
 Figura II.15

II.3 FASE DE MANTENIMIENTO

El mantenimiento del software ha sido caracterizado por Canning (28) como un "iceberg", pues normalmente se piensa que sólo lo que se ve inmediatamente del sistema es lo que existe. Pero realmente hay una cantidad enorme de problemas potenciales y de esfuerzo detrás del sistema. El mantenimiento de un sistema computarizado puede llevarse alrededor del 60% de todo el esfuerzo en el desarrollo de la organización. El porqué de tal cantidad, se podría explicar con la respuesta que da Rochkind (29): "Los programas de computadora están siempre cambiando. Hay tópicos que cambiar, características que añadir y optimizaciones que hacer... Junto a los problemas cuyas soluciones requieren ser cambiadas en un primer momento, el hecho de los cambios crean a su vez problemas adicionales...".

El mantenimiento del software es algo más que "corregir errores". Se puede definir el mantenimiento describiendo cuatro actividades que se llevan a cabo una vez entregado el sistema para su uso:

1) La primera actividad de mantenimiento ocurre tomando en cuenta que es imposible asumir que las pruebas del software han descubierto todos los errores latentes en el software del sistema computarizado. El proceso que incluye el diagnóstico y la corrección de uno o más errores es llamado "mantenimiento correctivo".

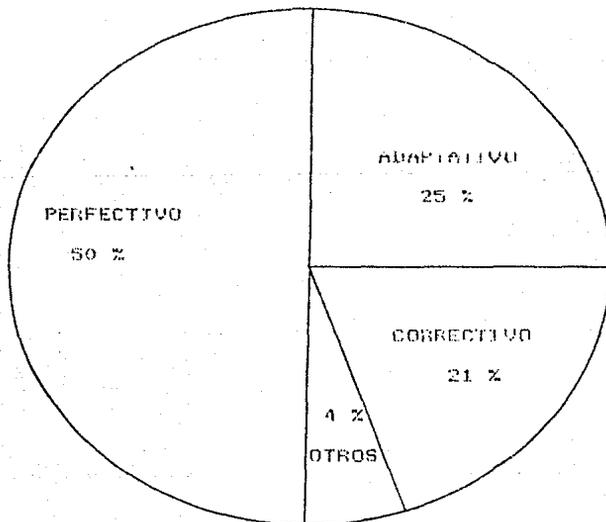
2) La segunda actividad que contribuye a definir el mantenimiento se debe al rápido cambio que sucede en cada aspecto de la computación: las nuevas generaciones de hardware, los nuevos sistemas operativos, equipo periférico y otros elementos del sistema que frecuentemente se actualizan o modifican. La vida útil de la aplicación del software, por otro lado, puede fácilmente sobrepasar los 10 años, siendo ya otro sistema ambiental el que lo rodea, y por tanto, exigiendo hacer grandes mejoras. Por tanto, el "mantenimiento adaptativo" es la actividad que modifica el software actualizándolo a los cambios que es sometido el sistema en caso de ser necesario.

3) La tercera actividad que puede ser aplicada a la definición del mantenimiento ocurre cuando un "paquete" de software tiene éxito. Mientras que el sistema de software es utilizado se reciben, por parte de los usuarios, recomendaciones para nuevas aplicaciones, modificaciones de las funciones ya existentes y mejoramientos en general. Para satisfacer dichos requerimientos se ejecuta entonces el "mantenimiento perfectivo". Esta actividad es la que se lleva mayor tiempo y esfuerzo en la fase del mantenimiento.

4) La cuarta actividad del mantenimiento ocurre cuando el software es modificado para mejorar tanto en el futuro mantenimiento como en su confiabilidad, o para proveer de una sólida base para futuros mejoramientos. También es llamado "mantenimiento preventivo", y esta actividad es relativamente rara en el ambiente del software.

Los términos usados para describir las primeras tres actividades son señaladas por Swanson (30). El cuarto término es comúnmente utilizado en el mantenimiento del hardware y otros sistemas físicos.

Aproximadamente la mitad de todo el mantenimiento del software es perfectivo, como se ilustra en la figura II.16; el porcentaje mostrado en la figura está basado en estudios de Lientz y Swanson (31) sobre 487 organizaciones de desarrollo de software. Estos datos dan la mejor información sobre las proporciones en el mantenimiento del "iceberg".



Distribución de las actividades en la
Fase de Mantenimiento
Figura II.16

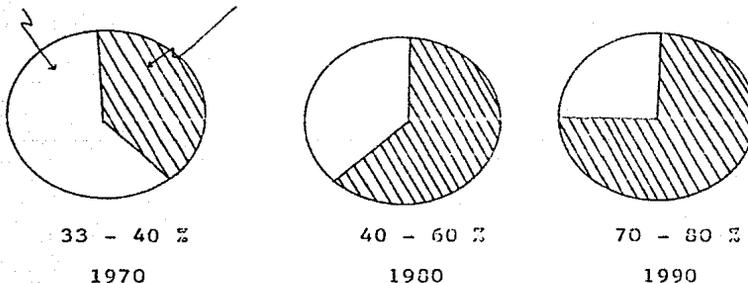
II.3.1 Características del mantenimiento

El mantenimiento del software había sido tomado como una fase de poca importancia. Hay muy poca literatura sobre el mantenimiento en relación con la que hay sobre la planeación o el desarrollo del software.

Para comprender las características del mantenimiento del software es conveniente considerarlo desde tres puntos de vista:

1. Las actividades requeridas para completar la fase de mantenimiento y el impacto de la investigación de la ingeniería de software sobre la eficacia de tales actividades.
2. El costo asociado con la fase de mantenimiento del software. Este se ha ido incrementando durante los últimos 20 años, como se ve ilustrado en la figura II.17 (2,pág.526).

Desarrollo Mantenimiento



Porcentaje del Costo del Mantenimiento respecto al Desarrollo del sistema
Figura II.17

3. Los problemas que frecuentemente se encuentran cuando se descuida el mantenimiento. Tales como el no tener documentado el sistema, o lo esté inadecuado; el que la mayoría del software no está diseñado para cambios.

II.3.2 Efectos del mantenimiento

La modificación del software es peligrosa. Muchas veces falla todo por cambiar sólo una proposición. Desafortunadamente cada vez que un cambio es introducido en un procedimiento lógico muy complejo, la posibilidad de error crece. La documentación del diseño y unas pruebas regresivas cuidadosas ayudan a eliminar los errores, sin embargo, es común que se produzcan errores debido al mantenimiento. Freedman y Weinberg (32) definen tres categorías principales de estos efectos:

1) Efectos en la codificación.

Sucede cuando se cambia algún texto. Una simple coma o punto puede tener consecuencias desastrosas. Es conveniente llevar en una bitácora un control sobre las modificaciones al texto con la fecha y el nombre de quien las realizó.

2) Efectos en los datos.

Estos efectos ocurren como una consecuencia de la modificación hecha en la estructura de la información que se maneja (33): la redefinición de constantes globales o locales, la redefinición de registros o archivos, el incremento o decremento en el tamaño de un arreglo, la inicialización de variables o banderas, etc.

3) Efectos en la documentación.

Estos errores ocurren cuando los cambios en el código fuente no son reflejados en la documentación del diseño o en el manual del usuario. Siempre que hay un cambio en el flujo de datos, en la estructura del software, en los procedimientos, o en cualquier otra característica, es necesario actualizar la documentación técnica. Una documentación que no esté actualizada puede incluso ser peor que no tenerla. Para un usuario, el software es tan bueno como lo sea la documentación (escrita o interactiva) que describa su uso.

II.3.3 Manual del usuario.

El "Manual del usuario" es un instructivo para instalar, operar y mantener el sistema o paquete de programación (Cfr. 23, pág.75). Dependiendo de la complejidad del sistema y de la extensión de su documentación, el manual del usuario puede ser presentado en un sólo documento, o bien, como un conjunto de guías que tratan en forma independiente los aspectos básicos de instalación, operación y mantenimiento:

a) Guía de Instalación.

Aquí se define la manera y los medios de transportar el sistema de programación al equipo de cómputo. Contendrá a su vez, el equipo y la programación necesaria para que el paquete de programación funcione apropiadamente, el procedimiento de instalación y la descripción de las pruebas que permitan constatar la correcta instalación.

b) Guía de operación.

Este manual orienta sobre la manera de utilizar el sistema, señalando diversos aspectos para su mejor uso, pudiendo contener cuatro aspectos:

- b.1 Un panorama general del sistema.
- b.2 La descripción de la interfase entre el hombre y la computadora.
- b.3 Recomendaciones sobre su uso.
- b.4 Un diccionario de términos que defina el significado de palabras y claves utilizadas en la operación del sistema en términos que el usuario pueda entender.

c) Guía de mantenimiento.

Es un método que define la manera en que el sistema fue realizado y hace referencia a los documentos de desarrollo de la programación que deben ser consultados cuando se deseen hacer correcciones, extensiones o adaptaciones. La descripción del sistema debe ser estructurada y objetiva.

II.4 RESUMEN

La metodología sugerida en este capítulo es la adoptada por Roger S. Pressman (2) con pequeñas modificaciones, y es equivalente en lo esencial a otros modelos desarrollados por diversos autores. En la tabla de la figura II.18 (Cfr. 23, pág.30) se presenta un resumen de las diversas fases enunciadas según el autor, pudiéndose observar las diversas nomenclaturas para cada fase así como su equivalente.

"Definición de Requerimientos" :

Roger S. Pressman: a) Definición del sistema
b) Planeación del software
c) Análisis de los requerimientos

Freeman : a) Necesidades
b) Análisis
c) Especificación

Boehm : a) Requerimientos del sistema
b) Requerimientos de la programación

Instituto de Investigaciones Eléctricas (IIE):
a) Objetivos
b) Análisis
c) Requerimientos

"Diseño"

Roger S. Pressman: a) Diseño preliminar
b) Diseño detallado

Freeman : a) Diseño de la arquitectura
b) Diseño detallado

Boehm : a) Diseño preliminar y detallado

IIE : a) Arquitectura
b) Diagrama de estructura
c) Detalle de módulos

"Desarrollo"

Roger S. Pressman: a) Codificación
b) Integración

Freeman : a) Implantación

Boehm : a) Codificación y depuración
b) Pruebas y pre-operación

IIE : a) Código
b) Planes de pruebas
c) Procedimientos de prueba

"Operación"

Roger S. Pressman: a) Pruebas y mantenimiento

Freeman : a) Mantenimiento

Boehm : a) Operación y mantenimiento

IIE : a) Operación y mantenimiento

Diversas nomenclaturas de las fases de la
ingeniería de Software según diversos autores
Figura II.18

II.5 TECNICAS Y HERRAMIENTAS EN LA INGENIERIA DE SOFTWARE

II.5.1 Su concepto.

Hasta ahora se ha señalado la necesidad de organizar el desarrollo de programas de computadora en tres fases. Este ordenamiento de actividades permite un refinamiento sistemático y progresivo (33) de los productos de programación y una adecuada administración de los recursos humanos y financieros. Ahora se exponen algunas de las diferentes técnicas que se emplean durante las diversas fases en el desarrollo sistemático del proceso de programación.

II.5.2 Programación Estructurada

La primera técnica estructurada discutida y practicada con amplitud fue la "Programación estructurada" (18). Está basada en los estudios realizados en 1960 por Dijkstra, Wirth, Bohm y Jacobini.

Se han dado diversas definiciones respecto al concepto de "programación estructurada", así, algunos la han limitado a señalar que la restricción de no utilizar instrucciones de ramificación del tipo "GO TO". Wirth, por su parte, la define de la siguiente manera: "la programación estructurada es la formulación jerárquica de estructuras de control de flujo anidadas e instrucciones computables para formar programas" (21).

Teniendo en cuenta varias definiciones, se han recopilado en la siguiente: "...la programación estructurada es la aplicación de métodos básicos de descomposición de problemas para establecer una estructura jerárquica fácilmente manejable, a través del proceso llamado refinamiento progresivo" (23.pág.133).

Para poder llevar a cabo la estructura jerárquica se vió la necesidad de eliminar por completo la instrucción "GO TO", y contar únicamente con una estructura de decisión y una sola instrucción para la repetición. Bohm (34) demostró matemáticamente que cualquier estructura de control puede ser sintetizada en términos de las estructuras de control IF-THEN-ELSE y DO-WHILE. Algunos han añadido la secuencia simple como la tercera estructura básica (35),(36).

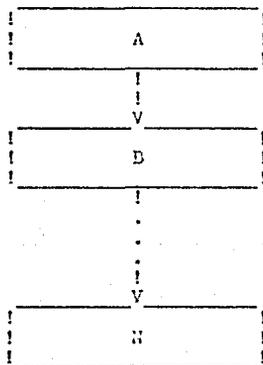
En la práctica, la programación estructurada es una herramienta que permite escribir programas claramente concebidos, más legibles y por tanto, con menos errores; sin embargo, no puede afirmarse que el mero uso de las estructuras básicas de control lleva natural y automáticamente a escribir programas estructurados; una serie de reglas mecánicas no puede ser sustituto de la claridad de pensamiento. Es cierto, sin embargo, que un

programa bien estructurado será concebido usando dichas estructuras en sus bloques de construcción (37). A continuación se presenta una breve descripción de cada una de estas estructuras.

1) Secuencia

La secuencia es una estructura de proceso que consiste en enunciar, una después de otra, una serie de instrucciones, es decir, indica que si no hay una estructura de control, el flujo del control pasa de una instrucción a la siguiente en secuencia. Una instrucción es una frase en lenguaje natural fácilmente traducible a código o una estructura de proceso (fig. II.19).

DIAGRAMA DE FLUJO



PSEUDOCODIGO

- Instrucción A

- Instrucción B

- Instrucción N

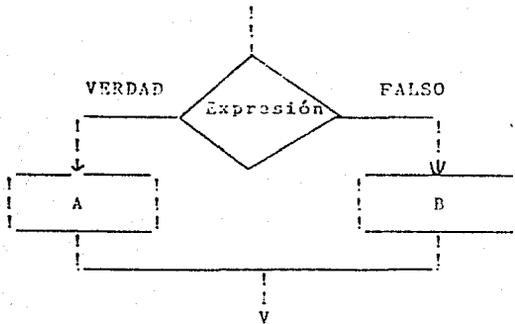
Estructura básica de control: SECUENCIA
Figura II.19

2) Decisión.

La decisión es una estructura básica de control en el proceso, que permite especificar alternativas en la ejecución de instrucciones dependiendo de una condición. Cuando la condición resulta verdadera, realiza lo especificado en el "ENTONCES" (THEN), si resulta falsa se ejecuta lo indicado en el "SINO" (ELSE) (fig. II.20)

DIAGRAMA DE FLUJO

PSEUDOCODIGO



SI Expresión
es verdadera
Ejecuta A
SINO
Ejecuta B
FIN (SI)

Estructura de control básica: DECISION
Figura II.20

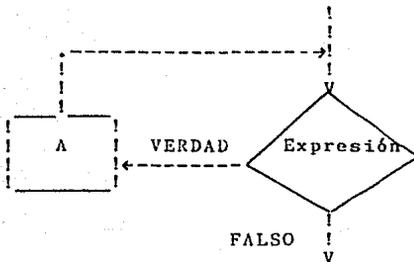
3) Repetición

La tercera estructura de control básica es la repetición, la cual permite la especificación de la ejecución iterativa de una serie de instrucciones, es decir, controla las veces a repetir de una serie de instrucciones. Existen varios tipos de repetición que se utiliza según las necesidades:

3.a) Estructura de repetición: "MIENTRAS" (DO-WHILE).

DIAGRAMA DE FLUJO

PSEUDOCODIGO



MIENTRAS Expresión es
verdadera
EJECUTA A
FIN (mientras)

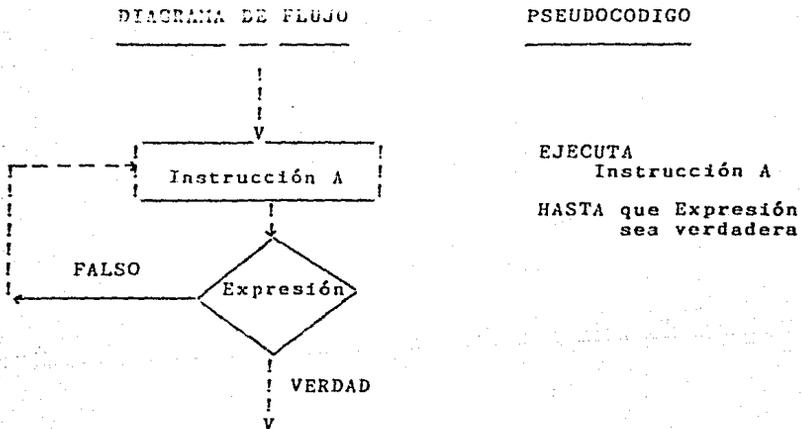
Estructura de repetición : MIENTRAS ("DO-WHILE")
Figura II.21

Si la condición es verdadera, las instrucciones se ejecutan. Posteriormente, la condición vuelve a ser evaluada: si resulta verdadera, las instrucciones se vuelven a ejecutar. El proceso MIENTRAS continúa hasta que la condición sea falsa.

3.b) Estructura de repetición: EJECUTA-HASTA ("DO-UNTIL")

Al ejecutarse la estructura EJECUTA-HASTA, las instrucciones se realizan, y al término de su ejecución la condición es evaluada, y si su valor es falso, las instrucciones se ejecutan de nuevo. Este procedimiento se repite hasta que la condición sea verdadera.

Una diferencia entre la estructura MIENTRAS y la estructura EJECUTA-HASTA es que esta última ejecuta las instrucciones al menos una vez (fig.II.22).

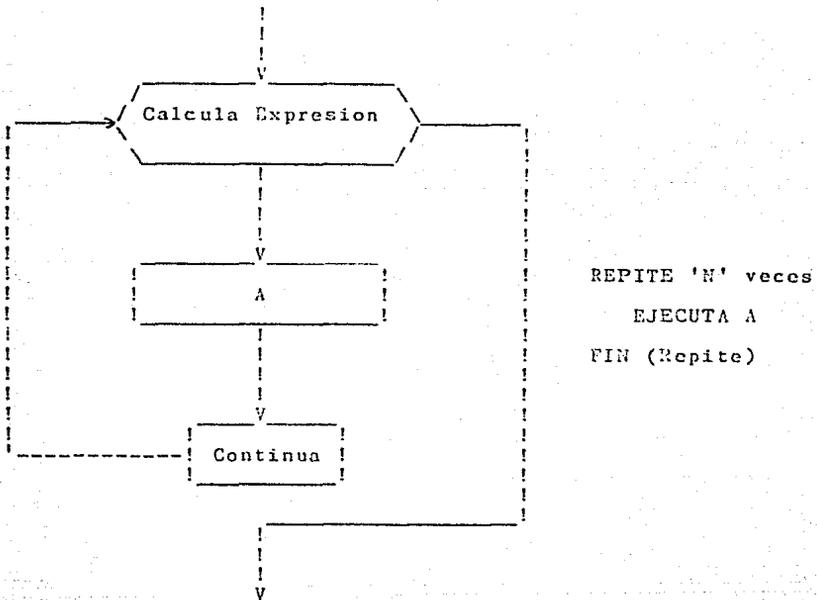


Estructura básica de control: EJECUTA-HASTA (DO-UNTIL)
Figura II.22

3.c) Estructura de repetición: REPITE ("REPEAT")

3.c.1) REPITE en su forma simple.

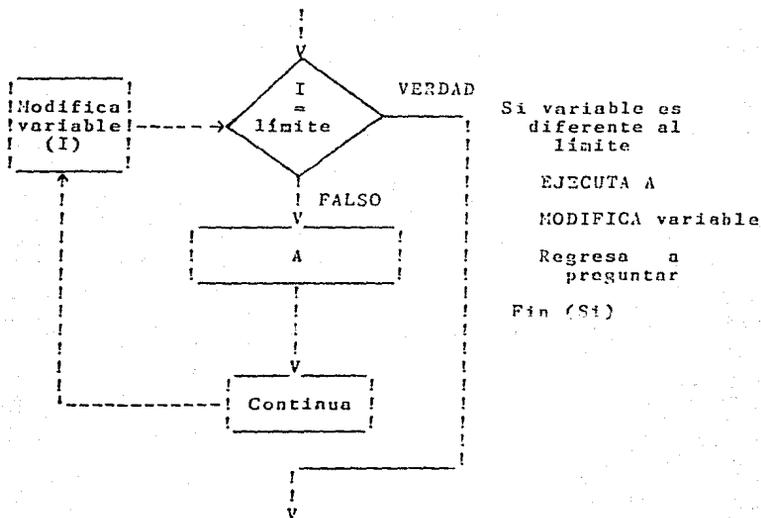
Al inicio de la ejecución del ciclo, se calcula el valor de la expresión que debe resultar un número entero. Este número representa la cantidad de veces que deberá repetirse el ciclo de las instrucciones. Si el valor de la instrucción es cero o negativo, las instrucciones no se ejecutan y se termina la estructura (fig.II.23).



Estructura básica de control: REPITE (simple)
Figura II.23

3.c.2) REPITE en su forma compuesta.

Al empezar la ejecución de la estructura REPITE la variable toma el valor de la expresión "inicio". Si este valor es mayor que la expresión "fin", las instrucciones no se ejecutan y la estructura se termina. Pero si el valor de la variable es menor que la de "fin", entonces las instrucciones se ejecutan, y una vez que éstas han sido procesadas, la variable se incrementa en un valor igual al de la expresión "incremento". Nuevamente se verifica si el valor de la variable es menor que el de "fin", en cuyo caso, las instrucciones se vuelven a ejecutar, de lo contrario, la estructura se termina (fig.II.24).



Estructura básica de control : REPITE (en forma compuesta)
 Figura II.24

II.5.3 Pseudocódigo.

El pseudocódigo, llamado también metacódigo o lenguaje para programación del diseño, es una notación que usando vocabulario de un lenguaje natural (p.ej. el Castellano o el Inglés) combina la sintaxis de otro lenguaje (p.ej. la sintaxis de la programación estructurada). Su diferencia con un lenguaje de programación de alto nivel es la utilización de frases y textos, que por no ser estrictamente lenguaje de programación no puede ser compilado.

Generalmente para el formato en las estructuras de control se anidan o se utiliza sangría para visualizar el control de las instrucciones. Un diseñador que ya domina un lenguaje de programación probablemente escribe el pseudo-código abreviando ese lenguaje particular. Se tienen varias ventajas al utilizarlo:

- * Se forma de una sintaxis fija de palabras clave para proveer a las declaraciones de datos, a los módulos y a las construcciones estructuradas.
- * Contiene una sintaxis libre en lo que respecta al lenguaje natural que utiliza para describir los procesos o instrucciones.
- * La declaración de los datos facilita que se incluyan estructuras de datos simple (escalares y arreglos) o complejas (listas ligadas o jerarquizada).
- * Una ventaja más es que puede ser creado, modificado, controlado y reproducido con facilidad con el uso de una terminal, un editor de textos y de una impresora asociada con el sistema de cómputo.

II.5.4 Técnicas de Arriba-hacia-Abajo (Top-Down).

El término "Arriba-hacia-Abajo" es utilizado para varias etapas en la Ingeniería de Software: para el diseño, la codificación, las pruebas y la integración.

a) En el diseño.

El diseño "Arriba-hacia-Abajo" es básicamente una descomposición del proceso el cual se enfoca sobre el flujo del control o sobre la estructura del control del programa (1, pág. 39).

El primer paso es estudiar todos los aspectos del sistema y dividirlo en un número independiente de funciones o módulos, tal vez de tres a diez. El segundo paso es subdividir cada módulo en otros módulos. El proceso se repite hasta que se obtienen módulos lo suficientemente pequeños para que sean codificados de una manera simple y clara.

Una característica importante del diseño "Arriba-hacia-Abajo" es que en cada nivel los detalles de diseño de niveles inferiores se encuentran "ocultos". Se definen sólo los datos recibidos, el control y las interfases que le siguen. Además, si una estructura de datos está contenida totalmente en un módulo de nivel inferior, no es necesario especificarla sino hasta que se alcance este nivel. En cambio, si algunos datos deben ser compartidos por varios módulos de un mismo nivel, entonces la estructura de datos debe ser definida antes de pasar a esos niveles inferiores.

b) En la codificación.

El uso de la técnica de "Arriba-hacia-Abajo" en la codificación, se comienza al detallar la estructura de control en un primer nivel de descomposición con su

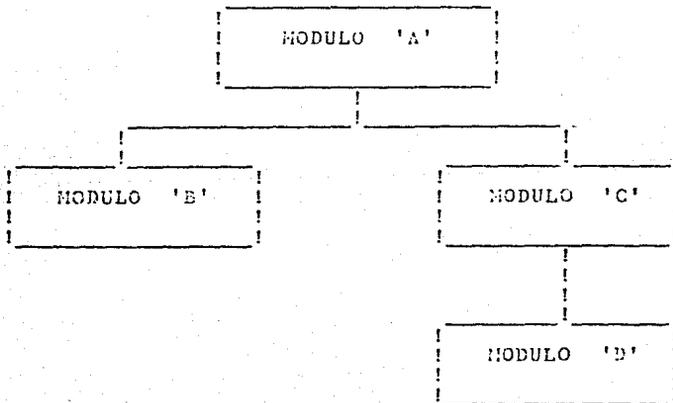
código. Y similarmente como durante el diseño, el código de control se va escribiendo conforme se pasa de nivel.

Es difícil saber el momento en que el módulo es lo suficientemente pequeño y ya no debe ser descompuesto. Algunas personas lo han definido como el tamaño de una hoja de salida de impresión, alrededor de 50 líneas. Harlan Mills (38) lo ha definido como lo suficientemente pequeño de tal manera que no se escape del control del programador.

c) En la integración y pruebas.

Teniendo una estructura de "Arriba-hacia-Abajo" (fig.II.25), la estrategia para realizar las pruebas, comienza con el módulo principal ("A") del programa de computadora. Para probar este primer módulo es necesario diseñar módulos "ciegos" que simulen las funciones de los módulos subordinados ("B" y "C").

La integración, a su vez, se hace añadiendo al módulo principal ("A") los módulos que le siguen inmediatamente ("B" y "C"), después de lo cual se realiza la prueba. Se procede a integrar un nivel inferior ("D"), y así sucesivamente hasta llegar a los últimos módulos.

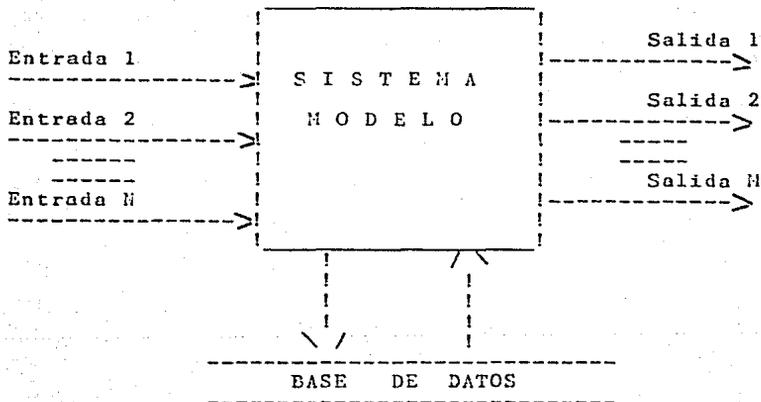


Estructura Arriba-hacia-Abajo
Figura II.25

II.5.5 Diagramas HIPO

Los diagramas HIPO, "Jerarquía más Entrada-Proceso-Salida" (Hierarchy plus Input-Process-Output), surgieron de de unas técnicas desarrolladas por IBM (39) para desarrollar el diseño del software y su documentación.

La característica primordial de estos diagramas es su inherente claridad para representar las relaciones entre los datos de entrada y salida con el procedimiento del software. Se ha llegado a extender estos diagramas en los llamados IPO/DB, los cuales toman en cuenta una base de datos (fig. II.26).



Diagramas HIPO (Hierarchy plus Input-Process-Output)
Figura II.26

El procedimiento del software se especifica en la "caja" o "burbuja" central, el cual es conectado con la información de la entrada, salida y de la base de datos. Esto permite al diseñador explicitar el flujo de la información asociada con el flujo del proceso.

Los diagramas HIPO son útiles en varias etapas de la Ingeniería de Software: al inicio y durante el desarrollo y revisión del diseño, para la realización de las pruebas y el mantenimiento.

Un completo conjunto de diagramas consiste en describir el sistema en tres niveles diferentes :

- (1) Un diagrama de contenido, el cual suministra de manera concisa las funciones del sistema.
- (2) Un diagrama global o general. Muestra en general la entrada, la salida, las actividades de procesamiento que se realizan en cada función.
- (3) Diagramas de detalle. Describen completamente las funciones específicas, utilizando tantos diagramas como sean necesarios, y complementando con notas si así se requiere.

Si el sistema es demasiado grande, se vuelve muy complejo utilizar gran cantidad de diagramas, por lo que se opta por representar sólo los principales niveles. En la mayoría de los casos los niveles son tan altos que no se puede empezar a codificar a partir de ellos, por lo que se podrá continuar con diagramas de flujo o pseudo-código como técnica intermedia antes de comenzar a codificar.

II.5.6 Modularidad

Básicamente, un módulo es "un subprograma de modesto tamaño el cual ejecuta independientemente una función específica" (1, pág.107). Un módulo contiene todos sus elementos y si se quita del sistema sólo afecta la única función ejecutada por el módulo. Similarmente, si se sustituye a un módulo por uno actualizado y se mantienen las mismas especificaciones de entrada y salida, el sistema debe de trabajar con muy poco o ningún cambio.

Aunque no es necesario que un diseño modular esté estructurado o sea con la técnica de "Arriba-hacia-Abajo", se observa claramente que si se diseña con la técnica, resulta invariablemente un diseño modular. Además, si se requiere que el programa sea estructurado, entonces la única-entrada y la única-salida ayudan para que las funciones sean independientes la una de la otra.

Se ha dicho que "la modularidad es el único atributo del software que permite que un programa sea intelectualmente manejable" (40), y es cierto, pues cuando el software es monolítico es ininteligible para el lector.

En resumen, se puede decir que la modularidad, para resolver complejos problemas, se basa en el principio "divide y vencerás". En tanto que dividimos el problema el esfuerzo por resolver cada parte es más fácil. Sin embargo, este proceso de dividir no se sigue indefinidamente, sino que hay un momento en que ya no resulta ventajoso seguir con él.

Es importante notar que un sistema puede ser diseñado modularmente aún si su implementación deba ser "monolítica"; situaciones tales como software de tiempo-real o en microprocesadores en los cuales la rapidez debe ser muy alta o en que la memoria es muy precaria. En tales situaciones puede y debe prevalecer la filosofía de la modularidad en el software diseñado.

a) Acoplamiento.

La medida de interdependencia entre los módulos es llamada acoplamiento. Un bajo índice de acoplamiento refleja módulos dependientes, y por tanto, difíciles de mantener. Un módulo fuertemente acoplado a otro no es transportable, es decir, resulta difícil usarlo en otro programa.

Un módulo es independiente cuando se puede usar en otros programas sin necesidad de cambiar su codificación. Un ejemplo son las subrutinas usadas en la mayoría de los lenguajes de programación para calcular una función trigonométrica.

Yourdon y Constantine (17) han desarrollado una clasificación del acoplamiento entre módulos para poder evaluar el diseño de diagramas de estructura; esta clasificación comprende seis tipos de acoplamiento:

- a.1) Acoplamiento por argumentos: la comunicación entre módulos es exclusivamente entre argumentos; es casi inevitable, pero se pueden disminuir al mínimo estos argumentos.
- a.2) Acoplamiento por estampado: resulta de comunicar como argumento alguna estructura de datos.
- a.3) Acoplamiento por control: ocurre cuando un módulo influye en la ejecución de otro a través de un código o una bandera.
- a.4) Acoplamiento por datos externos: Existe cuando dos o más módulos usan la misma estructura de datos: uno para leer información y el otro para escribir y uno más para modificar.
- a.5) Acoplamiento por datos globales: se presenta cuando un grupo de módulos comparte un área global de memoria. Por ejemplo en FORTRAN se usa el "COMMON" o en COBOL el "DATA DIVISION".
- a.6) Acoplamiento por contenido: Es cuando un módulo puede modificar la secuencia de instrucciones de otro módulo. Esto no sucede en lenguajes como COBOL o FORTRAN sino del tipo ensamblador.

b) Cohesión.

La cohesión, también conocida como solidez, robustez o coherencia de un módulo, es una medida de la fuerza con que los elementos están unidos dentro de un módulo. Estos elementos son las instrucciones o grupos de ellas.

Entre más robusto se presente un módulo, es más independiente y requiere menos acoplamiento con otros, es decir, un módulo es idealmente robusto si ejecuta sólo una función bien definida.

Myers (41) utiliza el término de robustez ("strength") para referirse a la relación que hay entre el control del programa y sus módulos, y así, crea una escala de siete niveles para medir dicha fuerza:

- b.1) Cohesión por función : el módulo resulta funcional si todos sus elementos contribuyen a ejecutar una y solo una función. Este tipo de cohesión es el ideal.
- b.2) Cohesión por información: es cuando el módulo tiene cohesión funcional pero relacionado con una estructura de datos o recursos comunes.
- b.3) Cohesión por comunicación: se presenta cuando un módulo realiza varias funciones, relacionando los datos de una con otra.
- b.4) Cohesión por procedimiento: es cuando se comete el error de incluir varias funciones en un módulo único. El control (no los datos, como en la cohesión por información) fluye de una función a otra.
- b.5) Cohesión por clase: es cuando el módulo ejecuta varias funciones secuencialmente pero con una débil relación entre ellas.
- b.6) Cohesión lógica: Se presenta cuando el módulo ejecuta una serie de funciones relacionadas que son activadas individualmente por un parámetro seleccionador que recibe el módulo.
- b.7) Cohesión por coincidencia: es la peor situación, la más baja cohesión, en donde los módulos ejecutan una función que no está claramente definida o se presentan varias funciones sin relación alguna.

En la práctica, no es necesario determinar con precisión el nivel de cohesión o acoplamiento en que están los módulos, sino más bien lo importante es esforzarse por conseguir una alta cohesión y un bajo acoplamiento así como el identificar cuando esté muy baja la cohesión o muy alto el acoplamiento, tal que su diseño sea modificado para mejorar la independencia entre los módulos.

C A P I T U L O I I I

PLANEACION DEL SISTEMA DE REINSCRIPCIONES

La fase de planeación, tal y como se contempla en el capítulo anterior, queda determinada por las etapas de definición, planeación del software y el análisis de los requerimientos. Ahora bien, estas tres etapas se abordan en el llamado "Reporte para la Especificación de los Requerimientos de Software"; a continuación se presenta dicho reporte para el caso concreto del Sistema de Reinscripciones.

REPORTE PARA LA ESPECIFICACION DE LOS REQUERIMIENTOS DE SOFTWARE EN EL SISTEMA DE REINSCRIPCIONES DE LA U.N.A.M.

1.0 Introducción

1.1 Definición del sistema

1.2 Descripción general

2.0 Descripción de la información

2.1 Diagrama de flujo de datos

2.2 Representación de la estructura de datos

2.3 Descripción de las interfaces externas

2.4 Descripción de las interfaces internas

3.0 Descripción de las funciones

3.1 Funciones

3.2 Narración del proceso

3.3 Restricciones del diseño

a) Factibilidad

b) Alcance del Software

c) Recursos

d) Restricciones del sistema

4.0 Criterio de validación

4.1 Clases de pruebas

4.2 Consideraciones especiales

5.0 Bibliografía

1.0 Introducción

El presente documento establece lo desarrollado durante la planeación del Sistema de Reinscripciones en la U.N.A.M. El documento sirve como un medio de comunicación entre el usuario, el analista, y los diseñadores responsables del sistema. La planeación sirve como base para el desarrollo y su mantenimiento posterior. En caso de discrepancia entre el presente documento y los resultantes del desarrollo del sistema, las diferencias deberán aclararse dando preferencia a los documentos precedentes.

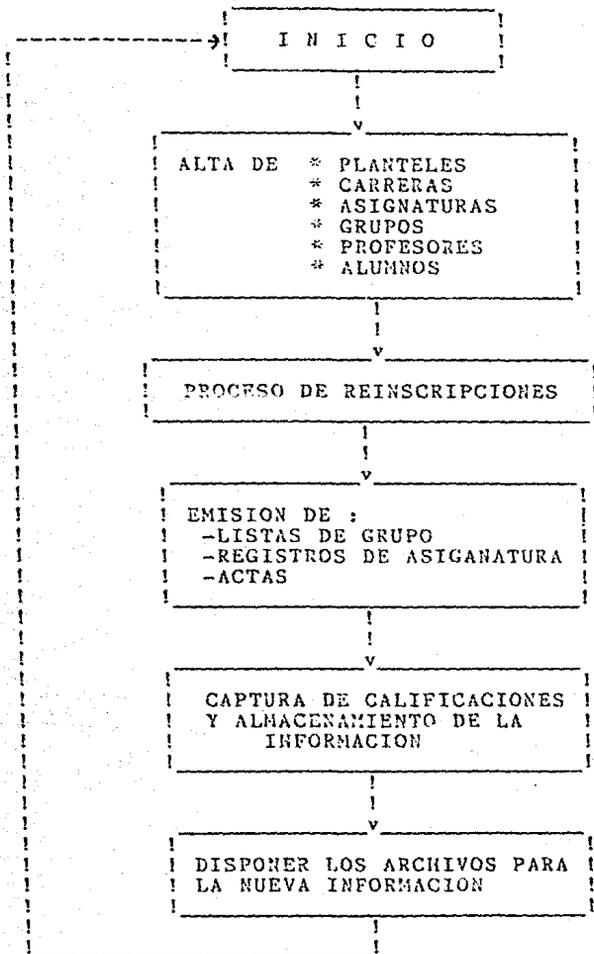
1.1 Definición del sistema

El sistema de reinscripciones tiene por objetivo el llevar a cabo la incorporación del alumno, dentro del sistema computarizado, a una asignatura-grupo solicitada por el mismo alumno. El sistema debe estar constituido de tal manera que permita su fácil manejo, control y mantenimiento.

1.2 Descripción general (fig. III.1).

El proceso de la reinscripción es una etapa dentro del ciclo que abarca todo un periodo escolar. Antes de proceder a su ejecución deben de haber sido ejecutados los procesos en que se dan de alta al sistema la información correspondiente a : Planteles, Carreras, Asignaturas, Grupos, Profesores y Alumnos. Una vez habiendo concluido estos procesos se procede a la reinscripción, donde se relacionan a los alumnos con las asignaturas y grupos solicitadas. Después es posible emitir varios tipos de reportes como "Listas de Grupo", o "Actas para examen", "Reportes de Asignaturas" por alumno, etc.

Al final del periodo escolar se lleva a cabo el proceso de calificar. Concluido éste, se guarda la información con los historiales académicos en la computadora mediante archivos magnéticos, y copiándolos en cintas magnéticas ; luego se quitan las relaciones del alumno con sus asignaturas para reiniciar un nuevo ciclo.



Ubicación del proceso de reinscripciones dentro del Sistema de Control Escolar en un periodo escolar
Figura III.1

2.0 Descripción de la Información.

2.1 Diagrama de flujo de datos.

En una primera instancia, se procede a realizar un diagrama de datos muy generalizado, con el fin de ir profundizando cada vez más en él. Un primer diagrama de flujo de datos se ilustra en la figura III.2.

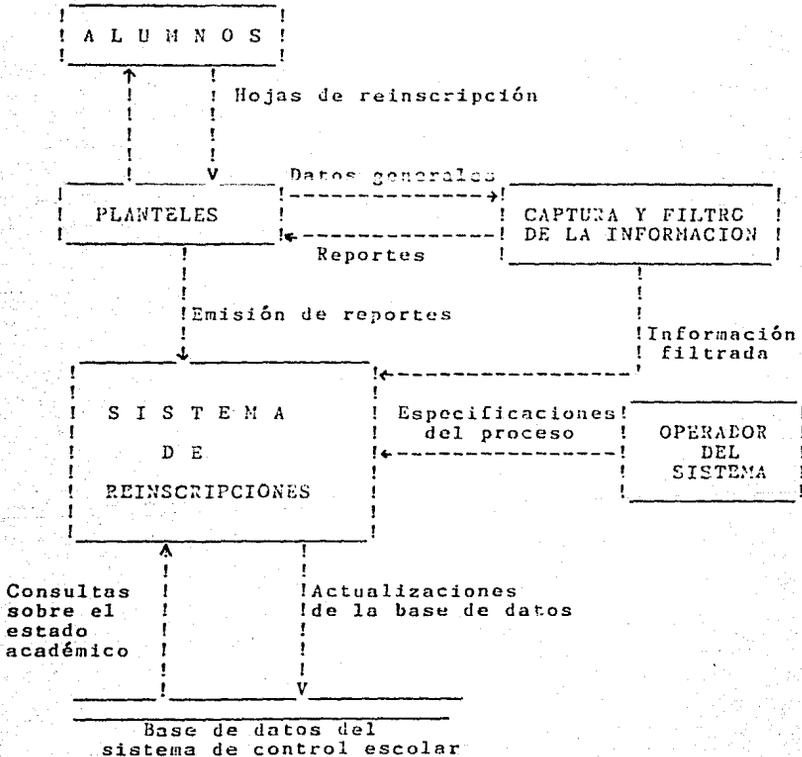
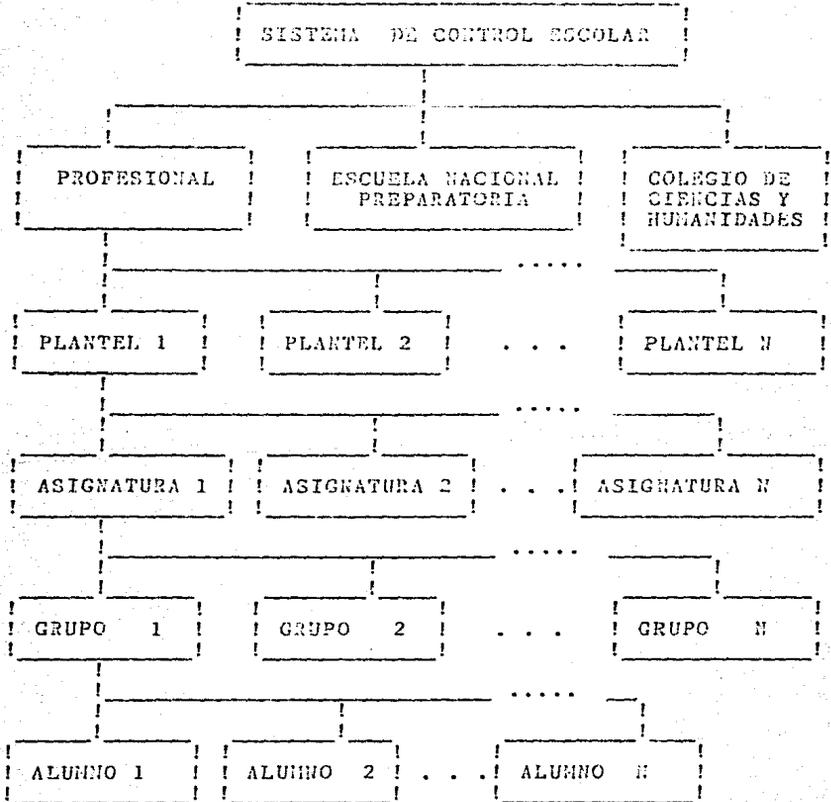


Diagrama del flujo de datos
Figura III.2

2.2 Representación de la estructura de datos.

Teniendo en cuenta que la misma estructura de datos rige para los tres planes de escolaridad, este es, Colegio de Ciencias y Humanidades, Escuela Nacional Preparatoria y Profesional, en la figura III.3 se representa de manera jerarquizada la información necesaria en este sistema. En una primera instancia se tienen los tres niveles académicos. Cada uno tiene varios planteles y en cada plantel se imparten una serie de asignaturas, donde cada una de ellas, a su vez, tiene varios grupos.



Estructura de datos del Sistema de Control Escolar
Fig. III.3

2.3 Descripción de las interfaces externas

a) Recepción de solicitudes.

Para lo cual es necesario la implantación de un sistema de captura de información, que se desarrolle durante la etapa del diseño, en donde se prevea conjuntar un formato adecuado con la información necesaria.

b) Reportes con el diagnóstico del proceso.

Un requerimiento necesario es el reporte de cada uno de los movimientos solicitados al sistema, así como de los datos estadísticos del mismo. Es necesario, por tanto, tener explícitos todos los mensajes que pudiera diagnosticar el reporte, estén los movimientos con error o sin él.

c) Procedimiento de ejecución

Esta es la interface con el usuario que pone en ejecución el inicio del proceso. Todos los pasos necesarios para su utilización deben estar descritos en el "Manual del Usuario", así como las instrucciones a seguir en caso de surgir un problema.

2.4 Descripción de las interfaces internas

El Sistema Escolar computarizado, para su uso en el sistema de reinscripciones, lo constituyen esencialmente cinco archivos:

- * Archivo de Planteles
- * Archivo de Asignaturas
- * Archivo de Grupos
- * Archivo de Profesores
- * Archivo de Alumnos

Aparte existen tres archivos internos que contienen información útil solamente para el manejo de la información, y mediante los cuales se puede tener acceso a la información de los archivos anteriores. En ellos está la estructura para la base de datos del sistema, y son:

- * Archivo de Dispersión de número de cuenta.
- * Archivo de Dispersión de Asignatura-Grupo.
- * Archivo con apuntadores a todos los archivos, denominado "Archivo de Celdas".

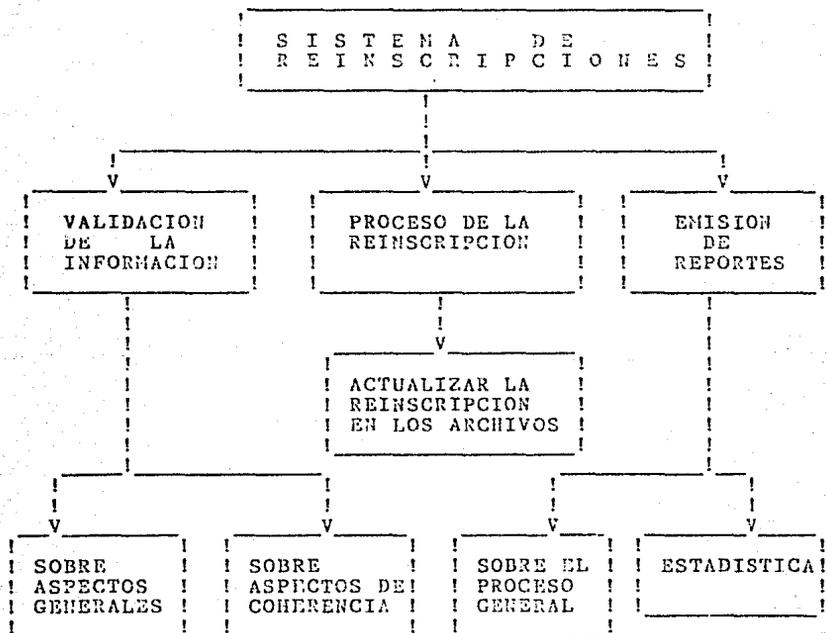
Dentro de los requerimientos están el conservar la estructura, así como modificar al mínimo los archivos principales. Es por ello que para llevar a cabo las reinscripciones es menester hacer uso principalmente del archivo de "celdas", mediante el cual quede ligado el alumno con la información restante.

3.0 Descripción de las funciones

3.1 Funciones.

El sistema de reinscripciones realiza básicamente tres funciones (fig. III.4):

- a) Revisión de la información (filtrado):
 - a.1) Sobre la información en sí misma.
 - a.2) Sobre la coherencia de la información.
- b) Actualización de los archivos computarizados con la nueva información.
- c) Emisión de reportes con el diagnóstico del proceso.



Funciones principales del sistema de Reinscripciones
Fig. III.4

3.2 Narración del proceso.

a) Recepción de la información

Las solicitudes, con la información de entrada al sistema, es recopilada por los planteles. Una de las formas de recabar esta información es mediante la utilización de hojas especiales para lectura óptica. Estas son escritas a lápiz por los mismos alumnos solicitando las asignaturas y grupos deseados. Una vez llenadas estas hojas, el plantel las envía a la Subdirección Técnica de la CAE, donde se procede a capturarlas en archivos magnéticos dentro de la computadora; no necesariamente debe de hacerse en dichas hojas, pues cabe recopilar la información en otro medio, como puede ser hojas formateadas o en cinta magnética.

b) Revisión de la información (filtrado)

Una vez recibida la información, el sistema lee la información y procede a verificar cada dato de ella. Como se observó, hay dos etapas de filtrado. La primera se limita a verificar que la información esté correcta de suyo, por ejemplo, revisa que si hay información numérica, ésta no venga de manera alfabética, etc. Y la segunda etapa, más compleja, se revisa la coherencia de la información, por ejemplo, busca que si un alumno se quiere inscribir a un determinado grupo, corresponda a la asignatura correspondiente, y que no la haya cursado en ordinario en más de dos ocasiones.

c) Actualización de los archivos computarizados.

En base al sistema total que contiene la información respectiva al estado académico de la Universidad, el "subsistema" de reinscripciones actualiza los nuevos movimientos debidos a la reinscripción. Se utiliza toda la estructura establecida en el manejo de la información almacenada.

d) Emisión de reportes con los resultados del proceso.

Después de tener un diálogo entre el departamento de sistemas y los diversos planteles, se desarrollan una serie de reportes en base a las necesidades señaladas; primero, de una manera general: número de reportes, contenido principal, etc; y en entrevistas posteriores se especifica sobre el formato detallado de salida más conveniente y sobre otros añadidos complementarios.

3.3 Restricciones del diseño.

a) Factibilidad

Su factibilidad económica esta justificada en base al gran servicio que representa para los planteles contar con este medio de control para su administración escolar. A su vez, la factibilidad técnica se limita a utilizar los recursos dispuestos por la Dirección de Cómputo para la Administración Académica para su utilización en la Coordinación de la Administración Escolar.

Respecto a la factibilidad legal el sistema se desarrolla en base a las disposiciones contempladas en el Reglamento General de Inscripciones de la UNAM.

b) Alcance del Software

Durante el desarrollo del software no se requiere de la computadora sino hasta la etapa de la codificación y posteriormente durante las pruebas.

El software desarrollado es capaz de realizar sus tareas tanto de manera interactiva como en procesos "por colas" (batch), aunque debido al volumen de información es recomendable esta última. Se requiere, como todo sistema computarizado, de tiempo de procesador y de recursos de entrada y salida. El proceso puede ser ejecutado a partir de una terminal conectada a la computadora.

Por otra parte, respecto al lenguaje de computadora para utilizarse durante la codificación, el más conveniente resulta ser el COBOL. Esto es debido a que siendo este "subsistema" una parte de un todo que está programado en COBOL, y es conveniente seguir utilizando el mismo, no tanto por costumbre, sino en vistas a un mejor acoplamiento y modularidad respecto a las diferentes rutinas comunes a todo el sistema; por ejemplo, la descripción en la especificación de los archivos, o rutinas de lectura y/o escritura en los mismos archivos.

Esta estandarización facilita la integración y una comprensión rápida del sistema, sin tener que gastar esfuerzo extra en tratar de explicar el funcionamiento de las nuevas rutinas.

c) Recursos

c.1) Recursos humanos.

Para llevar a cabo este sistema es necesario realizar diferentes funciones correspondientes a varias personas:

- * Director responsable de llevar a cabo el sistema, coordinando la ejecución de las diferentes fases. En este caso determinado, fue el Jefe de la "Unidad de Registro y Control Escolar" (URCE). Las demás personas que integran dicha unidad coadyuvan de una manera indirecta a la realización de cualquier proyecto al prestar su asesoramiento.
- * Analista que desarrolle las etapas de planeación y diseño.
- * Programador que lleve a cabo las fases de codificación y de pruebas. Tanto esta función como la anterior se asignó a un analista de sistemas de la URCE.
- * Persona encargada de ejecutar el proceso, ya durante la implementación, en base a un manual de procedimientos. Para esta tarea se asigna a una persona del "Departamento de Producción" de la misma URCE.

c.2) Recursos de Hardware

Estos recursos consisten en todo el equipo de cómputo físicamente hablando del que se dispone. A grandes rasgos es el siguiente:

- * Equipo Burroughs 6300.
- * Ocho unidades dobles de disco removible o "pack", y doce unidades sencillas. Cada uno de estos discos tiene capacidad de más de 8,000 Megabytes.
- * Seis unidades de lectura/escritura de cinta magnética.
- * Doce terminales conectadas al equipo B-6800. Cuatro de las cuales son microcomputadoras B-20, que pueden trabajar independientemente.
- * Dos lectoras ópticas de marcas reflexivas.
- * Dos minicomputadoras HP-1000 asociadas a las lectoras de marcas reflexivas.
- * Servicio de codificación, captura y microfilmación.
- * Cuatro impresoras.

c.3) Recursos de Software.

El equipo B-6800 cuenta con una gran variedad de infraestructura en software. Entre los recursos más utilizados están:

- * El sistema operativo del equipo, que permite una versatilidad en el tipo de ejecución para los procesos.
- * Un editor de comandos (CANDE) que facilita la codificación y las pruebas en línea directa con la computadora.
- * Los compiladores, que en este caso se utiliza el de COBOL, que añaden a esta función la de ofrecer un "cruce de variables" (Cobol Cross Reference).
- * El software estandarizado del sistema total, que permite utilizar partes o rutinas de él sin afectarlo. En este caso se utilizaron las rutinas de lectura y escritura a los archivos principales, así como la descripción de los archivos más comunes.
- * WFL: Lenguaje de flujos de trabajo (Work Flow Language): que permite crear toda una programación en el control y verificación de los procesos, facilitando la concatenación de muy diversas tareas.

d) Restricciones del sistema.

Los requerimientos más importantes que se contemplan dentro del sistema se encuentran los siguientes:

- d.1) Es requisito que el sistema pueda reinscribir en cualquiera de los tres planes de escolaridad: Profesional, Escuela Nacional Preparatoria y Colegio de Ciencias y Humanidades.
- d.2) La reinscripción debe llevarse a cabo en cualquier momento dentro del periodo escolar sin interferir con otros procesos del sistema de Control Escolar (procesos tales como modificaciones a la misma reinscripción, actualización de las historias académicas, etc).
- d.3) Doble Inscripción. Llevar a cabo lo estipulado por el artículo 27 del Reglamento General de Inscripciones de la U.N.A.M.: "Ningún alumno podrá ser inscrito más de dos veces en una misma asignatura. En caso de no acreditarla, sólo podrá hacerlo en examen extraordinario, de acuerdo con los

dispuesto en el capítulo III del Reglamento General de Exámenes".

- d.4) El sistema debe de contemplar el poder reinscribir tanto en grupos de ordinario como de extraordinario.
- d.5) Es posible poder cursar asignaturas de un plantel diferente al asignado, pero es necesario tener controlado a que plantel corresponde.
- d.6) Para el caso del plantel de Medicina, debe aceptar el poder reinscribirse varias veces a una misma asignatura en un mismo período, siempre y cuando haya aprobado su asignatura antecedente.

4.0 Criterio de Validación

4.1 Clases de Pruebas

El sistema debe de cumplir con los requerimientos especificados, y para ello se le somete a diversas pruebas.

Se comienza por probar cada módulo, tanto respecto al código como al cumplimiento de su función específica. Una vez alcanzada la confiabilidad de cada módulo, se procede a integrarlos para probarlos conjuntamente. Se llega así hasta la prueba más importante, correspondiente a la integración total, que consiste en simular el proceso, integrando el sistema de reinscripciones al sistema total de Control Escolar. Para ello se dispone de todo el entorno -computadora y archivos- y se ejecuta un proceso "especial": se construyen unos datos de entrada con casos típicos de errores o con situaciones complejas de tal manera que queden cubiertas las posibles infiltraciones de error. Cada uno de los posibles errores debe quedar contemplado y probado.

Una vez ejecutado el proceso simulador, se procede a una revisión total del Sistema de Control Escolar a fin de comprobar la los movimientos realizados.

4.2 Consideraciones especiales

Sin embargo, las pruebas se enfrentan a dos dificultades principales:

a) Respecto al tiempo para las pruebas:

El Sistema de Control Escolar, a lo largo de todo el período escolar, se maneja mediante tareas discontinuas, es decir, hay intervalos de tiempo entre ellas. Sin

embargo, en un mismo día hay múltiples tareas a procesar, tareas que realizan actualizaciones a los diversos archivos. Por tanto, no es posible "parar" o suspender los procesos para dedicarle al "nuevo" subsistema un período de pruebas, pues el calendario escolar ocupa la mayor parte del tiempo del año. El período de vacaciones resulta ideal para ejecutar los procesos de prueba.

b) Respecto al volumen de información.

Se podría pensar que para probar el sistema se copiara el sistema y sobre él se hicieran las pruebas. Sin embargo, es tan grande la cantidad de información que resulta en ocasiones imposible físicamente realizar la operación de copiado. Es más conveniente simular el sistema mediante archivos pequeños que contengan las mismas características.

c) Respecto a la seguridad y confiabilidad.

Existe un grave peligro en probar con los archivos vigentes del Sistema de Control Escolar, pues la información que se maneja es muy delicada e importante como para exponerla a ser perdida. En caso de perder información resulta muy costoso recuperarla y en ocasiones imposible. Es necesario contar con un máximo de seguridad para no afectar la información que se maneja.

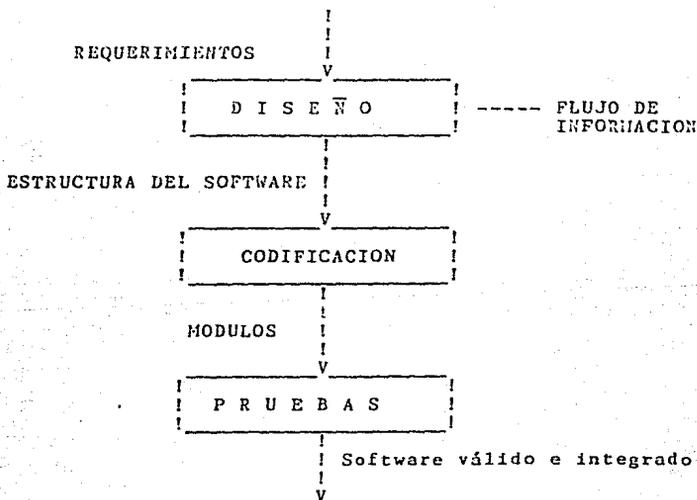
5.0 Bibliografía

- * Series B7000/B6000
CANDE User's Manual
1977 Burroughs Corporation, Detroit, USA.
- * The B6700 WFL Primer
Donald J. Gregory
Gregory Publishing Company
Sonoma, California 1979, USA.
- * Series B7000/B6000
I/O Subsystem
Reference Manual
1977 Burroughs Corporation, Detroit, USA.
- * Ansi-74 Reference Manual
Relevant to Mark 3.2 release
1980, Burroughs Corporation, Detroit, Michigan
U.S.A.

C A P I T U L O I V

DESARROLLO DEL SISTEMA DE REINSCRIPCIONES

El desarrollo de un sistema constituye la segunda fase en el proceso de su creación y consta de tres etapas: diseño, codificación y pruebas (fig. IV.1). A diferencia de la planeación del sistema, ahora se detallan las diversas características tomando como fundamento lo señalado en la fase anterior.



Etapas en la fase del desarrollo
Figura IV.1

IV.1 DISEÑO DEL SISTEMA DE REINSCRIPCIONES

La etapa del diseño queda completamente detallada en base al formato del documento "Especificación del diseño" cuyo índice se presenta en la sección II.2.1 y que se desarrolla a continuación.

ESPECIFICACION DEL DISEÑO PARA EL SISTEMA DE
REINSCRIPCIONES EN LA UNAM

1.0 Alcance del diseño

- 1.1 Objetivos y papel del software como un elemento del sistema
- 1.2 Interfaces
- 1.3 Principales funciones del software
- 1.4 Definición de la base de datos externa
- 1.5 Limitaciones y restricciones

2.0 Descripción del entorno

- 2.1 Descripción de los archivos externos
- 2.2 Diagrama de la estructura de datos

3.0 Descripción del diseño

- 3.1 Descripción del flujo de datos
- 3.2 Diagrama de la estructura del proceso

4.0 Descripción de los módulos

- 4.1 Módulo: CONTROL-DE-REINSCRIPCIONES
- 4.2 Módulo: PROCESO-DE-REINSCRIPCIONES
- 4.3 Módulo: FILTRO-REGISTRO-MAESTRO
- 4.4 Módulo: VERIFICA-REGISTRO-DETALLE
- 4.5 Módulo: REVISION-CON-REINSCRITAS
- 4.6 Módulo: REvisa-DOBLE-REINSCRIPCION
- 4.7 Módulo: REINSCRIBE-ASIGNATURAS
- 4.8 Módulo: REINSCRIBE-A-OTRO-PLANTEL

5.0 Interfase externa

- 5.1 Diagrama de la interfase con la base de datos
- 5.2 Mensajes
- 5.3 Paquetería

6.0 Equipo y Lenguaje

7.0 Recuperación de fallas

8.0 Certificación de calidad

9.0 Apéndices

- Apéndice A: Estructura física de los archivos
- Apéndice B: Diferentes tipos de "celdas"

1.0 Alcance del diseño

1.1 Objetivos y papel del software como un elemento del sistema

El presente documento intenta ser una continuación del documento "Especificación de los requerimientos de software en el Sistema de Reinscripciones de la U.N.A.M.", por tanto, los objetivos del sistema son los mismos: llevar a cabo la incorporación del alumno, dentro del sistema computarizado, a las asignatura-grupo que haya solicitado el mismo alumno.

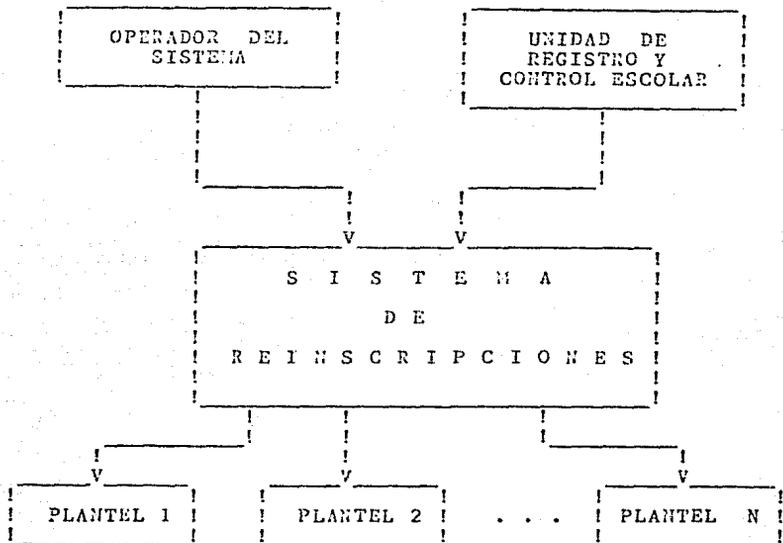
El sistema desarrollado en software constituye sólo un elemento más dentro del proceso completo de reinscripciones:

- Habiéndose iniciado el proceso con la repartición de hojas de reinscripciones a los alumnos, se recogen una vez llenadas.
- Se procede después a su almacenamiento en archivos computarizados, ayudándose de la lectora óptica.
- Ya almacenada la información, se hace uso del Sistema de Reinscripciones, el cual mediante su ejecución, realiza las diversas asignaciones, y a su vez, emite reportes sobre el estado del proceso que son entregados a los planteles correspondientes.

1.2 Interfaces

a) Interface humana (fig. IV.2):

- Interface remota: son los alumnos y el personal administrativo encargado de la obtención y entrega de solicitudes, que una vez procesada reciben los diagnósticos con los resultados del proceso.
- Interface próxima: es el operador responsable de la ejecución del proceso.
- El personal responsabilizado de darle mantenimiento al sistema (Unidad de Registro y Control Escolar).

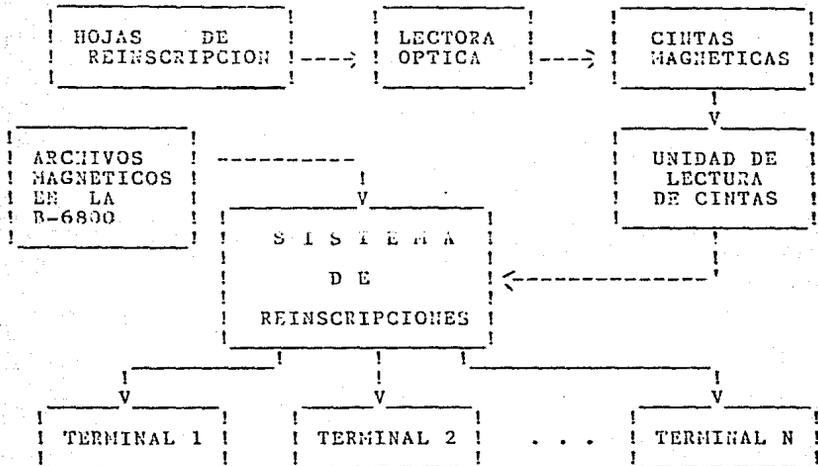


Interfaces humanas con el Sistema de Reinscripciones
 Figura IV.2

b) Interfaces con el hardware (fig. IV.3)

- Lectora óptica donde se lee la información proveniente de los planteles para quedar guardada en archivos magnéticos en la computadora B-6800. Las mismas hojas especiales para lectura óptica pueden considerarse como un elemento más en este ámbito.
- Unidades de lectura/escritura para cintas magnéticas. En caso de haber sido el medio de captura para las solicitudes las cintas magnéticas, se procede a su lectura para conservar la información en archivos magnéticos.
- Impresoras. Para emitir los diversos diagnósticos se precisa de una impresora y de papel "stock" blanco con el logotipo de la U.N.A.M.

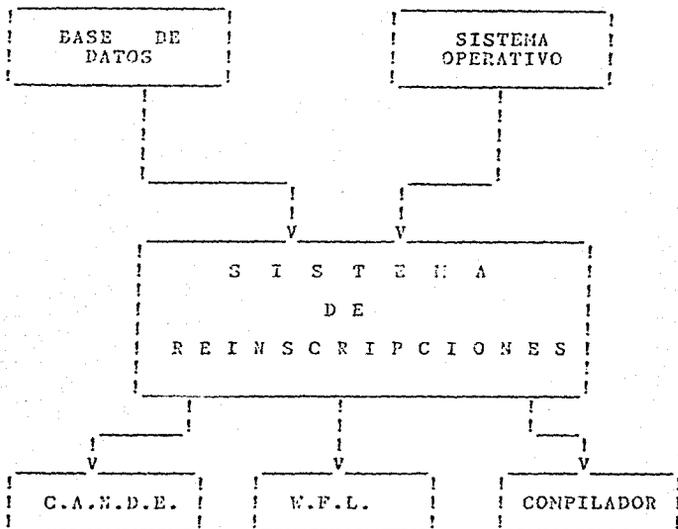
- Terminales de Video conectadas a la B-6800. Utilizada principalmente para ejecutar desde ella el proceso. También se puede monitorear el desarrollo del proceso desde ella.



Interfaces del Sistema con el Hardware
Figura IV.3

c) Interface con el Software (fig. IV.4)

- Base de datos. No sólo está íntimamente relacionado con la base de datos de la CAE, sino que viene a formar parte de ella.
- Sistema operativo. Como todo sistema computarizado, su ejecución está controlado por un sistema operativo que regula sus recursos y necesidades.
- El compilador: para interpretar el lenguaje y dejar preparados los programas en un lenguaje interpretable por el procesador específico de la computadora.
- Software de utilidad: la computadora cuenta con herramientas en software que sirven para manejar los procesos y programas, entre otros, el CANDE (Command AND Edit) y el WFL (Work Flow Lenguaje).



Interface del Sistema con el Software
Figura IV.4

1.3 Principales funciones del software

a) Revisión de la información (filtrado).

La información debe llegar a través de un formato específica en el cual se contengan los datos necesarios para la reinscripción. A cada dato se le aplicará, según el caso, una revisión apropiada: entre los principales campos que se capturan están los correspondientes al número de cuenta, la clave del plantel y de la(s) asignatura(s) pedidos.

En un segundo momento, la información que haya pasado el primer filtro, se somete a otra revisión pero ahora respecto a la situación académica del alumno. Por ejemplo, se revisa si ya está dada de alta la asignatura a la que se desea reinscribir, y en caso afirmativo, se observa si lo está en ordinario o en extraordinario.

b) Actualización de los archivos computarizados.

Esta función es la principal del proceso y en ella se lleva a cabo propiamente la reinscripción del alumno.

Una vez que la información ha pasado los diferentes filtros, se procede a "ligarla" al alumno, grabando en los diversos archivos los datos actualizados que son afectados al tener una reinscripción nueva.

c) Emisión de reportes.

La tercera función del software consiste en la emisión de una serie de reportes sobre los resultados del proceso. Como se señaló durante la planeación, su número y formato es acordado entre los planteles y la URCE, habiéndose establecido tres tipos de reportes: un reporte general en el que se señala el diagnóstico sobre cada uno de los movimientos efectuados (con o sin error), los reportes estadísticos que señalan las cifras totales, y tercero, un reporte estadístico sobre los errores ocurridos en el proceso.

1.4 Definición de la base de datos externa

El proceso de reinscripciones, al ser una parte del sistema total del control escolar, requiere un conocimiento detallado de la base de datos con que se cuenta. El diseño deberá ser completamente compatible con esta base de datos, y su descripción se especifica en la sección 2.0 del presente reporte.

1.5 Limitaciones y restricciones

Puesto que el proceso de reinscripciones no sólo consulta a los archivos de la base de datos, sino que los actualiza, esto conlleva un incremento considerable de registros, pudiéndose presentar una saturación del espacio físico en la computadora. Caben, por tanto, tomar medidas preventivas como sería que el proceso mandara unos mensajes de alarma al respecto, para tomar las precauciones convenientes.

El diseño del programa debe considerar en su aspecto lógico un margen de tolerancia respecto al límite de las variables y arreglos que se utilizan, para que su mantenimiento sea mínimo.

El sistema no está restringido a una computadora específica, pues al utilizar el lenguaje COBOL estandard, es factible su procesamiento en aquellas que contengan el compilador respectivo y la capacidad adecuada. Sin embargo, sí está limitado a utilizarse únicamente en la Coordinación de la Administración Escolar, tanto por lo específico del problema como por medidas de seguridad.

2.0 Descripción del entorno

2.1 Descripción de los archivos externos

a) Estructura del archivo: Plantel

- * Contenido: - Clave del Plantel
- Nombre del Plantel
- Información estadística:
 - + Alumnos reinscritos
 - + Total de alumnos
 - + Total de asignaturas
 - + Total de grupos

* Descripción física : Apéndice A.1

b) Estructura del archivo: Asignatura

- * Contenido: - Clave de asignatura
- Nombre de la asignatura
- Número de créditos
- Semestre en que se cursa
- Plan de estudios
- Número de grupos asignados

* Descripción física : Apéndice A.2

c) Estructura del archivo: Carrera

- * Contenido: - Clave de carrera
- Nombre de la carrera
- Plan (semestral o anual)
- Número de créditos obligatorios
- Número de créditos optativos

* Descripción física : Apéndice A.3

d) Estructura del archivo: Grupo

- * Contenido: - Clave de grupo
- Número de alumnos
- Número de alumnos reinscritos de otro plantel

* Descripción física : Apéndice A.4

e) Estructura del archivo: Alumnos

- * Contenido: - Nombre del alumno
- Número de cuenta
- Clave de plantel
- Clave de Carrera
- Número de asignaturas reinscritas
- Clave del plantel de procedencia
- Fecha de nacimiento

- Sexo
- Nacionalidad

Descripción física : Apéndice A.5

f) Estructura del archivo: Asignaturas doblemente inscritas

Contenido: - Número de cuenta
- Clave de Asignatura

Descripción física : Apéndice A.6

2.2) Diagrama de estructura de datos (fig. IV.5)

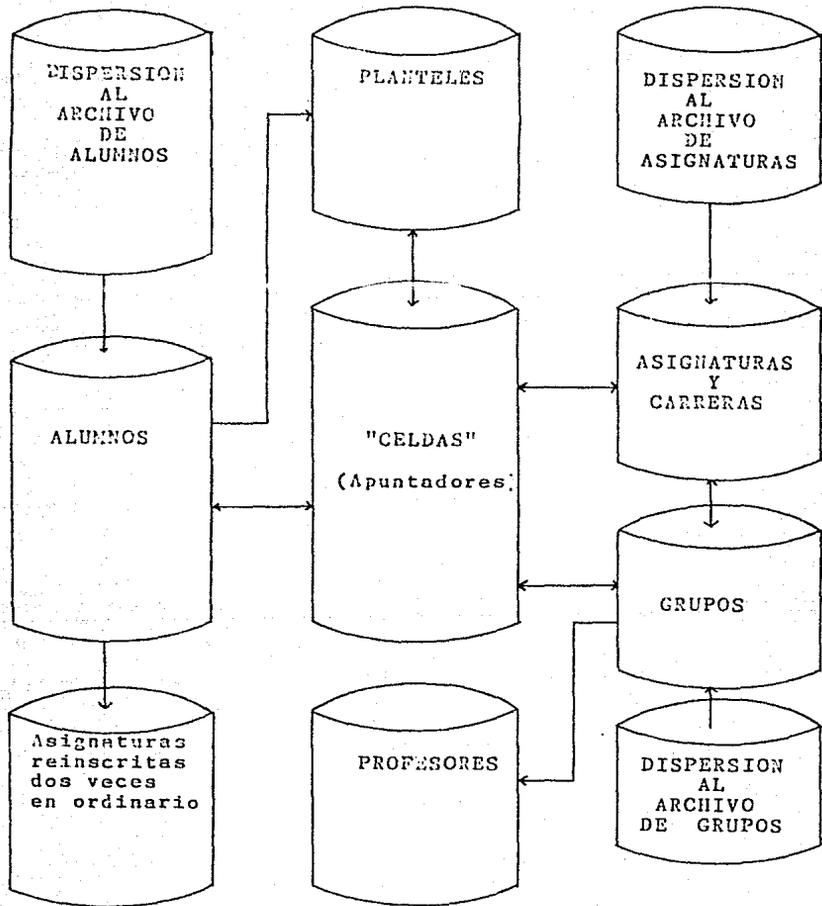
Este diagrama viene a complementar la estructura de datos señalada en el capítulo anterior (fig. III.3), pues muestra las diferentes maneras por las que se puede tener acceso a los diferentes archivos que constituyen la base de datos.

Todos los archivos pueden ser accedidos directamente ya sea de manera secuencial o aleatoria. Sin embargo, debido a la magnitud tan grande de los archivos (por ejemplo, el archivo de alumnos contiene aproximadamente 100,000 registros), se utiliza un método de dispersión para que en caso de búsqueda se encuentre la información de una manera eficiente. Son tres los archivos a los que se les puede acceder vía dispersión: alumnos, asignaturas y grupos. La información sobre las carreras está incluida en el archivo de asignaturas y se maneja de manera similar a éstas.

Cabe hacer mención sobre un archivo auxiliar, denominado "Archivo de Celdas", que sirve de enlace entre todos los demás archivos. Este archivo contiene diversos tipos de registros, no todos son iguales. No obstante, todos los registros tienen la misma estructura: un atributo y tres apuntadores. Los apuntadores podrán señalar a los diversos archivos, o incluso a otro registro del mismo "Archivo de Celdas", dependiendo del tipo de registro que se trate. En el proceso de reinscripciones intervienen tres tipos registros denominados:

- Registro tipo: "Cursando-Alumno"
- Registro tipo: "Grupo-Alumno"
- Registro tipo: "Otro-Plantel-Alumno"

Más adelante se explicitarán con mayor detalle estos tres registros, en la parte 4.7 y 4.8 del presente reporte. Además en el apéndice E.1 se muestran los diversos tipos de "celdas" que contiene el archivo.



Estructura del Sistema Computarizado del Sistema de Registro y Control Escolar
Figura IV.5

3.0 Descripción del diseño

3.1 Descripción del flujo de datos

El diagrama de flujo de datos (fig. III.2) presenta diversos tipos de datos que a continuación se describen:

a) Datos de Entrada.

Para esta etapa se diseñó un formato que sirviera para la obtención de los datos necesarios, el cual consiste en dos tipos de registro para cada alumno: el primero, llamado "registro-maestro" contiene los datos fijos y generales sobre el alumno; el segundo llamado "registro-detalle" especifica los datos sobre la asignatura-grupo. Sólo el segundo tipo de registro puede repetirse, dependiendo del número de reinscripciones que se soliciten.

REGISTRO MAESTRO (Reg-MI)

Información :

- Clave de identificación: "MI"
- Número de cuenta
- Nombre del alumno
- Plantel del alumno
- Nivel de estudios
- Nombre del alumno

REGISTRO DETALLE (Reg-DI)

Información :

- Clave de identificación: "DI"
- Número de cuenta
- Clave plantel al cual se reinscribe
- Nivel de estudios
- Clave de la carrera
- Clave de la asignatura
- Clave del grupo

Para el caso en que los planteles capturen con otro formato, se requiere simplemente de un programa que cambie dicho formato al descrito anteriormente.

b) Especificaciones para el proceso

Estos datos consisten en las generalidades del proceso y a los cuales se deben de ajustar los datos de entrada, pues de no coincidir se procede a rechazarlos. Estos datos son dos principalmente:

- Nivel académico de donde provienen, pudiendo referirse únicamente a tres: Profesional, Escuela Nacional Preparatoria o Colegio de Ciencias y Humanidades.

- El plantel al cual se desean reinscribir.

c) Consultas sobre el estado académico del alumno.

Consiste en la lectura de información necesaria para la verificación con los datos de entrada y es:

- Las asignatura-grupo en que está reinscrito el alumno.
- Las asignaturas que el alumno ya cursó por ordinario en dos o más ocasiones.
- El nombre del alumno.
- El nombre de la asignatura solicitada.

d) Datos para actualizar.

Corresponde a la información nueva que va a ser grabada en los archivos, y es de dos tipos:

- Contadores que son actualizados dependiendo del número de movimientos procesados.
- Nuevos registros que son añadidos a los archivos debido a las nuevas asignaturas reinscritas.

e) Emisión de reportes.

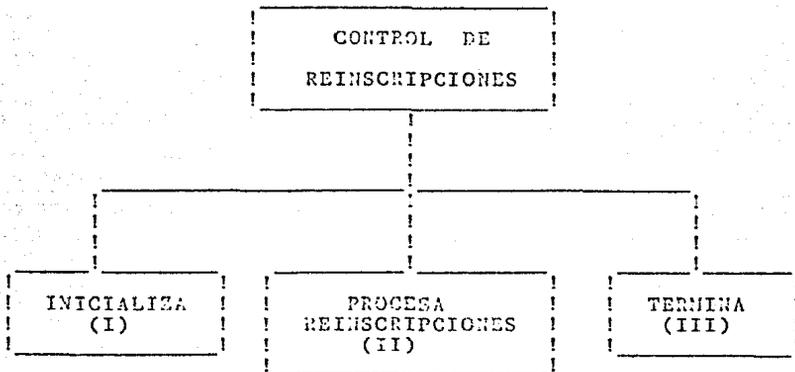
- Reporte general. Debe contener para cada movimiento los siguientes datos:

- a) Clave de plantel
- b) Clave de la carrera
- c) Clave de la asignatura solicitada
- d) Clave del grupo solicitado
- e) Nombre del plantel
- f) Nombre de la asignatura
- g) Nombre del alumno
- h) Número de cuenta del alumno
- i) Observaciones sobre el movimiento

- Reporte sobre el resumen estadístico del proceso. Contiene las cifras respecto al número total de movimientos efectuados precisando el número de alumnos que provienen de otro plantel.
- Reporte estadístico que contiene el número de veces que aconteció cada tipo de error.

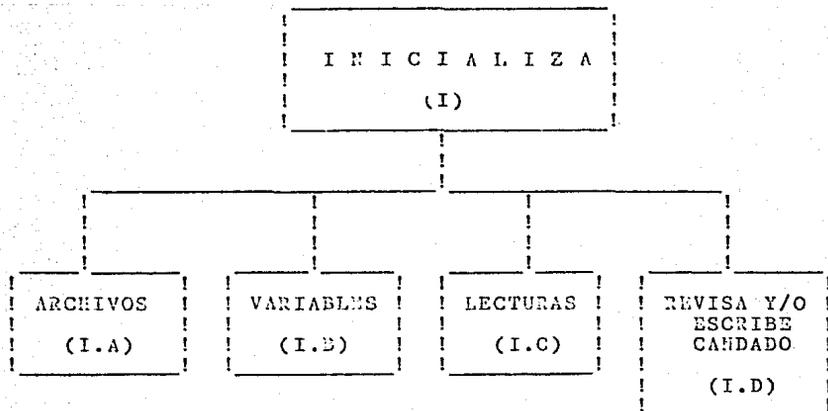
3.2 Diagrama de la estructura del proceso

El diagrama general consta de un bloque: CONTROL-DE-REINSCRIPCIONES. Este divide sus tareas en tres fases muy sencillas:

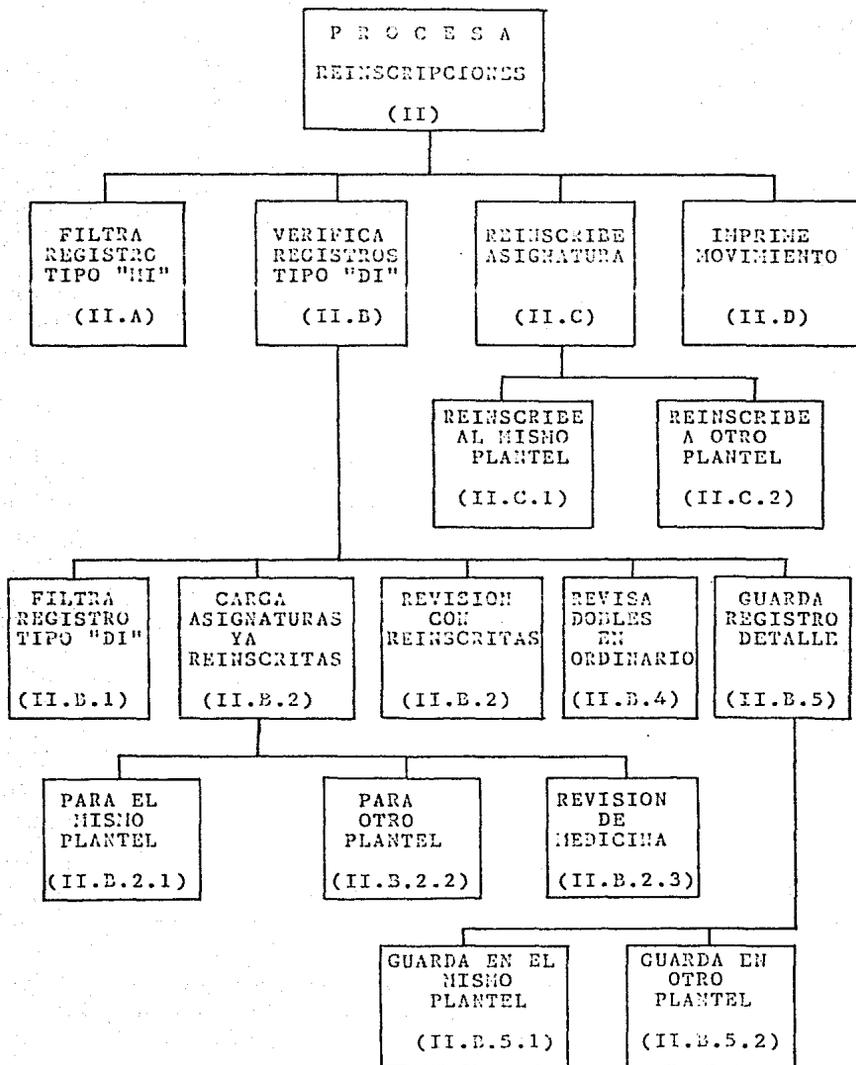


Cada una de las fases contiene además diversas funciones que controla. Detallando cada una se tiene:

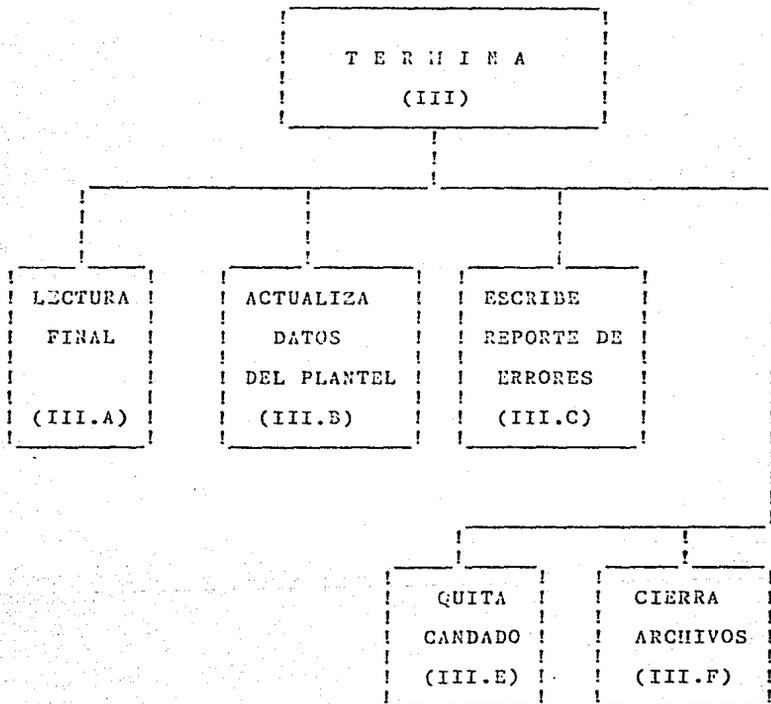
I) Primera fase: INICIALIZA



II) Segunda fase: PROCESO-DE-REINSCRIPCIONES



III) Tercera fase: TERMINA



4.0 Descripción de los módulos

A continuación se presentan el detalle de los módulos que se consideraron más significativos. Para indicar el detalle de los módulos se utiliza un pseudo-código con un español estructurado (sección II.4.3), a fin de facilitar su posterior codificación.

4.1 Descripción del módulo : CONTROL-DE-REINSCRIPCIONES

a) Reseña del proceso

El presente módulo es el encargado de coordinar la secuencia de las diferentes tareas. Está en el más alto nivel de jerarquía y para facilitar su trabajo clasifica sus tareas en tres fases:

- I) Fase de inicio, en la cual se manda inicializar variables, nombres de archivos a utilizar, banderas, y se hace la verificación del correcto estado de los archivos.
- II) Fase de la reinscripción, en la cual se manda ejecutar de manera iterativa la reinscripción de los alumnos, y termina hasta que se acaban los datos de entrada.
- III) Fase de terminación, en donde una vez procesados los diversos movimientos, se procede a ejecutar las actualizaciones finales con las cantidades totales y se cierran los archivos, poniéndolos a disposición para otros procesos.

b) Descripción de sus interfaces.

Recibe dos parámetros:

- 1o) El nivel académico de las reinscripciones que se desean procesar. Sólo puede ser de tres tipos:
 - "ENP" : Escuela Nacional Preparatoria
 - "CCH" : Colegio de Ciencias y Humanidades
 - "PRO" : Profesional
- 2o) La clave del plantel al cual se requiere reinscribir. Dicha clave consta de tres caracteres, identificando fácilmente el plantel preciso.

4.2 Descripción del módulo II: PROCESO-DE-REINSCRIPCIONES

a) Reseña del proceso

Cada vez que se ejecute esta rutina, será porque los datos indican que hay un nuevo alumno por reinscribirse. La rutina la constituyen cuatro tareas principales:

- Se comienza por inicializar variables generales, y con la información del "registro-maestro" se manda efectuar una primera revisión a la rutina FILTRO-DE-REGISTRO-MAESTRO (II.A).
- En caso de no presentar errores se procede a revisar todos los "registro-detalle" correspondientes al "registro-maestro" mediante la rutina VERIFICA-REGISTRO-DETALLE (II.B).
- Si una vez revisados todos los "registro-detalle" se cuenta con al menos uno correcto, se procede a mandar darlo de alta por la rutina REINSCRIBE-ASIGNATURA (II.C).
- Por último, se hayan o no dado de alta reinscripciones, se procede a mandar imprimir toda la información del alumno mediante la rutina IMPRIME-MOVIMIENTO (II.D).
- Y una vez ejecutada la impresión, se lee del archivo de datos de entrada a otro alumno a fin volver a entrar a la presente rutina con un nuevo alumno. En caso de que se hayan terminado los datos se indica mediante una bandera para que se proceda a dar por terminado el proceso.

b) Descripción de sus interfaces.

Entrada: 1) Recibe la información del "registro-maestro".

2) Lee todos los "registro-detalle" que corresponden al "registro-maestro".

Salida: 1) Manda la información de cada movimiento para que se imprima en el diagnóstico correspondiente.

2) Envía la indicación sobre la terminación de los datos, marcando una bandera.

c) Detalle del módulo.

```
!
!
! Empieza: PROCESO-DE-REINSCRIPCIONES
!
! Inicializa variables
!
! SI el dato es de un registro-maestro ENTONCES
!   EJECUTA FILTRO-REGISTRO-MAESTRO
!   SI estuvo correcta el registro ENTONCES
!     IMPRIME el registro maestro
!     EJECUTA LECTURA-REGISTRO-DETALLE
!     REPITE VERIFICA-CARGA-REGISTRO-DETALLE
!     HASTA que terminen Registros-Detalle
!     SI hay una o más correctas ENTONCES
!       EJECUTA REINSCRIBE-ASIGNATURAS
!     SINO
!       INDICA los errores
!       ESCRIBE los errores
!     FIN (SI)
!   SINO
!     IMPRIME el registro señalando el error
!     FIN (SI)
! SINO
!   EJECUTA eliminación de registros-detalle
!     que no contienen registro-maestro
! FIN (SI)
!
!
! SI hay datos y no se tiene registro-maestro
!   ENTONCES
!     LEE nuevo registro de entrada
!   FIN (SI)
!
!
! Termina: PROCESO-DE-REINSCRIPCIONES
!
```

Detalle del módulo II : Proceso de Reinscripción

Figura IV.7

4.3 Descripción del módulo II.A: FILTRO-REGISTRO-MAESTRO

a) Reseña del proceso

El dato más importante que proporciona el "registro-maestro" es el número de cuenta del alumno, por lo que la presente rutina está orientada a verificar su validez:

- Se revisa en primer lugar que no contenga caracteres no numéricos.
- La corrección del dígito de control.
- Revisa que siga como alumno en la universidad (que no haya sido dado de baja).
- En caso de que su plantel no corresponda al que pertenece, si no es de nivel profesional, no se acepta la reinscripción.

b) Descripción de sus interfaces.

Entrada: 1) El registro maestro del alumno
2) La información sobre el alumno que contiene la base de datos.

Salida: 1) Una bandera que se prende en caso de encontrarse un error y que sirve para cancelar toda su reinscripción.

c) Detalle del módulo.

En la figura IV.3 se muestra el detalle del presente módulo.

```

Empieza: FILTRO-REGISTRO-MAESTRO
INICIALIZA la bandera como correcta.
SI el número de cuenta es numérico ENTONCES
  CALCULA dígito de control correcto
  SI dígito de control está correcto ENTONCES
    EJECUTA Localización-del-Alumno
    SI se encontró al alumno ENTONCES
      EJECUTA Lee-Registro-Alumno
      SI es alumno activo ENTONCES
        SI plantel-alumno es igual al que
          se quiere reinscribir ENTONCES
            SI es de profesional el alumno
              ENTONCES es correcto
            SINO
              MARCAR tipo de error
              MARCAR bandera
              FIN (SI)
          SINO
            MARCAR tipo de error
            MARCAR bandera
            FIN (SI)
        SINO
          MARCAR tipo de error
          MARCAR bandera
          FIN (SI)
      SINO
        MARCAR tipo de error
        MARCAR bandera
        FIN (SI)
    SINO
      MARCAR tipo de error
      MARCAR bandera
      FIN (SI)
  SINO
    MARCAR tipo de error
    MARCAR bandera
    FIN (SI)
Termina: FILTRO-REGISTRO-MAESTRO

```

Detalle del módulo II.A: Filtro del Registro Maestro

Figura IV.8

4.4 Descripción del módulo II.B: VERIFICA-REGISTRO-DETALLE

a) Reseña del proceso

La finalidad del módulo es obtener, para cada alumno, todos los registros detalle que tenga correctos, a fin de almacenarlos para que sean procesados en otro módulo. Es así que desde este módulo se controlan diversos tipos de revisiones al registro detalle:

- * Se verifica que el registro-detalle corresponda al registro-maestro.
- * Se manda la ejecución de una primera revisión, FILTRO-REGISTRO-DETALLE (II.B.1), el cual detecta:
 - La corrección de la clave del plantel
 - La corrección y existencia de la asignatura
 - La corrección y existencia del grupo
- * Se manda ejecutar la rutina CARGA-ASIGNATURAS-REINSCRITAS (II.B.2) que almacena la información sobre las asignaturas ya reinscritas del alumno en turno en un arreglo, a fin de poder, en el siguiente módulo, comparar con mayor eficiencia que si se leyera en disco en cada de ellas.
- * Se manda la ejecución de un módulo que revisa la asignatura y el grupo solicitados con las asignaturas y grupos que ya tiene el alumno reinscritas: REVISION-CON-REINSCRITAS (II.B.3).
- * Se manda revisar que las asignaturas solicitadas no hayan sido cursadas por el alumno en ordinario en dos o más ocasiones, pues de ser así, se rechaza la reinscripción: REvisa-2-EN-ORDINARIO (II.B.4).
- * Una vez hechas todas las revisiones, en caso de cumplir satisfactoriamente los requisitos, se almacenan los datos del registro-detalle a un arreglo que tiene la información correcta de un mismo alumno.
- * Se lee otro registro para que la rutina lo vuelva a revisar en el proceso iterativo.
- * Se imprime el registro-detalle. En caso de que se encuentre algún error se señala en el reporte.

b) Descripción de sus interfaces

- Entrada: 1) El módulo lee los registro detalle a fin de revisarlos.
- 2) El registro maestro le sirve únicamente para revisar que correspondan los registros detalle con el maestro.

- Salida: 1) El módulo entrega los registro detalle correctos en un arreglo.
- 2) Manda ejecutar la impresión de todos los registros detalle que se leen.

c) Detalle del módulo.

```

Empieza: Verifica y Carga Registro-Detalle
Inicializa con cero la variable "contador-error"
SI es registro-detalle ENTONCES
  SI alumno igual al registro maestro ENTONCES
    EJECUTA FILTRO-REGISTRO-DETALLE
    SI está correcta ENTONCES
      EJECUTA CARGA-ASIGNATURAS-REINSCRITAS
      REPITE REVISION-CON-REINSCRITAS HASTA
        que se terminen las reinscritas
      EJECUTA REvisa-2-EN-ORDINARIO
      SI está correcto ENTONCES
        EJECUTA GUARDAR-REGISTRO-DETALLE
        EJECUTA IMPRESION-REGISTRO
      SINO
        MARCA error
        EJECUTA IMPRESION-REGISTRO
      FIN (SI)
    SINO
      MARCA error
      EJECUTA IMPRESION-REGISTRO
    FIN (SI)
  SINO
    MARCA error
    EJECUTA IMPRESION-REGISTRO
  FIN (SI)
SINO
  MARCA fin-de-rutina
FIN (SI)
EJECUTA Lectura-Registro
SI ya no hay datos ENTONCES
  MARCA la terminación de datos
FIN (SI)
Termina: Verifica y Carga Registro-Detalle

```

Detalle del módulo II.B: Verifica y Carga Registro-Detalle

Figura IV.9

4.5 Descripción del módulo II.B.3: REVISION-CON-REINSCRITAS

a) Reseña del proceso

Este módulo es iterativo, y se ejecutará tantas veces como asignaturas reinscritas tiene cada alumno. Se revisa que la asignatura no esté repetida, y en caso de estarlo, se verifica cómo es el grupo para aceptar o no la reinscripción. Los diferentes casos se muestran en la siguiente tabla:

GRUPO POR REINSCRIBIR		GRUPO REINSCRITO		PROCEDE
ORDINARIO	EXTRAORD.	ORDINARIO	EXTRAORD. CON SIN ACTA ACTA	ALTA
***		***		NO
***			***	SI
***			***	SI
	***	***		SI
	***		**	NO
	***		***	SI

Existe una situación especial con la Facultad de Medicina (plantel 012). En este caso se puede aceptar la reinscripción en ordinario aunque la asignatura por reinscribir ya esté reinscrita también en ordinario, siempre y cuando se haya cursado en un grupo antecedente que se tiene como requisito. El presente módulo detecta que se trata de ese caso especial y manda ejecutar otro módulo especial : REVISION-MEDICINA (II.B.3.1).

b) Descripción de sus interfaces

Entrada : 1) Recibe la información del registro detalle, esto es, la asignatura y el grupo al que se desea reinscribir.

- 2) El arreglo en que se contienen todas las asignaturas en que se está reinscrito.

Salida : 1) Señala en una bandera si hubo o no error al revisar con las asignaturas reinscritas.

- 2) Envía la información del registro detalle al módulo que revisa el caso especial de medicina.

c) Detalle del módulo.

```

!-----!
! Empieza: Revisión con Reinscritas !
! SI asign-X-reinscribir = asign-reinscrita ENTONCES !
! SI grupo-X-reinscribir es extraordinario ENTONCES !
! SI grupo ya reinscrito es extraordin. ENTONCES !
! EJECUTA lectura de grupo reinscrito !
! SI grupo reinscrito tiene acta ENTONCES !
! si procede el movimiento !
! SINO !
! MARCA error !
! FIN (SI) !
! SINO !
! continua con otra asignatura !
! FIN (SI) !
! SINO !
! SI grupo reinscrito es extraordinario ENTONCES !
! continua con otra asignatura !
! SINO !
! SI el plantel es de Medicina ENTONCES !
! EJECUTA REVISION-MEDICINA !
! SI encontró error ENTONCES !
! MARCA error !
! SINO !
! Procede el movimiento !
! FIN (SI) !
! SINO !
! MARCA error !
! FIN (SI) !
! FIN (SI) !
! FIN (SI) !
! TERMINA: REVISION-CON-REINSCRITAS !
!-----!

```

Detalle del módulo II.5.3: REVISION-CON-REINSCRITAS
Figura IV.10

4.0 Descripción del módulo: REVISA-DOBLE-REINSCRIPCION
(II.B.4)

a) Reseña del proceso.

Este módulo iterativo se ejecuta tantas veces como asignaturas reinscritas dos veces en ordinario haya tenido el alumno en su historia académica. Se comparara la asignatura por reinscribir con cada una de sus asignaturas que tiene como doblemente reinscritas en ordinario y que se leen en un archivo. En caso de encontrar una asignatura con la misma clave en el archivo leído, el movimiento no procede, tal y como se precisa en la etapa de análisis.

Para ejecutar este módulo, presupone el diseño y la creación del archivo que contiene debidamente ordenadas todas las asignaturas que están en las condiciones señaladas, así como el tipo de acceso hacia éste.

b) Descripción de sus interfaces.

Entrada: 1) Recibe la información del registro detalle para hacer referencia a la clave de la asignatura por reinscribir y compararla.

2) Efectua las lecturas correspondientes al archivo que contiene las claves de las asignaturas doblemente reinscritas en ordinario.

Salida: 1) Marca en una bandera si el movimiento procede o no.

c) Detalle del módulo

```
!-----!  
! Empieza: REVISA-DOBLE-REINSCRIPCION !  
! !  
! EFECTUA lectura de archivo !  
! SI hay error de lectura ENTONCES !  
! MARCA error !  
! SINO !  
! EFECTUA REVISION-CON-DOBLES-REINSCRITAS !  
! HASTA que se terminen las del registro !  
! leído !  
! MUEVE siguiente apuntador de registro del !  
! archivo para continuar leyendo !  
! FIN (SI) !  
! !  
! TERMINA: REVISA-DOBLE-REINSCRIPCION !  
!-----!
```

Detalle del módulo II.B.4: REVISA-DOBLE-REINSCRIPCION
Figura IV.11

4.7 Descripción del módulo II.C : REINSCRIBE-ASIGNATURAS

a) Reseña del proceso

Una vez que las asignatura-grupo que se desean reinscribir han pasado por todos los filtros y verificaciones, llegan a este módulo, el cual va a ejecutar propiamente la reinscripción. Esta consiste en relacionar mediante apuntadores la nueva asignatura con la base de datos ya establecida, así como incrementar las variables que contabilizan el número de asignaturas reinscritas en un plantel.

En el archivo de alumnos (Bloque I), cada registro corresponde a un alumno diferente. Uno de los campos de este registro contiene un apuntador al archivo de "celdas" (Bloque IV), el cual liga a todas sus asignaturas reinscritas. Este archivo de "celdas" tiene diversos tipos de registros. Todos los registros, como se señala en la fase anterior, consta de cuatro campos: un atributo y tres apuntadores. En el caso de la reinscripción de un alumno a su plantel -luego se verá cuando lo hace a otro plantel- se utilizan dos tipos de registro en el archivo de "celdas":

1) Registro tipo: "Cursando-Alumno"

CLAVE DE	APUNTADOR	APUNTADOR	APUNTADOR
ASIGNATURA	AL	AL	AL
	ARCHIVO	REGISTRO	SIGUIENTE
	DE	TIPO	REGISTRO
	ASIGNATURAS	GRUPO-ALUMNO	

2) Registro tipo: "Grupo-Alumno"

CLAVE DE	APUNTADOR	APUNTADOR	APUNTADOR
GRUPO	AL	AL	AL
	ARCHIVO	ARCHIVO	SIGUIENTE
	DE	DE	REGISTRO
	ALUMNOS	GRUPOS	

Y el alumno tendrá tantos registro tipo "Cursando-Alumno" como asignaturas reinscritas, cada una de ellas acompañada de otra celda del tipo "Grupo-Alumno". Al momento de la reinscripción se asignan, por tanto, tantos pares de registros como nuevas asignaturas haya de reinscribirse, haciendo las modificaciones pertinentes a los apuntadores a fin de que queden convenientemente "religados".

Para mostrar el proceso de la reinscripción se esquematiza el ejemplo de un alumno que está reinscrito en dos asignaturas y desea reinscribirse a otra. El estado inicial del alumno es como sigue:

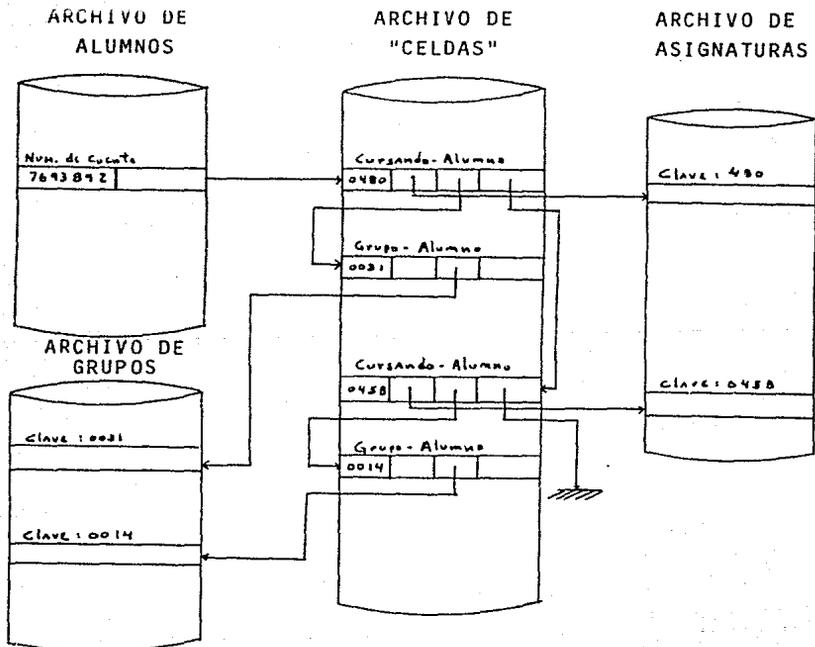
Alumno : José Martínez Colín

Número de cuenta: 7693892-6

Plantel: 011 (Facultad de Ingeniería)

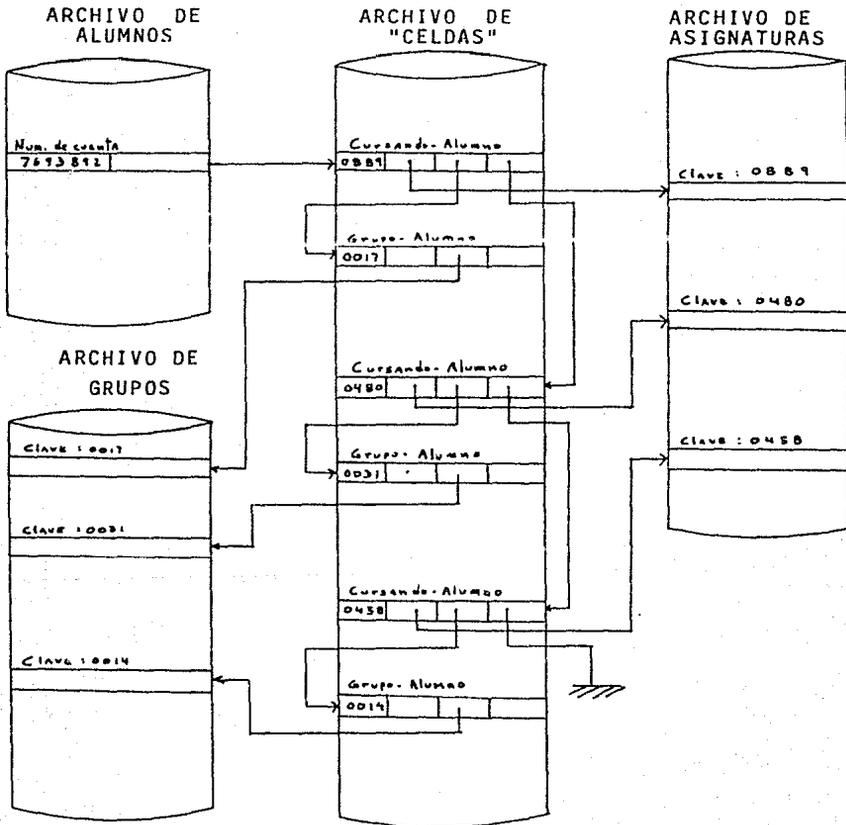
Asignaturas reinscritas :

- 1) Clave asignatura: 0480. (Métodos Numéricos)
Clave de Grupo: 0031
- 2) Clave asignatura: 0458. (Mecánica I)
Clave de Grupo: 0014



Estado original del alumno "antes" de reinscribirse
Figura IV.12

Una vez que se ha efectuado la reinscripción, a la asignatura "0889" y al grupo "0017", el archivo de "celdas" queda modificado al tener dos registros nuevos, así como también se cambian algunos apuntadores como se aprecia en el siguiente esquema:



Estado académico del alumno "después" de reinscribirse
 Figura IV.13

b) Descripción de sus interfaces

Entrada:

- 1) Un arreglo con las asignaturas, ya revisadas, para que proceda su reinscripción.

Entrada/Salida:

- 1) Archivo de alumnos: para modificar el apuntador a su primera asignatura reinscrita.
- 2) Archivo de "celdas", en donde se "crean" dos nuevos registros para cada nueva asignatura.
- 3) Archivo de grupos: se incrementa el número de alumnos reinscritos en el grupo.

c) Detalle del módulo.

```
!
!
! Empezar: REINSCRIBE-ASIGNATURAS
!
! SI reinscripción es al mismo plantel ENTONCES
! EFECTUA lectura del grupo
! INCREMENTA uno al contador de reinscritos
! EFECTUA Escritura del registro del grupo
!
! (* Crea celda tipo "Grupo-Alumno" *)
!
! MUEVE Clave-Grupo al atributo-de-celda
! MUEVE Apuntador del archivo de personas al
! Primer apuntador de la celda
! MUEVE Apuntador del archivo de grupos al
! Segundo apuntador de la celda
! MUEVE Apuntador del siguiente alumno del
! grupo al tercer apuntador de la celda
! EFECTUA Escritura de la nueva celda
!
! (* Crea la celda tipo "Cursando-Alumno" *)
!
! MUEVE Asignatura al atributo-de-celda
! MUEVE Apuntador del archivo de asignaturas
! al Primer apuntador de la celda
! MUEVE Apuntador a la celda tipo Grupo-Alumno
! al Segundo apuntador de la celda
! MUEVE Apuntador de la siguiente asignatura
! reinscrita al Tercer apuntador de la celda
! EFECTUA Escritura de la nueva celda
!
! SINO
! EFECTUA REINSCRIPCION-A-OTRO-PLANTEL
! FIN (SI)
!
! Termina: REINSCRIBE-ASIGNATURAS
!
```

Detalle del módulo II.C : REINSCRIBE-ASIGNATURAS
Figura IV.14

4.3 Descripción módulo II.C.1 REINSCRIBE-A-OTRO-PLANTEL

a) Descripción del proceso.

Así como el alumno tiene "ligadas" sus asignaturas reinscritas mediante un apuntador en su registro del archivo de alumnos, tiene también otro apuntador que relaciona sus asignaturas reinscritas a otro plantel.

El proceso cambia, debido a que se necesita otra "celda" que relacione todas las asignaturas que correspondan a un sólo plantel. Se necesita, por tanto, tantas celdas como planteles a los que esté reinscrito. La celda que indica el plantel al cual se reinscribe se llama: "Otra-Plantel-Alumno", y esta conformada de la siguiente manera:

CLAVE DE	APUNTADOR	APUNTADOR	APUNTADOR
PLANTEL	A LA CELDA	A LA CELDA	A LA CELDA
	TIPO:	SIGUIENTE	SIGUIENTE
	CURSANDO-	DEL MISMO	DEL MISMO
	ALUMNO	TIPO	TIPO
		(vía plantel)	(vía alumno)

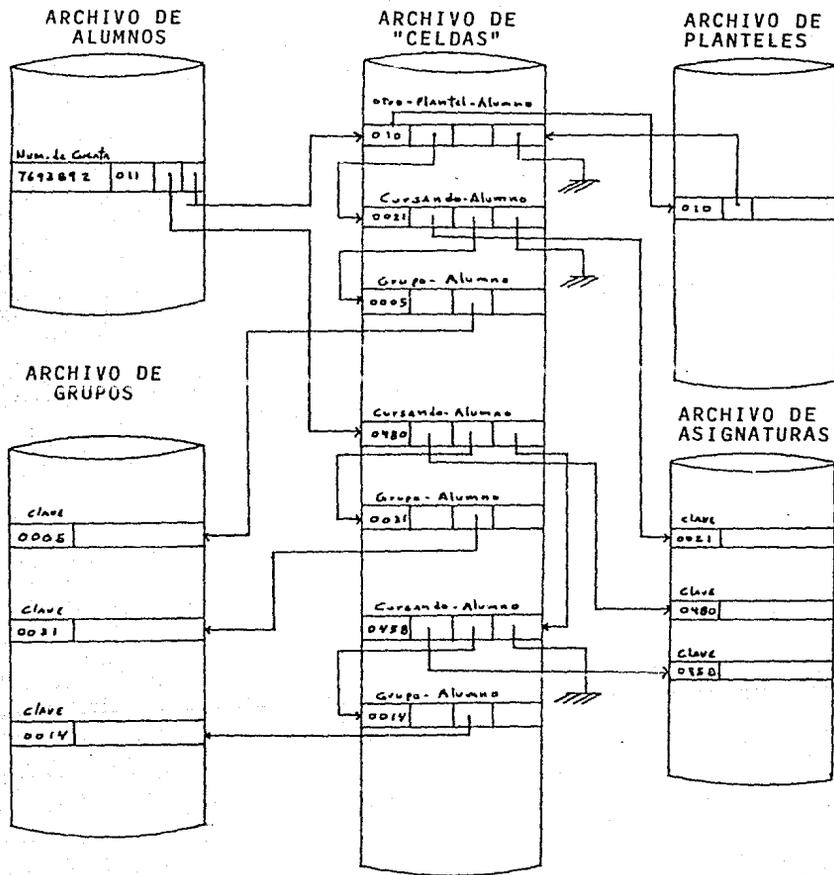
A este tipo de celda se le relacionan todas las asignaturas reinscritas que tiene el alumno al plantel que se indica en el atributo de la celda. Esta relación es mediante el primer apuntador, que al relacionarse con la celda tipo "Cursando-Alumno", todas las siguientes relaciones son exactamente igual que para el caso de la reinscripción en el mismo plantel.

En caso de que el alumno tenga reinscripciones en más de dos planteles, se le relaciona mediante el tercer apuntador de la celda, en caso negativo, este apuntador está en cero.

Cada plantel tiene relacionados a todos los alumnos que siendo de otros planteles, están reinscritos en él. Para ello utiliza el segundo apuntador de la celda, mediante la cual los liga. En cambio, el tercer apuntador de la celda, liga los planteles a los cuales se ha reinscrito el alumno.

Es así que todo el procedimiento del módulo anterior REINSCRIBE-ASIGNATURA, se utiliza añadiéndole la creación de la nueva celda descrita.

Una manera esquemática de visualizar la relación de los archivos cuando cursa asignaturas en otro plantel se presenta en la figura IV.15 .



Relación entre los archivos de Control Escolar cuando hay reinscripción en otro plantel
 Figura IV.15

b) Descripción de las interfaces

Entrada:

- 1) Recibe un arreglo con las asignaturas, ya revisadas, para que proceda su reinscripción.

Entrada/Salida:

- 1) Archivo de Alumnos: para modificar el apuntador a su primera asignatura reinscrita.
- 2) Archivo de "celdas", en donde se "crean" dos nuevos registros para cada nueva asignatura reinscrita en el plantel del alumno.
- 3) Archivo de grupos: se incrementa el número de alumnos reinscritos en el grupo.

c) Detalle del módulo.

En la figura IV.16 se muestra el bloque detallado.

Empieza: REINSCRIBE-ASIGNATURAS-A-OTRO-PLANTEL

EFFECTUA lectura del grupo

INCREMENTA uno al contador-reinscritos-otro-plantel

EFFECTUA Escritura del registro del grupo

(* Crea celda tipo "Grupo-Alumno" *)

MUEVE Clave-Grupo al atributo-de-celda

MUEVE Apuntador del archivo de personas al

Primer apuntador de la celda

MUEVE Apuntador del archivo de grupos al

Segundo apuntador de la celda

MUEVE Apuntador del siguiente alumno del

grupo al Tercer apuntador de la celda

EFFECTUA Escritura de la nueva celda

(* Crea la celda tipo "Cursando-Alumno" *)

MUEVE Asignatura al atributo de la celda

MUEVE Apuntador del archivo de asignaturas

al Primer apuntador de la celda

MUEVE Apuntador de la celda tipo Grupo-Alumno

al Segundo apuntador de la celda

MUEVE Apuntador de la siguiente asignatura

reinscrita al Tercer apuntador de la celda

EFFECTUA Escritura de la nueva celda

SI tiene asignaturas reinscritas en el otro plantel

(* Modifica celda tipo: "Otra-Plantel-Alumno" *)

MUEVE Clave de plantel al atributo de la celda

MUEVE Apuntador de la celda tipo cursando-alumno

al Primer apuntador de la celda

MUEVE Apuntador del siguiente alumno de otro

plantel al Segundo apuntador de la celda

MUEVE Apuntador de la siguiente celda tipo OTRO-

PLANTEL-ALUMNO al Tercer apuntador de la

celda

SINO

(* Crea celda tipo: "Otro-Plantel-Alumno" *)

MUEVE Apuntador de la celda tipo cursando-alumno

al Primer apuntador de la celda

FIN (SI)

EFFECTUA Escritura del registro de la celda

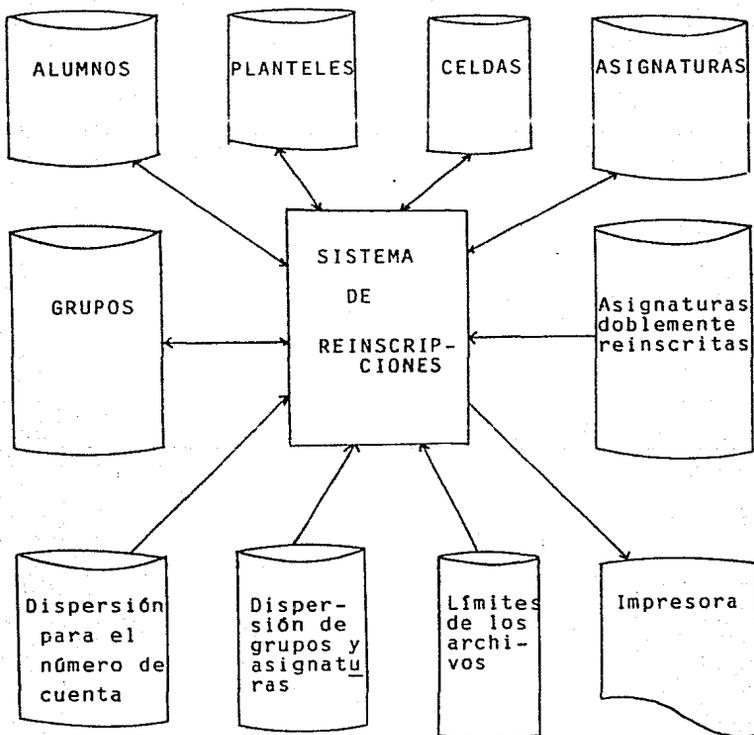
Termina: REINSCRIBE-ASIGNATURAS-A-OTRO-PLANTEL

Detalle del módulo II.C.1:
REINSCRIBE-ASIGNATURAS-A-OTRO-PLANTEL
Figura IV.16

5.0 INTERFASE EXTERNA

5.1 Diagrama de la interfase con la base de datos

El sistema de reinscripciones tiene gran comunicación con varios de los archivos del Sistema de Control Escolar; en unos casos solamente consulta los archivos y en otros los modifica. En la figura IV.17 se muestra esta interrelación de los archivos con el programa de reinscripciones.



Interfases externas del Sistema de Reinscripciones
Figura IV.17

5.2 Mensajes

El sistema tiene dos tipos de mensajes: aquellos que señalan un tipo error durante el proceso o aquellos que indican una circunstancia especial.

a) Mensajes sobre errores.

Existen dos tipos de errores: aquellos que señalan un error en un movimiento de la reinscripción y aquellos que señalan un error general del proceso. Los primeros simplemente rechazan el movimiento, en cambio, los segundos descontinúan todo el proceso al imposibilitar continuar con dicho error. A continuación se explicitan dichos errores.

a.1) Errores sobre los movimientos.

Cualquiera de estos errores implica que el movimiento señalado ha sido rechazado; la mayoría de los textos de los mensajes se entienden por sí mismos, sin embargo se explicitan para su mejor comprensión.

- 1) "CLAVE PLANTEL NO NUMERICO" : La clave del plantel sólo debe contener caracteres numéricos.
- 2) "CLAVE PLANTEL NO EXISTE " : En caso de que el plantel no haya sido dado de alta en los archivos.
- 3) "GRUPO YA TIENE ACTA " : Si el grupo ya ha sido evaluado, imposibilita la reinscripción a éste.
- 4) "NUM DE CUENTA NO NUMERICO" : Es un error de formato al poner un carácter no numérico.
- 5) "NUM. DE CUENTA INCORRECTO" : En caso de que el dígito de control esté incorrecto.
- 6) "NUM. DE CUENTA NO EXISTE " : En caso de que no se tenga registrado el número de cuenta
- 7) "ASIGNATURA NO NUMERICA " : La clave de la asignatura no debe contener caracteres no numéricos.
- 8) "GRUPO NO EXISTE " : En caso de que no esté registrado el grupo.
- 9) "YA INSCRITO EN GPO : _____" : Si el alumno ya está inscrito en la asignatura solicitada, se imprime el grupo reinscrito.

- 10) "NO CORRESPONDE EL PLANTEL" : En caso de Preparatoria o Colegio de Ciencias y Humanidades no procede reinscribirse en otro plantel.
- 11) "ASIGNATURA NO EXISTE " : En caso de que la asignatura no ha sido dada de alta en los archivos.
- 12) "GRUPO INCORRECTO " : Sucede si la clave del grupo contiene caracteres incorrectos.
- 13) "ALUMNO DADO DE BAJA " : Un alumno dado de baja por cualquier razón, nunca podrá volver a reinscribirse.
- 14) "ALUMNO INCORRECTO " : Sucede si el registro del archivo de alumnos contiene información incorrecta
- 15) "ALUMNO INEXISTENTE " : Sucede en caso de que no se encuentre al alumno en los archivos.
- 16) "NO PROCEDE POR SER EXALUM" : El alumno pasa a ser exalumno una vez que termina el tiempo correspondiente para cursar sus estudios, imposibilitando así, su reinscripción.
- 17) "FALTAN REGISTROS 'DI' " : Sucede cuando sólo hubo registro-maestro, sin ningún registro-detalle.
- 18) "DETALLE CON 'MI' ERRONEA " : En caso de que el registro-maestro contenga al menos un error, ninguno de sus registros-detalle procede.
- 19) "ART.27 (2 CALIF. EN HA) " : Consiste en la violación al artículo 27, en ya que se han cursado al menos dos veces la asignatura en ordinario. En este caso esas dos asignaturas ya se tienen registradas en la historia académica del alumno.
- 20) "ART.27 (HA+INSC.SEN.ANT.)" : Al igual que en el caso anterior, pero con la diferencia de que en sus asignaturas doblemente cursadas, una fue cursada en el semestre inmediato anterior.
- 21) "EL GRUPO ESTA CANCELADO " : En un grupo, que en un principio fue dado de alta, pero se canceló, no es posible reinscribirse.

a.2) Errores sobre el proceso.

Como se mencionó, estos errores suspenden el proceso violentamente, señalando la causa. Estos errores

imposibilitan la ejecución del proceso por tratarse de situaciones generales. Se termina de una manera violenta, no enviando a la rutina de terminación. Esto es a propósito para que los procesos, independientes de la reinscripción, que se ven afectados con el proceso incompleto no sean ejecutados, sino tomen las medidas preventivas. Estos errores son los siguientes:

- 1) "OPCION FAN. NO IDENTIFICADA": En caso de que el parámetro que señala el nivel de estudios esté incorrecto.
- 2) Cualquier error en la lectura inicial o posterior de los archivos, así como en su escritura se indica:
 - "ERROR EN LA LECTURA INICIAL DEL BLQ-XX "
 - "ERROR AL LEER EN BLQ-XX "
 - "ERROR AL ESCRIBIR EN BLQ-XX "
 - "ERROR EN LA ESCRITURA FINAL DEL BLQ-XX"

En donde en vez de XX, se pone la clave del archivo utilizado.

- 3) Si hay problemas al leer o escribir de una manera aleatoria debido a que el índice o llave rebasa los límites físicos del archivo para accederlo, se indica:

- "INDICE MUY GRANDE EN XXXX ": En donde 'XXXX' muestra en que archivo se hizo el acceso.

- 4) El control al acceso para modificar la base de datos se hace mediante la escritura en un pequeño archivo de una clave o "candado". Si ha sido escrita, se están modificando, de no estarla hay acceso libre; la indicación del mensaje dice:

- " C A N D A D O P U E S T O "

a.3) Mensajes especiales.

Este tipo de mensajes sólo indican ciertas particularidades de los movimientos y a diferencia de los anteriores, no implican el rechazo del movimiento efectuado. Se tienen diez mensajes:

- 1) "INICIA REINSCRIPCION DE LA XXX " : En donde "XXX" significa el nombre de la escuela o plantel en donde se harán las reinscripciones. Este mensaje se pone sólo una vez al inicio del proceso.

- 2) "ALUMNO DE OTRO PLANTEL" : En caso de que el alumno sea de otro plantel al cual se hacen las reinscripciones. En caso de alumnos de nivel profesional no se rechaza el movimiento, sólo se indica.
- 3) "EXTRAORDINARIO NUM. XX" : En donde "XX" significa el número de veces que el alumno ha cursado en extraordinario la asignatura solicitada.

5.3 Paquetería

El sistema utiliza, y es del todo recomendable que lo haga, ciertas rutinas del Sistema de Control Escolar que ya están probadas y son usadas en otros programas. Principalmente son las rutinas referentes a las lecturas y escrituras de los archivos más importantes. Así también es conveniente utilizar los mismos formatos y las variables - en cuanto que sea posible- del sistema general. Las rutinas utilizadas son las siguientes:

- c.1) LEE-PERS : Lectura del archivo de alumnos.
- c.2) LEE-ESC : Lectura del archivo de planteles.
- c.3) LEE-MAT : Lectura del archivo de asignaturas.
- c.4) LEE-CPO : Lectura del archivo de grupos.
- c.5) LEE-CELDA: Lectura del archivo de apuntadores.
- c.6) LEE-DCTA : Lectura del archivo de dispersión de número de cuenta.
- c.7) ESC-PERS : Escritura al archivo de alumnos.
- c.8) ESC-ESCU : Escritura al archivo de planteles.
- c.9) ESC-GPC : Escritura al archivo de grupos.
- c.10) ESC-CELDA: Escritura al archivo de apuntadores.
- c.11) BUSCA-CVE-MAT : Rutina que hace la dispersión de la clave de la asignatura para dispersarla.
- c.12) BUSCA-CTA: Rutina que realiza la dispersión del número de cuenta.

6.0 EQUIPO Y LENGUAJE

Equipo : Computadora Burroughs 6000

Lenguaje: COBOL

7.0 Recuperación de fallas

El programa que realiza las reinscripciones, se activa mediante un pequeño programa de comandos al sistema, en donde en caso de fallas del equipo (procesador o equipo periférico), es capaz de volver a reactivarlo una vez que el equipo restablece su funcionamiento. Sin

embargo, puesto que es posible que hayan quedado los movimientos incompletos, es preciso tener una copia del estado inicial del proceso para que cuando se reactive el sistema, se reinicie el proceso desde su comienzo.

3.0 Certificación de calidad

Las necesidades que se plantearon en la "Especificación de los requerimientos" se cubren por los diferentes módulos:

Sobre las funciones a realizar (Sección III.3.1):

- a) La revisión de la información se hace en los siguientes módulos:

II.A : FILTRO-REGISTRO-MAESTRO
II.B : VERIFICA-Y-CARGA-REGISTRO-DETALLE
II.B.1 : FILTRO-REGISTRO-DETALLE
II.B.3 : REVISION-COM-REINSCRITAS

- b) La actualización de los archivos computarizados con la nueva información:

II.C : REINSCRIBE-ASIGNATURAS
III.B : ACTUALIZA-PLANTEL

- c) La emisión de reportes con el diagnóstico del proceso:

III.C : ESCRIBE-REPORTES-TOTAL
III.D : EMITE-REPORTE-ERRORES

Sobre las restricciones a observar (Sección III.3.3.d):

- a) La revisión de las asignaturas doblemente reinscritas:

II.C.3 : REvisa-DOBLE-REINSCRIPCION

- b) El poder reinscribir en grupos de extraordinario:

II.B.3 : REVISION-COM-REINSCRITAS

- c) Sobre la reinscripción a otro plantel del correspondiente al del alumno:

II.C.1 : REINSCRIBE-ASIGNATURAS-A-OTRO-PLANTEL

- d) Para el caso especial de la reinscripción de la Facultad de Medicina:

II.B.3.1 : REVISION-MEDICINA

9.0 Apéndices

APENDICE A: ESTRUCTURA FISICA DE LOS ARCHIVOS

A continuación se presentan el contenido de los archivos principales del Sistema de Registro Escolar, pudiéndose remitir a listados del mismo sistema para una mejor descripción de ellos.

a.1) Archivo de planteles

CONTENIDO	LONGITUD	TIPO
Nombre del plantel	32	alfanumérico
Número de alumnos-asignatura	6	numérico
Número de asignaturas	4	numérico
Número de grupos	5	numérico
Número de alumnos en el plantel	5	numérico
Número de alumnos de otro plantel	5	numérico
Número de alumnos reinscritos en el semestre	5	numérico
Número de transacciones solicitadas	5	numérico
Número de transacciones realizadas	5	numérico
Contador de planteles	90	numérico
Apuntador a oficio	6	numérico
Apuntador a departamento	7	numérico
Apuntador a Alumnos	7	numérico
Apuntador a Maestros	7	numérico
Apuntador a los alumnos que han cursado en otro plantel	7	numérico
Apuntador a los alumnos que cursan en otro plantel	7	numérico

Características:

Longitud del registro = 299 caracteres (al contenido del registro se le añaden campos en blanco para su próxima utilización)

Longitud del bloque = 299 caracteres (1 registro)

a.2) Archivo de asignaturas

CONTENIDO	LONGITUD	TIPO
Clave de la asignatura	4	numérico
Créditos	2	numérico
Semestre en que se cursa	2	numérico
Nivel de estudios	1	numérico
Apuntador a los grupos asignados	7	numérico
Número de grupos asignados	2	numérico
Departamento al que corresponde	2	numérico
Nombre de la asignatura	28	alfanumérico
Apuntador a las otras asignaturas	7	numérico

Características:

Longitud del registro lógico = 64 caracteres
Longitud del registro físico = 640 caracteres (diez registros lógicos)
Longitud del bloque = 640 caracteres (1 reg. físico)

A.3) Archivo de carreras

CONTENIDO	LONGITUD	TIPO
Registro primero:		
Clave de la carrera	4	numérico
Nombre de la carrera	28	numérico
Registro segundo:		
Clave del plan de estudios	1	numérico
Número de semestres que abarca	2	
Número del total de créditos obligatorios	3	numérico
Número del total de créditos optativos	3	numérico

Características:

Para la información de las carreras se utilizan los dos registros descritos. Estos datos se encuentran en el mismo archivo que las asignaturas, aunque para efectos del manejo de la información se consideren como dos archivos diferentes.

Longitud del registro lógico = 64 caracteres

Longitud del registro físico = 640 caracteres (diez registros lógicos)

Longitud del bloque = 640 caracteres (1 reg. físico)

A.4) Archivo de grupos

CONTENIDO	LONGITUD	TIPO
Clave del grupo	4	alfanumérico
Número de alumnos	6	numérico
Número del cupo inicial	6	numérico
Apuntador al maestro	6	numérico
Apuntador al primer alumno (celdas)	7	numérico
número de alumnos inscritos de otro plantel	2	numérico
Apuntador a la asignatura	5	numérico
Apuntador a celda tipo GPO-ALUMNO	7	numérico
Apuntador al archivo de actas	4	numérico

Características:

Longitud del registro lógico = 64 caracteres

Longitud del registro físico = 640 caracteres (diez registros lógicos)

Longitud del bloque = 640 caracteres (1 reg físico)

A.5) Archivo de Alumnos

CONTENIDO	LONGITUD	TIPO
Nombre del alumno	32	alfanumérico
Clave de las causa de ingreso	2	numérico
Fecha del último movimiento	6	numérico
Apuntador a su resumen académico	6	numérico
Número de reinscritas en ordinario	2	numérico
Número de reinscritas en extraordinario	2	numérico
Número de cuenta	8	numérico
Clave del plantel	3	numérico
Clave del departamento	2	numérico
Apuntador a celda: Cursando-Alumno	7	numérico
Apuntador a celda: Otra-Plantel-Alumno	7	numérico
Clave del plantel de procedencia	7	numérico
Fecha de nacimiento	6	numérico
Apuntador a celda: Alumno-Plantel	7	numérico
Estado del alumno (baja, alta, etc)	1	numérico
Apuntador al archivo de asignaturas doblemente reinscritas en ordinario	7	numérico
Año de primer ingreso	2	numérico
Clave del sexo del alumno	1	numérico
Clave de Nacionalidad	1	numérico

Características:

Longitud del registro lógico = 104 caracteres

Longitud del registro físico = 1040 caracteres (diez registros lógicos)

Longitud del bloque = 1040 caracteres (1 reg. físico)

A.6) Archivo de las asignaturas doblemente reinscritas en ordinario

CONTENIDO	LONGITUD	TIPO
Número de cuenta	3	numérico
Clave de asignatura	4	numérico (se repite once veces)

Características:

Longitud del registro = 72 caracteres (el lógico es igual al físico)

Longitud del bloque = 720 caracteres (10 registros)

A.8) Archivo de celdas

CONTENIDO	LONGITUD	TIPO
Atributo	4	alfanumérico
Primer apuntador	7	numérico
Segundo apuntador	7	numérico
Tercer apuntador	7	numérico

Características:

Longitud del registro lógico = 28 caracteres

Longitud del registro físico = 1600 caracteres (cien registros lógicos)

Longitud del bloque = 1600 caracteres (1 registro físicos)

APENDICE B.1 : DIFERENTES TIPOS DE CELDAS

	TIPO DE CELDA	ATRIBUTO	APUNT-1	APUNT-2	APUNT-3
I)	ALUMNO PLANTEL	CLAVE DE PLANTEL	APUNT. A ALUMNO	X	SIGUIENTE
II)	GRUPO ASIGNATURA	CLAVE DE GRUPO	APUNT. A GRUPO	APUNT. A CELDA ANTERIOR	SIGUIENTE
III)	CARRERA PLANTEL	CLAVE DE CARRERA	NÚMERO DE ASIGN.	APUNT. A CELDA ASIGN-GPO	SIGUIENTE
IV)	CURSANDO ALUMNO	CLAVE DE ASIGN.	APUNT. AL ARCH. ASIGN.	APUNT. A CELDA CPO-ALUM	SIGUIENTE
V)	ASIGNATURA CARRERA	CLAVE DE ASIGN.	APUNT. AL ARCH. ASIGN.	APUNT. A CELDA ANTERIOR	SIGUIENTE
VI)	GRUPO ALUMNO	CLAVE DE GRUPO	APUNT. AL ARCH. ALUMNO	APUNT. AL ARCH. GRUPO	SIGUIENTE
VII)	OTRO PLANTEL ALUMNO	CLAVE DEL PLANTEL	APUNT. A CELDA CURS-ALUM	SIGUIENTE (PLANTEL)	SIGUIENTE (ALUMNO)
VIII)	NOMBRE ASIGNATURA	CLAVE DE PLANTEL	APUNT. AL ARCH. ASIGN.	X	SIGUIENTE
IX)	OTRO PLANTEL	CLAVE DE PLANTEL	APUNT. AL ARCH. ALUMNO	APUNT. A SIG. PLANTEL	SIGUIENTE

Descripción de los diversos tipos de "celdas":

I) ALUMNO-PLANTEL:

Relaciona en forma encadenada a todos los alumnos de un mismo plantel.

II) GRUPO-ASIGNATURA:

Encadena a todos los grupos asociados a una misma asignatura.

III) CARRERA-PLANTEL:

Relaciona todas las carreras de un plantel (actualmente sólo se utiliza una carrera).

IV) CURSANDO-ALUMNO:

Encadena todas las asignaturas que cursa un alumno en su plantel correspondiente.

V) ASIGNATURA-CARRERA:

Encadena todas las asignaturas asociadas a una carrera (actualmente son encadenadas a la única carrera).

VI) GRUPO-ALUMNO:

Encadena a todos los alumnos que están reinscritos a una misma asignatura-grupo.

VII) OTRO-PLANTEL-ALUMNO

Encadena las asignaturas que cursa un alumno registrado en un plantel, que no corresponde al que se inscribió en un principio.

VIII) NOMBRE-ASIGNATURA:

Se utiliza para ligar las asignaturas con su nombre.

IX) OTRO-PLANTEL:

Encadena a un plantel todos los alumnos que están cursando asignaturas en él, pero no están reinscritos en dicho plantel.

IV.2 CODIFICACION

Una vez especificado el diseño del sistema, la segunda etapa a seguir en el desarrollo es la codificación.

De los dos enfoques postulados (sección II.2.2), se prefirió seguir el llamado "conservador", es decir, en primer lugar se decidió terminar con todo el diseño para después proceder a las etapas de codificación y pruebas. Se decidió por este enfoque por tener un programa de actividades muy preciso: período de reinscripciones, período de modificación e inscripción, calificación de asignaturas, etc.

IV.2.1 Estilo de codificación

Se trató de buscar en el código la simplicidad y la claridad. Para ello se utilizaron los nombres mnemotécnicos para las variables y rutinas, comentarios que hicieran más explícito el desarrollo del programa fuente, y se siguió un orden en la ubicación de las rutinas de acuerdo a la estructura del proceso (sección IV.I, parte 2.4). También se procuró hacer la declaración de las variables teniendo en cuenta un orden alfabético, y la utilización de condicionales simples y positivos.

Respecto a las entradas al Sistema de Reinscripciones se desarrollaron rutinas de validación, y la estructura del lenguaje permite la identificación de la terminación de datos en la lectura, sin ningún problema.

Para la salida se desarrollan los diversos formatos, constituyendo los diversos tipos de reporte. Ya en el diseño se presentaron los datos básicos que deben de contener, ahora sólo sigue elaborar el formato preciso para que ese tipo de diagnóstico sea codificado.

- 1o) El diagnóstico general presenta el detalle del proceso, es decir, se imprime cada movimiento que entra al sistema y la resolución que se toma respecto a él.
- 2o) El diagnóstico resumido, presenta en una sola hoja el resumen estadístico de todo el proceso a fin de visualizar de manera rápida el resultado de las reinscripciones ejecutadas.
- 3o) El tercer tipo de reportes indica, también en una sola hoja, la contabilización de los tipos de errores, y así, señala cuantos veces se presentó cada error.

A continuación se presenta la codificación de los tres tipos de reportes respectivamente.

```

(2 24 C C / .....*****
0065 C C L **          EN C A B E Z A D O S          D E          I M P R E S I O N          **
0097 C C C O .....*****
0099 C C C O .....*****
0041 C C C O 01  TIT -UNAM.                                PIC X(04) VALUE SPACES.
0047 C C C O 03  FILLER                                PIC X(02) /-
0072 C C C O 03  FIA-IMP                               PIC X(03) /-
0073 C C C O 03  PES-IMP                               PIC 9(02) -
0074 C C C O 03  PNC-IMP                               PIC X(32) - VALUE SPACES.
0075 C C C O 03  FILLER                                PIC X(74) VA "UNIVERSIDAD
0076 C C C O 03  FILLER                                "AL AUTONOMA DE MEXICO".
0077 C C C O 03  FILLER                                PIC X(05) VA "HOJA "
0078 C C C O 03  FILLER                                PIC ZZ,ZZ9.
0079 C C C O 03  NUM-HOJA
0080 C C C O
0081 C C C O 01  TIT -CAE.                                PIC X(46) VALUE SPACES.
0082 C C C O 01  FILLER                                PIC X(68) VA "COORDINACION DE LA
0083 C C C O 03  FILLER                                "ADMINISTRACION ESCOLAR".
0084 C C C O 01  TIT -ST.                                PIC X(56) VALUE SPACES.
0085 C C C O 03  FILLER                                PIC X(58) VA "SUBDIRECCION TECHN
0086 C C C O 03  FILLER                                "CA".
0087 C C C O 01  ENCA -1.                                PIC X(51) VALUE SPACES.
0088 C C C O 03  FILLER                                PIC X(30) VALUE
0089 C C C O 03  FILLER                                "DIAGNOSTICO DE REINSCRIPCION".
0090 C C C O 03  FILLER                                PIC X(51) VALUE SPACES.
0091 C C C O
0092 C C C O 01  ENCA -2.                                PIC X(46) VALUE SPACES.
0093 C C C O 03  FILLER                                PIC X(01) VALUE "C".
0094 C C C O 03  FILLER                                PIC 9(03) - VALUE "J "
0095 C C C O 03  ESC-IMP                               PIC X(03) -
0096 C C C O 03  FILLER                                PIC X(32) -
0097 C C C O 03  ACW-ESC-IMP                          PIC X(47) - VALUE SPACES.
0098 C C C O 03  FILLER
0099 C C C O
0100 C C C O 01  GUIONES.                                PIC X(132) VALUE ALL " - ".
0101 C C C O 03  FILLER
0102 C C C O
0103 C C C O
0104 C C C O 01  ENCA -3A.                               PIC X(29) VALUE SPACES.
0105 C C C O 03  FILLER                                PIC X(11) VALUE "NUMERO
0106 C C C O 03  FILLER                                PIC X(10) VALUE "CVE".
0107 C C C O 03  FILLER                                PIC X(04) VALUE "CVE"
0108 C C C O 03  FILLER                                PIC X(78) VALUE SPACES.
0109 C C C O
0110 C C C O 01  ENCA -3B.                               PIC X(152) VALUE
0111 C C C O 03  FILLER                                "----- OBSERVACIONES -----> H
0112 C C C O 03  FILLER                                "HOJA     MOV     PLT     CARR     ASIG     SPO <
0113 C C C O 03  FILLER                                "----- NOMBRE     ASIGNATURA -----> CUE
0114 C C C O 03  FILLER                                "NTA     -D <----- NOMBRE     ALUMNO
0115 C C C O 03  FILLER                                "-----> "
0116 C C C O
0117 C C C O
0118 C C C O
0119 C C C O

```

```

10200000 *****
10200000 **
10200000 **
10200000 **
10200000 *****
10200000
10200000 C1 DETALLE
10200000 C3 D-MENSAJE
10200000 C3 D-D-MENSAJE-2 PIC X(19)
10200000 C3 D-CONTEXT PIC ZZ9
10200000 C3 D-MENSAJE-3 PIC X(07)
10200000 C3 D-NUM-SEC PIC ZZZZ9
10200000 C3 FILLER PIC X(02) VALUE SPACES
10200000 C3 D-TIPO-TAR PIC X(02)
10200000 C3 FILLER PIC X(02) VALUE SPACES
10200000 C3 D-PLANTEL PIC X(02)
10200000 C3 FILLER PIC X(02) VALUE SPACES
10200000 C3 D-CARRERA PIC 9(02)
10200000 C3 FILLER PIC X(02) VALUE SPACES
10200000 C3 D-ASIGNATURA PIC 9(04)
10200000 C3 FILLER PIC X(04) VALUE SPACES
10200000 C3 D-ERUPO PIC X(04)
10200000 C3 FILLER PIC X(01) VALUE SPACES
10200000 C3 D-NOMBRE-AT PIC X(20)
10200000 C3 FILLER PIC X(01) VALUE SPACES
10200000 C3 D-NUM-CUENTA
10200000 C3 D-NUM-CTA PIC 9(07)-9
10200000 C3 FILLER PIC X(01) VALUE SPACES
10200000 C3 D-NOMBRE PIC X(52)
10200000
10200000 D1 DET-ERROR
10200000 C3 NOP-ERROR-IMP PIC X(25)
10200000 C3 FILLER PIC X(12) VALUE SPACES
10200000 C3 ASTER-IMP PIC X(01) OCCURS 62 TIMES
10200000 C3 FILLER PIC X(33) VALUE SPACES
10200000
10200000 L1 ESPACIOS
10200000 C3 FILLER PIC X(132) VALUE SPACES
10200000
10200000 G1 ENC-ERROR
10200000 C3 FILLER PIC X(47) VALUE SPACES
10200000 C3 FILLER PIC X(40) VALUE "RELACION"
10200000 C3 FILLER "DEL ERROR"
10200000 C3 FILLER PIC X(45) VALUE SPACES
10200000
10200000 C1 DET-ERR-1
10200000 C3 FILLER PIC X(47) VALUE SPACES
10200000 C3 FILLER PIC X(30) VALUE "OCURRENCIAS NO"
10200000 C3 FILLER "MAYOR DEL ERROR"
10200000 C3 FILLER PIC X(55) VALUE SPACES
10200000
10200000 L1 DET-ERR-2
10200000 C3 FILLER PIC X(49) VALUE SPACES
10200000 C3 TOTAL-ERROR-IMP PIC ZZZZ9
10200000 C3 FILLER PIC X(03) VALUE SPACES
10200000 C3 NOMBRE-ERROR-IMP PIC X(20)
10200000 C3 FILLER PIC X(43) VALUE SPACES
10200000
10200000 D1 MSG-012
10200000 C3 FILLER PIC X(19) VALUE
10200000 C3 D-INS-CICLO "INSERITC EN ASIG.: "
10200000 C3 FILLER PIC ZZ9
10200000 C3 FILLER PIC X(07) VALUE "VECES "

```

```

108300 *****
108400 *****
108500 *****
108600 *****
108700 *****
108800 *****
108900 *****
109000 *****
109100 *****
109200 *****
109300 *****
109400 *****
109500 *****
109600 *****
109700 *****
109800 *****
109900 *****
110000 *****
110100 *****
110200 *****
110300 *****
110400 *****
110500 *****
110600 *****
110700 *****
110800 *****
110900 *****
111000 *****
111100 *****
111200 *****
111300 *****
111400 *****
111500 *****
111600 *****
111700 *****
111800 *****
111900 *****
112000 *****
112100 *****
112200 *****
112300 *****
112400 *****
112500 *****
112600 *****
112700 *****
112800 *****
112900 *****
113000 *****
113100 *****
113200 *****
113300 *****
113400 *****
113500 *****
113600 *****
113700 *****
113800 *****
113900 *****
114000 *****

```

Line	Code	Description	Field	Value
109000	G1	ENC A-CTR-1.		
109100	03	FILLER	PIC X(46)	VALUE SPACES.
109200	03	FILLER	PIC X(37)	VALUE "ALUMNUS REINDECK"
109300	03	FILLER	PIC X(47)	VALUE SPACES.
109500	C1	ENC A-CTR-2.		
109600	03	FILLER	PIC X(46)	VALUE SPACES.
109700	03	FILLER	PIC X(01)	VALUE "N".
109800	03	DISP-CONVE-ESC	PIC X(03)	VALUE " ".
109900	03	DISP-CONVE	PIC X(02)	VALUE " ".
110000	03	DISP-NUMBR-ESC	PIC X(32)	VALUE SPACES.
110100	03	FILLER	PIC X(46)	VALUE SPACES.
110300	G1	ENC A-CTR-3.		
110400	03	FILLER	PIC X(45)	VALUE SPACES.
110500	03	FILLER	PIC X(37)	VALUE "ALUMNUS REINDECK"
110600	03	FILLER	PIC X(45)	VALUE SPACES.
110700	03	FILLER	PIC X(35)	VALUE "ALUMNUS REINDECK"
110800	03	FILLER	PIC X(45)	VALUE SPACES.
110900	D1	ENC A-CTR-4.		
111000	03	FILLER	PIC X(45)	VALUE SPACES.
111100	03	FILLER	PIC X(35)	VALUE "ALUMNUS REINDECK"
111200	03	FILLER	PIC X(45)	VALUE SPACES.
111300	G3	FILLER	PIC X(35)	VALUE "ALUMNUS REINDECK"
111400	G3	FILLER	PIC X(45)	VALUE SPACES.
111500	C1	ENC A-CTR-5.		
111600	03	FILLER	PIC X(46)	VALUE SPACES.
111700	03	FILLER	PIC X(28)	VALUE "NUMERU DE REGI"
111800	03	FILLER	PIC X(28)	VALUE "NUMERU DE REGI"
111900	03	DISP-TOT-REG-MI	PIC ZZZ,ZZ9	VALUE " ".
112000	03	FILLER	PIC X(47)	VALUE SPACES.
112100	G1	ENC A-CTR-6.		
112200	03	FILLER	PIC X(46)	VALUE SPACES.
112300	03	FILLER	PIC X(28)	VALUE "NUMERU DE REGI"
112400	03	FILLER	PIC X(28)	VALUE "NUMERU DE REGI"
112500	03	FILLER	PIC ZZZ,ZZ9	VALUE " ".
112600	03	DISP-TOT-REG-DI	PIC ZZZ,ZZ9	VALUE " ".
112700	03	FILLER	PIC X(47)	VALUE SPACES.
112800	G1	ENC A-CTR-7.		
112900	03	FILLER	PIC X(63)	VALUE SPACES.
113000	03	FILLER	PIC X(13)	VALUE "PROCEDIERON :".
113100	03	FILLER	PIC ZZZ,ZZ9	VALUE " ".
113200	03	DISP-PROCEDIERON	PIC ZZZ,ZZ9	VALUE " ".
113300	03	FILLER	PIC X(45)	VALUE SPACES.
113400	D1	ENC A-CTR-8.		
113500	03	FILLER	PIC X(50)	VALUE SPACES.
113600	03	FILLER	PIC X(10)	VALUE "NO PROCEDIERON"
113700	03	FILLER	PIC X(10)	VALUE "NO PROCEDIERON"
113800	03	FILLER	PIC ZZZ,ZZ9	VALUE " ".
113900	03	DISP-CON-ERROR	PIC ZZZ,ZZ9	VALUE " ".
114000	03	FILLER	PIC X(49)	VALUE SPACES.

```

114100/*****
114200/*****
114300**          D E T A L L E S          P A R A          I M P R E S I O N          **
114400**          C O N T .          **
114500/*****
114600 C1 ENCA-CTR-9.
114700 C3 FILLER PIC X(45) VALUE SPACES.
114800 C3 FILLER PIC X(07) VALUE "TOTALES".
114900 C3 FILLER PIC X(06) VALUE SPACES.
115000 C3 FILLER
115100 C1 ENCA-CTR-10.
115200 C3 FILLER PIC X(45) VALUE SPACES.
115300 C3 FILLER PIC X(34) VALUE "PRUMEDIC DE REI
115400 C3 FILLER "NSC. POR ALUMNO :".
115500 C3 DISP-PGRC-INSC-X-ALUM PIC ZZ,99.
115600 C3 FILLER PIC X(49) VALUE SPACES.
115700 C1 ENCA-CTR-11.
115800 C3 FILLER PIC X(45) VALUE SPACES.
115900 C3 FILLER PIC X(34) VALUE "PORCENTAJE DE R
116000 C3 FILLER "EINSC. CORRECTAS :".
116100 C3 DISP-PGRC-INSC-UK PIC ZZ,9982.
116200 C3 FILLER PIC X(47) VALUE SPACES.
116300 C1 TIT-UNAM-DI.
116400 C3 FILLER PIC X(04) VALUE SPACES.
116500 C3 FIA-IMP-DI PIC 9(02)/.
116600 C3 RES-IMP-DI PIC X(03)/.
116700 C3 ANIO-IMP-DI PIC 9(02)/.
116800 C3 FILLER PIC X(52) VALUE SPACES.
116900 C3 FILLER PIC X(74) VA "UNIVERSIDAD NACIO
117000 C3 FILLER "NAL AUTONOMA DE MEXICO".
117100 C3 FILLER PIC X(05) VA "HOJA ".
117200 C3 NUR-HOJA-DI PIC ZZ,ZZ9.
117300 C1 ENCA-DI.
117400 C3 FILLER PIC X(31) VALUE SPACES.
117500 C3 FILLER PIC X(89) VALUE "ALUMNOS QUE VIO
117600 C3 FILLER "LAN EL ART. 27 DEL REGLAMENTO G
117700 C3 FILLER "ENERAL DE INSCRIPCIONES".
117800 C3 FILLER PIC X(32) VALUE SPACES.

```

IV.2.2 Integración

De las dos estrategias señaladas en la sección II.2.3 se prefirió seguir la "integración incremental", es decir, aquella que va añadiendo los nuevos módulos a los ya existentes. Aunque ya se tenía todo el diseño del proceso, se fue probando poco a poco a fin de encontrar los errores que casi inevitablemente siempre se presentan.

Aunque es importante esta fase por permitir encontrar los diversos errores, se considera innecesario presentar los diversos errores que se fueron presentando al momento de hacer la integración y simplemente se presentan a continuación los módulos más importantes ya codificados en su forma definitiva. Estos módulos son los mismos que se presentaron en pseudo-código en la sección IV.1.2.

114566/ PROCEDIME DIVISION USINS NIVEL-ACADEMICO, PLANTEL.

```
114500*****
114700**          M O D U L O      P R I N C I P A L          **
114800**          C O N T R O L   D E   R E I N S C R I P C I O N E S   **
114900**          *****
115000*****
115200 PROGRAM-PRINCIPAL SECCION.
115300 INICIO.
115400
115500**** PROSESJ DE INICIALIZACION *****
115600 PERFORM INICIALIZA-ARCHIVOS
115700 PERFORM INICIALIZA-VARIABLES-GENERALES
115800 PERFORM INICIALIZA-LECTURAS
115900 PERFORM REVISAR-CANTIDAD
116000
116100
116200
116300**** PROSESJ DE REINSCRIPCIONES *****
116400 PERFORM PROSESJ-DE-REINSCRIPCIONES UNTIL 10-MAY-1970
116500
116600
116700
116800**** PROSESJ DE TERMINACION *****
116900 PERFORM LECTURAS-FINALES
117000 PERFORM ACTUALIZA-DATOS-DEL-PLANTEL
117100 PERFORM ESCRIBIR-REPORTES-TOTAL
117200 PERFORM ESCRIBIR-REPORTES-DE-ERRORES
117300 PERFORM QUITAR-CANDADO
117400 PERFORM CIERRE-DE-ARCHIVOS
117500
117600
117700
117800
117900
117800 STOP RUN.
```

```

136000/*****
137100/ **      PROCESO DE REINSCRIPCIONES      ( MODULO II )      **
137200/ **
137300/*****
137300/ PROCESO-DE-REINSCRIPCIONES.
137300/
137700/   PERFORM INICIALIZA-VARIABLES-ALUMNO.
137710/
138300/   IF RG-TIPO-M
138300/     PERFORM FILTRJ-REIM
138700/     IF NO-HAY-ERROR AND REGISTRO-CORRECTA-OK
138800/       MOVJ RG-CUENTA TO M-CUENTA
138900/       MOVE R-ESCUELA TO M-ESCUELA
139000/       MOVE R-DEPAR TO RG-ESCUELA
139100/       MOVE R-NOMPE TO RG-NOMPE
139300/     PERFORM IMPRESION-REGISTRO
139500/     READ LECTORA AT END
139600/     MOVE NO-W TO HAY-DATOS
139700/   ELSE
139900/     MOVE CON-REG-LEIDOS TO CON-REG-ANTERIOR
140000/     PERFORM VERIFICA-CARGA-REG-DETALLE
140100/     UNTIL NO-HAY-REG-DET OR NO-HAY-DATOS
140200/     IF CON-<INSC > ZLRC
140300/       PERFORM REINSCRIBE-ASIGNATURAS VARYING
140400/         I-INS FROM MSG-VA-ASC + 1 BY 1
140500/         UNTIL I-INS > IA-INS CRITAS
140600/       PERFORM ESC-PERS
140700/     ELSE
140800/       IF CON-REG-LEIDOS = CON-REG-ANTERIOR
140900/         MOVE MSG-ERROR (1) TO D-MENSAJE
141000/         WRITE LINEJ FROM DETALLE AFTER 1
141100/         ADD 1 TO TOT-ERR (17)
141200/         MOVE SPACES TO DETALLE
141300/       NOELSE
141400/       ENDIF
141500/     ENDR END
141600/   ELSE
141700/     MOVE RG-ESCUELA TO M-ESCUELA
141800/     PERFORM IMPRESION-REGISTRO
141900/     PERFORM ELIMINA-DETALLE UNTIL NO-HAY-REG-DET OR
142000/       NO-HAY-DATOS
142100/   ENDIF
142200/ ELSE
142300/   MOVE "NO" TO REGISTRO-CORRECTA
142400/   IF RG-TIPO-CK
142500/     MOVE MSG-ERROR (2) TO D-MENSAJE
142600/     PERFORM IMPRESION-REGISTRO
142700/     PERFORM ELIMINA-DETALLE UNTIL NO-HAY-REG-DET OR
142800/       NO-HAY-DATOS
142900/   ELSE
143000/     MOVE MSG-ERROR (3) TO D-MENSAJE
143100/     ADD 1 TO TOT-ERR (1)
143200/     PERFORM IMPRESION-REGISTRO.
143300/   ENDIF
143400/ ENDIF.
143500/
143600/ IF NO-HAY-DATOS OR RG-TIPO-M
143700/   NEXT SENTENCE
143800/ ELSE
143900/   READ LECTORA AT END
144000/   MOVE NO-W TO HAY-DATOS.
144100/ ENDREAD
144200/ ENDIF.
144300/

```

```

142 300 *****
142 700 *****
142 700 ** RUTINA PARA FILTRAR EL REGISTRO TIPO "MAESTRO" MODULO II.A **
142 700 *****
142 700 *****
142 100 FILTRO-PLG-II.
142 300 ** FILTRO DEL NUMERO DE CUENTA ***
142 400
142 700 MOVL SI-H TO REGISTRO-CORRECTA
142 800 IF RG-CUENTA IS NUMERIC
142 900 MOVE 3-CTA TO D-NUM-CTA
142 900 COMPUTE DIG-NVO = (DIG (1) + DIG (3) + DIG (5) + DIG (7))
142 900 * 7 + (DIG (2) + DIG (4) + DIG (6)) * 3
142 900 IF DIG-NVO = DIG (8)
142 100 MOVE RG-CTA TO CUENTA-DISP3
142 200 PERFORM BUSCA-CUENTA
142 300 IF ST-E ST+ = 1
142 400 MOVE R-AP-DNUCTA TO I-PERS
142 500 PERFORM LEE-PERS
142 600 IF R-AJM-OK
142 700 IF R-ES-ALUM
142 800 MOVE R-ESCUELA TO RG-ESCUELA
142 900 IF RG-ESC = CVE-NUM-PLT
142 900 IF NEXT SENTENCE
142 100 ELSE
142 200 MOVE HSG-ERRDR (4) TO D-MENSAGE
142 300 IF ENP OR CCH
142 400 ELSE MOVE NO-W TC REGISTRO-CORRECTA
142 500 ELSE NEXT SENTENCE
142 600 ENDIF
142 700 ELSE
142 800 ENDIF
142 900 ELSE
142 100 ADD 1 TO CON-ERR
142 110 IF R-ES-BAJA
142 120 MOVE 13 TO NUM-ERR(CON-ERR)
142 130 ELSE
142 140 MOVE 14 TO NUM-ERR(CON-ERR)
142 150 ENDIF
142 160 ELSE
142 170 ENDIF
142 180 ELSE
142 190 ADD 1 TO CON-ERR
142 200 MOVE 15 TO NUM-ERR(CON-ERR)
142 210 ENDIF
142 220 ELSE
142 230 ADD 1 TO CON-ERR
142 240 IF R-DCTA-EXAL
142 250 MOVE 16 TO NUM-ERR(CON-ERR)
142 260 ELSE
142 270 MOVE 3 TO NUM-ERR(CON-ERR)
142 280 ENDIF
142 290 ELSE
142 300 ENDIF
142 310 ELSE
142 320 ADD 1 TO CON-ERR
142 330 MOVE 3 TO NUM-ERR(CON-ERR)
142 340 ENDIF
142 350 ELSE
142 360 ADD 1 TO CON-ERR
142 370 MOVE 4 TO NUM-ERR(CON-ERR)
142 380 ENDIF.
142 390 *****

```

```

156300/*****
156400**  RUTINA PARA VERIFICAR Y CARGAR EL REGISTRO TIPO "DETALLE" **
156500**                                "CUCO 11.3" **
156600*****
156700 VERIFICA-CARGA-REG-DETALLE.
156800
156900 MOVE ZERO TO CON-ERR
157000 IF RG-TIPO=OK
157100 IF R3-TIPO=M
157200 MOVE NO-W TO HAY-REG-D
157300 ELSE
157400   ADD 1 TO C04-REG-LEIDOS
157500   IF R1-CUENTA = M-CUENTA
157600     PERFORM FILTRO-REGISTRO-DETALLE
157700     IF NO-HAY-ERROR
157800       PERFORM CARGA-ASIGNATURAS-REINSCRITAS
157900       UNTIL ES-1-A-VEZ
158000       MOVE 1 TO NUM-INSC-CICLO
158100       PERFORM REVISION-COM-REINSCRITAS
158200       VARYING I-INS FROM 1 BY 1 UNTIL
158300         I-INS > IND-INSCRITAS OR CON-ERR > J
158400       MOVE R-A-DBLE-INSC TO A-BLE-DBLE-INS
158500       PERFORM REVIS-2-EN-ORDINARIO
158600       UNTIL A-BLE-COR-E-INSC = J
158700       OR RG-CPC-EXT
158800       OR M-ESC NOT = CVE-NUM-PLT
158900       IF NO-HAY-ERROR
159000         PERFORM GUARDAR-REGISTRO-DETALLE
159100         PERFORM IMPRESION-REGISTRO
159200       ELSE
159300         PERFORM IMPRESION-REGISTRO
159400       ENDIF
159500     ELSE
159600       PERFORM IMPRESION-REGISTRO
159700     ENDIF
159800*   ELSE
159900*     PERFORM IMPRESION-REGISTRO
160000*   ENDIF
160100*   ELSE MOVE NO-W TO HAY-REG-D
160200*   ENDIF
160300* ENDIF
160400* ELSE
160500*   MOVE "CLAVE DE REGISTRO ERRONEO >>>" TO O-MENSAGE
160600*   ADD 1 TO CON-REG-LEIDOS
160700*   ADD 1 TO TOT-ERR (1)
160800*   MOVE "NO" TO REGISTRO-CORRECTA
160900*   PERFORM IMPRESION-REGISTRO-
161000* ENDIF.
161100* IF NO-HAY-REG-DET
161200*   NEXT SENTENCE
161300* ELSE
161400*   READ LECTORA AT END
161500*   MOVE NO-W TO HAY-DATOS.
161600* ENDREAD
161700* ENDIF

```

```

173900 *****
173800 *  RUTINA QUE REVISLA LA REINSCRIPCION CON LOS DATOS DE ENTRADA *
173700 *  "MODULO" II.3.3 *
174000 *****
174100 REVISION-CON-REINSCRITAS.
174200
174300 IF RG-ASIS = T-ASIS(I-INS)
174400 IF K1-GPO-EXT
174500 IF T-GPO-1 (I-INS) = "E"
174600 MOVE T-AP-GPO (I-INS) TO I-GPO
174700 PERFORM LOG-GPO
174800 R-AP-VALAS = 0
174900 MOVE T-GPO (I-INS) TO GPO-YA-INSCRIPC
175000 ADD 1 TO CON-ERR
175100 MOVE 9 TO NUM-ERR (CON-LEPR)
175200 ELSE
175300 MOVE I-GPO-WAIT TO I-GPO
175400 ENDIF
175500 ELSE
175600 NEXT SENTENCE
175700 ENDIF
175800 *
175900 *
176000 *
176100 *
176200 *
176300 *
176400 *
176500 *
176600 *
176700 *
176800 *
176900 *
177000 *
177100 *
177200 *
177300 *
177400 *
177500 *
177600 *
177700 *
177800 *
177900 *
178000 *
178100 *
178200 *
178300 *
178400 *

```


IV.3 PRUEBAS DEL SISTEMA DE REINSCRIPCIONES

A continuación se presentan las dos técnicas señaladas en el capítulo 2 sección II.2.3, con las que se realizaron diversas pruebas.

IV.3.1) Análisis de Entrada-Salida.

Esta metodología permite diseñar casos de pruebas a partir de la especificación escrita del programa. Es así que se puede ir probando una por una cada especificación. Esta técnica ve al programa como una "caja negra", lo que importa son los resultados. Para ello se utiliza un programa del "Sistema de Control Escolar" el cual permite consultar la base de datos de una manera clara y precisa. Esto permite en las pruebas tener por un lado la información original antes de ejecutar el proceso, y una vez ejecutado, observar las modificaciones que se esperaban obtener. Por ejemplo, se consulta la información de la reinscripción de un alumno y el número de alumnos que tienen los grupos a los que se desea reinscribir; después se ejecuta el proceso y se vuelve a consultar la reinscripción del alumno, la cual deberá de contener las nuevas reinscripciones, y el contador de alumnos de los grupos también deberá estar modificado según las reinscripciones hechas en él.

Para poder ejecutar estas pruebas, se precisa haber copiado toda la información del "Sistema de Control Escolar" en cintas magnéticas, a fin de poder modificar todos los archivos sin importar, dado el caso, que se modifiquen erróneamente. Como es difícil que haya un periodo de inactividad, se escoge un fin de semana o un periodo de vacaciones en el cual no se ejecuten tareas. Esto tiene la ventaja de que se cuenta con el sistema tal y como se utiliza normalmente, dándose al mismo tiempo las pruebas funcionales y las pruebas de implementación (Cap.II, sección II.2.3).

La entrada para estas pruebas se escogen de tal manera que haya datos correctos e incorrectos. Los correctos deben, por un lado, ser diagnosticados en los reportes como tales, y por otro lado, deben de actualizar la base de datos correctamente. A los datos incorrectos se les debe de diagnosticar en el reporte el error del que se trata y de ninguna manera pueden modificar al sistema. Se selecciona un caso típico por cada condición que dé origen a una entrada inválida; no probando todas estas condiciones simultáneamente en un solo caso de prueba, sino en varios, pues un error puede originar que no se verifiquen otros tipo de errores.

También otra prueba que indica la confiabilidad del sistema es la que se hace comparando los resultados del sistema original de reinscripciones con el nuevo proceso. Para ello se copia el Sistema de Control Escolar, y por un lado se procesa el programa original de reinscripciones y por otro lado, se procesa el nuevo sistema. Se comparan los resultados a fin de que queden las mismas actualizaciones. Sin embargo, estas pruebas solo son respecto a las funciones que se ejecutaban en el anterior sistema, las implementaciones, por no ser contempladas en el anterior sistema, se prueban de otra manera.

Otro tipo de pruebas consiste en suponer que las especificaciones no son interpretadas correctamente ya sea por accidente o por que dan por presupuesto requisitos que parecen obvios. Es decir, se formulan datos de cada especificación, requiriéndose de preferencia, que el mismo usuario sea quien pida esas pruebas específicas a fin de que quede satisfecho con los resultados obtenidos. Se puede hacer cada especificación por separado y también combinada con otras, tratando abarcar el máximo número de posibilidades que pueden presentarse.

Los requisitos principales que se prueban son:

- Se diseñan entradas para los tres planes de escolaridad: Preparatoria, Colegio de Ciencias y Humanidades y Profesional.
- Se hacen las pruebas en diversas etapas del ciclo escolar.
- Se verifica que no acepte reinscripciones a las asignaturas que hayan sido cursadas en más de dos ocasiones en ordinario.
- Se forman datos que contengan asignaturas de ordinario y de extraordinario.
- Se verifican que las reinscripciones de alumnos a otro plantel del que le corresponden se contabilicen como tales, sólo en el caso de profesional, rechazando la reinscripción para los otros casos.
- Se prueba el caso especial de la Facultad de Medicina, en que sólo se acepta reinscribirse a la misma asignatura si ya aprobó esa asignatura en el grupo antecedente.

IV.3.2) Por cobertura de lógica.

Los casos de prueba que se utilizan se planean a partir del diseño y el código del programa. Es así que se prueba propiamente el código fuente del programa. Y dado que la mayoría de los errores de código ocurre en los puntos de decisión relacionados con las estructuras lógicas de repetición y decisión del programa (23,p.179), se procede a verificar estas instrucciones. Para llevar a cabo estas pruebas ayuda el conceptualizar el programa de cómputo como un conjunto de rutas entre el inicio y la terminación del programa.

El método consiste en formular apropiadamente casos de prueba para invocar, al menos una vez, todas las posibles combinaciones de estado en las condiciones de cada punto de decisión. Es así que conviene ir probando módulo por módulo hasta agotar el programa. Para ello, y dado que el diseño así lo plantea, se construyen tantos datos diferentes como posibilidades de errores o mensajes se puedan presentar.

Los errores presentados en el diseño a veces se presentan en más de una ocasión en el código por lo que se tienen que contemplar todos los casos por los que pudiera recorrer el flujo del programa.

Como se menciona en el diseño, el programa cuenta con rutinas estandar que ya fueron probadas anteriormente, por lo que se puede omitir su verificación.

Por último, antes de dar por visto bueno el sistema, procede a un tipo de prueba que sirve para saber si el progr responde a las diversas entradas como se espera. Para ello modifica el programa invalidando toda escritura a los archivos, es decir, se anulan las instrucciones que contengan algún "WRITE" convirtiéndolas en comentarios mediante un asterisco -para el caso de que se utiliza COBOL-, pudiendo entonces observar los diagnósticos como las "rutas" que siguió el programa p cada caso.

C A P I T U L O V

MANTENIMIENTO DEL SISTEMA DE REINSCRIPCIONES

La tercera y última fase en la constitución de un sistema es la del mantenimiento. Esta fase se caracteriza por cuatro tipos de mantenimiento (Cfr.cap.II.3). El "Mantenimiento Correctivo" se lleva a cabo cuando al ejecutar el proceso se encuentran errores que no se manifestaron durante el período de pruebas y que es preciso corregir. Normalmente estos errores son pequeños, pues los más comunes e importantes ya fueron corregidos en la fase anterior.

Se hace necesario llevar una bitácora en donde queden registrados:

- Los errores corregidos (nombre de rutina, número de instrucción, mostrar la instrucción antes y después de la modificación, etc.)
- El autor de la corrección
- La fecha en que se realizó.

Esta bitácora sirve para remitirse a la historia del programa, sin que queden como "presupuestos" las modificaciones que va sufriendo el sistema a través de su vida.

Tanto el "Mantenimiento Adaptativo" como el anterior se van realizando a través del tiempo en que el proceso se ejecuta, es decir, en un principio no hay nada escrito, sino que conforme a los errores que se presentan o a los cambios que sufre el sistema, tanto en software como en hardware, se van recopilando las notas pertinentes. Por ejemplo, cambios de computadora, cambios de sistema operativo, cambios en instrucciones del lenguaje de COBOL, etc. Normalmente los cambios externos al sistema que se llevan a cabo son en vistas a un mejoramiento, por lo que habrá que llevar a cabo, sin temor, las modificaciones necesarias.

El "Mantenimiento Perfectivo" corresponde a las nuevas aplicaciones que se le introducen al sistema. Para ello se hace imprescindible recurrir tanto al "Reporte para la Especificación de los Requerimientos de Software" como al reporte de la "Especificación del Diseño", en donde se tiene que anexar las nuevas funciones o modificaciones introducidas. Como todo sistema, una vez que funciona regularmente, se van descubriendo nuevas aplicaciones o la conveniencia de pequeñas modificaciones que no alteran sustancialmente los objetivos iniciales pero que, sin embargo, es necesario recabar por escrito, aún por insignificante que parezcan dichas modificaciones.

El "Mantenimiento Preventivo" se efectúa mirando a futuro, es decir, se pueden definir normas para el mantenimiento en base a la experiencia adquirida en mantener el presente sistema y previendo posibles cambios que pudieran presentarse. Por ejemplo, el cambio de la entrada de los datos, o el cambio de computadora, o prever la creación de nuevas unidades o plantelas, nuevas carreras, o incluso, su aplicación a otras universidades.

Muchas veces, debido a la cantidad de tareas por ejecutarse en un departamento de cómputo, se le da menos prioridad al mantenimiento preventivo, dejando las nuevas funciones o modificaciones que se presenten para el momento en que se requiera, remitiendo estas actualizaciones al "Mantenimiento Perfectivo".

El principal documento que condensa la fase del mantenimiento de un sistema es el "Manual del usuario", el cual es un instructivo para instalar, operar y mantener el sistema (Cfr. cap.II.3.3). Este manual viene a complementar la documentación anterior y es, a la vez, el más recurrido. Ello es debido a que los otros dos reportes son para consultar especificaciones internas del sistema y no su operación. A continuación se presenta dicho manual con sus tres partes quedando definido el mantenimiento.

No hay que olvidar, sin embargo, que un sistema, para que continúe "vivo", nunca se termina de procurarle los diversos tipos de mantenimiento.

MANUAL DEL USUARIO
SISTEMA DE REINSCRIPCIONES EN LA UNAM

INDICE

I) Guía de Instalación

- I.1 Equipo y programación necesaria
- I.2 Procedimiento de instalación
- I.3 Descripción de las pruebas

II) Guía de operación

- II.1 Panorama general
- II.2 Descripción de sus interfaces
 - a) Entradas al proceso
 - b) Salidas del proceso
- II.3 Procedimiento de ejecución
 - a) Modo Directo
 - b) Modo Indirecto
- II.4 Glosario de términos

III) Guía de mantenimiento

- III.1 Documentación sobre el sistema
- III.2 Recomendaciones para su uso

Apéndices

- A. Formas para la captura de los datos de entrada.
 - A.1 Formato general
 - A.2 Hoja para la lectura óptica de los datos de entrada.
- B. Diagnósticos del sistema
 - B.1 Reporte general
 - B.2 Reporte resumen
 - B.3 Reporte resumen sobre los errores ocurridos en la ejecución.
 - B.4 Bitácora del sistema operativo mostrando una ejecución del proceso.
- C. Relación de los planteles con sus claves.
- D. Formas para la modificación a la reinscripción.
- E. Listado del programa de reinscripciones.

I. Guía de Instalación

I.1 Equipo y programación necesarios.

El Sistema de Reinscripciones fué desarrollado por la Unidad de Registro y Control Escolar bajo la supervisión de la Subdirección Técnica de la Coordinación de la Administración Escolar. Fué desarrollado para ser utilizado en la "Dirección de Cómputo para la Administración Académica", la cual cuenta con una computadora D-6800 con sus periféricos, que en 1986 tenía la configuración de la figura V.1. Para la instalación del sistema se requiere que se tenga el programa que reinscribe en la unidad de disco junto a la base de datos del Sistema Escolar. El programa que lleva las reinscripciones recibe el nombre de RE/C/O/REINSCRIBE/MIDI, en donde las siglas significan:

RE	:	Registro Escolar
G	:	General: para tres planes de escolaridad
O	:	Es el programa objeto, ya compilado
REINSCRIBE	:	Indica la función del programa.
MIDI	:	Hace referencia a que recibe registros tipo Maestro y tipo Detalle

El Sistema requiere para su funcionamiento que estén presentes varios archivos, tanto para leer como para escribir en ellos:

- 1) P/E/*/1 : Información sobre los alumnos.
- 2) P/E/*/2 : Información sobre las carreras, asignaturas y grupos.
- 3) P/E/*/3 : Información sobre los planteles.
- 4) P/E/*/4 : Contiene las "celdas" que relacionan los diversos archivos.
- 5) P/E/*/7 : Contiene los apuntadores al archivo de los alumnos para utilizar la dispersión del número de cuenta.
- 6) P/E*/LIMITES : Contiene las cifras que marcan el límite físico de los archivos.
- 7) P/E*/DCBLE/INSC : Contiene las asignaturas que han sido cursadas dos o más veces en ordinario.
- 8) El archivo de datos, cuyo nombre es variable y se asigna al momento de ejecutar el proceso.

El nombre de los archivos contiene un "*" que significa: "P" para profesional, "E" para preparatoria y "C" para el Colegio de Ciencias y Humanidades.

I. Guía de Instalación

I.1 Equipo y programación necesarios.

El Sistema de Reinscripciones fué desarrollado por la Unidad de Registro y Control Escolar bajo la supervisión de la Subdirección Técnica de la Coordinación de la Administración Escolar. Fué desarrollado para ser utilizado en la "Dirección de Cómputo para la Administración Académica", la cual cuenta con una computadora D-6800 con sus periféricos, que en 1986 tenía la configuración de la figura V.1. Para la instalación del sistema se requiere que se tenga el programa que reinscribe en la unidad de disco junto a la base de datos del Sistema Escolar. El programa que lleva las reinscripciones recibe el nombre de RE/G/O/REINSCRIBE/MIMI, en donde las siglas significan:

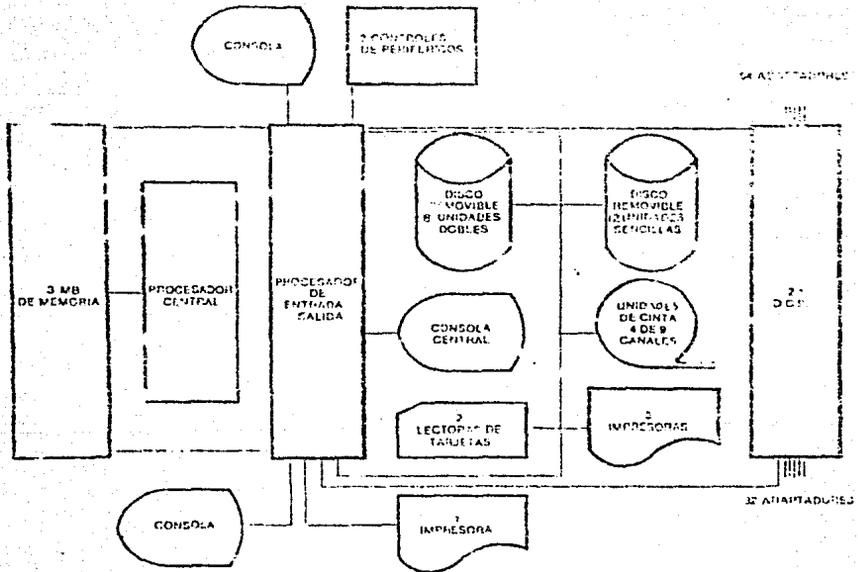
RE : Registro Escolar
G : General: para tres planes de escolaridad
O : Es el programa objeto, ya compilado
REINSCRIBE : Indica la función del programa.
MIMI : Hace referencia a que recibe registros tipo Maestro y tipo Detalle

El Sistema requiere para su funcionamiento que estén presentes varios archivos, tanto para leer como para escribir en ellos:

- 1) P/E/*/1 : Información sobre los alumnos.
- 2) P/E/*/2 : Información sobre las carreras, asignaturas y grupos.
- 3) P/E/*/3 : Información sobre los planteles.
- 4) P/E/*/4 : Contiene las "celdas" que relacionan los diversos archivos.
- 5) P/E/*/7 : Contiene los apuntadores al archivo de los alumnos para utilizar la dispersión del número de cuenta.
- 6) P/E/*/LIMITES : Contiene las cifras que marcan el límite físico de los archivos.
- 7) P/E/*/DCBLE/INSC : Contiene las asignaturas que han sido cursadas dos o más veces en ordinario.
- 8) El archivo de datos, cuyo nombre es variable y se asigna al momento de ejecutar el proceso.

El nombre de los archivos contiene un "*" que significa: "P" para profesional, "E" para preparatoria y "C" para el Colegio de Ciencias y Humanidades.

CONFIGURACION DEL EQUIPO B-6300



Configuración de la B-6300
Figura V.1

I.2 Procedimiento de instalación.

tanto los archivos que se utilizan junto al sistema, como el mismo programa de reinscripciones se almacenan en cintas magnéticas, sirviendo de protección, pues en caso de cualquier error o falla del equipo se cuenta con una copia que debe estar actualizada. Por ello es recomendable tener un calendario de protecciones para que en cualquier momento dado se pueda restablecer el sistema sin graves pérdidas.

El programa de reinscripciones se compila guardándose el programa objeto resultante. Para su utilización se procede a tener presente todos los archivos poniéndolos de las cintas magnéticas al disco de la computadora. No obstante, siendo muy requerida la utilización del programa, conviene tenerlo presente permanentemente en el área asignada en disco para el Sistema de Control Escolar.

La instalación del sistema no conlleva grandes modificaciones puesto que su desarrollo y sus pruebas fueron ejecutadas en la misma computadora B-6300, donde se lleva a cabo la ejecución. Se utilizan los mismos periféricos y dispositivos de entrada y salida. Es así que en cuanto queda completa la fase de pruebas, prácticamente queda instalado el sistema.

I.3 Descripción de las pruebas

Las pruebas realizadas y que constatan su correcta instalación fueron realizadas en la fase del desarrollo. Ahora, en el momento de la ejecución, ya no es preciso volverlas a hacer. Mas si se desean, tanto para volver a probar el sistema como para saber si ya se probaron determinadas situaciones se puede remitir a la fase anterior.

En dado caso que se desee probar al sistema de nuevo, se puede servir del programa que se utilizó para probar los diferentes diagnósticos. Este programa es en todo igual al definitivo, excepto en unos cambios respecto a las instrucciones de escritura a los archivos, las cuales se omiten para no modificar el sistema. Las escrituras a los archivos de la impresora se conservan, y gracias a los diagnósticos emitidos, es fácil observar si el proceso se llevará a cabo correctamente. Ello puede servir para saber si la información de entrada viene con el formato adecuado, sin la necesidad de interrumpir las diferentes tareas con accesos a los archivos. El nombre del programa es: RE/C/O/REINSCRIBE/MIDI/FILTRO, y la manera de ejecutarlo es la misma que se indica para un proceso

II. Guía de operación

II.1) Panorama general

A) Objetivo

La finalidad del Sistema de Reinscripciones es llevar a cabo la incorporación del alumno a un plantel-asignatura-grupo dentro del sistema computarizado de la Coordinación de la Administración Escolar (CAE). Esta reinscripción deberá poder ser realizada en los tres tipos de escolaridad: Colegio de Ciencias y Humanidades, preparatoria y profesional.

B) Generalidades

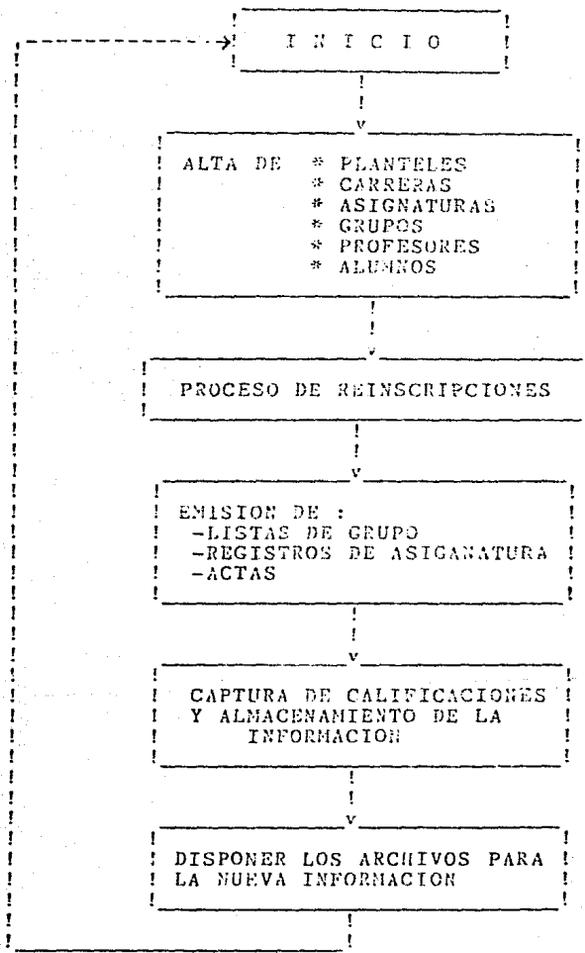
Este sistema es una parte del conjunto de tareas que se llevan a cabo en un periodo escolar. Generalmente es de las primeras funciones que se realizan antecediendo sólo por el proceso que da de alta los nuevos grupos, asignaturas, planteles y/o alumnos. Sin embargo este proceso se puede ejecutar en cualquier momento, y de hecho, las reinscripciones a las asignaturas en grupos de extraordinario se lleva a cabo hasta al final del periodo escolar. En la figura V.2 se muestra la ubicación del sistema durante el periodo escolar.

C) Requisitos que contempla.

- + Impide la reinscripción a las asignaturas doblemente reinscritas:
- + El sistema puede reinscribir tanto en grupos de ordinario como de extraordinario:
- + Es posible realizar la reinscripción a otro plantel del correspondiente al del alumno:
- + Para la Facultad de Medicina se puede llevar a cabo la reinscripción, aún en el caso de que esté reinscrito en la misma asignatura siempre y cuando haya aprobado el curso antecedente.

D) Especificaciones.

- + Lenguaje : COBOL
- + Líneas de código : 3100
- + Número de rutinas : 30
- + Sistema operativo : versión 3.4



Ubicación del proceso de reinscripciones dentro del Sistema de Control Escolar en un periodo escolar
 Figura V.1

II.2 Descripción de sus interfaces.

Como se apunta en el "Reporte de Especificaciones" el sistema consta de tres tipos de comunicación con el exterior:

A) Entradas al proceso

Los datos de entrada al sistema deberán estar también en el disco de la computadora. Sin embargo, no importa para el proceso el origen de los datos, éstos pueden venir de tarjetas perforadas, hojas de lectura óptica, cinta magnética, etc. Lo que importa es que vengan bajo el formato preciso. Dicho formato, con sus características se presenta en el apéndice A.1. Sin embargo, la forma de captura más generalizada y que tiene gran confiabilidad y rapidez es la hoja para lectura óptica (apéndice A.2), que el mismo alumno llena a lápiz con los datos deseados para su reinscripción. Lo que hace años se llevaba a cabo mediante tarjetas perforadas se ha ido actualizando y mejorando mediante estas hojas, hasta que hoy en día es raro hacer uso de dichas tarjetas.

B) Salidas del proceso

Son cuatro tipos de reportes con el diagnóstico del proceso los que emite el sistema:

B.1) DIAGNOSTICO GENERAL:

Muestra movimiento por movimiento el resultado del proceso. Para ello muestra todos los principales datos del alumno (nombre, número de cuenta, plantel, carrera), así como los correspondientes a su reinscripción (clave y nombre de la asignatura y el grupo). En caso de ocurrir un error durante la reinscripción, se indica de que tipo se trata. El anexo B.1 presenta una muestra de este diagnóstico.

B.2) DIAGNOSTICO ESTADISTICO DEL PROCESO:

Presenta las cifras totales del proceso tanto para las reinscripciones realizadas por alumnos del mismo plantel como para los alumnos de otro plantel (apéndice B.2). En dicho reporte se indica lo siguiente:

- + El número total de reinscripciones correctas e incorrectas
- + El número de "Registros Maestro" correctos e incorrectos

- + El número de "Registros Detalle" correctos e incorrectos
- + Porcentaje de reinscripciones correctas
- + Promedio de reinscripciones por alumno

B.3) DIAGNOSTICO ESTADISTICO SOBRE ERRORES:

Muestra el diagnóstico con el resumen del número de ocurrencias de cada tipo de error que se presentó durante el proceso. En el apéndice B.3 se muestra un ejemplo.

B.4) DIAGNOSTICO ESPECIFICO:

Señala cada uno de los alumnos que violan el art. 27 del "Reglamento General de Inscripciones", es decir, aquellos que dejaron reinscribirse a una asignatura que ya la habían cursado en ordinario en dos o más ocasiones.

De los tres últimos reportes, se emiten dos diagnósticos exactamente iguales, uno se requiere en el plantel correspondiente del proceso y el otro reporte se necesita como parte de la bitácora de los procesos llevados a cabo en el departamento.

También es importante la información que ofrece la bitácora del sistema operativo. Sin embargo, esta información es de uso más restringido y no interesa a los planteles, sino al mismo departamento de cómputo a fin de ir aprovechando mejor los recursos (apéndice B.4). Este listado muestra, por un lado, el "Diagnóstico Estadístico del proceso", y por otro, la información referente a los recursos utilizados durante el proceso. La información más útil es la que indica los siguientes datos:

- El tiempo de procesador
- El tiempo de utilizar recursos de entrada/salida
- La hora de inicio y fin de la tarea
- Las líneas impresas
- La memoria utilizada
- Las anomalías, si las hubo, durante el transcurso del proceso
- Los mensajes que se mandan escribir desde el programa

II.3 Procedimiento de ejecución

A) Modo de hacerlo

El programa de reinscripciones se puede poner en ejecución de dos maneras: ya sea mediante el proceso interactivo por terminal (modo directo), o mediante el modo indirecto llamado proceso "por colas" (Batch). Para las dos maneras se utilizan comandos del "Lenguaje para flujos de trabajos" (Work-Flow Language o mejor conocido como "WFL"):

A.1) MODO DIRECTO:

Desde una terminal se manda a la computadora la instrucción de ejecutar el proceso, indicando el archivo de datos de entrada:

```
WFL RUN RE/G/O/REINSCRIBE/MIDI ("FAM",PLANTEL)
FILE LECTORA(KIND=DISK,
FILETYPE=7,
TITLE=RE/D/REINSCRIBE/PLANTEL/FECNA)
```

donde:

+ "FAM" : indica el tipo de nivel de estudios de las reinscripciones que se van a procesar, constando este parámetro de tres caracteres alfabéticos en que sólo se reconocen tres tipos:

"PRO" para nivel profesional
"ENP" para la Escuela Nacional Preparatoria
"CCH" para el Colegio de Ciencias y Humanidades

+ PLANTEL : indica la clave de plantel al cual se desean hacer las reinscripciones. En el apéndice C.1 se encuentra la relación de planteles con sus claves a los cuales el sistema permite reinscribir.

Como se observa en el ejemplo de la instrucción, los datos de entrada están asignados a disco -KIND=DISK-, siendo ello lo más común si se tiene en cuenta la enorme cantidad de reinscripciones que se procesan regularmente. Aunque es posible introducir los datos directamente por terminal, asignando el archivo de datos a la misma terminal -KIND=RENOTE-, no resulta práctico ni recomendable, pues no hay manera de comprobar la corrección de los datos de entrada una vez procesado el sistema. Al ejecutarse esta instrucción se tiene la desventaja de "bloquear" la terminal, es decir, impide el manejo de la terminal para otras tareas. Para ello está el "Modo Indirecto".

A.2) MODO INDIRECTO

Esta es la manera más común y recomendable para ejecutar el proceso, pues aparte de no interrumpir la utilización de la terminal, se pueden precisar los recursos a utilizar: prioridad en el sistema operativo, la oportunidad de enviar varios procesos que se ejecuten uno después de otro sin la continua intervención de alguien, y sobre todo la facilidad de poder recomenzar en caso de presentarse una falla en el sistema. El modo de hacerlo es preparar un pequeño archivo con instrucciones de WFL (el archivo es tipo "JOB"), y "encolarlas" (con la instrucción START) en la lista de espera que tiene el sistema operativo. Un ejemplo de dicho archivo es el siguiente:

```
JOB "REINSCRIPCION" BEGIN
```

```
CLASS = 50
```

```
TASK = T1, T2, ..., Tn
```

(*Se etiquetan las tareas a realizar en el JOB*)

```
RUN RE/G/O/T1 (T1)
```

```
RUN RE/G/O/T2 (T2)
```

```
.....
```

```
.....
```

```
RUN RE/G/O/REINSCRIBE/MIDI ("EMP", 021); (TN)
```

```
FILE LECTORA (KIND=DISK,
```

```
FILETYPE=7,
```

```
TITLE=RE/G/D/REINSCRIBE/190327);
```

```
.....
```

```
END JOB
```

III) Guía de Mantenimiento

III.1 Documentación sobre el sistema.

En base a la división establecida para la creación de un sistema, es muy conveniente que de cada fase del proceso, se emita un reporte. En caso de requerir hacer cualquier tipo de corrección, extensión o adaptación, se debe consultar los diversos tipos de documentación que se han emitido, y apuntar en ellos lo realizado. Los tres reportes que tiene el Sistema de Reinscripción son los siguientes:

* FASE DE PLANEACION:

"Reporte para la especificación de los requerimientos de software para el Sistema de Reinscripciones en la UNAM".

* FASE DE DESARROLLO:

"Especificación del Diseño para el Sistema de Reinscripciones en la UNAM".

* FASE DE MANTENIMIENTO:

"Manual del Usuario para el Sistema de Reinscripciones en la Unam".

El "Manual del usuario" (el presente documento), incluye el listado del programa y una bitácora que se actualiza cada vez que se realiza una modificación al sistema o simplemente para apuntar recomendaciones para mejorar el sistema. Esta parte del manual se revisa periódicamente a fin de ir previendo las diversos tipos de mantenimiento. El controlar las diferentes versiones del código, redundará en beneficio inmediato para su usuario.

III.2 Modificaciones al Sistema

Es común el caso de que por un error al capturar los datos de entrada o porque un alumno se equivoque al solicitar un grupo que no desca, o por alguna otra causa, se requiera modificar la reinscripción. Cuando ello sucede, existe un sistema muy afín denominado "Sistema de Modificación a la Reinscripción". Lo que hace es ejecutar altas, bajas y (o) cambios de la reinscripción. El programa que realiza las modificaciones se denomina RE/G/O/FBCOL y utiliza como datos de entrada unas formas especiales denominadas FBCOL, en donde se especifica la información necesaria para realizar la modificación. En el apéndice E se muestra una de estas formas. Para mayor información sobre este sistema conviene consultar los reportes específicos.

III.3 Bitácora sobre el mantenimiento

Todas las modificaciones que se realicen en el Sistema de Reinscripciones han de ser puestas por escrito en una bitácora que contenga la información necesaria para acudir a ella en caso de comprobación o de simple consulta. La información mínima indispensable sobre las modificaciones son:

- Nombre del programa
- Nombre del departamento encargado
- Nombre de la persona que la ejecuta
- Fecha de inicio
- Fecha en que se entrega el sistema ya modificado
- Tipo de mantenimiento: Adaptativo, Perfectivo o Prevent
- Módulos afectados
- Líneas de código afectadas
- Finalidad que se persigue al hacer la modificación
- Observaciones

De preferencia, estas modificaciones deben de ser confrontadas con los tres tipos de documentación, a fin de poder contemplar el problema con una visión más general y de dar una solución óptima. Si la modificación es muy grande conviene detallarla en los tres reportes. En caso de ser menor, basta que se indique en el "Manual del Usuario", y puede también indicarse en el mismo programa fuente, señalándolo en el lugar de la modificación mediante algún comentario.

10/MAR/87

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
 SUPLECIÓN TÉCNICA
 DIAGNÓSTICO DE REINSCRIPCIÓN
 E0163 FAC DE MED VETERINARIA Y ZOOTEC.

PÁG. 10



RESERVACIONES	NUMERO CJA	CVE POV	PLI	CARR	ASIG	GPC	NOMBRE	ASIGNATURA	CUENTA	D	NOMBRE	ALUMNO
		DI	016	21	0061	100	TRENCA	ZOOTECNICA				
		DI	016	21	0062	100	AGRICULTURA					
		DI	016	21	0063	100	AGRICULTURA					
	2,336	MI	016	21	0667	100	CLINICA	POCINA	740137	-9	ALMEIDA	MUCHI ALONSO LECTER
		MI	016	21	0668	100	CLINICA	CAMINA				
		MI	016	21	0669	100	ZOOTECNICA	DE LAS AVES				
		MI	016	21	0608	100	ADMINI	DE EMPRESAS AGROPEC				
	417	MI	016	21	0667	100	INMUNOLOGIA		740143	-3	CASTRO	VARGUEZ MIGUEL YVEL
		MI	016	21	0668	100	CLINICA	CLINIC				
		MI	016	21	0669	100	ZOOTECNICA	DE LAS AVES				
		MI	016	21	0206	100	ADMINI	DE EMPRESAS AGROPEC				
	1,422	MI	016	21	0667	100	VIRULO	DE ENFERMEDADES VIRAL	740173	-2	AREOLA	JIMENEZ JOSE MARIE
		MI	016	21	0668	100	SELECCION	DE PLANTAS VETERINARIA				
		MI	016	21	0669	100	ADMINI	DE EMPRESAS AGROPEC				
		MI	016	21	0442	100	ZOOTECNICA	DE LAS AVES				
		MI	016	21	0127	100	ECONOMIA	ZOOTECNICA				
	321	MI	016	21	0282	100	INMUNOLOGIA		740240	-1	ERAY	ARANDA ENRIQUE
	505	MI	016	21	0667	100	CLINICA	DE LAS AVES	740255	-7	BARRIOS	ARTEAGA MA CRUZ
		MI	016	21	0668	100	CLINICA	DE LAS AVES				
		MI	016	21	0127	100	ECONOMIA	ZOOTECNICA				
		MI	016	21	0667	100	CLINICA	DE ANIMALES DE ZOOLOGIC				
		MI	016	21	0668	100	CLINICA	POCINA				
	907	MI	016	21	0165	100	FISIOLGIA	VETERINARIA	740294	-4	BAUTISTA	OLGA RUFFA
		MI	016	21	0166	100	INMUNOLOGIA					
		MI	016	21	0167	100	ADMINI	DE EMPRESAS AGROPEC				
	401	MI	016	21	0442	100	PATOLGIA	ESPECIAL	740283	-2	EFACAMONTE	MORANDEZ FELISA
		MI	016	21	0667	100	CLINICA	DE LAS AVES				
		MI	016	21	0668	100	CLINICA	DE LAS AVES				
		MI	016	21	0669	100	ADMINI	DE EMPRESAS AGROPEC				
	2,992	MI	016	21	0604	100	AGRICULTURA		740325	-5	CANALES	FIGUEROA ANTONIO

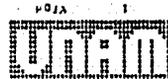
ASIGNATURA NO NUMERICA
INSCRIPCION NO PROCEDE >>>>

ASIGNATURA NO NUMERICA
INSCRIPCION NO PROCEDE >>>>

Apéndice 3.3 : Reporte estadístico sobre los errores
ocurridos

10/04/77

UNIVERSIDAD NACIONAL DE INGENIERIA
SUBDIRECCION TECNICA
RELACION DE ERRORES
OCURRENCIAS NOBRE DEL ERROR



5	LLAVE DE REGISTRO ERRONEO
5	LLAVE PLANTEL NO NUMERICO
5	CLAVE PLANTEL NO EXISTE
10	GRUPO YA TIENE ACTA
11	NUMERO DE CUENTA NO NUMERICO
7	NUM. DE CUENTA INCORRECTO
5	NUM. DE CUENTA NO EXISTE
30	ASIGNATURA NO NUMERICA
51	GRUPO NO EXISTE
15	ASIGNATURA YA INSCRITA
4	NO CORRESPONDE EL PLANTEL
35	ASIGNATURA NO EXISTE
40	GRUPO INCORRECTO

Apéndice C : Relación de plantelas con su clave

*** PROFESIONAL ***

UNIDAD: CIUDAD UNIVERSITARIA

001	FACULTAD DE ARQUITECTURA
002	ESCUELA NACIONAL DE ARTES PLASTICAS
003	FACULTAD DE CIENCIAS
004	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES
005	FACULTAD DE QUIMICA
006	FACULTAD DE CONTADURIA Y ADMINISTRACION
007	FACULTAD DE DERECHO
008	FACULTAD DE ECONOMIA
009	ESCUELA NACIONAL DE ENFERMERIA Y OBSTETRICIA
010	FACULTAD DE FILOSOFIA Y LETRAS
011	FACULTAD DE INGENIERIA
012	FACULTAD DE MEDICINA
013	ESCUELA NACIONAL DE MUSICA
014	FACULTAD DE ODONTOLOGIA
015	ESCUELA NACIONAL DE TRABAJO SOCIAL
016	FACULTAD DE MEDICINA VETERINARIA Y ZOOCTECIA
019	FACULTAD DE PSICOLOGIA

UNIDAD: CUAUTITLAN

105	FACULTAD DE QUIMICA
106	FACULTAD DE CONTADURIA
111	FACULTAD DE INGENIERIA
116	FACULTAD DE VETERINARIA
118	FACULTAD DE AGRICULTURA

UNIDAD: ACATLAN

201	ESCUELA DE ARQUITECTURA
203	ESCUELA DE ACTUARIA
204	ESCUELA DE CIENCIAS POLITICAS
207	ESCUELA DE DERECHO
208	ESCUELA DE ECONOMIA
210	ESCUELA DE FILOSOFIA
211	ESCUELA DE INGENIERIA
240	ESCUELA DE COMPUTACION

UNIDAD: IZTACALA

303	ESCUELA DE BIOLOGIA
309	ESCUELA DE ENFERMERIA
312	ESCUELA DE MEDICINA
314	ESCUELA DE ODONTOLOGIA
319	ESCUELA DE PSICOLOGIA

UNIDAD: ARAGON

401	ESCUELA DE ARQUITECTURA
404	ESCUELA DE CIENCIAS POLITICAS
407	ESCUELA DE DERECHO
408	ESCUELA DE ECONOMIA
410	ESCUELA DE FILOSOFIA
411	ESCUELA DE INGENIERIA
420	ESCUELA AGROPECUARIA

UNIDAD: ZARAGOZA

503	ESCUELA DE BIOLOGIA
505	ESCUELA DE QUIMICA
509	ESCUELA DE ENFERMERIA
512	ESCUELA DE MEDICINA
514	ESCUELA DE GONTOLOGIA
519	ESCUELA DE PSICOLOGIA

*** ESCUELA NACIONAL PREPARATORIA ***

021	PLANTEL 1 : GABINO BARREDA
022	PLANTEL 2 : ERASMO CASTELLANOS
023	PLANTEL 3 : JUSTO SIERRA
024	PLANTEL 4 : VIDAL CASTAÑEDA Y N.
025	PLANTEL 5 : JOSE VASCONCELOS
026	PLANTEL 6 : ANTONIO CASO
027	PLANTEL 7 : EZEQUIEL A. CHAVEZ
028	PLANTEL 8 : MIGUEL E. SCHULTZ
029	PLANTEL 9 : PEDRO DE ALBA

*** COLEGIO DE CIENCIAS Y HUMANIDADES ***

031	PLANTEL PARQUE VIA
032	PLANTEL NAUCALPAN
033	PLANTEL VALLEJO
034	PLANTEL ORIENTE
035	PLANTEL SUR

CONCLUSIONES

En el presente trabajo se cumplen dos objetivos: Primero, el análisis de una metodología para el desarrollo de sistemas de computadora (capítulo II), y segundo, la aplicación de dicha metodología, consistente en tres fases, a un problema concreto y real como lo es la reinscripción en la Universidad Nacional Autónoma de México (capítulos III, IV y V). Es en este segundo objetivo donde se constata la necesidad de contar con toda una infraestructura en el desarrollo de proyectos en el ámbito computacional. Si bien, se aplicó la teoría de la Ingeniería de Software a un sólo caso específico, es fácil inferir en base a lo presentado en la literatura utilizada, que se puede utilizar como teoría en cualquier tipo de proyecto computacional. Se recomienda sobre todo su utilización en las diversas partes del mismo Sistema de Control Escolar.

En cualquier sistema computacional se hace imprescindible el recurso a una metodología para el desarrollo de sus proyectos. No sólo se evitan pérdidas de recursos sino que se obtienen considerables ventajas, entre ellas:

- Evitar un sobre costo en el desarrollo del proyecto
- Obtener un producto eficiente y confiable
- Cumplir con un programa de actividades
- Tener bajo control todo el proyecto, independientemente de los creadores del sistema.

Sucede en muchos casos que al terminar un proyecto, no hay nada escrito sobre él y puesto que ya se está ejecutando, se prefiere empezar con un nuevo proyecto que dejarlo terminado, provocando con ello un descontrol sobre el sistema. Al adoptar las técnicas establecidas, bien pudiera parecer que se "complica" el sistema y se retrasa ante la necesidad de ir registrando cada fase, sin embargo, no es así, pues al ir elaborando reportes de su planeación y desarrollo, una vez que se termina el proyecto, queda todo perfectamente documentado. Además, una vez que se ha utilizado por primera vez cierta metodología, se tiene la experiencia para las demás ocasiones, facilitando su elaboración.

Vale la pena seguir un orden en la elaboración de cualquier tarea, pues por molesto que resulte, ayuda a perfeccionar tanto al trabajo elaborado como al ejecutor de él.

Bibliografía (Referencias)

- (1) Shoeman, Martin L.,
"Software Engineering",
International Student Edition
McGraw-Hill International Book Co.,
2a edición, 1985
- (2) Pressman, Roger S.,
"Software Engineering: a practitioner's approach",
International Student Edition,
McGraw-Hill International Book Co.,
3a edición, 1984
- (3) Gemignani, M.,
"Law and the Computer", CDI Publishing Company,
Boston, 1982
- (4) Sukert, A., and A. Goel,
"Error Modeling Applications in Software Quality
Assurance" en "Proceeding of the Software Quality and
Assurance Workshop", ACM,
San Diego, CA, Noviembre 1978, pp. 33-38
- (5) Hills, H. D.,
"On the Statistical Validation of Computer Programs",
FSC 72:6015, IBM Federal Systems Division, 1972
- (6) Musa, J.,
"Software Reliability Measurement", en "The Journal
of Systems and Software", vol. 1, 1980, pp.223-241
- (7) Putnam, L.,
"A General Empirical Solution to the Macro Software
Sizing and Estimating Project", en IEEE:
"Transactions on Software Engineering", vol. 4,
No.4, 1978, pp.345-361
- (8) Esterling, R.,
"Software Manpower Costs: A Model", Datamation, Marzo
1980, pp.164-170
- (9) Putnam, L.,
"Software Cost Estimating and life Cycle Control", en
IEEE Computer Society Press, 1980, pp.324-328
- (10) Basili, V.,
"Models and Metrics for Software Management and
Engineering", en IEEE Computer Society Press,
1980, pp. 4-9
- (11) Norden, P.,
"Useful Tools for Project Management", en Putnam,
Ibid, pp.216-225

- (12) Riggs, J.,
"Production Systems Planning, Analysis and Control,
3a edición, Wiley, 1981, pp. 205-223
- (13) Wiest, J., and F. Levy,
"A Management Guide to PERT/CPM", 2a edición, ed.
Prentice-Hall, 1977
- (14) Putnam, L., Ibid, p.15
- (15) Yourdon E. and Constantine, L.L.,
"Structured Design", Yourdon Press, New York, 1975
- (16) Yourdon, E.,
"How to Manage Structured Programming",
New York, Yourdon Inc., 1976
- (17) Yourdon, Edward,
"Techniques of Program Structure and Design",
ed. Prentice-Hall, Englewood Cliffs, N.J.,1979
- (18) De Marco, T.,
"Structured Analysis and System Specification",
ed. Prentice-Hall, 1979
- (19) Taylor, E. S.,
"An Interim Report on Engineering Design",
Massachusetts Institute of Technology, Cambridge,
Massachusetts, 1959
- (20) Dijkstra, E. W.,
"A Discipline of Programming",
Prentice-Hall, 1976
- (21) Wirth, N.,
"Program Development by Stepwise Refinement",
Communications of the ACM,
Abril 1971
- (22) Weinberg, G.,
"The Psychology of Computer Programming",
Van Nostrand, 1971
- (23) Gerez V., Mier H., Nieva R., Rodríguez G.,
"Desarrollo y Administración de Programas de
Computadora (Software)",
Compañía Editorial Continental, S.A., México,
Segunda impresión,
Septiembre de 1985
- (24) Kernighan, B., y P. Plauger,
"The Elements of Programming Style",
McGraw-Hill, 1974

- (25) Shneiderman, B.,
"Software Psychology",
Winthrop Publishers, 1980
- (26) Wasserman, A.,
"User Software Engineering and the Design of
Interactive Systems", en "Proceedings of the Fifth
International Conference on Software Engineering",
IEEE, San Diego, CA, Marzo 1981, pp. 387-393
- (27) Myers, G.,
"The Art of Software Testing",
Wiley, 1979
- (28) Canning, R.,
"The Maintenance 'Iceberg' ",
EDP Analyser, vol. 10, No.10, Octubre, 1972
- (29) Rochkind, H.,
"The Source Code Control System",
IEEE Transactions on Software Engineering,
vol.1, No.4, Diciembre 1975, pp. 364-370
- (30) Swanson, E. B.,
"The Dimensions of Maintenance", Proceedings of the
Second International Conference on Software
Engineering, IEEE, Octubre 1976, pp.492-497
- (31) Lientz, B. y E. Swanson,
"Software Maintenance Management",
Addison-Wesley, 1980
- (32) Freedman, D. y G. Weinberg,
"Techniques of Program and System Maintenance",
G. Parikh, Winthrop Publishers, 1981
- (33) Wirth, N.,
"On The Composition of Well Structured Programs",
Computing Surveys, Diciembre 1974
- (34) Bohm, C. y Jacopini, G.,
"Flow Diagram, Turing Machines and Languages with two
Formation Rules", Communications of the ACM,
Mayo, 1966, pp. 366-371
- (35) "Improved Programming Technologies, an overview",
International Business Machines Corp.,
Technical publications, 1974
- (36) Jensen R. W. y Tonies C.C.,
"Software Engineering",
Prentice-Hall, 1979

- (37) Kerningham, B.W. & Plauger, P.J.,
"The Elements of Programming Style",
McGraw-Hill, 1974
- (38) Hills, Marlan D.,
"Software Development",
Proceedings of the Second International Conference on
Software Engineering, IEEE, New York, Octubre 1976,
vol. II, p.79
- (39) Stay, J.F.,
"HIPO and Integrated Program Design",
IBM Systems Journal, vol. 15, No.2, 1976, pp.143-154
- (40) Myers, G.,
"Composite Structured Design",
Van Nostrand, 1978
- (41) Myers, Glenford J.,
"Software Reliability Principles and Practice",
Wiley, New York, 1976
- (42) Inoue Kubo, Motoki
"Ingeniería de programación para el programa de estima-
ción de carga horaria para el Centro de Información y
Control en tiempo real del Sistema Eléctrico Nacional"
Tesis de la Facultad de Ingeniería,
México, D.F., 1981