



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
" ARAGON "

"DISEÑO DE UN SISTEMA DE INFORMACION PARA
EL PROCESO DE INSCRIPCIONES DE ALUMNOS EN
LA E.N.E.P. ARAGON"

FALE... GEN

T E S I S

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

MARTIN ORDOÑEZ ROSALES

1989



POR MI RAZA HABLARA EL ESPIRITU



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.

INTRODUCCION I

C A P I T U L O 1

ELEMENTOS NECESARIOS PARA EL DISEÑO DEL SISTEMA

1.1 FUNDAMENTOS 1

 1.1.1 REQUERIMIENTOS DEL SISTEMA 10

 1.1.2 SISTEMA DE INFORMACION 10

 1.1.3 ORGANIGRAMA DE UN EMPRESA 11

1.2 PRIMERA ACCION EN DISEÑO DE UN SISTEMA DE INFORMACION

 1.2.1 FASE DE PLANEACION 14

1.3 SEGUNDA ACCION EN EL DISEÑO DEL SISTEMA

 1.3.1 FASE DE PLANEACION 16

1.4 DISEÑO DEL SISTEMA 17

 1.4.1 PERSONAL 24

 1.4.2 EQUIPO 26

 1.4.3 ESTRUCTURA DE DATOS 27

C A P I T U L O 2

GENERACION Y EMISION DE INFORMACION PARA EL PERIODO DE INSCRIPCION

2.1 DIAGRAMA DE SISTEMA 31

2.2 PROCESO DE PREINSCRIPCION 32

 2.2.1 HISTORIAL ACADEMICO 32

2.3 INICIACION DE FUNCIONAMIENTO DEL SISTEMA 35

2.4 FILTRACION DEL ARCHIVO DE HISTORIA ACADEMICA 36

2.5	PRIMER FILTRO (OBTENCION DE PROMEDIOS)	37
2.5.1	PROGRAMA GENERADOR DE PROMEDIOS	37
2.5.2	FORMATO DEL ARCHIVO HIST-ACA.DAT.....	39
2.6	SEGUNDO FILTRO (CONVERSION DE FORMATO).....	44
2.6.1	PROGRAMA SEPARADOR DE REGISTROS	44
2.7	DECLARACION DE ARCHIVOS RAIZ Y LLAVES.....	48
2.8	ARCHIVO MAESTRO.....	49
2.9	ARCHIVO DE DETALLE O ARCHIVO DE LLAVE	51
2.10	CREACION DE PRIMERA BASE DE DATOS.....	53
2.11	ASIGNACION DE PROMEDIOS.....	57
2.11.1	PERIODO SEMESTRE IMPAR	57
2.12	ASIGNACION POR PROMEDIOS	61
2.12.1	PERIODO SEMESTRE PAR	61
2.12.2	PROGRAMA MEZCLADOR	62
2.13	ASIGNACION DE NUMEROS DE SORTEO ALEATORIAMENTE.....	66
2.13.1	PROGRAMA MEZCLADOR	66
2.13.2	PROGRAMA ASIGNADOR DE NUMEROS ALEATORIOS ...	67
2.14	CLASIFICACION POR DIA DE INSCRIPCION.....	71
2.15	CREACION DE SEGUNDA BASE DE DATOS.....	73
2.16	GENERACION DE REPOTES.....	77
2.17	HOJA DE LECTURA OPTICA	77
2.18	IMPRESION DE HOJAS DE LECTURA OPTICA PARA INSCRIPCION.....	80
2.18.1	PROGRAMA PARA REALIZAR LA IMPRESION DE HOJAS DE LECTURA OPTICA	81

CAPITULO 3

CUANTIFICACION Y CUALIFICACION DE LA INFORMACION OBTENIDA DURANTE LA ETAPA DE LA INSCRIPCION.....	83
3.1 FILTROS DE DEPURACION Y VALIDACION PARA EL ARCHIVO DE INSCRIPCION.....	86
3.1.1. COMPACTACION.....	86
3.1.2. VALIDACION.....	88
3.2 CLASIFICACION Y ARCHIVO DE INSCRIPCION POR CARRERAS	91
3.3 CLASIFICACION POR GRUPOS.....	92
3.4 BUSQUEDA DE NOMBRES A LOS GRUPOS CLASIFICADOS	100
3.5 ORDENAMIENTO ALFABETICO	106
3.6 GENERACION DE FORMATO PARA LA IMPRESION DE LAS LISTAS DE ASISTENCIA.....	107
3.7 OBTENCION DE LISTAS DE ALUMNOS CON ERRORES EN SU INSCRIPCION.....	109
3.8 OBTENCION DE ESTADISTICAS DE INSCRIPCION DE ALUMNOS EN LAS MATERIAS GRUPO.....	111
3.9 GRAFICAS DE ESTADISTICAS DE INSCRIPCION	111
3.10 GENERACION DE TIRAS DE MATERIAS PROVISIONALES	113
3.11 GENERACION DE ARCHIVOS DIRECTOS.....	114
3.12 CLASIFICACION DE CLAVES MATERIAS-GRUPO POR ALUMNO INSCRITO	116
3.13 BUSQUEDA DE NOMBRES	117
3.14 BUSQUEDA DE NOMBRES DE MATERIAS.....	117
3.15 FORMATO PARA TIRAS DE MATERIAS PROVISIONALES.....	121

CAPITULO 4

PROCESO DE AJUSTE Y ACTUALIZACION DEFINITIVA	122
4.1 ARCHIVO DE AJUSTE	128
4.2 VALIDACION	129
4.3 INICIALIZACION DE LOS ARCHIVOS DIRECTOS.....	131
4.4 CONVERSION DE ARCHIVO SENCUENCIAL DE INSCRIP- CION A UN ARCHIVO DIRECTO.....	133
4.5 BUSQUEDA DE ALUMNOS Y REALIZACION DE AJUSTE Y/O ACTUALIZACION	134
4.5.1 MOVIMIENTO DE ALTA	136
4.5.2 MOVIMIENTO DE BAJA	138
4.5.3 MOVIMIENTO DE CAMBIO	140

CAPITULO 5

PERSPECTIVA DE EVOLUCION DEL SISTEMA	143
5.1 MEJORAS	146
5.2 ADAPTACION	147
5.2.1 DATOS DE ENTRADA Y ARCHIVOS	148
5.2.2 EQUIPO Y SISTEMA OPERATIVO.....	148
5.3 CORRECCIONES	150
5.3.1 ARREGLOS DE EMERGENCIA	150
5.3.2 ARREGLOS PROGRAMADOS	150

CONCLUSIONES	151
--------------------	-----

APENDICE "A" PROGRAMAS DEL SISTEMA	153
--	-----

APENDICE "B" DIRECCIONAMIENTO DE ARCHIVOS	213
---	-----

BIBLIOGRAFIA	218
--------------------	-----

En este trabajo de tesis, realizo la descripción y filosofía de diseño del SISTEMA DE INFORMACION PARA EL PROCESO DE INSCRIPCIONES DE ALUMNOS EN LA ENEP ARAGON, detallando además, algoritmos, criterios y políticas de desarrollo utilizados en la creación y funcionamiento del mismo.

Este sistema pretende cubrir lo mejor posible, la imperiosa necesidad de independización y autosuficiencia en lo que al aspecto de procesamiento y/o generación de la información requiere la ENEP ARAGON a nivel interno (aunque en ciertos aspectos, es a nivel externo también) para el periodo anterior y posterior a la inscripción de sus alumnos.

La tesis se divide en CINCO capitulos y dos apéndices complementarios, el contenido en forma general es el siguiente:

CAPITULO I. ELEMENTOS NECESARIOS PARA EL DISEÑO DEL SISTEMA

En este capítulo, se trata el tema de sistemas en general, ahondando en lo concerniente a *sistemas de información*.

La teoría fundamental acerca de estos sistemas y sus fases de diseño y desarrollo se cubren en forma global y a manera de sustentación teórica del proyecto de tesis, para lo cual, se consultó a varios autores y se sustrajo lo mas sobresaliente e importante de cada una de sus aportaciones al tema.

CAPITULO 2. GENERACION Y EMISION DE INFORMACION PARA EL PERIODO DE INSCRIPCION

En este capitulo, se describe la forma en que se obtienen los números de sorteo y la impresión de las hojas de lectura óptica, información vital para la inscripción de los alumnos.

CAPITULO 3. CUANTIFICACION Y CUALIFICACION DE LA INFORMACION OBTENIDA DURANTE LA ETAPA DE LA INSCRIPCION

Se detalla la manera en la cual se realiza el procesamiento de la información recabada durante las inscripciones, para la obtención de las listas de asistencia de grupos, las tiras de materia provisionales, estadísticas de cupos y grupos, gráficas de población, etc.

CAPITULO 4. PROCESO DE AJUSTE Y ACTUALIZACION DEFINITIVA

En este capitulo, se engloba el periodo posterior a la inscripción, en el cual es necesario realizar un ajuste a la inscripción original, el proceso necesario para los cambios es analizado y explicado aquí.

CAPITULO 5. PERSPECTIVAS DE EVOLUCION DEL SISTEMA

Como todo sistema, este debe evolucionar para mantenerse vigente y funcional. en este capitulo, se desglosan puntos importantes acerca de las perspectivas a mediano y largo plazo del sistema.

APENDICE A. PROGRAMAS

La parte fundamental del sistema lo constituye el software utilizado, por ello en este apéndice se presentan los programas que conforman al sistema.

APENDICE B. DIRECCIONAMIENTO DE ARCHIVOS

La explicación y desarrollo de algunos métodos de direccionamiento de archivos importantes y utilizados en el sistema son explicados.

Los objetivos primordiales de este trabajo de tesis son:

- Diseñar un sistema capaz de procesar toda la información generada por las inscripciones en la ENEP ARAGON.

- Independizar a la ENEP ARAGON en cuanto al proceso de inscripciones de alumnos se refiere.

- Lograr un ahorro en tiempo y dinero para el proceso de inscripciones.

- Evitar un uso inadecuado de recursos humanos y materiales en la ENEP ARAGON en cuanto a la información referente a los alumnos y/o inscripciones de los mismos.

**DISEÑO DE UN SISTEMA DE INFORMACION PARA
EL PROCESO DE INSCRIPCIONES DE ALUMNOS
EN LA ENEP ARAGON**

D) ELEMENTOS NECESARIOS PARA EL DISEÑO DE UN SISTEMA

1.1 FUNDAMENTOS

Para elaborar o diseñar un "sistema de información", primero se debe definir o concretar el significado del mismo.

SISTEMA

Conjunto de elementos conectados entre sí y que interactúan de modo que forman una unidad orgánica.

En informática, la palabra "sistema" se utiliza precisamente para indicar varios componentes, tanto de hardware¹ como de software² (tales como la unidad central, las terminales, las memorias periféricas, el software de base y los programas específicos) que, en su conjunto constituyen un "sistema de proceso de datos".

Si se definió el concepto de sistema, el concepto de información no puede dejarse a la deriva por ello se da a continuación:

INFORMACION:

En sentido general indica la adquisición de nuevos conocimientos

Basándose en la moderna teoría de la información de Shannon el término información es todo lo que reduce la incertidumbre entre diversas alternativas posibles

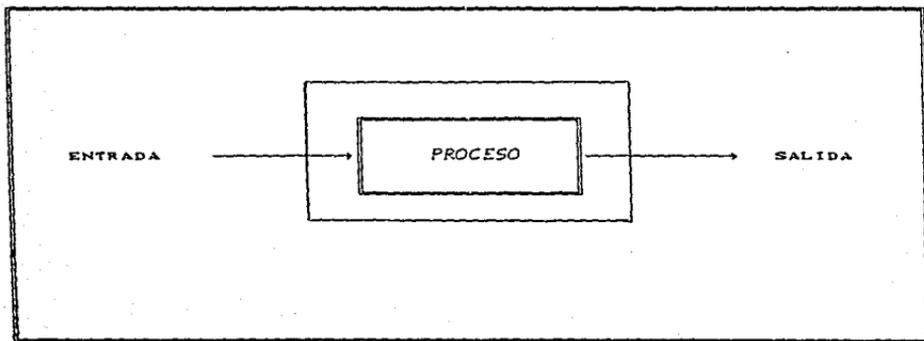
En informática el término información suele ser sinónimo de datos

1 Elementos físicos.

2 Programas, archivos, datos.

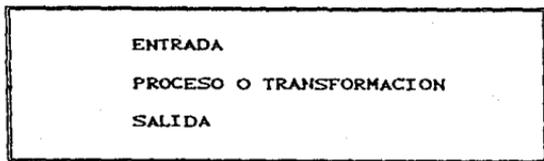
En el cuadro anterior se han expresado las definiciones mas comunes que se manejan para el término información. En el sistema planteado, se conjugarán las definiciones para darle un enfoque adecuado al concepto.

El diagrama básico o más general de un sistema es el siguiente:



Según la figura anterior, un sistema en general se compone de

3 partes fundamentales:



ENTRADA:

Aquí se dan los elementos y/o información a procesar por el sistema.

PROCESO O TRANSFORMACION:

Tomando este bloque como una caja negra, en esta parte del sistema, se aplicarán los cambios y operaciones necesarias sobre el elemento de entrada para lograr así, obtener un resultado previsto o deseado.

SALIDA:

Una vez aplicado sobre la entrada la acción de la parte de transformación o proceso, se obtendrá un tercer elemento que contendrá la información y elementos ya procesados o transformados.

El diagrama básico de un sistema como el presentado, corresponde a uno de malla abierta o de lazo abierto.

En los sistemas de malla abierta, la salida obtenida no tiene una regulación posible a través del sistema en sí. Esto significa que, si existe una variación en la salida con respecto a lo esperado, no podrá realizarse un ajuste automático dentro del sistema para eliminar el error generado.

Lo anterior es la característica principal de un sistema de malla abierta, la cual a su vez, casi siempre es su mayor desventaja.

Un segundo tipo de sistema se muestra en la figura 1.2. Este sistema es conocido como de malla cerrada, lazo cerrado, bucle cerrado o retroalimentado.

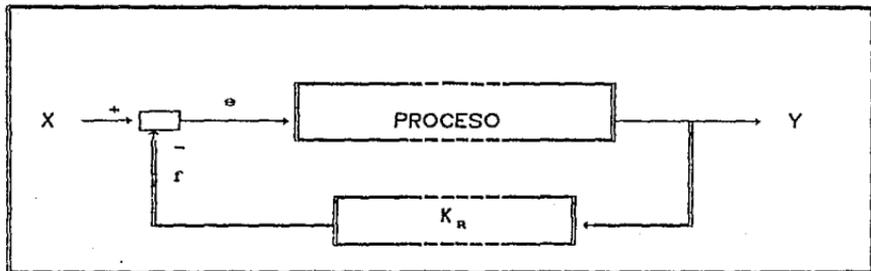


FIGURA 1.2

Las partes que lo conforman son las siguientes:

X Entrada:

Se introducen los elementos y/o información a procesar o transformar por el sistema.

Y Salida:

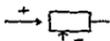
La misma definición que para el sistema de malla abierta.

P Transformación, Proceso, Planta:

Ya definido anteriormente.

K_R Escalamiento de lazo de retroalimentación ó ganancia de retroalimentación:

Para poder realizar un comparación entre la salida y la entrada, es necesario aplicarle un escalamiento o proporcionamiento estándar de entrada/salida.



Punto o nodo suma:

En esta parte se realiza la comparación de la salida contra la entrada o viceversa. El signo invertido sirve para lograr la comparación.

E Señal de error:

Al realizar la comparación entre la entrada y la salida, se genera una tercer señal, que indica o muestra si un error o desviación de la salida con respecto a la entrada existe. Para determinar si el sistema funciona totalmente controlado, la señal de error debe ser nula.

F Señal de retroalimentación:

Es la señal de salida ya escalada para poder compararla con la entrada.

Las ecuaciones del sistema son:

$$Y = P X$$

La salida es igual a la acción
de transformación sobre la entrada.

$$F = K_R Y$$

Para poder comparar la salida contra la entrada, es necesario generalmente, aplicar sobre la primera un factor de conversión que será un escalamiento para que la magnitud de la salida sea análoga a la de la entrada.

$$E = X - F$$

$$E = X - K_R Y$$

La señal de error es igual a la comparación de la señal de entrada con la de salida ya escalada.

En un sistema de malla cerrada, el control depende de sí mismo. O sea que el sistema se autoregulará en cuanto comience a funcionar y se le aplique una entrada y parámetros de escalamiento adecuados.

En resumen, un sistema puede ser retroalimentado o no retroalimentado. El sistema retroalimentado aventaja al segundo principalmente en que puede auto-regularse una vez que se pone en marcha su funcionamiento.

El diseño del sistema en este trabajo, se realiza tomando en cuenta como factor primordial la retroalimentación de la información.

Al proceso de obtención de información resultante de la comparación de los rendimientos reales con estándares y del análisis de elementos de control sobre las desviaciones generadas en la práctica, se le conoce como retroalimentación.

Un sistema de cualquier índole, cumplirá con requerimientos para funcionar de manera adecuada y en su caso, continuar vigente.

1.1.1 REQUERIMIENTOS DEL SISTEMA

- a) Ser un sistema abierto
(interacción con el medio exterior)
- b) Contar con un estándar de rendimiento útil y por encima de otros sistemas.
- c) Tener manera de comparar el rendimiento real con el estándar.
- d) Establecer un método de retroalimentación.
- e) Ser autoregulable o autoajustable..

A continuación se describirá en que consiste un sistema de información:

A partir de este punto, todo lo que se diga acerca de sistemas, se referirá única y exclusivamente a los de información. Por lo tanto, se definirá antes que cualquier otra cosa lo que es un Sistema de información:

1.1.2 SISTEMA DE INFORMACION

Conjunto de todos los procedimientos y dispositivos (computadoras y otros) implicados en el proceso, almacenamiento y distribución de la información en una organización.

En el sistema de información, se procesará como es obvio, la entrada, que consiste en los archivos de datos a utilizar, de este modo, se obtendrá o generará información, reportes y otras salidas necesarias al sistema y al usuario del mismo.

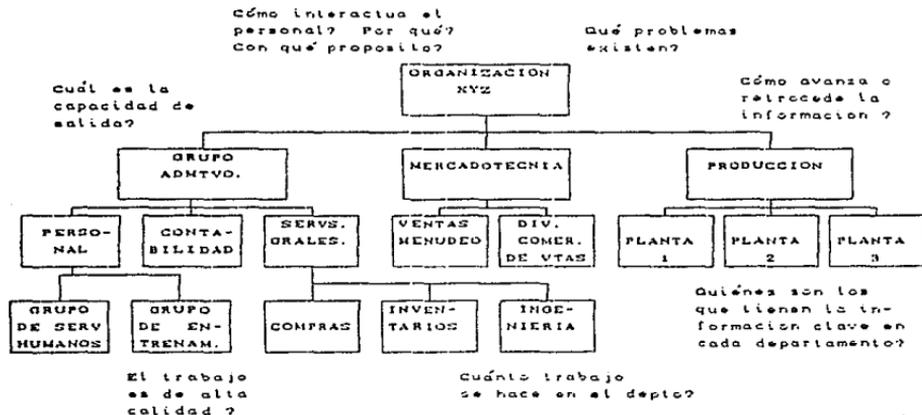
Para conformar un sistema de información, se realizan subdivisiones a manera de módulos interrelacionados y que funcionan como subsistemas. cada subsistema se ocupa de un procedimiento específico, esto es, el sistema de información se desglosa en equipo específico (hardware), programas, archivos (software) y procesos.

Englobando el total de los subsistemas arriba mencionados, se tiene una APLICACION de sistemas de información.

Como los sistemas de información apoyan a otros sistemas, lo primero a estudiar en el análisis ,es al sistema como un todo y luego enfocarse en detalles de los mismos. A menudo, se utilizan en la empresa o institución organigramas, para mostrar las relaciones formales entre los elementos del sistema, tales como divisiones, departamentos, secciones, oficinas y personal.

El organigrama (ver figura) no mostrará el íntimo funcionamiento de la manera en que opera el sistema de la empresa o institución, ya que la información presentada es muy reducida.

1.1.3 ORGANIGRAMA DE UNA EMPRESA



Para el diseñador del sistema, es importante tomar en cuenta por ende, lo siguiente:

1) CANALES FORMALES

¿ Qué interacciones existen entre el personal y los departamentos, pero que no aparecen en el organigrama o en los procedimientos de operación previamente establecidos?

2) INTERDEPENDENCIAS

¿ En qué otros departamentos y componentes de la empresa o institución se encuentra una dependencia específica?

3) EL PERSONAL CLAVE Y LAS FUNCIONES

¿Qué individuos y elementos del sistema son más importantes para su existencia exitosa?

4) RELACIONES CRITICAS DE COMUNICACION

¿ Cómo circulan la información y las instrucciones entre los componentes de la empresa y como interactúan las diferentes áreas con las demás?

Los anteriores planteamientos, marcan la importancia de investigar y analizar como funcionan las compañías o instituciones.

Para el desarrollo del nuevo sistema, el análisis debe identificar las características importantes que sean necesarias. Determinándose así, la forma de operación del sistema y subsistemas, las entradas requeridas y las salidas a producir. En esta etapa, se establece el trabajo que se realizará por computadora y cuál manualmente.

Para lograr lo anterior, es necesario pasar por ciertas etapas

de análisis de sistemas, las cuales son mejor conocidas como fases, esto se detalla a continuación:

12 PRIMERA ACCION EN DISEÑO DE UN SISTEMA DE INFORMACION

12.1 FASE DE PLANEACION

En la planeación de un sistema de información, lo primero a llevar a cabo es determinar de una manera simple y concisa, el enunciado del problema que se solucionará, mencionando junto con él, las limitantes y alcances que se tendrán al tratar de solventar el problema.

Se tiene que agregar además, una descripción del marco de referencia en donde se encuadrará al sistema, sumando a esto, un análisis de las metas que se cumplirán al implantar el nuevo sistema.

Cuando se realiza el enunciado del problema, este tendrá que ser entendido total y sobriamente, aparte de tener un conocimiento adecuado acerca del entorno del problema.

Como técnicas a utilizar por parte del diseñador del sistema para lograr ese "entendimiento claro" arriba mencionado, tendrán:

ENTREVISTAS CON EL FUTURO USUARIO DEL SISTEMA
OBSERVACION DE LAS TAREAS PROBLEMATICAS
DESARROLLO DE LAS TAREAS ACTUALES

El diseñador debe tomar en cuenta diferentes puntos de vista, enfoque y/o necesidades, lo cual redundará en la precepción del área del problema. Además, considerar el poco conocimiento de las posibilidades de una computadora del propio usuario.

Sucede con frecuencia, que los sistemas de información se diseñan para aliviar un síntoma y no la enfermedad en sí. Si el problema no es estudiado, analizado y comprendido hasta dominarlo, como se mencionó antes, muy probablemente se tendrá esta situación, lo cual no solucionará al problema, trayendo consigo esto, que el sistema sea inoperante o en otras palabras, inútil o inservible.

1.3 SEGUNDA ACCION EN EL DISEÑO DE UN SISTEMA

1.3.1 FASE DE PLANEACION

El segundo paso o acción a realizar en la planeación de un sistema de información, es determinar lo apropiado de una solución computacional. Además de ser eficaz en términos de costo, un sistema computacional debe aceptarse social y políticamente.

Para ser funcional en términos de costo, un nuevo sistema de información debe proporcionar los mismos servicios e información (por lo menos) que el sistema antiguo, usando menos tiempo y personal, o proporcionar servicios e información que antes eran inaccesibles.

Ahora, la atención se centra en las funciones de los principales subsistemas del sistema computacional.

Un sistema computacional, está formado por: Los subsistemas del personal, equipo y de productos de programación, más las interconexiones entre ellos.

Las funciones que debe realizar cada subsistema principal se deben identificar, además de establecer las interacciones entre subsistemas para determinar las restricciones en el desarrollo y operación de cada subsistema principal.

Las limitaciones especifican número y tipo de equipos, cantidad y habilidades del personal, y características del producto de programación como: Funcionamiento, precisión y nivel de

confiabilidad. La asignación adecuada de funciones, entre equipo, programación y personal puede dificultarse durante la planeación preliminar, tal vez sea necesario desarrollar primero un análisis detallado. No obstante, debe intentarse la definición preliminar de las funciones de los subsistemas principales.

1.4 DISEÑO DEL SISTEMA

EL diseño del sistema determina como un sistema logrará lo que tiene que lograr; involucra la configuración de los componentes de Software y Hardware del sistema para que después de su instalación, el sistema satisfaga completamente las especificaciones de sistemas establecidas al final de la fase de planeación y análisis de sistemas.

Como se mencionó anteriormente, las tres primeras acciones primordiales a realizar para el diseño de un sistema de

información, son:

1) Determinar de manera simple el objetivo por el que se diseñará el sistema (con sus limitantes y marco de referencia)

2) Determinar lo que es apropiado para una solución computacional (costo, aceptación, ventajas, etc.)

3) Atención a los subsistemas del sistema (personal, equipo, estructura de datos, etc.)

D) Para poner en marcha el diseño del sistema de información de este trabajo, se determina el objetivo del mismo:

Obtener un sistema de información completo, funcional, veraz, flexible y de bajo costo para el proceso de inscripciones de alumnos en la Escuela Nacional de Estudios Profesionales IIRGON.

Según se menciona en el objetivo, el sistema de información contemplará la resolución de cómo manejar toda la información necesaria para llevar a cabo las inscripciones de alumnos en la ENEP Aragón de manera eficaz.

El sistema no consiste en que el alumno se inscriba por computadora ó se le asignen grupos de manera automática, sino en que la información generada sea procesada de una manera rápida, útil y eficaz.

No se pretende modificar el procedimiento de trámites de solicitud de inscripción, ya que este atañe a la UNIDAD ACADEMICA (Departamento de Servicios Escolares).

El que un alumno se inscriba o se le asigne su grupo u horario, se contempla dentro de las perspectivas de evolución del sistema.

El sistema de información se diseña para sustituir al anterior, el cual ha estado vigente casi en su totalidad desde que la ENEP ARAGON fue inaugurada y puesta a funcionar, como una dependencia descentralizadora de ciertas carreras universitarias a

nivel licenciatura.

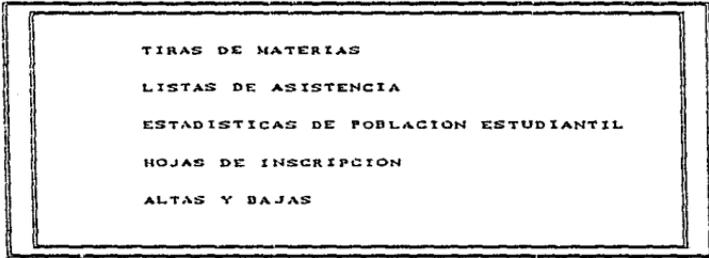
Como un paréntesis se mencionarán a continuación las carreras que se imparten en la ENEP ARAGON:

Ingeniería en Computación
Ingeniería Mecánica - Eléctrica
Ingeniería Civil
Licenciado en Derecho
Licenciado en Sociología
Licenciado Periodismo y Comunicación Colectiva
Licenciado en Relaciones Internacionales
Licenciado en Pedagogía
Licenciado en Economía
Licenciado en Planif. para el Des. Agrop.
Licenciado en Arquitectura
Licenciado en Diseño Industrial

El antiguo sistema de información ha tenido que ir "parchándose" a medida que alguna nueva carrera se impartiera, o a medida que la población escolar fuera creciendo. Provocando "fracturas" en su funcionamiento, así como una muy marcada lentitud en el proceso de obtención de resultados que tenían que generarse manualmente por un gran número de personas y tiempo, lo cual es poco adecuado. Trayendo consigo esto, la circulación innecesaria de la información "confidencial" de los alumnos, además de una

burocratización inútil del sistema, debido al incremento de las personas por las que tenía que pasar la información.

Las anteriores son algunas de las razones por las cuales el viejo sistema no entrega resultados vitales en el tiempo preciso. Resultados tales como:



TIRAS DE MATERIAS
LISTAS DE ASISTENCIA
ESTADISTICAS DE POBLACION ESTUDIANTIL
HOJAS DE INSCRIPCION
ALTAS Y BAJAS

Provocando que exista un atraso en la obtención de información útil para el alumno, para la unidad académica, para las coordinaciones de cada carrera y para los profesores.

II) El diseño del sistema de información para el proceso de inscripciones en la ENEP ARAGON, contempla el obtener a su debido tiempo toda la información que es el resultado del proceso de inscripción. Además intenta o propone una independización muy notoria de la ENEP ARAGON con respecto a otras dependencias de la

UNAM^a en cuanto a la utilización de equipos de cómputo ajenos a la propia ENEP ARAGON.

Con el diseño del nuevo sistema, se plantea la opción de que la ENEP ARAGON sea virtualmente autosuficiente en cuanto al manejo de información sobre su alumnado. Además de explotar en un porcentaje mucho mayor, las capacidades del equipo instalado en su propio centro de cómputo, las cuales no habían sido utilizadas con el viejo sistema.

El costo de utilización del sistema nuevo es muy bajo en comparación con el sistema viejo, ya que el equipo a usar es el mismo instalado desde 1984 en la ENEP ARAGON, por lo cual no se pagarán capacitación foránea, personal dedicado y especializado para realizar el enlace :

C A E ¹ - E N E P A R A G O N P I T A G O R A S - E N E P A R A G O N

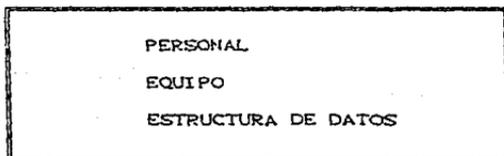
Con el sistema propuesto en este trabajo, se puede evitar inclusive, la contratación y en consecuencia la derogación

² Tales como la Coordinación de Administración Escolar y la Dirección General de Administración Escolar

supérflua de salarios a personal de tiempo completo que tuviera que trabajar en CAE o en PITAGORAS. los cuales tendrian que cubrir la necesidad constante de obtención de información y comunicación continua con otros subsistemas del viejo sistema.

El sistema en cuestión, no propone desplazar a trabajador o empleado alguno de su puesto, antes bien, otorga libertad a los jefes de unidades y departamentos, de asignar al personal bajo su mando² a tareas que hagan aún más óptimo el funcionamiento conjunto del sistema de inscripciones y otros sistemas de información que interactúan con el primero.

III) Para que el diseño y posteriormente la implantación del sistema de información pudiera ser un éxito, se analizan 3 factores de gran importancia :



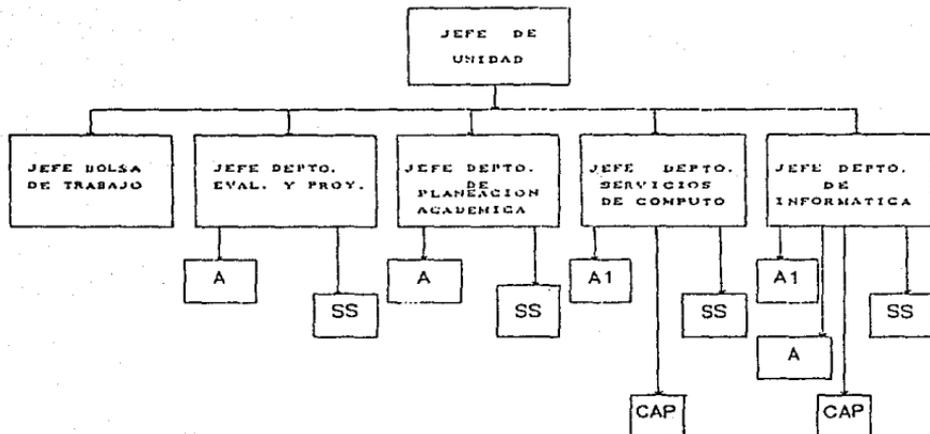
2

Secretarias, técnicos, capturistas, etc.

1.4.1 PERSONAL

El departamento encargado de la generación de todo el procesamiento por computadora de la información recabada durante las inscripciones, es el de INFORMATICA, departamento que pertenece a la UNIDAD DE PLANEACION (ver la figura de la siguiente página).

ORGANIGRAMA DE LA UNIDAD DE PLANEACION



A : AYUDANTE DE PROFESOR

A1: ANALISTA DE SISTEMAS

SS: PRESTADOR DE SERVICIO SOCIAL

CAP: CAPTURISTA

El personal actual es adecuado en número, y en lo que al DEPARTAMENTO DE INFORMATICA se refiere, al ser su personal, egresados o pasantes de la carrera de Ingeniería en Computación, tendrán todos ellos, la capacidad de análisis y conocimientos requeridos para el manejo del sistema.

En cuanto al personal requerido para el enlace de información con la UNIDAD ACADEMICA y las coordinaciones, sigue siendo el mismo que con el antiguo sistema, con la diferencia que existe mayor flexibilidad en cuanto a las ocupaciones que desempeñará cada quien dentro del funcionar del nuevo sistema, ya que los reportes y estadísticas generados, facilitarán y agilizarán su labor.

1.4.2 EQUIPO

El equipo de cómputo a utilizar para el sistema, es una minicomputadora HP-1000 (HEWLETTE PACKARD modelo 1000), con 30 estaciones de trabajo.

El equipo HP-1000 no es el ideal, dadas sus características, para operar un sistema de la naturaleza del que se trata en este trabajo, ya que no es el adecuado ni recomendado por los propios fabricantes de HP, puesto que por la magnitud del sistema, la cantidad de información y su naturaleza, es recomendado por los

asesores de HP, utilizar una computadora de la familia 3000 y no una 1000, ya que esta es la antecesora de la familia 9000, las cuales se enfocan o diseñan con el fin de ser utilizadas en una aplicación propia del área de control de procesos por computadora (CAD/CAM³ por ejemplo) y no para procesos administrativos.

Pero como una de las premisas que forman parte del objetivo del sistema, es obtener bajo costo y utilizar en alto porcentaje y de manera óptima los recursos propios de la ENEP ARAGON, se encuadra al sistema propuesto, desde su inicio, contemplando las posibilidades y capacidades del equipo existente. Logrando con esto, un ahorro considerable en cuanto a costo se refiere.

Con el sistema, se trata de maximizar el rendimiento a obtener, pero con mínimo esfuerzo por parte del personal y equipo con que cuenta la ENEP ARAGON.

Como alternativa y de manera auxiliar, se utilizará en el sistema, el equipo de computadoras personales y periféricos con que se cuenta en el departamento de INFORMATICA.

14.3 ESTRUCTURA DE DATOS

Como recursos para tener una adecuada estructura de datos, se

³

CAD: Diseño asistido por computadora.

CAM: Manufactura asistida por computadora.

manejan en el sistema:

ARCHIVOS SECUENCIALES
ARCHIVOS DIRECTOS
BASE DE DATOS RELACIONAL
ARREGLOS UNI Y BIDIMENSIONALES

La base de datos utilizada con el equipo HP-1000, es de tipo relacional, llamada IMAGE-1000. Esta base de datos es muy útil para facilitar el tratamiento de los grandes volúmenes de información que es necesario manejar para el funcionamiento del sistema.

El acceso a los archivos secuenciales o directos, se realiza a través de programas escritos o codificados en su mayoría, en lenguaje FORTRAN 77. Constituyendo una especie de manejador de archivos, actuando conjuntamente con la base de datos.

Los algoritmos a utilizar para el sistema, se cuentan en gran variedad, dada la necesidad de realizar muchas conversiones y procesos sobre los archivos originales.

FORTAN 77 es el lenguaje utilizado en casi la totalidad del sistema. Esto se debe a que es el lenguaje que presenta mayor flexibilidad y utilerías para el equipo HP-1000. Dada su naturaleza, el FORTRAN 77 es una herramienta idónea para utilizarse

en programación de control de procesos y es por eso que la minicomputadora HP-1000 (con fines de control), cuenta con mayor potencialidad mediante el lenguaje mismo.

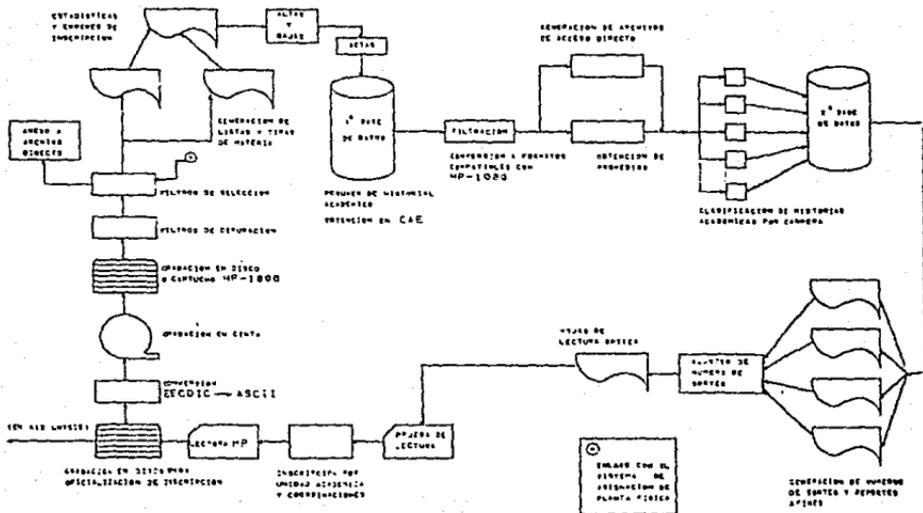
Aprovechando estas características al máximo, se diseñó el sistema.

CAPITULO 2

GENERACION Y EMISION DE INFORMACION PARA EL PERIODO DE INSCRIPCION

A continuación se presenta un diagrama que esquematiza el flujo de información y el ciclo del sistema propuesto, por este trabajo de tesis.

2.1 DIAGRAMA DEL SISTEMA



Este capítulo, se avocará al estudio y desarrollo del sistema en su primera fase, la cual se encuadra dentro del proceso de preinscripción.

2.2 PROCESO DE PREINSCRIPCION

2.2.1 HISTORIAL ACADEMICO

El historial académico, es un archivo que se actualiza semestre a semestre y que contiene información trascendental acerca del desarrollo académico de todo estudiante perteneciente a la UNAM.

Los datos contenidos dentro del historial académico son tales como, el nombre, número de cuenta del alumno¹, clave del plantel inscrito, semestre cursado, año y semestre de primer ingreso a licenciatura, último semestre inscrito, materias cursadas, materias acreditadas, materias no aprobadas, número de calificaciones con

¹ El número de cuenta es la clave de identificación del alumno para cualquier trámite que éste realice.

MB, B, S, NA y NP obtenidas a lo largo de la carrera², materias acreditadas en ordinarios, en extraordinarios y alguna información adicional.

El historial académico en forma de documento, se le entrega al alumno cada semestre para que este, se encuentre informado veraz y confiablemente sobre su situación académica.

Para propósitos del sistema que nos interesa en este trabajo, al historial académico no se le tomara como un documento, sino como una serie de datos en forma continua que representaran toda la información pertinente al desarrollo académico de cada alumno y para ello se le identificará como RESUMEN DE HISTORIA ACADEMICA.

En la DGAE³, se genera el archivo de resumen de historia académica, correspondiente a cada semestre escolar concluido.

La fuente de información principal para realizar la

² La escala correspondiente numérica para las calificaciones literales, es la siguiente:

MB	-	10
B	-	8
S	-	6
NA, NP	-	Sin equivalencia numérica

³ Dirección General de Administración Escolar

actualización del historial académico es el acta oficial donde se asientan las calificaciones de los alumnos de cada grupo-materia, al finalizar el curso correspondiente. Estas calificaciones son almacenadas en un archivo de datos que luego será procesado y se conjugará con el anterior historial académico para obtener un nuevo archivo actualizado del historial del alumno y de esta forma poder contar con un resumen actualizado del mismo.

Otra fuente de información para la obtención del nuevo historial académico, la constituye el acta adicional, donde se realizan correcciones de calificaciones que se suscitan posteriormente a la entrega de actas ordinarias, calificaciones obtenidas en exámenes extraordinarios⁴, etc.

⁴ Obteniendo estas a través de su acta oficial correspondiente.

Para una visualización más clara de lo anterior ver la figura que se muestra a continuación:



El sistema propuesto en este trabajo, toma como entrada los datos que le son proporcionados por el resumen del historial académico correspondiente al cuadro anterior.

2.3 INICIACION DE FUNCIONAMIENTO DEL SISTEMA

La ENHEP ARAGON, a través de la UNIDAD DE PLANEACION y LA UNIDAD ACADÉMICA, contacta con la DGAE para realizar la solicitud del archivo correspondiente al resumen de historia académica.

La Unidad de Planeación a través del DEPARTAMENTO DE INFORMÁTICA, realiza con ayuda de la DGAE, el proceso de grabación

en cinta magnetica del archivo de historia academica.

Una vez que el archivo de historia academica es obtenido en el formato adecuado, se procesara esta informacion utilizando el equipo existente en la ENEP ARAGON, por lo que se realizaran procesos de filtracion y conversión de formatos, lo que conforma el primer objetivo en la etapa de preinscripción, que consiste en obtener los numeros de sorteo para asignación de día de inscripción a los alumnos de las diferentes carreras. Todo lo anterior tomando como base la secuencia indicada por la figura correspondiente al ciclo del sistema (ver pagina 31).

2.4 FILTRACION DEL ARCHIVO DE HISTORIA ACADEMICA

Es el proceso que permite realizar una estandarización de la información de la historia academica para poder validar, cuantificar y cualificar esta.

El archivo original de historia academica, es un archivo secuencial con un número total de registros igual al numero de alumnos que realizaron su reinscripción o primera inscripción en el semestre anterior a la generación de la última actualización del mismo.

2.5 PRIMER FILTRO (OBTENCION DE PROMEDIOS)

El primer filtro realizado al archivo original grabado en código ASCII, sirve para obtener el promedio numerico del alumno. Como entrada se tiene el archivo de historia academica llamado HIST_ACA.DAT y como salida, se tendra el archivo conteniendo los promedios de cada alumno. El programa que se utiliza para esto es SOR2.FTN.

A continuación se detallarán las características basicas utilizadas en este programa:

2.5.1 PROGRAMA GENERADOR DE PROMEDIOS

A) Toma como fuente el archivo Hist_aca.dat, y se obtiene como salida un archivo con los promedios calculados.

B) Se pide en el programa, la clave del plantel (Ingenieria civil, Ingenieria en computación, etc) y el rango de promedios a obtener en el archivo de salida. De esta manera, se puede clasificar por promedios a los archivos generados por plantel.

C) En el sistema, los nombres de archivos generados tienen

como inicio la identificación del plantel correspondiente y a continuación el rango de las calificaciones contenidas.

EJEMPLO:

DER9 . DAT

Significa:

- ARCHIVO CORRESPONDIENTE A DERECHO.
- LAS CALIFICACIONES CONTENIDAS ESTAN EN UN RANGO DE 9.00 A 10.00

ING7 . DAT

Significa:

- ARCHIVO CORRESPONDIENTE A INGENIERIA.
- LAS CALIFICACIONES CONTENIDAS ESTAN EN UN RANGO DE 7.00 A 7.99

D) El programa realiza el conteo del numero de alumnos que aparecen con un promedio menor a 6 ó no tienen calificaciones en su historial académico. Si se desea, los archivos obtenidos por rango de calificaciones, se mezclan (realizando un proceso conocido como MERGE) en uno solo. Obteniendo así, un archivo general de

determinada carrera, pero ordenado por rangos de calificaciones⁵.

2.5.2 FORMATO DEL ARCHIVO HIST_ACADAT

CAMPO:

a b c d e f g h i j k l m n o p q r s t u

CARACTERES:

32 8 3 2 2 2 2 2 10 2 2 2 2 2 2 2 2 3 3 3 3

⁵ La mezcla o fusión de los archivos de cada carrera por rangos de calificaciones, se realiza por medio del editor de línea de la HP-1000.

DESCRIPCION DE CAMPOS:				
CAMPO	DESCRIPCION	LONGITUD	VARIABLE	TIPO
a	Nombre de alumno	32	ALGO	alfanumerico
b	No. de cuenta	8	NCTA	alfanumerico
c	Clave plantel	3	CPL	entero
d	Clave carrera	2	CCA	entero
e	Ingreso a licenc.	2	ING	entero
f	Calif. con MB	2	MB	entero
g	Calif. con B	2	B	entero
h	Calif. con S	2	S	entero
i	Espacios en blanco	10	BLAN	alfanumerico
j	Calif. con NA	2	NA	entero
k	Calif. con NP	2	NP	entero
l	Revalidadas	2	REV	entero
m	Mat. acreditadas	2	ACRE	entero
n	Mat. aprobadas en or.	2	APOR	entero
o	Mat. aprobadas en ex.	2	APEX	entero
p	Mat. reprob. en ord.	2	REOR	entero
q	Mat. reprob. en ext.	2	REEX	entero
r	Créd. obligatorios	4	COBL	entero
s	Créd. optativos	4	COPT	entero
t	Periodo inicial cursado en lic.	4	PERIN	entero
u	Periodo de última inscripción	4	PEUL	entero

De acuerdo al artículo 19 de la legislación universitaria, sólo se obtendrán los promedios de los alumnos que tengan derecho a una nueva reinscripción, por ello, cada semestre que se generen los promedios, se modificará el parámetro que sirve para determinar si el alumno tiene derecho a reinscripción o no. Este parámetro es el año de ingreso y el periodo inicial cursado en el nivel de licenciatura, logrando con esto, un punto de referencia para

contabilizar los semestres que han transcurrido desde el inicio de los estudios a nivel licenciatura por parte del alumno.

Los archivos con promedios obtenidos tienen el siguiente formato:

CAMPO:																					
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
CARACTERES:																					
32	8	3	2	2	2	2	2	10	2	2	2	2	2	2	2	2	4	4	4	4	6

Especificación de los campos:

Los campos de los archivos generados, tienen casi el mismo formato que el archivo original, con la diferencia de que el nuevo archivo contiene un campo adicional que sirve para representar

allí, el promedio obtenido:

DESCRIPCION DE CAMPOS:				
CAMPO	DESCRIPCION	LONGITUD	VARIABLE	TIPO
a	Nombre de alumno	32	ALGO	alfanumérico
b	No. de cuenta	8	NCTA	alfanumérico
c	Clave plantel	3	CPL	entero
d	Clave carrera	2	CCA	entero
e	Ingreso a licenc.	2	ING	entero
f	Calif. con MB	2	MB	entero
g	Calif. con B	2	B	entero
h	Calif. con S	2	S	entero
i	Espacios en blanco	10	BLAN	alfanumérico
j	Calif. con NA	2	NA	entero
k	Calif. con NP	2	NP	entero
l	Revalidadas	2	REV	entero
m	Mat. acreditadas	2	ACRE	entero
n	Mat. aprobadas en or.	2	APOR	entero
o	Mat. aprobadas en ex.	2	APEX	entero
p	Mat. reprob. en ord.	2	REOR	entero
q	Mat. reprob. en ext.	2	REEX	entero
r	Cred. obligatorios	4	COBL	entero
s	Cred. optativos	4	COPT	entero
t	Periodo inicial cursado en lic.	4	PERIN	entero
u	Periodo de ultima inscripción	4	PEUL	entero
v	Promedio	8	PIRO	real

La variable introducida en este archivo, que contiene el promedio es: PIRO

CAMPO: V
 DESCRIPCION: Promedio del alumno
 LONGITUD: 8
 VARIABLE: PIRO
 TIPO: REAL

La formula para obtener el promedio es la siguiente:

$$F1 = \text{FLOAT} (MB)$$

$$F2 = \text{FLOAT} (B)$$

$$F3 = \text{FLOAT} (S)$$

$$\text{PIRO} = (F1*MB + F2*B + F3*S) / (F1+F2+F3)$$

Se convierte a número real el valor de la variable de calificaciones correspondiente, para que el promedio pueda contener parte fraccionaria.

VARIABLES DE USO GENERAL

VARIABLE	DESCRIPCION	TIPO
LAC	PROMEDIO MENOR A TOMAR EN CUENTA	ENTERA
LA2	PROMEDIO MAYOR A TOMAR EN CUENTA	ENTERA
CPL, LLL	CLAVE DE PLANTEL A OBTENER SUS PROMEDIOS	ENTERA
N2	# DE ALUMNOS A LOS QUE SE OBTIENE PROMEDIO	ENTERA
KL	# DE ALUMNOS CON PROMEDIO MENOR QUE 6	ENTERA

El listado del programa se puede consultar en el apéndice "A", (PROGRAMA 1)

2.6 SEGUNDO FILTRO (CONVERSION DE FORMATO)

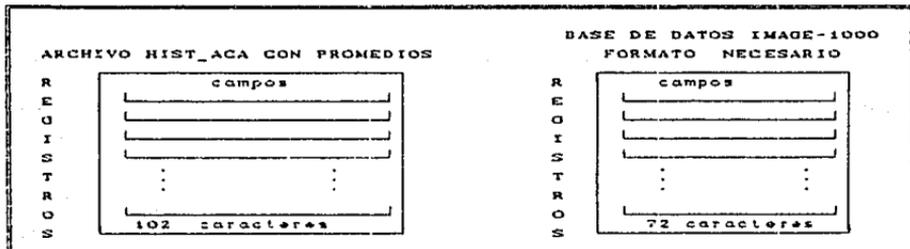
2.6.1 PROGRAMA SEPARADOR DE REGISTROS

El segundo filtro a aplicar al archivo de historia académica, ya con promedios, consiste en modificar el formato de los registros del archivo a uno compatible con los parámetros que maneja la base de datos IMAGE 1000.

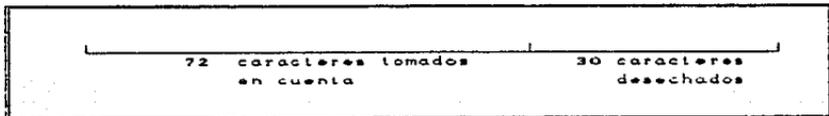
El cambio de formato es necesario ya que cada registro del archivo de historia académica con promedios, tiene una longitud fija de 102 caracteres, mientras que la base de datos maneja longitudes máximas de 72 caracteres por registro y/o registro con un número par de caracteres.

Aunque parece incongruente el que un registro tenga longitud de 102 caracteres y la base de datos maneje longitudes máximas de 72 caracteres, las características y facilidades de la base de datos permiten reconocer los campos faltantes en un registro, tomándolos del siguiente.

El diagrama a continuación, esquematiza lo anterior:



Si se introdujera el archivo sin modificar su formato, sucedería esto:

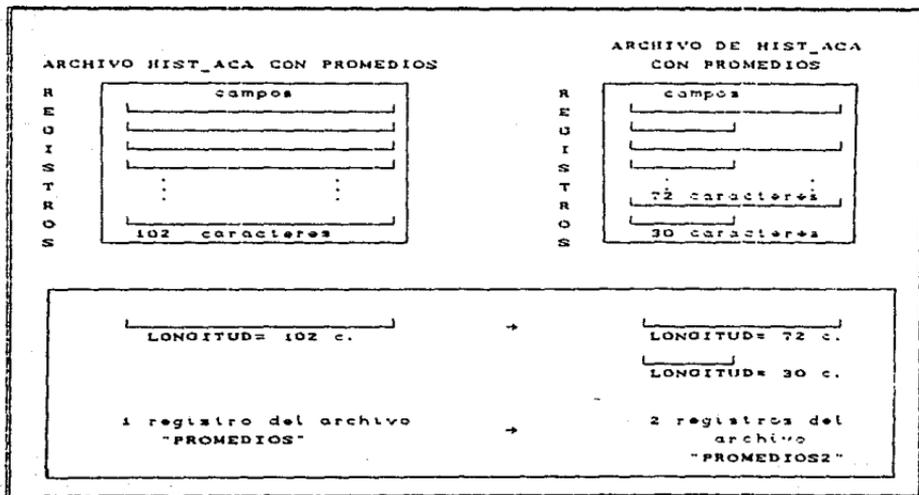


Trayendo como consecuencia, la pérdida de información y una interpretación totalmente errónea de la que sí es tomada en cuenta.

Como se mencionó anteriormente, la base de datos IMAGE 1000,

permite reconocer por bloques, la información a introducir en ella. Por lo que, el registro original se divide o "parte" en 2 registros independientes, de la siguiente manera:

PARTICION DEL REGISTRO DEL ARCHIVO HIST_ACA:



Cuando más tarde, se introduzca el archivo "PROMEDIOS2" a la base de datos, se reconocerán de manera implícita los campos del registro de longitud igual a 72 caracteres y los del siguiente, de longitud igual a 30 caracteres. Los espacios existentes después del

caracter 72, en el primer registro, no son tomados en cuenta.

El programa que se utiliza para realizar la división del registro en dos, maneja las variables que se usan en el primer filtro, menos las declaraciones de rango de calificaciones.

Se hace una lectura del total del registro (102 caracteres) y se graba en el nuevo archivo, partiéndolo a partir del caracter 73.

FORMATO DEL REGISTRO ORIGINAL
(ARCHIVO PROMEDIOS1)

CAMPO:																					
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
CARACTERES:																					
32	8	4	2	2	2	2	2	10	2	2	2	2	2	2	2	2	4	4	4	4	8

FORMATO DE REGISTROS GENERADOS
(ARCHIVO PROMEDIOS2)

CAMPO:												
a	b	c	d	e	f	g	h	i	j	k	l	m
CARACTERES:												
32	8	4	2	2	2	2	2	10	2	2	2	2

1er. REGISTRO
LONGITUD = 72 CARACTERES

formarán los archivos de datos (RAIZ y LLAVES).

Se hace esto como se indica:

ARCHIVO RAIZ O ESTRUCTURA

Es la estructura con la cual se dá de alta la base de datos.

a) nombre de la base: código de seguridad: cartucho

b) niveles de acceso:

capturistas
otros
diseñador

c) variables (items):

nombre
tipo
longitud
(accesos permitidos)

d) conjuntos (sets):

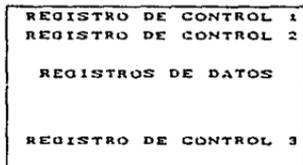
nombre::cartucho, tipo
llave
capacidad

2.8 ARCHIVO MAESTRO

El archivo obtenido anteriormente, "PROMEDIOS2", formará parte del archivo raíz (archivo de datos principal que conformará el conjunto o set de la base de datos, el cual será declarado a

detalle 'detail'). El resto lo componen 3 registros, conocidos como de control, dispuestos 2 al inicio del archivo y 1 al final.

ARCHIVO MAESTRO



REGISTRO DE CONTROL 1:

Contiene:

*Nombre de base: código, cartucho, acceso
de datos prioritario*

REGISTRO DE CONTROL 2

Contiene:

*\$ SET: nombre del conjunto declarado
en el archivo de control*

REGISTROS DE DATOS

Contiene:

*Historia académica con promedios por
alumno, divididos en 2 registros*

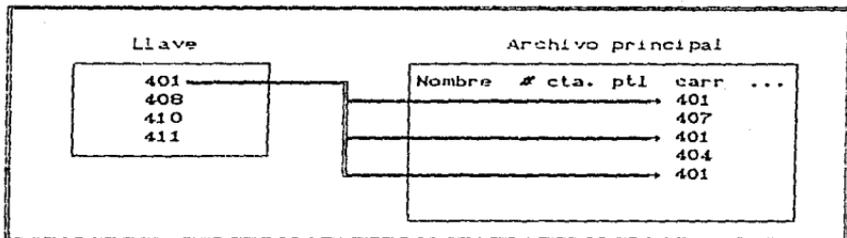
REGISTRO DE CONTROL 3

Contiene:

\$ END

2.9 ARCHIVO DE DETALLE O ARCHIVO DE LLAVE

Es el archivo que contiene los datos que se asignaran a la(s) variable(s), y que servira para hacer la relación^d entre este archivo y el archivo principal.



La llave especificada relaciona el archivo principal con el de llaves y así se puede relacionar aquellos registros pertenecientes a la base de datos, que cumplan con la llave o llaves utilizadas, evitando búsquedas secuenciales.

^d Recordar que IMAGE 1000 es una base de datos relacional.

ARCHIVO LLAVE

REGISTRO DE CONTROL 1
REGISTRO DE CONTROL 2
REGISTROS DE DATOS
REGISTRO DE CONTROL 3

REGISTRO DE CONTROL 1:

Contiene:

*Nombre de base: código, cartucho, acceso
de datos prioritario*

REGISTRO DE CONTROL 2

Contiene:

*\$ SET: nombre del conjunto declarado
en el archivo de control*

REGISTROS DE DATOS

Contiene:

Claves de carrera que servirán como llaves.

REGISTRO DE CONTROL 3

Contiene:

\$ END

ESTRUCTURA DE LA PRIMER BASE

DE DATOS.

\$CONTROL:;
 BEGIN DATA BASE:BADAT1:001:26;
 LEVELS:

1 CAP;
 10 OTROS;
 15 HOR;

ITEMS:

ALGO, X32(10,15);
 HCTA, X8(10,15);
 CPL, X4(10,15);
 CCA, X2(10,15);
 INC, X2(10,15);
 HB, X2(10,15);
 B, X2(10,15);
 S, X2(10,15);
 BLAH, X10(10,15);
 HA, X2(10,15);
 HP, X2(10,15);
 REV, X2(10,15);
 ACRE, X2(10,15);
 APOR, X2(10,15);
 APEX, X2(10,15);
 REOR, X2(10,15);
 REEX, X2(10,15);
 COBL, X4(10,15);
 COPT, X4(10,15);
 PERIN, X4(10,15);
 PEUL, X4(10,15);
 PIRO, X2(10,15);

SETS:

NAME:SUB11:126,M;
 ENTRY:CPL(1);
 CAPACITY:10;
 NAME:SUB12:126,D;
 ENTRY:

ALGO,
 HCTA,
 CPL(SUB11),
 CCA,
 INC,
 HB,
 B,
 S,
 BLAH,
 HA,
 HP,
 REV,
 ACRE,
 APOR,
 APEX,
 REOR,
 REEX,
 COBL,
 COPT,
 PERIN,
 PEUL,
 PIRO;

CAPACITY:9000;
 END.

2.10 CREACION DE PRIMERA BASE DE DATOS

Una vez obtenidos o creados los archivos principal y llaves, es necesario darlos de alta en la base de datos, creando primero ésta por medio del archivo de control¹.

A) DECLARACION DE LA BASE DE DATOS:

Se deberá utilizar el comando

```
DBDS NOM:cartucho,1
```

¹ Puede conocerse como archivo de declaraciones y parámetros.

GENERACION Y EMISION DE INFORMACION

Donde:

DBDS	DATA BASE DEFINITION STRUCTURE (DEFINICION DE ESTRUCTURA DE LA BASE DE DATOS)
NOM	NOMBRE DEL ARCHIVO DE CONTROL
CARTUCHO	DIRECCION LOGICA DEL CARTUCHO DE URABACION
I	VISUALIZACION DE LOS DATOS EN PANTALLA

Resultado:

El resultado obtenido al usar el comando anterior es:

<p>- SE DECLARA UNA BASE DE DATOS CON:</p> <p>nombre cartucho codigo de seguridad niveles de acceso campos, longitud, tipo registros bloques requeridos errores generados</p>

CONSTRUCCION DE LA BASE DE DATOS.

B) ALTA DE ARCHIVO DE LLAVES.

Se utiliza el comando:

DBBLD, NOM2::cartucho, 1, ADD

DONDE:

DBBLD	ABREVIACION DE DATA BASE BUILD (CONSTRUCCION DE LA BASE DE DATOS)
NOM2	NOMBRE DEL ARCHIVO MAESTRO O LLAVE
CARTUCHO	CARTUCHO DE GRABACION
1	VISUALIZACION EN PANTALLA
ADD	AÑADIR EL ARCHIVO A LA BASE DE DATOS

Al ejecutarse el comando anterior, se desplegarán los datos y errores en caso de existir al tratar de dar de alta el archivo llave.

Si existen errores, se repetirá el proceso después de corregirlos.

Si no existen errores, se despliega un mensaje indicativo.

C) ALTA DE ARCHIVO MAESTRO (A DETALLE).

Se utiliza el comando:

DBBLD, NOM::cartucho.1,ADD

Con el mismo significado que en B), pero modificando NOM2 por NOM3 (nombre del archivo principal).

Una vez que se ha creado la base de datos y dado de alta toda la informacion que va a contener, se procede a efectuar la generacion de reportes que seran los siguientes:

R E P O R T E S

- | |
|---|
| - ORDENACION ALFABETICA (DE NOMBRES) |
| - ORDENACION NUMERICA ASCENDENTE
DE PROMEDIOS (MENOR A MAYOR) |
| - ORDENACION NUMERICA ASCENDENTE
(DE PERIODO DE PRIMERA INSCRIPCION) |
| - OTROS ORDENAMIENTOS NECESARIOS |

Para consultar los listados de reportes, ver el Apndice

"A" (reportes).

La asignación de números de sorteo, se propone de dos maneras:

Pueden formarse a partir de la ordenación numérica ascendente correspondiente a los promedios o de una manera aleatoria, así que primero se analizará y detallará la asignación de números de sorteo por promedio y luego la asignación aleatoria.

2.11 ASIGNACION DE PROMEDIOS

2.11.1 PERIODO SEMESTRE IMPAR

No se asigna número de sorteo a alumnos de primer ingreso.

Para realizar la asignación de números de sorteo, tomando en cuenta el promedio, se parte de la obtención del reporte de ordenación numérica ascendente de promedios. A este reporte, se le conocerá como archivo de promedios ascendentes, al cual se le aplicará un proceso consistente en la asignación secuencial de un número en orden descendente. El máximo número se asignará al primer registro, y el número 1 se asignará al último registro del archivo

secuencial. esto es:



Ejemplo:

ARCHIVO DE PROMEDIOS ASCENDENTES

NOMBRE:
PROINO

CARTUCHO:
26

promedios		
_____	,	0.00
_____	,	0.01
_____	,	0.02
_____	,	0.10
_____	,	0.20
_____	,	0.30
.	.	.
.	.	.
.	.	.
_____	,	10.00

1000 registros

ARCHIVO DE NUMERACION DESCENDENTE
 PROMEDIOS ASCENDENTES

NOMBRE:
 SORINO

CARTUCHO:
 26

máximo
 número

mínimo
 número

# asignado	promedios	} 1000 registros
1000	6.00	
999	6.01	
998	6.02	
997	6.10	
996	6.20	
995	6.30	
.	.	
.	.	
.	.	
1	10.00	

Máximo número= Número de registros en el archivo.

Mínimo número= Número 1.

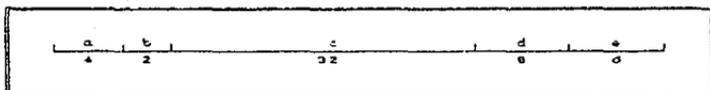
El programa que asigna la numeración descendente, a promedios ascendentes es NUMERA_DOWN.FTN (ver apéndice "A" programa 3).

En caso de que exista un "empate" en los promedios (que sea el mismo), el programa asigna el número de sorteo de acuerdo a como se presenta el archivo original, en forma secuencial y aleatoria totalmente.

La función del programa, es leer cada registro conteniendo la información propia a:

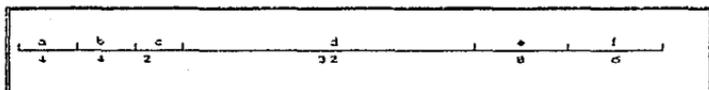
CLAVE PLANTEL	(a)
CLAVE CARRERA	(b)
NOMBRE	(c)
NUMERO DE CUENTA	(d)
PROMEDIO	(e)

FORMATO PARA ARCHIVO CON PROMEDIOS ASCENDENTES



Hecha la lectura, se asigna el número correspondiente y se graba junto con los datos anteriores:

FORMATO PARA EL ARCHIVO CON NUMERACION DESCENDENTE



Significado:
 a= Número asignado
 b= Clave plantel
 c= Clave carrera
 d= Nombre
 e= Número de cuenta
 f= Promedio

DESCRIPCION DE CAMPOS:

CAMPO	DESCRIPCION	LONGITUD	VARIABLE	TIPO
a	Número asignado	4	NUSOR	entero
b	Clave plantel	4	CPL	entero
c	Clave carrera	2	CCA	entero
d	Nombre	32	NOM	alfanumérico
e	Número de cuenta	8	NCTA	entero
f	Promedio	8	PRO	real

2.12 ASIGNACION POR PROMEDIOS

2.12.1 PERIODO SEMESTRE PAR

La condición o restricción para este proceso, es que se deben generar y asignar números de sorteos a los alumnos de primer ingreso y de reingreso.

Si el periodo de inscripciones corresponde a un semestre impar, los alumnos que en el anterior semestre fueron catalogados como de primer ingreso, deberán tener asignación de número de sorteo. El problema planteado con esto, es que para la fecha de generación de números de sorteos, la historia académica de estos alumnos, no existe aun y por ello, no tienen promedio a obtener.

Debido a lo anterior, se deberán seguir cualquiera de las 2 opciones siguientes:

- a) Inscripción en el primer día a todos estos alumnos.
- b) División y repartición del total de estos alumnos en igual número, en cada día de inscripción.

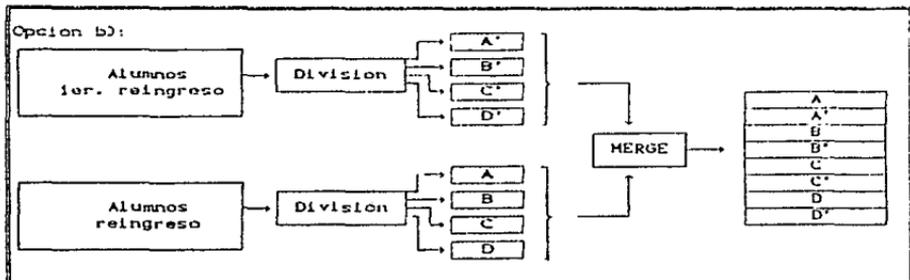
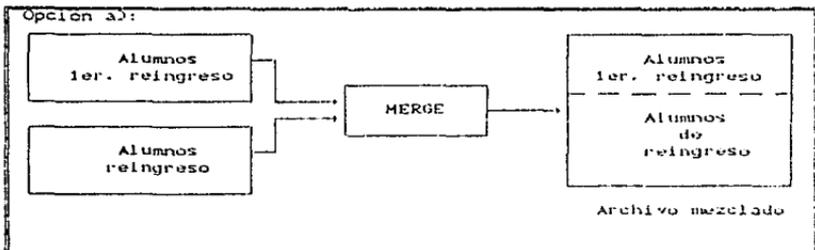
Para realizar esto, se utiliza un programa mezclador de archivos, con lo que se logra "unir" el archivo de alumnos de primer reingreso, con los demás.

2.12.2 PROGRAMA MEZCLADOR

NOMBRE:

EDITOR DE LINEA DE HP-1000

El programa al ejecutarse, realiza la fusion de los archivos citados y crea uno solo:



DESCRIPCION DE CAMPOS:				
CAMPO	DESCRIPCION	LONGITUD	VARIABLE	TIPO
a	Nombre de alumno	32	ALGO	alfanumerico
b	No. de cuenta	8	NCTA	alfanumerico
c	Clave plantel	3	CPL	entero
d	Clave carrera	2	CCA	entero
e	Ingreso a licenc.	2	ING	entero
f	Calif. con MB	2	MB	entero
g	Calif. con B	2	B	entero
h	Calif. con S	2	S	entero
i	Espacios en blanco	10	BLAN	alfanumerico
j	Calif. con NA	2	NA	entero
k	Calif. con NP	2	NP	entero
l	Revalidadas	2	REV	entero
m	Mat. acreditadas	2	ACRE	entero
n	Mat. aprobadas en or.	2	APOK	entero
o	Mat. aprobadas en ex.	2	APEX	entero
p	Mat. reprob. en ord.	2	REOR	entero
q	Mat. reprob. en ext.	2	KEEX	entero
r	Cred. obligatorios	4	COBL	entero
s	Cred. optativos	4	COPT	entero
t	Periodo inicial cursado en lic.	4	PERIN	entero
u	Periodo de ultima inscripcion	4	PEUL	entero

IID Archivo de alumnos de reingreso

Es el archivo obtenido por promedios ascendentes.

Formato:

CAMPO:																						
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	
CARACTERES:																						
32	8	3	2	2	2	2	2	2	10	2	2	2	2	2	2	2	2	3	3	3	3	8

COMO SE PUEDE OBSERVAR, EL FORMATO ES EL MISMO QUE EL ANTERIOR, PERO AGREGANDO UN CAMPO DE 6 CARACTERES DEDICADO AL PROMEDIO.

III) El archivo de salida de este proceso, es el archivo con el total de alumnos de determinada carrera.

formato:

ES EL MISMO QUE EL ANTERIOR.

Si se desea que los alumnos de primer ingreso tengan su inscripción en el primer día asignado, se seleccionaran todos los alumnos indicados y se grabarán en los primeros registros del archivo a generar, enseguida, se grabarán los alumnos de reingreso.

En caso contrario, si se desea que los alumnos de primer ingreso se distribuyan a lo largo de los días de inscripción, o sea que se inscriba a un bloque de alumnos de primer ingreso en cada día, entonces es necesario que se seleccionen a todos ellos, se obtenga su total y se divida este entre el número de días de inscripción. De esta manera, se clasificarán en tantos bloques como días sean, luego se grabarán alternativamente un bloque de alumnos de reingreso y uno de primer ingreso, repitiendo este proceso hasta

el termino del archivo de entrada.

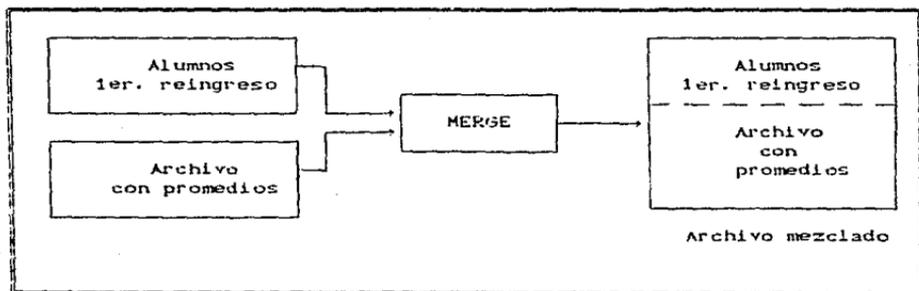
2.13 ASIGNACION DE NUMEROS DE SORTEO ALEATORIAMENTE

Para la obtención de asignación aleatoria de los numeros de sorteo, se mezclan los archivos con promedios de alumnos de reingreso y el archivo con alumnos de 1er. ingreso, no importando su orden.

Teniendo preparado el archivo mezclado, se utiliza un programa generador de numeros aleatorios, para realizar la asignacion de los numeros de sorteo.

2.13.1 PROGRAMA MEZCLADOR

El programa para mezclar los dos archivos, realiza lo siguiente:



2.13.2 PROGRAMA ASIGNADOR DE NUMEROS ALEATORIOS

El programa para realizar la asignación aleatoria de los números de sorteo, se realiza utilizando la función disponible en FORTRAN para generar números aleatorios, para ello se lee el archivo de entrada, el cual es la mezcla del archivo de promedios descendentes con el de alumnos de 1er. reingreso. Se asigna el número aleatorio, el día y fecha de inscripción y se almacena o graba todo el registro resultante en un archivo de acceso directo, esto se hace para que en caso de existir un número repetido, se asigne el siguiente número y si este ya fue asignado, incrementar

el número en uno, hasta que se localice uno no asignado previamente.

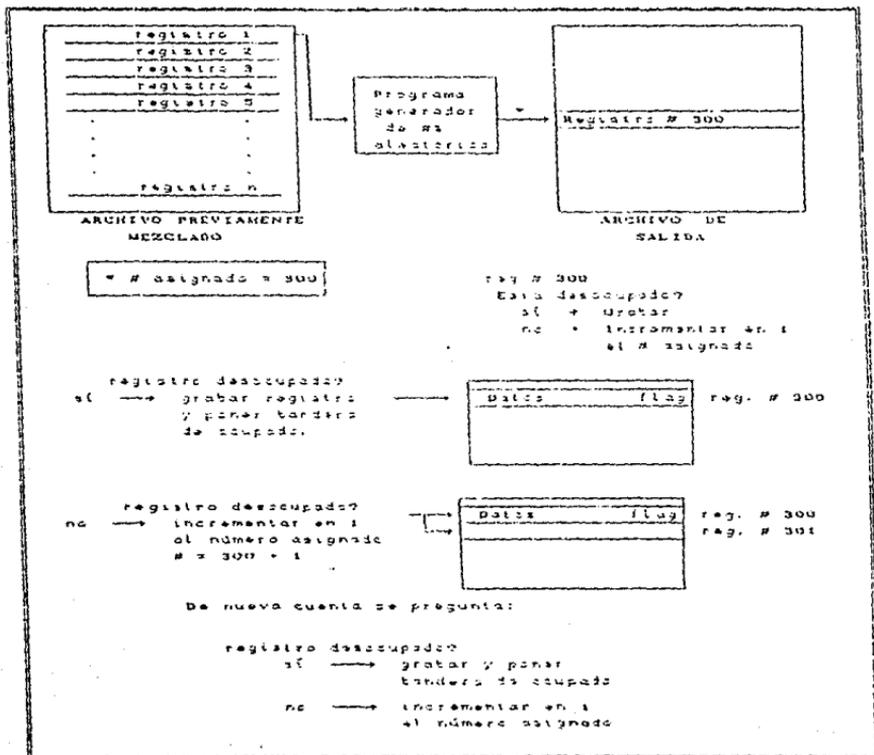
Ejemplo:

1er. registro	_____	→	No. asignado = 300
2o. registro	_____	→	No. asignado = 210
3er. registro	_____	→	No. asignado = 300

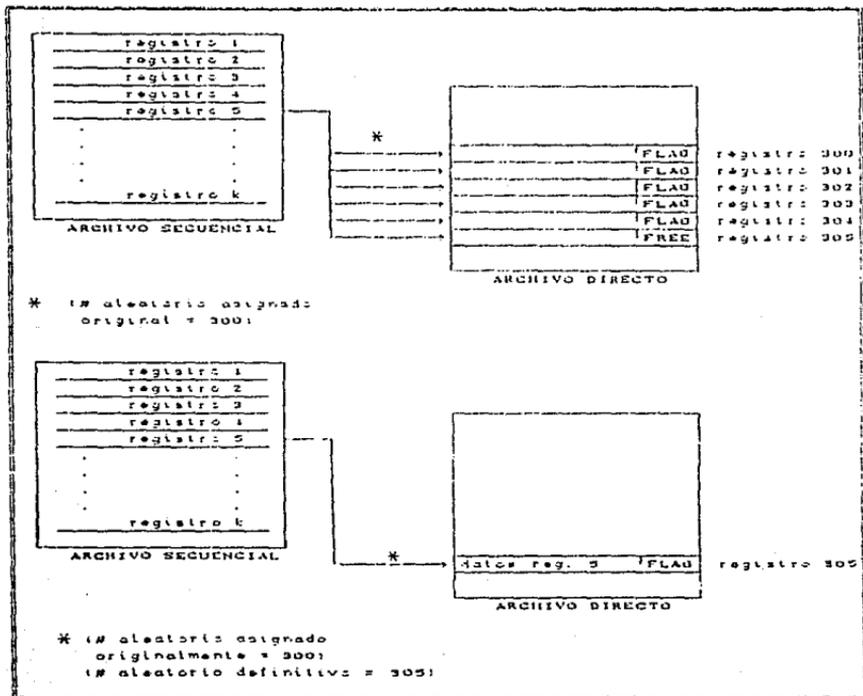
Como el 3er. registro tiene asignado el mismo número de sorteo que el primero, se le asignará el siguiente número y así el registro número 3 leído, tendrá asignado el # 301. En caso de que se verifique que el 301 ha sido asignado previamente a otro registro, se incrementa el número 301 en uno, para tratar de asignar el número 302, si este está libre, se graba el registro de entrada en el registro número 302 del archivo directo, si no, se incrementará en uno el número, hasta encontrar un registro disponible.

El concepto utilizado para saber si un registro está disponible o no, es la utilización de una bandera o indicador de vacío u ocupado.

El siguiente diagrama ejemplifica lo anterior:

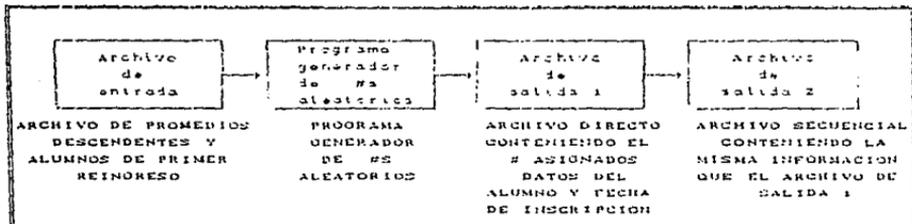


Si el registro desocupado es el 305, hasta ese registro se detiene el proceso.



Si se trataran de grabar los registros de salida en un archivo de acceso secuencial, no podrian saltarse registros y la

modificación del primer número asignado sería muy difícil de realizar.



Listado del programa, en el apéndice "A" (programa 4).

Después de obtener el archivo directo, este se convierte a uno secuencial, el cual estará en disponibilidad de introducirse a la segunda base de datos.

2.14 CLASIFICACION POR DIA DE INSCRIPCION

Al obtener el archivo de numeración descendente (ya sea para semestre par o impar, por promedios o aleatorio), se clasificará de acuerdo a las fechas, número de días de inscripción y número de

alumnos por carrera, para así determinar cuales alumnos se inscribirán en determinado día.

Las fechas oficiales de inscripciones son proporcionadas por la Unidad Académica a la Unidad de Planeación y esta, los turna al Departamento de informática, con ello se estará en disponibilidad de llevar a cabo la clasificación mencionada.

Por ejemplo, si la Unidad Académica, informa que las fechas de inscripciones regulares serán:

Primer día -----	Lunes	21 de octubre
Segundo día -----	Martes	22 de octubre
Tercer día -----	Miércoles	23 de octubre
Cuarto día -----	Jueves	24 de octubre

Se tomará como información de entrada, que existen 4 días de inscripción y la fecha de cada uno de ellos, así, el programa encargado de clasificar por día de inscripción a los alumnos, dividirá el número total de alumnos por carrera entre el número de días de inscripción:

# de alumnos por día	=	# total de alumnos por carrera	/	# de días de inscripción
-------------------------	---	--------------------------------------	---	-----------------------------

ESTRUCTURA DE LA SEGUNDA BASE

\$CONTROL:;
BEGIN DATA BASE:BADAT3:003:26;
LEVELS:

DE DATOS

1 CAP;
10 STOS;
15 MOR;

ITEMS:

NOM, X32(10,15);
CTA, X8(10,15);
CPL, X4(10,15);
CCA, X2(10,15);

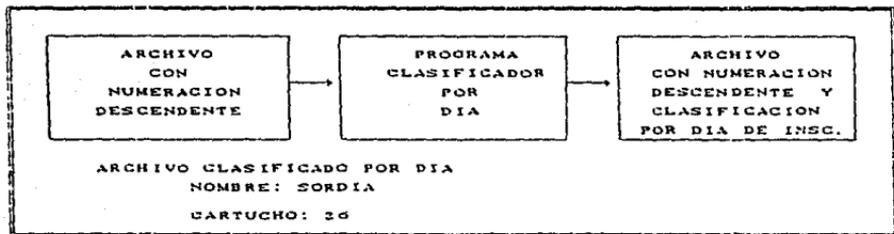
SETS:

NAME:SUB31::26,M;
ENTRY:CCA(1);
CAPACITY:10;
NAME:SUB32::26,D;
ENTRY:

NOM,
CTA,
CPL,
CCA(SUB31);

CAPACITY:9000;
END.

Además de la clasificación, asignará el día y la fecha de inscripción correspondiente. El programa encargado de realizar esta clasificación es PON_DIA.FTN, que realiza la siguiente transformación:



Listado del programa en el apéndice "A" (programa 5).

2.15 CREACION DE SEGUNDA BASE DE DATOS

Una vez obtenido el archivo clasificado por día de inscripción, es necesario crear una segunda base de datos, para poder generar los reportes finales conteniendo los números de

Sorteo asignados.

De nueva cuenta, se crearán los archivos de control, principal y llave.

ARCHIVO RAIZ

NOMBRE UTILIZADO: MRAIZ2

CARTUCHO: 26

Siguiendo los parámetros a indicar en este archivo, son análogos a los de la primer base de datos:

a) Nombre de la base: código de seguridad: cartucho

b) Niveles de acceso:

capturistas
otros
diseñador

c) Variables:

nombre
tipo
longitud
(accesos permitidos)

d) Conjuntos

nombre::cartucho, tipo
llave
capacidad

NOMBRE DE LA SEGUNDA BASE

DE DATOS: BADAT2

CODIGO DE SEGURIDAD: 002

CARTUCHO: 26

ARCHIVO LLAVE

Se crea el archivo llave conteniendo lo siguiente:

ARCHIVO LLAVE

REGISTRO DE CONTROL 1
REGISTRO DE CONTROL 2

REGISTROS DE DATOS

REGISTRO DE CONTROL 3

REGISTRO DE CONTROL 1:

Contiene:

*Nombre de base: codigo, cartucho, acceso
de datos prioritario*

REGISTRO DE CONTROL 2

Contiene:

\$ SET: nombre del conjunto declarado
en el archivo de control

REGISTROS DE DATOS

Contiene:

Claves de carrera, que servirán como llaves.

REGISTRO DE CONTROL 3

Contiene:

\$ END

ARCHIVO PRINCIPAL

El archivo principal, es el archivo clasificado por día
INGDIA::26 más los registros de control.

R de C 1

R de C 2

archivo clasificado por día

R de C 3

CREACION Y ALTA DE 2A. BASE DE DATOS

De igual forma que para la primer base de datos, el alta de la
segunda se lleva a cabo. Los comandos utilizados son en este caso:

DBDS, nombre 2a base :: cartucho, 1

DBBLD, nom_arch_llave :: cartucho,1,ADD

DBBLD, nom_arch_maestro :: cartucho,1,ADD

2.16 GENERACION DE REPORTES

Para la obtención de los reportes definitivos de números de sorteo, se utiliza los siguientes generadores:

- GENERADOR DE REPORTE POR DIA
- GENERADOR DE REPORTE POR NOMBRE
- GENERADOR DE REPORTE POR # DE SORTEO
- GENERADOR DE REPORTE POR CARRERA, DIA Y NOMBRE
- GENERADOR DE REPORTE POR PROMEDIOS

Los listados de estos generadores se muestran en el apéndice "A".

Hasta aquí, se ha detallado el proceso de asignación de números de sorteo, la cual se realiza ciclicamente cada semestre.

2.17 HOJA DE LECTURA OPTICA

¿Que es la hoja de lectura óptica?

El único medio, por el cual el alumno realmente queda inscrito

en determinadas materias-grupo, es la hoja de lectura optica, ya que ésta contiene el formato y especificaciones normalizadas para que pueda almacenarse y procesarse en una computadora la informacion necesaria para la inscripcion.

Los demas documentos expedidos, tales como tiras de materia provisionales, ordenes de pago de reinscripcion, de cambio de turno y otros, sirven para un control interno y administrativo de la escuela y/o como comprobante de haber realizado cierto movimiento de reinscripcion para futuras aclaraciones.

La hoja de lectura optica tiene una distribucion y presentacion previamente aceptada, para que en cada alveolo u ovalo se represente un numero que servira para identificar:

NUMERO DE CUENTA
PLANTEL
CARRERA
MATERIAS
GRUPOS

Luego por medio de un dispositivo de entrada, llamado lector óptico, se aprovecha la reflexion de la luz en la hoja y se interpreta como informacion válida para la computadora.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
DIRECCIÓN GENERAL DE ADMINISTRACIÓN ESCOLAR
SOLICITUD DE REINSCRIPCIÓN

H O J A

NOBRE DEL ALUMNO

PRIMER GRADO

PLANTIL

NUMERO DE CUENTA				PLANTIL				FAVOR DE LEER CUIDADOSAMENTE EL INSTRUCTIVO				
1	0	0	0	0	0	0	0	0	0	0	0	UTILICE LAPIZ DEL NUMERO 2
2	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	

ES RESPONSABILIDAD DEL ALUMNO EL LLENADO DE ESTA HOJA. EN CASO DE ERROR NO PROCEDERA LA INSCRIPCION

INSTRUCTIVO

1. ANOTE CUIDADOSAMENTE LAS CLAVES DE LAS ASIGNATURAS Y LOS GRUPOS Y LOS QUE HA SIDO AUTORIZADO SU PRIMER GRADO
2. RELLENE COMPLETA Y ANIMAMENTE EL DIVIDIO QUE CORRESPONDA A CADA UNO DE LOS INDICIOS MENCIONADOS. (Ver ejemplo en el ejemplo)
3. UTILICE UN CUADRO PARA CADA ASIGNATURA Y SIGA EL ORDEN NUMERICO DE LOS CUADROS.
4. EN CASO DE ERRORES HAGA COMPLETAMENTE SIN DARNOS LA HOJA. SI HUBIERA NECESARIO, SOLICITE OTRA HOJA Y COMPLETELA CORRECTAMENTE.

EJEMPLO

1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

FIRME EN ESTE LUGAR

FIRMA DEL ALUMNO

CLAVE DE ASIGNATURA				NUMERO DE GRUPO			
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

CONTINUA EN LA PAGINA SIGUIENTE

MADE - 111

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

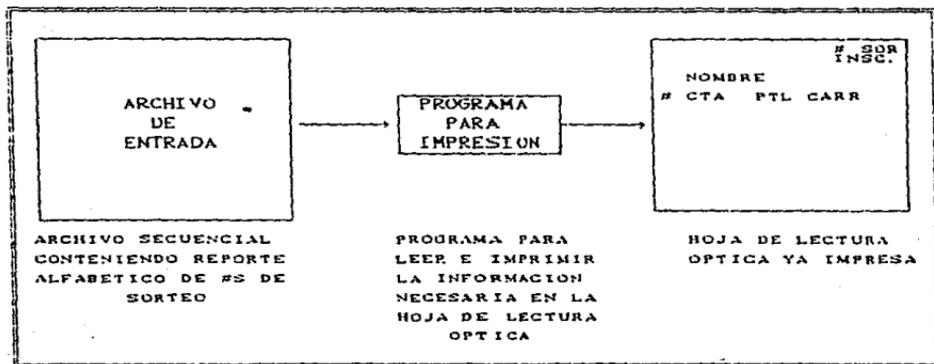
2.18 IMPRESION DE HOJAS DE LECTURA OPTICA PARA INSCRIPCION

Debido a la alta probabilidad de error, existente al dejar que el alumno realice el llenado de los alveolos correspondientes al numero de cuenta, clave de plantel y clave de carrera, se realiza el proceso de impresion de estos campos, por medio de la computadora, lo cual reduce la posibilidad de error de llenado y por ende de la no lectura² de esta informacion, lo que trae como consecuencia que el alumno no quede inscrito. Asi, el alumno queda como responsable unicamente, del correcto llenado de los campos de la hoja correspondientes a los grupos y claves de materia, en los cuales desea inscribirse.

Ademas de la reduccion de posibilidad de error de lectura, con la impresion de las hojas de lectura optica, se obtiene un mayor control acerca de los alumnos que realizaran o pueden realizar su reinscripcion, ya que estas se imprimiran por orden alfabetico, con el numero de sorteo asignado, asi como con la fecha o dia correspondiente de inscripcion. Esto facilita enormemente el tramite administrativo de inscripcion por medio de hoja de lectura optica y evita un costo elevado en la distribucion de las hojas.

² Realizada mas tarde por un lector optico en la DGAE.

El diagrama a bloques del proceso de impresión de las hojas de lectura óptica es el siguiente:



2.18.1 PROGRAMA PARA REALIZAR LA IMPRESION DE HOJAS DE LECTURA OPTICA

La impresión de número de cuenta y plantel se logra por medio de la conversión de estos valores a una matriz de 10 renglones y 13

columnas:

	8	0	2	5	2	8	4	4	4	1	1	3	2
	1	2	3	4	5	6	7	8	9	10	11	12	13
5													
8	3					3							
7													
6				3									
5													
4							3	3	3				
3												3	
2			3		3								3
1										3	3		
0		3											

Como se vió, el proceso de impresion de las hojas de lectura óptica, es muy útil y adecuado para una institución tal como la ENEP APAGON. La aplicación de este proceso puede realizarse en forma inmediata, logrando así, gozar al máximo de las ventajas obtenidas con el mismo (programa 8, apéndice A)

**CUANTIFICACION Y CUALIFICACION DE LA INFORMACION
OBTENIDA DURANTE LA ETAPA DE LA INSCRIPCION**

Una vez que se han obtenido los reportes correspondientes a numeros de sorteo, además de la impresión de las hojas de lectura optica, para el total de alumnos que están en disponibilidad de realizar su reinscripción, se entregan estos a la Unidad Académica, la cual lleva la responsabilidad del manejo y control de la inscripción de alumnos.

En fechas proximas a la realización de la inscripción, se reparte a los alumnos, la papeleria necesaria para poder realizar su pago de inscripción, planes de estudio y otras indicaciones necesarias.

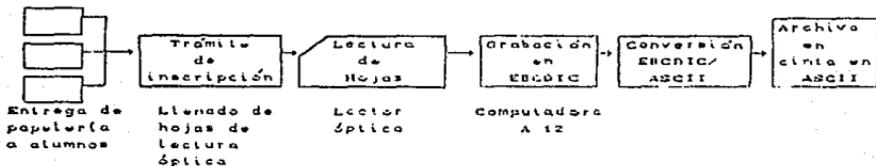
En los días fijados para la inscripción, se recopila toda la información necesaria para que los alumnos se den de alta en las materias-grupo que hayan seleccionado. Cada coordinación de carrera junto con el Departamento de Servicios Escolares (dependiente de la Unidad Académica), realizan las actividades necesarias para los trámites de preinscripción así como de la propia inscripción.

Una vez que han transcurrido los días para la inscripción normal, se da un día para inscripción extemporánea, para aquellos alumnos que por diversas causas no realizaron su trámite en la fecha correspondiente.

El total de hojas de lectura óptica, conteniendo la

información correspondiente a toda la inscripción de la FNEP ARAGON, se entrega al DEPARTAMENTO DE INFORMATICA, el cual se encargara de realizar la lectura optica.

Para la lectura, se trasladan las hojas a la DGAE, donde se hace ésta, hoja a hoja por medio de un lector óptico. Esta información se almacena en una cinta magnética en código EBCDIC, el cual se transformará a código ASCII. Tal código, como se vio al inicio del capítulo II, es el que utiliza la HP-1000.



Los jobs o trabajos, necesarios para convertir el archivo de inscripción en código EBCDIC a uno en código ASCII, se diseñan para trabajar en la Computadora A-12.

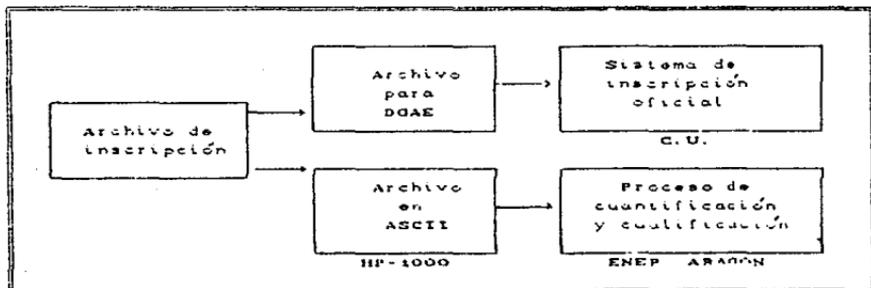
Estos jobs, se manejan en un ambiente de trabajo mediante sistema operativo CANDE¹.

La cinta conteniendo el archivo de inscripción se trasladara a

¹ Es un sistema operativo utilizado en la computadora A12 de la DGAE.

la ENEP ARAGÓN, para que se almacene en disco o cartucho de la HP-1000 y pueda ser procesado.

El archivo resultante obtenido de la lectura de las hojas, es turnado de manera oficial, para que la DGAE, por medio de su sistema de inscripción, lleve a cabo el alta de los alumnos en sus respectivas materias-grupo:

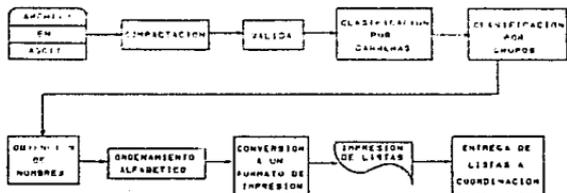


3.1 FILTROS DE DEPURACION Y VALIDACION

PARA EL ARCHIVO DE INSCRIPCION

PROCESO DE OBTENCION DE LISTAS DE ASISTENCIA

Para la obtención de las listas de asistencia de toda la ENFP ARAGON, se lleva a cabo la secuencia indicada en el diagrama:



3.1.1 COMPACTACION

Una vez que el *DEPARTAMENTO DE INFORMATICA* cuenta con el archivo de inscripcion, se procede a aplicarle una serie de filtros

de depuración y validación para que éste, quede listo para su cuantificación y cualificación.

El primero de los filtros referidos, consiste en compactar la información contenida en el archivo de inscripción original, ya que éste, presenta un formato de la siguiente manera:

<u>M I</u>	caracteres 1 y 2 : identificación del archivo de inscripción.
<u>8 8 0 0 2 2 6 6 2 2 0 0 4 4 4 4</u>	caracteres 3 a 18 contiene el número de cuenta con cada dígito repetido
<u>4 4 1 1 1 1</u>	caracteres 19 a 24 contiene la clave del plantel con cada dígito repetido
<u>3 3 2 2</u>	caracteres 25 a 28 contiene la clave de carrera con cada dígito repetido

Así, si el número de cuenta del alumno es por ejemplo:

80262844

Su clave de plantel es:

411

Y la clave de carrera es:

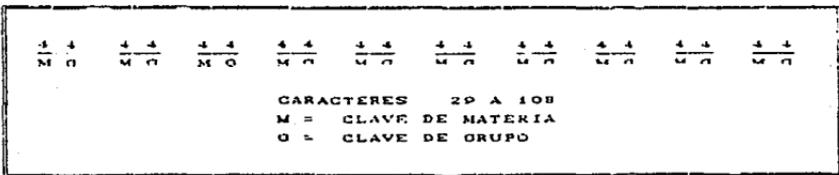
32

El registro aparecería de la siguiente forma:

MI8800226622884444441113322

Esto se hace, para que en dado caso que un dígito no fuera leído, el otro sí lo fuera, reduciendo la posibilidad de error y con ello, que el alumno en cuestión sí sea tomado en cuenta.

Los campos dedicados a las materias, contienen cada uno un total de 8 caracteres, 4 para la clave de materia y 4 para el grupo correspondiente, después de cada campo, existe un espacio en blanco, el total de campos posibles es igual a 10 por alumno, para que éste pueda inscribirse a un número igual de materias-grupo:



El programa utilizado para realizar la compactación del archivo es: PAS01.FTN (Apéndice 'A' programa 7)

3.12 VALIDACION

Una vez compactado el archivo de inscripción, se procede a realizar la validación del mismo, para esto, se utiliza un

programa para verificar que en todos los campos del archivo exista información de tipo numérico y no presente caracteres nulos o asteriscos (*), los cuales aparecen cuando el alumno no llenó en forma correcta los óvalos correspondientes.

El archivo original, presenta información parecida a la siguiente (en el caso de errores de llenado):

802628444113200**3102 ** 41205 00612103 00142106 ERROR ERROR

En los campos correspondientes a materia-grupo, número uno y dos, el alumno al llenar incorrectamente los óvalos, originó que aparecieran asteriscos y blancos en estos campos, sustituyendo al número que el alumno deseaba apuntar.

Los caracteres asterisco y blancos ocasionan un error en los subsiguientes procesos, debido a que los valores que se toman en cuenta son de tipo numérico y con el error, estos aparecen con valores de tipo alfanumérico (* ó espacio en blanco). Al tratar de realizar asignaciones numéricas con valores alfanuméricos, se presenta una incongruencia en los datos y por ello, el proceso no puede continuar.

Por lo anterior, el programa de validación elimina los campos

erróneos, colocando en ellos tantos ceros como tenga caracteres el campo. En este caso, como el campo materia-grupo consta de 8 caracteres, el número de ceros colocados es de 8.

Haciendo referencia al ejemplo del alumno que llenó mal su hoja, el registro correspondiente quedaría así:

802628444113200000000 00000000 00612103 00142105
ANULADO ANULADO

Como puede observarse, los campos erróneos fueron llenados con ceros, o sea fueron validados.

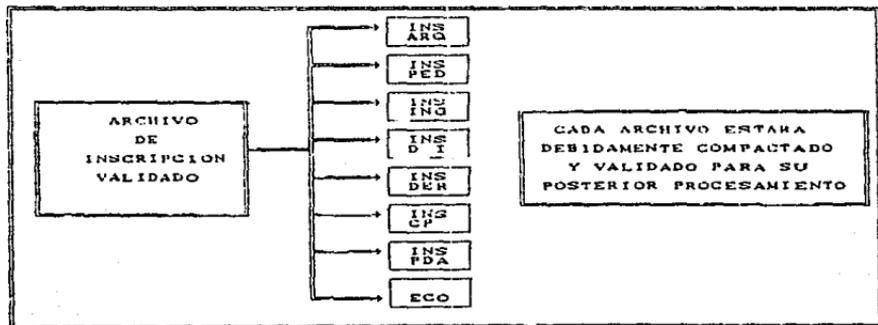
Una vez sustituidos en todo el archivo los campos erróneos, se obtiene un archivo de inscripción con datos válidos únicamente. La sustitución por ceros, se hace debido a que estos no se tomarán en cuenta en los siguientes procesos a realizar.

El listado del programa de validación se presenta en el apéndice "A". (*Programa 8*)

3.2 CLASIFICACION DE ARCHIVO DE INSCRIPCION POR CARRERAS

Al haber validado el archivo de inscripción, se procede a realizar la clasificación de los archivos de inscripción por carreras, los cuales servirán para lograr la obtención de las listas de asistencia de todos los grupos de la ENEP ARAGON.

La clasificación de archivos será de la siguiente forma:



Consultar el apéndice "A" para ver el listado del programa que clasifica carreras (programa Clas_ca.ftn programa 9).

3.3 CLASIFICACION POR GRUPOS

El siguiente proceso en la obtención de la listas de asistencia, es el realizar la clasificación de alumnos por grupos inscritos. Para ello, se utiliza un programa que genera una serie de arreglos, conteniendo: El número de cuenta, plantel, clave de carrera y número del alumno inscrito en cada grupo.

La utilización de arreglos para este programa, es esencial, ya que se evitan búsquedas secuenciales para cada grupo y que se realicen tantas lecturas del archivo de inscripción, como materias-grupo existan en cada carrera.

La clasificación por grupos del archivo, debe realizarse de la siguiente manera:

Formato del archivo de inscripción:

<u>M</u>	<u>I</u>									
<u># de cuenta</u>										
<u>p</u>	<u>t</u>	<u>l</u>	<u>C</u>	<u>a</u>						
<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>	<u>4 4</u>
<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>	<u>M O</u>

Si por ejemplo, el archivo de inscripción constara de 5

alumnos (5 registros) y estos se presentarán como a continuación:

8620313341126 00602201 00000000 00622207 00632202 09942202 00000000
00000000 00000000 00000000 00000000 00000000

8435882841126 00622202 04582202 09942202 00602201 00592103 00000000
00000000 00000000 00000000 00000000 00000000

8606566741132 00612701 00632202 00622207 04582203 00192205 00000000
00000000 00000000 00000000 00000000 00000000

8616650641132 04582203 00622202 00632202 00612205 00192302 00000000
00000000 00000000 00000000 00000000 00000000

8511314341126 00602202 00622201 00632202 04582202 09942202 00000000
00000000 00000000 00000000 00000000 00000000

La clasificación por grupos se debe realizar como sigue:

Materia: 0019 Grupo: 2205
Alumnos inscritos:
86065667

Materia: 0061 Grupo: 2205
Alumnos inscritos:
86166506

Materia: 0019 Grupo: 2302
Alumnos inscritos:
86166506

Materia: 0062 Grupo: 2207
Alumnos inscritos:
86203133
86065667

Materia: 0059 Grupo: 2103
Alumnos inscritos:
84358828

Materia: 0063 Grupo: 2202
Alumnos inscritos:
86203133
86065667
86166506
85113143

Materia: 0060 Grupo: 2201
Alumnos inscritos:
86203133
84358828
86065667

Materia: 0458 Grupo: 2202
Alumnos inscritos:
84358828
85113143

Materia: 0060 Grupo: 2202
Alumnos inscritos:
84358828

Materia: 0458 Grupo: 2203
Alumnos inscritos:
86065667
86166506

Materia: 0082 Grupo: 2201
Alumnos inscritos:
85113143

Materia: 0062 Grupo: 2202
Alumnos inscritos:
86166506
85113143

Materia: 0994 Grupo: 2202
Alumnos inscritos:
86203133
84358828
85113143

Como se puede observar, la clasificación por grupos debe considerar una total aleatoriedad en la materia-grupo presentada en cada campo declarado, puesto que el orden encontrado, así como la misma materia-grupo, no serán las mismas para cada alumno. Además, el número de materias grupo en que se inscribe cada alumno, también es variable. Por lo tanto, no existe forma de asignar u ordenar previamente el archivo, de tal manera que pueda clasificarse secuencialmente.

Si se deseara hacer una clasificación del archivo, grupo por grupo, se implican una serie de problemas tales como los siguientes:

- 1) Búsquedas secuenciales de cada alumno inscrito en una materia-grupo específica.
- 2) Tiempo exagerado de procesamiento al hacer un número de búsquedas igual al número de materias existentes.
- 3) Creación de demasiados archivos al clasificar por materia-grupo.
- 4) Falta de capacidad de almacenamiento para el número tan grande de archivos que se crearían clasificados por materia-grupo.

Tomando en cuenta los problemas anteriores, se concluye fácilmente, que no es operante el utilizar un proceso que implique

búsquedas secuenciales. por tanto. la utilización de arreglos o vectores se hace necesario. ya que estos funcionan de manera análoga a un archivo directo.

Los arreglos se usarán como se indica a continuación:

1) Lectura del total del registro por alumno.

formato del registro de inscripción:

2 caracteres para identificación	MI
8 caracteres de número de cuenta	CTA
3 caracteres de plantel	PTL
2 caracteres de carrera	CARR
Repetido 10 veces:	
4 caracteres de clave de materia ... (PRIMER SUBINDICE)	MATE
4 caracteres de clave de grupo	GRUPO

2) Asignación en el arreglo de primer subíndice igual a la clave de materia y segundo el # de alumno inscrito en esa materia. los cuales servirán como punteros para señalar un alumno o materia en específico.

de cuenta
plantel
grupo
alumno

} DATOS A ALMACENAR EN ARREGLOS

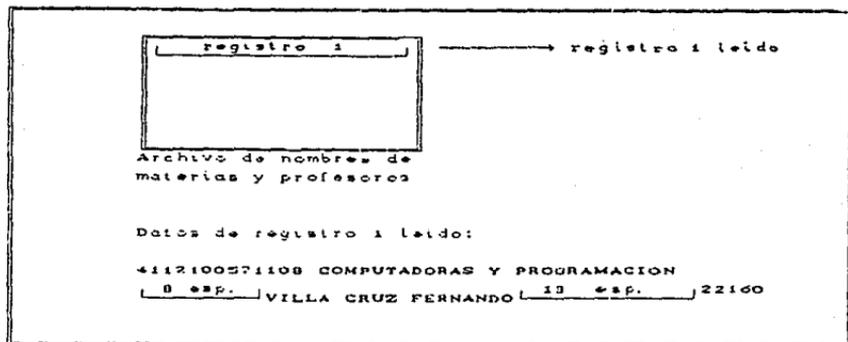
CUANTIFICACION Y CUALIFICACION

3) La asignación de los arreglos se lleva a cabo con una lectura UNICA secuencial del archivo de inscripción. Así al terminar de leer el archivo, se habrán asignado a todas las materias-grupo indicadas, el total de alumnos inscritos en ellas.

4) Solo se asignan o toman en cuenta, las materias-grupo con cierta terminación previamente señalada. Esto es, se clasifican las materias-grupo terminadas en 01,02,03,51,51.53, etc. sin importar si son correspondientes a los grupos 1101, 1201, 1301, 1202, 1302, 1402, 1103, 1203, 1303, 1151, 1251, 1351, 1172, 1252, 1352, 1153, 1253, o 1353, según corresponda.

5) Una vez realizada la asignación de los arreglos o matrices, se hace una lectura secuencial del archivo correspondiente a materias y profesores de cada carrera, identificando la terminación del número de grupo y verificando que pertenezcan a determinada materia, para luego unir el arreglo correspondiente con su nombre de materia, profesor, grupo, plantel, carrera, etc.

Ejemplo:



De los datos anteriores, se identifica la clave de materia y la terminación del grupo:

CMAT = 0057

Terminación del gpo. = 08

Si la terminación del grupo no corresponde a la predeterminada, el registro es ignorado y se lee el siguiente.

Al identificar la clave y terminación, se puede tener referencia sobre la matriz en particular del grupo en cuestión, recordando que el primer subíndice es la clave de materia y el segundo es la terminación de número de grupo.

Para el ejemplo, se accederá a la matriz llamada A, con primer subíndice igual a 57 y segundo subíndice igual a 8:

CALUC 57, 8)

Para determinar el número de alumnos existente en este grupo, es por lo que se accesa esta matriz. El valor contenido en esta, es igual al número de alumnos registrados en la materia con clave 0057 y terminación de Número de grupo 08.

Si el valor de CALUC(57,8) es 30, significa que se registraron 30 alumnos en la materia-grupo.

Para saber cuales son los números de cuenta de los alumnos registrados en esta materia-grupo, se utiliza otra matriz "C" que contiene como primer subíndice la clave de la materia en cuestión y como segundo subíndice, el número de alumnos registrados en ella.

Se utiliza un ciclo de índice automático "DO", para ir obteniendo los números de cuenta clasificados desde el primero hasta el último, que en este caso sería el 30.

Al detectar una materia con terminación de grupo predeterminada, se graba el registro completo, en un archivo secuencial. Posteriormente, al utilizar el ciclo de índice automático, se grabarán los números de cuenta clasificados para la materia-grupo en cuestión, todos en el mismo archivo secuencial donde se grabó el registro de nombre materia y profesor.

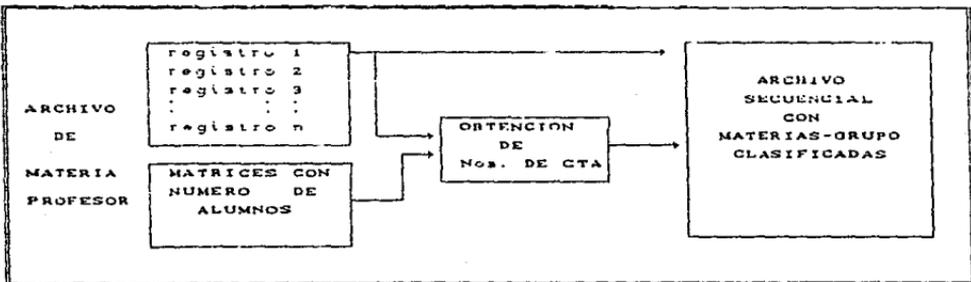
Una vez grabado el anterior grupo, se procede a localizar en el archivo de materias y profesores, una nueva materia-grupo con la

terminación de grupo predeterminada y de nueva cuenta, se tomará la clave de materia como primer subíndice y terminación de grupo como segundo subíndice, para apuntar a la matriz correspondiente que contiene el número de alumnos inscritos en esa materia-grupo.

A continuación se procede con la grabación del registro leído de materia-profesor y los números de cuenta clasificados para la materia-grupo.

El proceso anterior se realizará tantas veces como registros del archivo materia-profesor cumplan con la terminación de grupo fijada, o hasta el fin de archivo de materia-profesor.

El diagrama siguiente, muestra el flujo de información:



VER APENDICE "A" PARA CONSULTAR EL LISTADO DEL PROGRAMA VERIFICA.FTN (PROGRAMA 10).

3.4 BUSQUEDA DE NOMBRES A LOS GRUPOS CLASIFICADOS

Al haber realizado la clasificación de los grupos por número de cuenta, es necesario buscar el nombre correspondiente a cada uno, ya que se localizan en un archivo independiente del archivo de inscripción.

Para lograr la obtención del nombre del alumno teniendo su número de cuenta, es necesario realizar un proceso previo e independiente del clasificatorio por grupos.

El archivo utilizado como fuente del nombre, será el de historia académica (de tipo secuencial). Este archivo se convertirá en uno de acceso directo con número de registro equivalente al número de cuenta y así poder apuntar a él en forma directa y no tener que realizar búsquedas secuenciales, evitando con ello, un tiempo de proceso excesivo.

En este punto, se presenta un inconveniente, el cual consiste en usar el número de cuenta como número de registro.

La capacidad de un archivo está limitada a 32767 registros y el número de cuenta consiste de 8 dígitos, lo cual hace imposible el direccionamiento con el número de cuenta real.

Dada la restricción anterior, es necesario utilizar una técnica de conversión de claves, llamada comúnmente "HASH".

En el apéndice "B" se explica en qué consiste este método.

El método utilizado, es el hash por división entre un número primo, el cual será el número de registro máximo a acceder.

De acuerdo a la población estudiantil observada en la ENEP Aragón, el número primo 15737, puede contener al número total de alumnos a inscribirse, teniendo una holgura de algunos miles de registros disponibles. Además, el número 15737, permite la reducción de colisiones y la segregación de registros utilizados. Logrando con esto, que el archivo directo sea lo más compacto posible (Pudiendo modificarse tal número, de acuerdo al crecimiento poblacional de la ENEP ARAGON).

Pasos para crear el archivo directo conteniendo nombre, clave de plantel, carrera, número de cuenta y bandera de ocupación:

1) Inicializar 3 archivos directos con 20000 registros. Grabando en cada uno de ellos lo siguiente:

32	blancos en el campo del nombre
5	blancos en el campo de plantel y carrera
00000000	en el campo de número de cuenta
ARA	en el campo de bandera

VARIABLE	LONGITUD	TIPO
ESCU	5	ALFANUMERICA
NOMBRE	32	ALFANUMERICA
CTAA	8	NUMERICA
BAN	3	ALFANUMERICA

ARCHIVOS A INICIALIZAR :

DIREC_INGE

DIREC_DERE

DIREC_ORAL

2) Separar el archivo de historia académica por Ingeniería, Derecho y otras².

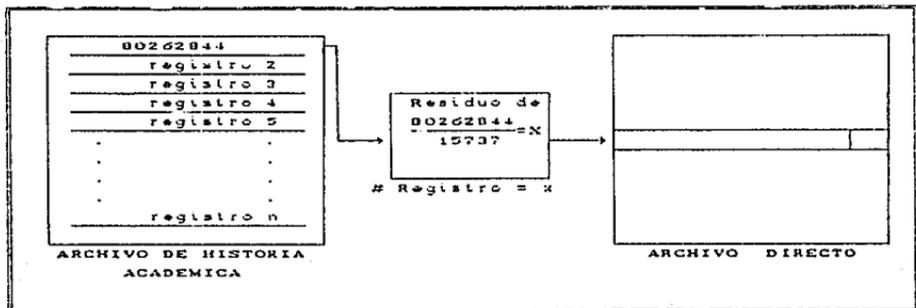
3) Leer un registro del archivo de historia académica, identificar el número de cuenta y transformarlo por hash a un número de registro adecuado.

Ejemplo

$$80262844 \text{ mod } 15737 = 4144$$

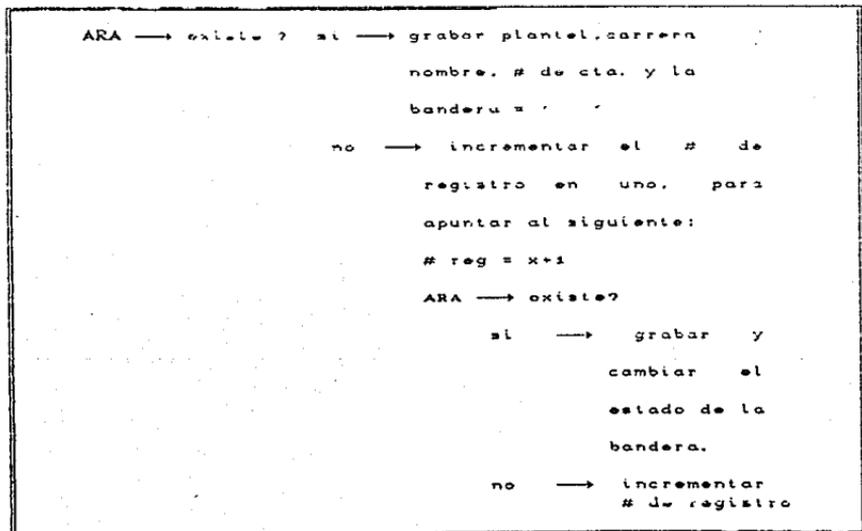
4) Accesar el archivo directo inicializado en el paso 1 apuntando al registro correspondiente:

² La separación se lleva a cabo para no saturar de sobremanera al archivo directo. Se crearan 3 archivos directos: direc_inge, direc_dere, y direc_gral.



5) Verificar que la bandera "ARA" exista.

Mientras la bandera no exista:



5) Repetir proceso indicado desde el paso 2 , terminar si hay fin de archivo.

Al detectarse el fin de archivo, se habrán dado de alta en el archivo directo, a todos los alumnos del archivo de historia académica. Otro archivo fuente para dar de alta, es el archivo de alumnos de primer ingreso. Realizando con el, exactamente el mismo proceso que con el de historia académica, y así anexarlo al archivo directo correspondiente.

En el apéndice "A", se encontrará el listado del programa dirar.ftn (programa 11).

Una vez generados los archivos directos, se podrá acceder a ellos para obtener el nombre de cualquier alumno dado de alta en ellos, con solo tener el número de cuenta del mismo.

El archivo clasificado por materias-grupo, con número de cuenta y nombre de materia, se leerá y procederá como a continuación:

- 1) Leer un registro.
- 2) Identificar el número de cuenta.
- 3) Realizar el hash con número de cuenta/15737 para obtener el número de registro a buscar en archivo directo.
- 4) Acceder al archivo directo correspondiente, apuntando al registro obtenido por hashing.
- 5) Leer el registro del archivo directo.
- 6) Comparar el número de cuenta leído del archivo directo con el del archivo clasificado por materias-grupo.

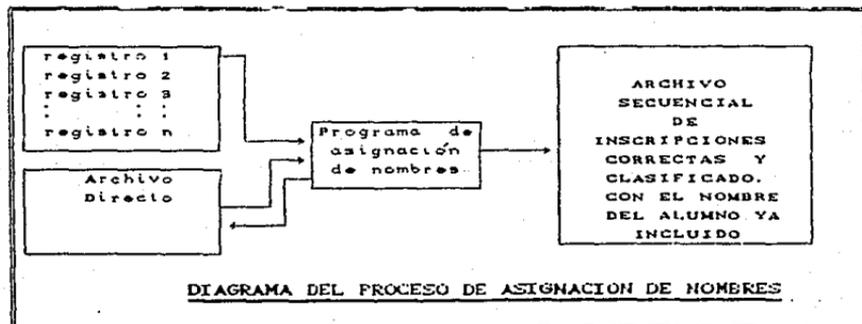
cla. directo = # cla clasificado?

si → El alumno se ha localizado.

Grabar en un archivo de salida el registro del archivo clasificado, además del nombre obtenido.

no → Incrementar # de registro del archivo directo, comparar y repetir el paso 6 tantas veces como sea necesario o hasta que se localice un registro vacío.

Ver el apéndice "A" para consultar el listado del programa busnom2.ftn (programa 12).



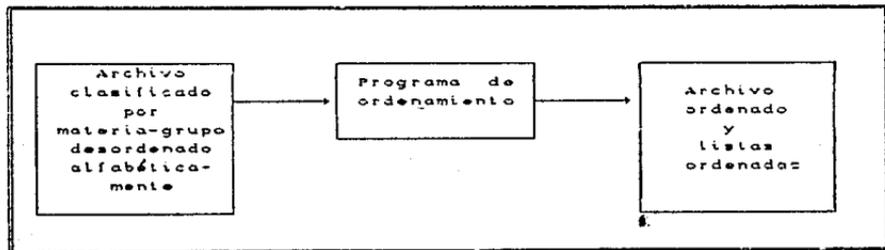
3.5 ORDENAMIENTO ALFABETICO

Cuando se ha obtenido el archivo clasificado por grupos, con números de cuenta y nombres de alumnos, es necesario ordenar cada materia-grupo alfabéticamente y tener un archivo clasificado por materia-grupo con los números de cuenta y nombres de alumnos en forma adecuada.

El programa para realizar el ordenamiento alfabético, está escrito en lenguaje Pascal, el cual permite una mayor velocidad de procesamiento en ordenamientos.

El nombre del programa es Ordena.Pas (Ver el apéndice "A" para consultar el listado del programa 13).

El proceso se esquematiza a continuación:



3.6 GENERACION DE FORMATO PARA LA IMPRESION DE LAS LISTAS DE ASISTENCIA

El siguiente paso en el procesamiento de la información, para la obtención de las listas de asistencia, es el aplicar un programa que "acomode" la información del archivo ordenado alfabéticamente por nombre, a un formato útil para la impresión de cada lista en las hojas diseñadas con tal fin.

El programa agrega únicamente información correspondiente a:

1) Créditos sobre la creación de cada lista, en este caso es el mensaje:

ENEP ARAGON UNAM DEPARTAMENTO DE INFORMATICA
--

- 2) Número de hoja
- 3) Número de página
- 4) Salto de hoja

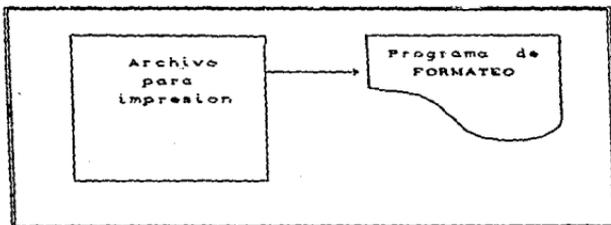
El programa para la obtención del archivo de impresión es FORM_LISTA.FTN (ver apéndice "A" para consultar el listado del programa 15).

Los archivos obtenidos son:

```

LING1 ... LING8  LCP1 ... LCP7  LPED1 ... LPED3  LARQ1 ... LARQ2
LING51...LING59  LCP51...LCP57  LPED51...LPED53  LARQ51...LARQ52
LDER1 ... LDER8  LDER51...LDFR57  LPDA1 ...LPDA51  LDII ... LDII2
  
```

Como último paso, se mandará a impresión el archivo generado de cada lista, ya separadas por plantel, carrera y materia-grupo.



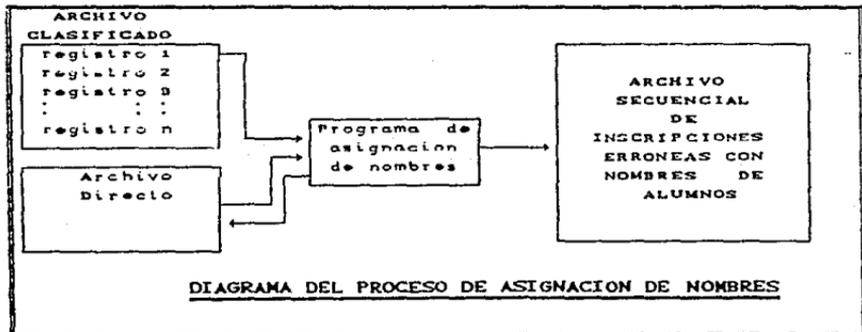
Las listas se entregan por parte de la Unidad de Planeación a la Unidad Académica, para que esta realice la repartición a cada una de las coordinaciones de carreras de la ENEP Aragón.

3.7 OBTENCION DE LISTAS DE ALUMNOS CON ERRORES EN SU INSCRIPCION

El archivo secuencial con registros correspondientes a aquellos alumnos con error en su inscripción, se tratará como si este correspondiera al archivo clasificado por materia-grupo, con números de cuenta.

De este modo, se leerá cada registro de este archivo y procederá a buscar el nombre del alumno cuyo número de cuenta sea el leído en el registro.

La búsqueda se realiza de igual manera que en el proceso de obtención de listas de asistencia, con la salvedad de que no todos los nombres podrán encontrarse, debido a que si un número de cuenta es erróneo, no podrá relacionarse el nombre correspondiente.



El ordenamiento alfabético se hace necesario, para realizar la impresión del archivo resultante y así lograr una mayor claridad, para que todo alumno pueda verificar si su inscripción fue correcta o incorrecta.

De ser incorrecta la inscripción, el alumno tendrá esta información en un tiempo por demás adecuado y útil, puesto que de inmediato podrá recurrir al Departamento de Servicios Escolares para realizar el trámite de inscripción correctamente.

Pasos a seguir para la obtención de la lista de alumnos con error en su inscripción:

- 1) Detectar aquellos alumnos con inscripción incorrecta.
- 2) Obtención de los nombres de alumnos correspondientes a los números de cuenta leídos.
- 3) Ordenamiento alfabético por nombre del archivo de inscripción errónea.
- 4) Impresión del archivo ordenado alfabéticamente.

3.8 OBTENCION DE ESTADISTICAS DE INSCRIPCION DE ALUMNOS EN LAS MATERIAS-GRUPO

El proceso para obtener los totales de alumnos inscritos en cada materia-grupo, se deriva de la obtención del archivo clasificado por materia-grupo con números de cuenta, utilizado en la obtención de las listas de asistencia. El programa generador de estadísticas, almacena el nombre de la materia, grupo, nombre de profesor, plantel, carrera y total de alumnos inscritos en ella.

El programa que obtiene las estadísticas de inscripción es ESTADISTICA.FTN y CUPOS_GPOS.FTN (Ver apéndice "A" para consultar el listado de los programas 16 y 17).

Además de las estadísticas mencionadas anteriormente, se genera una estadística total por carreras y semestres en general.

3.9 GRAFICAS DE ESTADISTICAS DE INSCRIPCION

La graficación de las estadísticas de inscripción total, se realiza, utilizando el "paquete" llamado STORY BOARD o QUATRO propio para computadora personal compatible con IBM¹.

¹ STORY-BOARD es un paquete conformado por 4 programas, el cual permite obtener además de otras muchas cosas, graficas de cuatro tipos: Linea, pastel, barras.
QUATRO es un paquete de hoja electronica, y capacidad de graficación, que permite importar datos en código ASCII y existe la comunicación entre HP-1000 y PC.

Con una gran facilidad, se obtendrán gráficas claras e ilustrativas del resultado de la inscripción en la ENEP Aragón. Las cuales darán puntos de apoyo para futuros procesos y/o mejoras necesarias a realizar de cualquier índole.

3.10 GENERACION DE TIRAS DE MATERIAS PROVISIONALES

Para este proceso, se retoma el archivo de inscripción validado INSGRAL, separado por carreras y de este se parte para la emisión de tiras de materias provisionales.

El proceso consiste en lo siguiente:

1) Generar archivos directos conteniendo, clave de plantel, clave de carrera, nombre de materia y nombre de profesor en cada registro.

Se generarán tantos archivos directos, como carreras existan en la ENEP Aragón.

2) Clasificar las claves de materia-grupo por alumno inscrito.

3) Buscar en el archivo directo de nombres, los indicados por el número de cuenta del registro del archivo de inscripción.

4) Buscar en los archivos directos de nombres de materias y profesor, los indicados por las claves de materia-grupos leídos de inscripción.

5) Grabar en archivos secuenciales, los archivos clasificados por alumno, con su respectivo nombre, además de la clave materia-grupo con nombre de materia y profesor correspondiente.

6) Dar un formato adecuado de impresión al archivo clasificado por alumno.

7) Imprimir tiras de materias provisionales.

3.11 GENERACION DE ARCHIVOS DIRECTOS

Para generar cada archivo directo conteniendo clave de carrera, plantel, nombre de materia y profesor, se recurre a los archivos secuenciales de alta de grupos, los cuales tienen la información anterior en sus registros.

Los archivos secuenciales son:

PLN401
PLN404
PLN407
PLN408
PLN410
PLN411

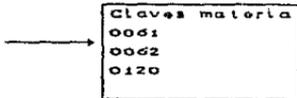
El número del registro en el archivo directo, estará dado por la clave de materia correspondiente, pero con una conversión de clave. Esta conversión de clave se hace para que el archivo directo sea:

- a) Compacto (que no existan registro vacíos)
- b) El archivo no tenga una longitud excesiva

La conversión de clave se realiza de la siguiente manera:

- 1) Se inicializa un contador con valor=1
- 2) Se lee el primer registro
- 3) Se graba la información del registro de archivo secuencial, en el # de registro = contador
- 4) Se incrementa el contador en uno y se procede a leer el siguiente registro del archivo secuencial, hasta el fin de archivo.

Ejemplo:



```

Claves materia
0001
0002
0120
  
```

```
Contador = 1
```

```
# reg (archivo directo) = 1
```

```
Información a grabar:
```

```
0001 Grupo Materia Profesor
```

```
Contador = Contador + 1
```

```
# reg = 2
```

```
Información a grabar:
```

```
0002 Grupo Materia Profesor
```

```
Contador = Contador + 1
```

```
# reg = 3
```

```
Información a grabar:
```

```
0120 Grupo Materia Profesor
```

De la manera anterior, se tendran tantos registros en archivo directo, como registros existan en el archivo secuencial. Además, el incremento de # de registro, sera unitario, compactando

Totalmente el archivo directo.

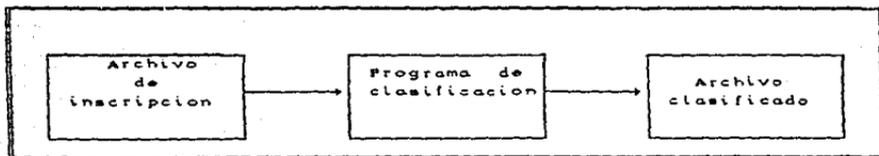
El programa generador de archivos directos se encuentra listado en el apéndice A (programa 18).

3.12 CLASIFICACION DE CLAVES MATERIAS-GRUPO POR ALUMNO INSCRITO

Se toma como archivo de entrada, el archivo secuencial de inscripción validado. Leyendo registro a registro y separando este en:

Numero de cuenta
Clave materia-grupo 1
Clave materia-grupo 2
Clave materia-grupo 3
Clave materia-grupo n

Obteniendo como archivo de salida, la clasificación por número de cuenta y sus materias-grupo solicitadas:



El listado del programa se puede consultar en el apéndice A (programa 19).

3.13 BUSQUEDA DE NOMBRES

Tal y como se realiza en el proceso de elaboración de listas de asistencia, se procederá a buscar el nombre correspondiente al número de cuenta leído en el archivo clasificado en el paso anterior.

Archivo de entrada (*archivo clasificado*)

Archivo de salida (*archivo clasificado con nombres*)
programa 20 (apéndice A).

3.14 BUSQUEDA DE NOMBRES DE MATERIAS

Es necesario asignar el nombre de la materia correspondiente a la clave de la materia leída en el archivo clasificado con nombres de alumnos.

La búsqueda se realiza de la siguiente manera:

- 1) Se inicializa un contador con valor igual a 1
- 2) Se lee al primer registro encontrado en el archivo, identificando la clave de materia.
- 3) Se genera un vector cuyo subíndice será igual al valor de

la clave de materia y contenido es el valor del contador.

A (cmat) = contador

Ejemplo:

Reg1 → Cmat = 0060 Cont = 1 → AC(60) = 1
Reg2 → Cmat = 0061 Cont = 2 → AC(61) = 2
Reg3 → Cmat = 0120 Cont = 3 → AC(120) = 3

Al tener generado este arreglo, se procede a leer el archivo secuencial clasificado por alumno, con claves de materia y grupo.

Se identifica la clave de materia:

Cmat →	# de cta	Nombre	Plantel	Carrera
Cmat →	0060 1101			
Cmat →	0061 1102			
Cmat →	# de cta	Nombre	Plantel	Carrera
Cmat →	0060 1102			
Cmat →	0120 1203			
Cmat →	# de cta	Nombre	Plantel	Carrera
.
.
.

Para la primer clave de materia, se toma esta como subíndice del arreglo, y su valor sera el número de registro a buscar en el archivo directo de nombres de materia.

En el ejemplo, la clave 0060, apuntará al elemento cuyo

subíndice es 00.

$$A(60) = 1$$

El valor 1, sirve para que se indique al registro # 1 del archivo directo de nombres de materias. De esta manera, se obtiene la información contenida en este registro, la cual podría ser:

0060 DIBUJO

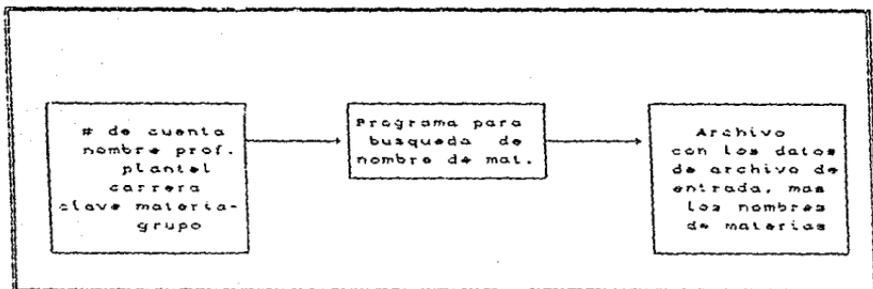
Con el nombre de la materia identificado, se graba en un archivo de salida, el cual contendrá lo siguiente:

# de cta	Ptl	Carrera	nombre de alumno
cmat	grupo	nombre materia	
cmat	grupo	nombre materia	
cmat	grupo	nombre materia	
# de cta	Ptl	Carrera	nombre de alumno
cmat	grupo	nombre materia	
cmat	grupo	nombre materia	
# de cta	Ptl	Carrera	nombre de alumno
.	.	.	.
.	.	.	.
.	.	.	.

En el caso de que la clave de materia (Cmat) sea igual a cero, en lugar de buscar el nombre de la materia, se grabará en el archivo de salida, un mensaje como el siguiente:

0000 0000 ERROR EN INSCRIPCION

Repetiendo el anterior proceso, hasta el fin de archivo de nombres de alumno y claves de materia, se habrá obtenido el archivo secuencial conteniendo toda la información necesaria para las tiras de materias provisionales, lo que se representa por el siguiente diagrama:



C * Programa para búsqueda de nombres de materias, en el apéndice A, programa 21 *

De esta manera, se concluye con la descripción del proceso requerido por el sistema para la elaboración de las tiras de materias provisionales. por último, se presenta el formato final de las mismas.

PROCESO DE AJUSTE Y ACTUALIZACION DEFINITIVA

Al haber realizado la impresión de las tiras de materias provisionales, se habrá terminado la etapa en la cual se considera que se lleva a cabo la parte más difícil y ardua del proceso de inscripción. Este proceso, como se ha podido observar, es largo y laborioso, por ello, es necesario dividirlo en varias etapas o fases de realización.

hecho esto, el sistema en conjunto, podrá aplicarse con un alto grado de eficiencia y funcionalidad.

A pesar de haber llevado a cabo hasta este punto una serie de procesos y secuencias con el fin de realizar la inscripción de alumnos en la ENEP ARAGON, se hace indispensable contar con un proceso de ajuste y actualización definitiva de la información recabada en los procesos previos del sistema de información de la

inscripción en la FNEP APAGON.

El proceso de ajuste y actualización, contempla el llevar a cabo una serie de cambios a los archivos de inscripción, obtenidos en las etapas previas.

Después de entregar a los profesores, las listas de asistencia correspondientes a su materia y a cada alumno su tira de materias temporal, así como los listados de errores en inscripción, se abre un periodo administrativo de ajustes y actualizaciones para los alumnos, el cual es conocido como periodo de altas y bajas.

En el periodo de altas y bajas, la UNIDAD ACADEMICA a través del departamento de Servicios Escolares, atiende las solicitudes de todo aquel alumno que por muy diversas causas, desee anular su inscripción en determinada(s) materia(s), inscribirse a cierta(s) materia(s) anteriormente no inscrito, o cambiar de grupo en una materia si inscrito pero en grupo no deseado.

Así, se generan dos movimientos de ajuste:

<p>ALTA de una materia en un grupo BAJA de una materia en un grupo</p>
--

La combinación de los dos movimientos, resulta en un tercero,

el cual se conoce como CAMBIO DE GRUPO:

CAMBIO DE GRUPO (Alta de materia X en grupo Y
Baja de materia X en grupo W)

Las causas de peticiones de ajustes y/o actualizaciones son muy diversas. por ejemplo:

- Horarios encimados (traslapados)
- Preferencia por otras materias
- Error en inscripción
- Preferencia por otros profesores
- No disponibilidad de tiempo en horario escogido
- Falta de conocimientos en la materia
- Problemas de índole personal, etc.

Las solicitudes se reciben en el periodo señalado para ello y se analizan posteriormente.

Dependiendo de los motivos y razones de cada solicitud, además de las posibilidades de aceptación de inscripción en

determinado grupo⁴, se rechazarán o aceptarán las solicitudes realizadas y posteriormente, se entregarán las solicitudes a los alumnos que las realizaron.

En caso de ser aceptadas, los alumnos solicitantes, llenarán una hoja de lectura óptica (color verde), similar a la hoja de lectura de inscripción normal (color azul). En esta hoja, se codifica la información correspondiente al movimiento deseado por el alumno. De esta manera, de nueva cuenta, el documento que realmente servirá para realizar el ajuste en la inscripción, es la hoja de lectura óptica (verde), tal y como en la inscripción normal.

Se realiza la recopilación del total de hojas de lectura óptica, durante el periodo indicado. Hecho esto, se turnan las hojas de lectura óptica verdes a la UNIDAD DE PLANEACION, la cual por medio del DEPARTAMENTO DE INFORMATICA, realizará la lectura óptica correspondiente, tal y como se realiza con las hojas de lectura óptica de la inscripción normal (azul)⁵.

Al realizar la lectura de hojas verdes, se obtiene un archivo secuencial conteniendo la información correspondiente a los ajustes y/o actualizaciones solicitadas por los alumnos.

⁴ Conocidas por las estadísticas generadas por el sistema propuesto en este trabajo de tesis.

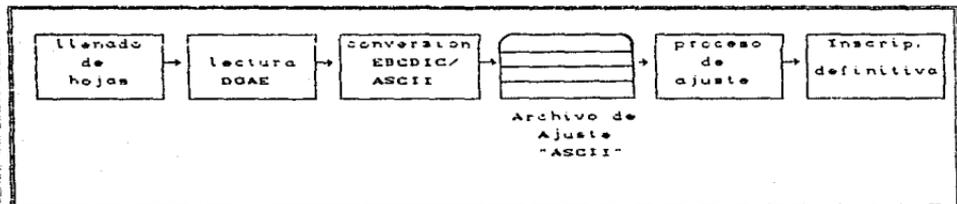
⁵ Proceso ya descrito en el capítulo IX.

De aquí en adelante, se llamará al archivo de ajustes y/o actualizaciones, "Archivo de ajuste".

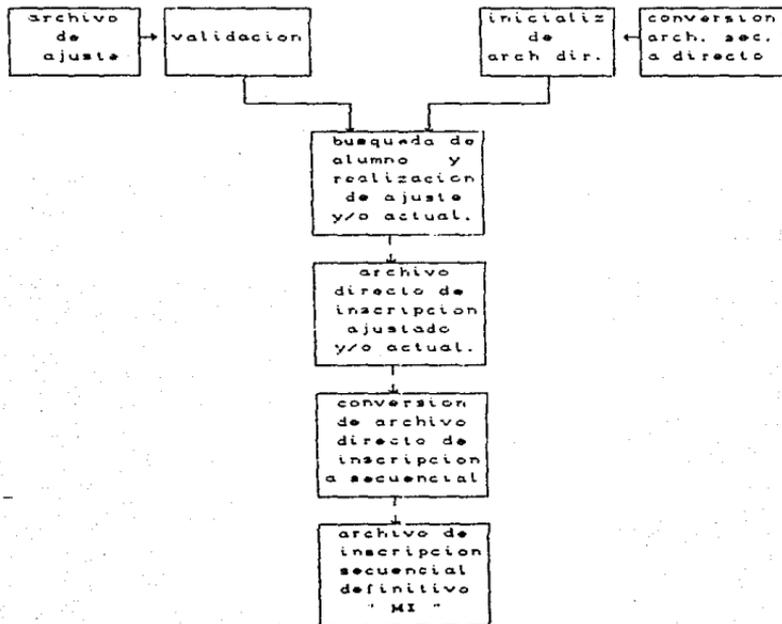
Una vez obtenido el archivo de ajuste, se seguirá con el proceso de ajuste y/o actualización definitiva.

El diagrama de ajuste es el siguiente:

RECOPIACION DE SOLICITUDES Y OBTENCION DE ARCHIVO DE AJUSTE



PROCESO DE AJUSTE Y/O ACTUALIZACION
 (ALTAS, BAJAS Y CAMBIOS)



4.1 ARCHIVO DE AJUSTE

El formato del archivo de ajuste es el siguiente:

# de cuenta,	plantel,	carr,	mat gpo A gpo B,
			cambio 1
mat gpo A gpo B,	mat gpo A gpo B,	mat gpo A gpo B,	
cambio 2	cambio 3	cambio 4	

DESCRIPCION DE CAMPOS:				
CAMPO	DESCRIPCION	LONGITUD	VARIABLE	TIPO
a	Número de cuenta	8	CTAA	alfanumerico
b	Clave de plantel	3	PTL	alfanumerico
c	Clave carrera	2	CARR	alfanumerico
d1	Materia	4	MOV(1:4)	alfanumerico
d2	Grupo de alta	4	MOV(5:8)	alfanumerico
d3	Grupo de baja	4	MOV(9:13)	alfanumerico

Los campos d1, d2 y d3, se repiten 5 veces, tantas como cambios posibles existen.

4.2 VALIDACION

El archivo de ajuste debe recibir un proceso de validación para que en él, no existan números de cuenta, materias y/o grupos erróneos. Si existe número de cuenta erróneo, se sustituirá por ceros y así no será tomado en cuenta.

Si una clave de materia o grupo es errónea, se sustituirá el total del campo D (d1,d2,d3) por ceros, para que no sea tomado en cuenta.

EJEMPLO:

Presentación de un posible registro del archivo de ajuste y su significado:

No.	cta	Ptl	Car	cve	altabaja	cve	alta
80262844	411	32		006012011202	00611201----		
				1er. mov.		2o. mov.	
				cve	baja	cve	altabaja
				0120----	1204	0128*	94112 1
				3er. mov.		4o. mov.	

Nota: Al leer el archivo, se necesita compactarlo previamente, ya que los dígitos correspondientes al número de cuenta, plantel y carrera están repetidos.

MOVIMIENTOS SOLICITADOS (con el ejemplo):

```
1er. movimiento --- Alta de materia 0000 en grupo 1201
                   Baja de materia 0000 en grupo 1202
2o. movimiento --- Alta de materia 0001 en grupo 1201
3er. movimiento --- Baja de materia 0120 en grupo 1204
4o. movimiento --- Error en el llenado de movimiento
```

El registro validado, quedará como sigue:

```
80262844 411 32 008012011202 00611201****
                   1er. mov.      2o. mov.

0120****1204 000000000000
                   3er. mov.      4o. mov.
```

Se sustituyen blancos por asteriscos (*) en los campos leídos correctamente.

Se sustituye el total de un campo leído incorrectamente, por ceros.

El programa de validación se presenta en el apéndice "A".
(Programa 22)

4.3 INICIALIZACION DE LOS ARCHIVOS DIRECTOS

Tal y como se realiza para los archivos de nombres, se inicializarán archivos directos para grabar en ellos la información correspondiente a los archivos de inscripción.

Se inicializan 3 archivos directos de inscripción:

1 para Derecho
1 para Ingeniería
1 para el resto de las carreras impartidas en ARAGON

En cada uno de ellos, se almacenaran los datos del archivo de inscripción según corresponda.

La división y/o utilización de 3 archivos, se hace para disminuir el número de colisiones a generarse por hash y así también, el tiempo de acceso al archivo directo.

El programa de inicialización esta presentado en el apéndice "A" (Programa 23).

DESCRIPCION DE CAMPOS:

CAMPO	DESCRIPCION	LONGITUD	VARIABLE	TIPO
a	Plantel	3	ESCU	alfanumerico
b	Carrera	2	ESCU	alfanumerico
c	No. de cuenta	8	CTAA	alfanumerico
d	Materia-grupo 1	8	MATEGPOCID	alfanumerico
e	Materia-grupo 2	8	MATEGPOCID	alfanumerico
f	Materia-grupo 3	8	MATEGPOCID	alfanumerico
g	Materia-grupo 4	8	MATEGPOCID	alfanumerico
h	Materia-grupo 5	8	MATEGPOCID	alfanumerico
i	Materia-grupo 6	8	MATEGPOCID	alfanumerico
j	Materia-grupo 7	8	MATEGPOCID	alfanumerico
k	Materia-grupo 8	8	MATEGPOCID	alfanumerico
l	Materia-grupo 9	8	MATEGPOCID	alfanumerico
m	Materia-grupo 10	8	MATEGPOCID	alfanumerico
n	Número de materias	2	NMATE	entero
o	Bandera	3	BAN	alfanumerico

Se graba:

```

- ESCU = 4 espacios en blanco
  CTAA = "00000000"
  MATEGPOCID = 8 espacios en blanco
  NMATE = "00"
  BAN = "MOR"

```

4.4 CONVERSION DE ARCHIVO SECUENCIAL DE INSCRIPCION A UN ARCHIVO DIRECTO

El archivo directo previamente inicializado, será depositario de la información contenida en el archivo secuencial de inscripción, para que en él, se realicen las modificaciones de altas y bajas. Se hace esto para poder localizar rápida y fácilmente a un registro en particular -aprovechando para ello- las características intrínsecas de un archivo aleatorio o directo.

De esta forma, se lee el archivo secuencial de inscripción registro a registro. Para obtener el direccionamiento, se identifica el número de cuenta del alumno en cuestión, se procede a una transformación de clave, por medio del método de hash^d, mediante la conversión por módulo de la división entre un número primo, el cual señala al número de registro de posible grabación. Si este registro está desocupado, se procede a grabar en él, la información del registro del archivo secuencial, si no está desocupado, se incrementa el número de registro hasta encontrar un registro desocupado, se graba la información y se cambia la bandera de registro desocupado "MOR" por la de ocupado (tres blancos).

^d Método analizado y detallado en el capítulo II y apéndice B.

4.5 BUSQUEDA DE ALUMNOS Y REALIZACION DE AJUSTE Y/O ACTUALIZACION

Se hace necesario, el contar con un programa que permita buscar y consultar el contenido de un registro en particular de cualquiera de los 3 archivos directos correspondientes a la inscripción.

En este programa, se utiliza de nueva cuenta el método de hash para localizar el registro contenedor de la información de un alumno en específico.

El nombre del programa para buscar un alumno en el archivo de inscripción es : BUSCADOR.FTN.

(*El listado se puede consultar en el apéndice A, programa 24*)

El programa anterior no realiza cambio o ajuste alguno a los archivos directos. El programa que si realiza los ajustes y/o actualizaciones, es el programa BUSNOM_ALTA.FTN, el cual, leyendo registro a registro el archivo secuencial de ajuste y utilizando "hash", busca y encuentra (si existe) el registro correspondiente al alumno sobre el cual se realizará el ajuste y/o actualización.

El ajuste se realiza de acuerdo a lo siguiente:

Una vez leído el registro del archivo de ajuste y localizado

el registro del archivo directo de inscripción, con el mismo número de cuenta:

Reg. de archivo de ajuste	8026284441132	006012011202	0061****1203
	ctaa		

Registro de archivo directo de inscripción:

41132	80262844	00601202	00611203	01201202
	ctaa2			

ctaa = ctaa2

Se tendrán 3 movimientos posibles:

a) Alta de materia en un grupo

(Identificado por clave de materia, gpo. alta y ****)

b) Baja de materia en un grupo

(Identificado por clave de materia **** gpo. baja)

c) Cambio de materia

Alta de materia en un grupo

Baja de materia en un grupo

(Identificado por clave de materia gpo. alta gpo. baja)

4.5.1 MOVIMIENTO DE ALTA

Para identificar que el alumno desea realizar un alta de materia en un grupo dado, el campo correspondiente al movimiento, debe presentarse así:

Caracteres	Contenido
1 a 4	Clave de materia
5 a 8	Clave de grupo solicitud
9 a 12	**** (4 asteriscos)

Se detecta el valor de la variable NMATE, almacenada en el archivo directo, si ésta tiene un valor menor a 10, entonces se puede proceder a la inscripción o alta del alumno en la materia, si no, entonces se despliega el mensaje:

```
ERROR: no se permiten mas de 10 materias !  
movimiento anulado !!!!
```

y el alta no se realiza.

En caso de que el alta si pudiera realizarse, se utiliza una rutina para detectar la posible repetición de inscripción en una materia, para anular el movimiento de alta solicitado, si sucediera la repetición.

Se comparan todas las materias ya inscritas por el alumno, si

alguna de ellas es igual a la solicitada para la alta, se despliega un mensaje como el siguiente:

ERROR EN ALTA: Ya existe inscripcion en esta materia !

El estado de la bandera es modificado para terminar el ciclo de comparaciones, realizando luego, la comparación con el estado de la bandera y así realizar o no el alta solicitada⁷.

Si se puede realizar el ajuste, se asigna a la materia siguiente (Nmate+1), la clave de materia y el grupo solicitado, para luego, incrementar el valor del contador de materias en una unidad.

MATEGPOC (NMATE + 1) = MOV (1:8)
 NMATE = NMATE + 1

De la manera anterior, se asigna en el vector de materias, el ajuste de inscripción deseado (Alta).

⁷ Bandera: variable "L", si L=0 no hay error, si L=1 error.

4.5.2 MOVIMIENTO DE BAJA

Cuando el ajuste o movimiento deseado, sea el de baja o anulación de inscripción en una materia-grupo, el campo de movimiento, debe tener la siguiente estructura:

Caracteres	Contenido
1 a 4	Clave de materia
5 a 8	**** (4 asteriscos)
9 a 12	Clave de grupo de baja

Se verifica que ocurra esta condición del registro de movimientos y luego se procede a comparar cada materia en la que esté inscrito el alumno con la materia solicitada a dar de baja.

Si se cumple la igualdad de la clave de materia, se compara la clave de grupo inscrito con la del solicitado, en el caso de no ser iguales, existe un error y se despliega un mensaje como el siguiente:

ERROR EN BAJA: No existe grupo !

Si el grupo es el mismo, se realizará la baja por medio del algoritmo siguiente:

Dependiendo de cual de las 10 materias se desee dar de baja, un contador de número de materia indicará la posición guardada por ella en el registro del archivo directo de inscripción. Así, desde este número hasta el número de materias total menos uno ($10-1=9$), se recorrerán los campos no modificados, para de este modo realizar la anulación de la materia inscrita. La materia número 10 tendrá ceros como contenido, para indicar que no hay inscripción en ella.

Se decrementa en uno el contenido del contador de materias, para tener el ajuste y/o actualización en forma correcta.

Por último, se modifica el estado de una bandera, para indicar la realización de la baja o no^a.

Si al término de las comparaciones de las claves de materia, no hubo cambio en el estado de la bandera, significa esto, que la materia-grupo a dar de baja, no existe en el registro del archivo

^a
 Bandera L=0 --> No hubo movimiento de baja
 Bandera L=999 --> Se realizó la baja

de inscripción y por lo tanto no es posible realizar la baja en una materia no inscrita. por eso, se desplegará el mensaje:

ERROR EN BAJA: No existe inscripción a tal materia !

4.5.3 MOVIMIENTO DE CAMBIO

Para identificar la solicitud de cambio de grupo en una misma materia, el registro debe presentar la siguiente estructura:

Caracteres	Contenido
1 a 4	Clave de materia
5 a 8	Clave de grupo a inscribir
9 a 12	Clave de grupo inscrito

Como se puede observar, el movimiento de cambio, es en realidad el proceso de ajuste y/o actualización, consistente en la conjugación de alta de una materia-grupo y baja de una materia-grupo, donde la materia es la misma en ambos casos.

Para el movimiento de cambio, se compara cada una de las materias inscritas con la solicitada, si se encuentra ésta, se asigna como nuevo grupo inscrito la clave del grupo a inscribir,

dejando intacta la clave de materia y logrando el cambio o ajuste deseado. La clave de grupo a dar de baja, se pierde y modifica al asignar la nueva clave de grupo. Al hacer el cambio, el estado de la bandera se modifica de L=0 a L=999.

En el caso de no encontrarse la materia solicitada, como materia inscrita, no se modifica el estado de la bandera de identificación^p y por lo tanto, al término de la búsqueda, se obtendrá un mensaje como el siguiente:

ERROR EN MOVIMIENTO: Materia inexistente !

Cualquiera de los 3 movimientos anteriores, se puede repetir o realizar. Según lo indique el campo de movimientos, es posible efectuar hasta un máximo de 5 movimientos por alumno.

Una vez realizados los ajustes y/o actualizaciones de un alumno, se procede a realizar los del siguiente. Así, hasta que se termine el archivo de ajuste y se tengan todos los movimientos efectuados.

(*El listado se presenta en el apéndice A, PROGRAMA 25*)

^p Si L=0 no se realizó el cambio
Si L=999 se realizó el cambio

Cuando se han hecho los movimientos al total de alumnos solicitantes, el archivo directo de inscripción estará totalmente ajustado y/o actualizado, de esta forma, se habrá obtenido un archivo de inscripción que podrá ser tomado como definitivo. De este modo, el único proceso por llevar a cabo, es el de convertir el archivo directo de inscripción a uno secuencial, esto con el fin de poder procesarlo posteriormente.

Con el archivo anterior, se puede obtener el archivo para elaboración de listas, tiras y actas definitivas de la inscripción de un semestre en particular en la ENEP Aragón y con ello, el proceso de ajuste y/o actualización definitiva dentro del sistema propuesto, se da por concluido.

PERSPECTIVAS DE EVOLUCION DEL SISTEMA

Al establecerse el sistema de información propuesto en este trabajo, se deberán tomar en cuenta diferentes aspectos que conformarán la evolución y mantenimiento del mismo.

El primer aspecto a considerar es el mantenimiento del sistema.

El mantenimiento del software suele necesitar de 40 a 80% y en algunos casos hasta 90% del esfuerzo total durante el ciclo de vida del establecimiento del sistema, estas actividades comprenden agregar mejoras al sistema, adaptar el producto para nuevos ambientes de proceso y corregir los problemas de los programas.

Una regla útil muy usada por la distribución del esfuerzo de las actividades de mantenimiento, es asignar 60% del tiempo a mejoras, 20% a adaptación y 20% a la depuración o corrección de problemas. Con una encuesta entre 487 instalaciones de procesamiento de datos, Lientz y Swanson

determinaron que el nivel del esfuerzo característico dedicado al mantenimiento de los programas era de aproximadamente 50 % del tiempo total del ciclo de vida del proyecto, y la distribución interna de este tiempo fue de 51.3% para mejoras, 23% para adaptaciones, 21.7% para reparaciones y 3.4% para arreglos varios (LIEBO)¹.

Estos porcentajes se desglosan en el siguiente cuadro:

ACTIVIDAD	% DEL ESFUERZO
Mejoras	51.3
Mayor eficiencia	4.0
Mejor documentación	5.5
Mejoras para el usuario	41.8
Adaptación	23.6
Datos de entrada y archivos	17.4
Equipo y sistema operativo	6.2
Correcciones	21.7
Arreglos de emergencia	12.4
Arreglos programados	9.3
Otros	3.4

DISTRIBUCION DEL ESFUERZO DE MANTENIMIENTO
(TOMADO DE LIEBO)

¹ Lientz, B., and E. Swanson: Software Maintenance Management: A study of the maintenance of computer application software in 487 data processing organizations, Addison-Wesley, Reading, MA, 1980.)

El mantenimiento para el sistema de información referido en este trabajo, ocupa un 40% a 50% del esfuerzo total.

En cuanto a la distribución del esfuerzo del mantenimiento del sistema propuesto, se tiene muy en cuenta el aspecto de mejoras, el cual tendrá un porcentaje cercano al 50%, adaptaciones un 25%, correcciones un 20% y otros 5%. el cuadro desglosado de actividades para la distribución de esfuerzo de mantenimiento, se muestra a continuación:

ACTIVIDAD	% DEL ESFUERZO
Mejoras	50.0
Mayor eficiencia	5.0
Mejor documentación	5.0
Mejoras para el usuario	40.0
Adaptación	25.0
Datos de entrada y archivos	15.0
Equipo y sistema operativo	10.0
Correcciones	20.0
Arreglos de emergencia	5.0
Arreglos programados	15.0
Otros	5.0

DISTRIBUCION DEL ESFUERZO DE MANTENIMIENTO
EN EL SISTEMA DE INFORMACION PARA EL
PROCESO DE INSCRIPCIONES EN LA ENEP ARAGON

5.1 MEJORAS

Mejoras para el usuario: Se pretende optimizar el sistema de información para que en un futuro, el alumno pudiera realizar su inscripción directamente en una terminal.

Mayor eficiencia : Se busca que el sistema funcione casi por si solo, sin tener un gran esfuerzo de mantenimiento dedicado. Lográndose esto, gracias a la mejora de rutinas de software, y de algoritmos utilizados, además de la depuración de los errores presentados con la implantación y puesta en marcha del sistema.

Mejor documentación: Realización de documentación a detalle de programas y/o rutinas utilizadas en el sistema, explicación de técnicas de programación utilizadas y aceptación de nuevas con su correspondiente documentación.

Obtención de documentación consistente en la historia de funcionamiento del sistema, problemas y vicisitudes, además de logros importantes con el. Esto con el fin de contar con una guía de referencia para futuras aplicaciones y/o mejoras al sistema.

5.2 ADAPTACION

Dentro del aspecto evolutivo del sistema, es importantísimo tomar en cuenta lo que se refiere al mantenimiento del mismo, en su enfoque de adaptación.

Debido al incesante y fugaz avance tecnológico en cuanto a computadoras, los cambios del medio ambiente en el que se desenvuelva el sistema son muy posibles, esto es, el equipo utilizado puede quedar dentro de la categoría de "obsoleto", por lo cual, dependiendo de las políticas de la ENEP-Aragón y sus posibilidades económicas, podría adquirirse un equipo de cómputo de características y potencialidades muy diferentes a las del actual. Es por esto, que el sistema, se ha contemplado desde su planteamiento, como un sistema flexible y adaptable, para que en caso de tener que funcionar en un ambiente diferente, lo hiciera con igual o mayor eficacia y rendimiento.

El aspecto contemplado de adaptación, cubre incluso, el hecho posible de que el sistema se exportara a otra institución que lo deseara utilizar. Trayendo consigo esto, la obvia necesidad de que el sistema sea funcional en condiciones muy diversas de equipo y personal (Medio ambiente), además de una variación en necesidad específica de la institución usuario. Por esto, el sistema ha sido planteado idealmente para que sea de utilidad general y no para operarse exclusivamente en la ENEP-Aragón.

El aspecto anterior, marca notoriamente en el plano evolutivo

del sistema. su ambiciosa tendencia de adaptabilidad.

5.2.1 DATOS DE ENTRADA Y ARCHIVOS

Los datos de entrada podrían en un momento dado, modificarse en su estructura. Incluso los archivos utilizados pueden ser diferentes a los usados con mayor frecuencia en la ENEP-Aragon actualmente.

Por lo anterior. el software cuenta con la facilidad de utilizar formatos de diferente índole a los usados comúnmente. El 15% de esfuerzo dedicado a este punto. es dado para contemplar la importancia de posibles modificaciones en los datos y archivos de entrada. con el fin de hacerles más funcionales y óptimos.

5.2.2 EQUIPO Y SISTEMA OPERATIVO

Como se mencionó anteriormente. el posible cambio del equipo utilizado. es contemplado por el sistema. aunque el cambio del sistema operativo. también se ha tomado en cuenta. Por ello. se han utilizado técnicas de programación. algoritmos y estructuras de datos. lo más generales posibles. Significa que estos puedan ser utilizables casi en cualquier equipo de la actualidad. desde una

red de computadoras personales, una minicomputadora y hasta un mainframe.

La misma adaptabilidad se ha buscado en lo que concierne al sistema operativo, esta propiedad se ve comprobada a corto plazo, ya que las versiones del sistema operativo que maneja la HP-1000 se reemplazan constantemente, trayendo consigo, modificaciones al modo de funcionar de la computadora y su software utilizado.

Otro aspecto considerado en el desarrollo evolutivo del sistema, es el utilizar un lenguaje de alto nivel diferente al usado en la actualidad por el sistema, debido a que con otro equipo, el lenguaje utilizado no fuera el idóneo.

Los programas del sistema, están escritos en el lenguaje Fortran 77, por las razones ya explicadas en el capítulo I.

La adaptabilidad en este caso, se puede observar en el propio sistema, ya que todos sus programas han sido diseñados tomando en cuenta los principios de la programación estructurada, mediante el diseño de arriba a abajo (TOP-DOWN), modular e independiente. Así que los algoritmos utilizados pueden ser transferidos o "traducidos" a cualquier otro lenguaje estructurado que maneje las estructuras básicas de secuencias, selección y repetición necesarias. De una manera tan simple, que no es obligatorio el que realice un estudio a fondo de los algoritmos utilizados, sino que se puede hacer una traducción casi "literal" de los mismos.

5.3 CORRECCIONES

Las correcciones tienen un lugar preponderante dentro del sistema. estas pueden o deben realizarse con un planteamiento previo o de forma totalmente inesperada. así se tendrán:

5.3.1 ARREGLOS DE EMERGENCIA

Al ser un sistema de información que cuenta con muchos aspectos innovadores e independizadores de otros sistemas de información. el implantarlo trae consigo día a día, errores u omisiones no previstos con anterioridad debido a su naturaleza, por ello. es necesario aplicar un arreglo o corrección de emergencia a aquel proceso del sistema que así lo requiera.

5.3.2 ARREGLOS PROGRAMADOS

El sistema de información no contiene todos los procesos o conexiones necesarias con otros sistemas. Estas limitaciones se han previsto desde los inicios de diseño del sistema. por lo cual. se han dejado como parte del entorno evolutivo del mismo. tomando el enfoque de arreglos programados al sistema para la posterior incorporación de funciones contempladas con anterioridad.

CONCLUSIONES

El sistema de información para el proceso de inscripciones de alumnos en la ENEP ARAGON, ha sido puesto en marcha a -- partir del semestre 88-II en su primera fase, que corresponde a la elaboración de números de sorteo y listas de asistencia de -- alumnos. A partir del semestre 89-I, se puso en marcha la segunda fase que consiste en la impresión de las hojas de inscripción, tiras de materia provisionales, errores en inscripciones y estadísticas de cupos de grupos.

El éxito obtenido desde la primera fase es notable, el -- ahorro previsto en tiempo, trámites y costos se ha logrado conforme lo planeado, y se ha beneficiado al alumno enormemente.

Las fallas u omisiones no previstas se han corregido a -- medida que se ha tenido un estudio de la historia de funcionamiento del sistema y se puede decir que los objetivos se han alcanzado satisfactoriamente.

La tercera fase del sistema que consiste del proceso de altas y bajas se encuentra en etapa de preparación y adecuación para su inmediata aplicación.

A medida que el sistema vaya madurando se prevee un mejoramiento en su funcionamiento, además de su crecimiento ya que el sistema no es estático, sino que incorporará procesos necesarios para la ENEP-ARAGON.

APENDICE A

PROGRAMAS DEL SISTEMA

PROGRAM NUMSOR(4,10)

```

C ***** PROGRAMA QUE SIRVE PARA OBTENER LOS PROMEDIOS DE *****
C ***** ALUMNOS ALMACENADOS EN EL ARCHIVO DE HISTORIAL *****
C ***** ACADENICO *****
C ***** EL ARCHIVO DE SALIDA, CONTENDRA A LOS ALUMNOS *****
C ***** DE PRIMER INGRESO DE ARCHIVO GENERAL *****
C ***** ELABORADO: MIERCOLES-12-ABRIL-1999 *****

      DIMENSION SINDER2(150)
C      DIMENSION PROM(50),CAL2(5,50),SINDER2(50)
C      COMMON/DAT02/CAL2(5,4000)
C      COMMON/DAT03/C
C      COMMON/DAT04/SINDER2(4000)
      INTEGER SINDER2
      INTEGER CLAS,D,E,F,G,H
      INTEGER P,MB,B,S,NA,N,IN,UN,PROM,CAL2
      INTEGER CPL,CCA,ING,REV,ACR,APOR,APEX,REOP,REEX
      INTEGER COBL,COPT,PERIN,PEUL,CONT
      CHARACTER NCTA*9
      CHARACTER BLAN*10
      CHARACTER *20 ENTRADA
      CHARACTER*20 SALIDA
      CHARACTER ALGO*32
      CHARACTER SINDER1(50)*32
      CHARACTER CAL1*30
      CALL RCPAR(1,ENTRADA)
      CALL RCPAR(2,SALIDA)
      OPEN(10,FILE=ENTRADA,STATUS='OLD')
      OPEN(8,FILE=SALIDA,STATUS='UNKNOWN')

      N=0
      30 READ(10,10,END=999)ALGO,NCTA,CPL,CCA,ING,MB,B,S,BLAN,NA,HP,REV,ACR
      1E,APOR,APEX,REOP,REEX,COBL,COPT,PERIN,PEUL
      10 FORMAT(A32,A8,I3,S12,A10,B12,4I3)
C      WRITE(1,*)# CTA = 'NCTA
      WRITE(1,*)ALGO,MB,B,S,NA,HP

C      ***** EN SEGUIDA, SE VERIFICA SI EL ALUMNO TIENE O NO DERECHO *****

C      IF<ING.LT.84.OR.<PERIN.LT.841.AND.NCTA(1:2).LT.'81')THEN
C      CONT=CONT+1
C      SINDER1<CONT>=ALGO
C      SINDER2<CONT>=PERIN
C      ENDIF

C      ***** SI EL ALUMNO NO TIENE DERECHO, NO SE OBTIENE SU PROMEDIO *****
C      IF<ING.LT.84.OR.<PERIN.LT.841.AND.NCTA(1:2).LT.'81')GOTO 30
      N=N+1
      P=0
      PROM=0
      F1=FLOAT(MB)
      F2=FLOAT(B)
      F3=FLOAT(S)
      F4=FLOAT(HP)
      F5=FLOAT(NA)
  
```

PROGRAMA 1-A

```

C      P=(MB+10+B+S+6.)
C      WRITE(1,*)'B = ',B
C      WRITE(1,*)'S = ',S
C      WRITE(1,*)'HA = ',HA

C      WRITE(1,*)'MB = ',MB
C      WRITE(1,*)'P = ',P
C      PROM=P/(MB+B+S)
C      PIRO=(MB+10+B+S+6.)/(MB+B+S)
C      PIRO=(F1+10.+F2+9.+F3+6.)/(F1+F2+F3)
C      PIRO=(F1+10.+F2+8.+F3+6.)/(F1+F2+F3)
C      IF (F1.GE.1.AND.(F2.EQ.0.AND.F3.EQ.0).AND.NA.EQ.0.AND.NP.EQ.0)
*      PIRO=10.00

C      WRITE(1,*)'PROMEDIO ES ==> ',PROM
C      IF(PROM.GE.LAC.AND.PRON.LT.LA2) THEN
C        WRITE(1,*)'PROMEDIO ES 9'
C          C=C+1
C          CLAS=1
C          CAL1=ALGO
C          CAL2=9
C
C      *****
C      GRABACION DE ARCHIVO CLASIFICACION : CLAS=1
C      *****
C      H2=H2+1
C      WRITE(1,*)H2
C      WRITE(3,200)ALGO,NOTH,CPL,CCP,ING,N2,E,S,BLAN,NA,NP,REV,ACRE,APOR
200  FORMAT(A32,A8,I4,S12,A10,S12,A14,F6.2)
C      WRITE(1,*)'NOMBRE ',ALGO
C      WRITE(1,*)'PROMEDIO ',PIRO
C      WRITE(1,*)'ALUMNO # ',H2

C      END IF
C      IF(PRON.LT.6)THEN
C        KL=KL+1
C      END IF
C      END IF
C      GOTO 30
999  WRITE(1,*)
C      WRITE(1,*)CAL1
C      WRITE(1,*)CAL2
C      WRITE(1,*)'NO REGISTRADOS ==>> ',KL

      STOP
      END

```

```

2 #FILES 0,3
3 PROGRAM NUMSOR(4,10)
4 DIMENSION SINDER2(150)
5 C DIMENSION PROM(50),CAL2(5,50),SINDER2(50)
6 C COMMON/DAT02/CAL2(5,4000)
7 COMMON/DAT03/C
8 C COMMON/DAT04/SINDER2(4000)
9 INTEGER SINDER2
10 INTEGER CLAS,D,E,F,G,H
11 INTEGER P,MB,B,S,HA,N,IM,UN,PROM,CAL2
12 INTEGER CPL,CCA,ING,REV,ACRE,APOR,APEX,REOR,REEX
13 INTEGER COBL,COPT,PERIN,PEUL,COHT
14 CHARACTER NCTA*8
15 CHARACTER BLAH*10
16 CHARACTER *20 ENTRADA
17 CHARACTER*20 SALIDA
18 CHARACTER ALGO*32
19 CHARACTER SINDER1(50)*32
20 CHARACTER CAL1*30
21 CALL RCPAR(1,ENTRADA)
22 CALL RCPAR(2,SALIDA)
23 OPEN(1,FILE=ENTRADA,STATUS='OLD')
24 OPEN(2,FILE=SALIDA,STATUS='UNKNOWN')
25
26 WRITE(1,*)'1 PARA IMPRESORA 2 PANTALLA'
27 READ(1,888)IMP
28 WRITE(1,*)'DAME CALIF MENOR'
29 READ(1,*)LAC
30 WRITE(1,*)'DAME CAL MAYOR'
31 READ(1,*)LA2
32 C READ(1,888)IMP
33 888 FORMAT(I6)
34 IF(IMP.EQ.1)UN=6
35 IF(IMP.EQ.2)UN=1
36 IM=UN
37 OPEN(UNIT=UN)
38 N=0
39 write(1,*)'dame la clave'
40 read(1,*)cpl
41 WRITE(1,*)'DAME LA CLAVE'
42 READ(1,*)LLLL
43 30 READ(10,10,END=999)ALGO,NCTA,CPL,CCA,ING,MB,B,S,BLAH,HA,NP,REV,ACR
44 1E,APOR,APEX,REOR,REEX,COBL,COPT,PERIN,PEUL,PIRO
45 10 FORMAT(A32,A8,I4,S12,A10,S12,A14,A6)
46
47
48 N2=N2+1
49 WRITE(1,*)N2
50 WRITE(8,200)ALGO,NCTA,CPL,CCA,ING,MB,B,S,BLAH,HA,NP,REV,ACRE,APOR
51 2,APEX,REOR,REEX,COBL,COPT,PERIN,PEUL,PIRO
52 200 FORMAT(A32,A8,I4,S12,A10,4I2,/,4I2,A14,A6)
53
54 GOTO 30
55 999 WRITE(1,*)
56 WRITE(1,*)'NO REGISTRADOS **>>'.KL

```

PROGRAMA 2-A

Page 2 NUMSOR

Opts: 77/LYI

Wed, Sep 14, 1988, 2:14 PM
/IHF006/SORTE0/SOR4.FTH57
58 STOP
59 ENDModule NUMSOR
FTH7X 5000/951229No errors Prog: 2013 Blank common: none
No warnings Save: none Local EMR: none

REPORT:

H1,DATE,19;
 H1,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",84;
 H1,"HOJA",110;
 H1,PAGENO,116,SPACE A1;
 H2,"E.N.E.P. ARAGON",64;
 H3,"<<UNIDAD DE PLANEACION>>",59;
 H3,"**DEPARTAMENTO DE INFORMATICA**",95,SPACE A2;
 H4,"PLANTEL",14;
 H4,"CARR",22;
 H4,"NOMBRE",47;
 H4,"# DE CTA",75;
 H4,"SORTED",87;
 H4,"FECHA",101;
 H4,"DIA",118;
 H5,"=====",14;
 H5,"=====",22;
 H5,"=====",60;
 H5,"=====",75;
 H5,"=====",87;
 H5,"=====",105;
 H5,"=====",122;

REPORTE POR NOMBRE

S,NOM;
 D1,CPL,13;
 D1,CCA,21;
 D1,NOM,60;
 D1,NCTA,75;
 D1,NUSOP,86;
 D1,FECHA,105;
 D1,DIA,122;
 END;

REPORT;
 H1,DATE,19;
 H1,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",84;
 H1,"HOJA",110;
 H1,PAGENO,116,SPACE A1;
 H2,"E.N.E.P. ARAGON",64;
 H3,"<<UNIDAD DE PLANEACION>>",59;
 H3,"**DEPARTAMENTO DE INFORMATICA**",95,SPACE A2;
 H4,"PLANTEL",14;
 H4,"CARR",22;
 H4,"NOMBRE",47;
 H4,"# DE CTA",73;
 H4,"SORTEO",87;
 H4,"FECHA",101;
 H4,"DIA",118,14;
 H5,"=====",14;
 H5,"=====",22;
 H5,"=====",60;
 H5,"=====",75;
 H5,"=====",87;
 H5,"=====",105;
 H5,"=====",122;
 S,NUSOR;
 D1,CPL,12;
 D1,CCA,21;
 D1,HOM,60;
 D1,HCTA,75;
 D1,NUSOR,86;
 D1,FECHA,105;
 D1,DIA,122;
 END;

REPORT;

REPORTE POR FECHA Y NOMBRE

H1,DATE,19;
H1,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",84;
H1,"HOJA",110;
H1,PAGENO,116,SPACE A1;
H2,"E.N.E.P. ARAGON",64;
H3,"<<UNIDAD DE PLANEACION>>",59;
H3,"**DEPARTAMENTO DE INFORMATICA**",95,SPACE A2;
H4,"PLATEL",14;
H4,"CARR",22;
H4,"NOMBRE",47;
H4,"# DE CTA",75;
H4,"SORTEO",87;
H4,"FECHA",101;
H4,"DIA",118;
H5,"=====",14;
H5,"====",22;
H5,"=====",60;
H5,"=====",75;
H5,"=====",87;
H5,"=====",105;
H5,"=====",122;
S,NOM;
S1,FECHA;
D1,CPL,12;
D1,CCA,21;
D1,NOM,60;
D1,HCTA,75;
D1,NUSOR,86;
D1,FECHA,105;
D1,DIA,122;
END;

REPORT;
 H1,DATE,19;
 H1,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",94;
 H1,"HOJA",110;
 H1,PAGEO,116,SPACE A1;
 H2,"E.N.E.P. ARAGON",64;
 H3,"<<UNIDAD DE PLANEACION>>",59;
 H3,"**DEPARTAMENTO DE INFORMATICA**",95,SPACE A2;
 H4,"PLANTEL",8;
 H4,"CARR",16;
 H4,"NOMBRE",41;
 H4,"# DE CTA",69;
 H4,"SORTEO",81;
 H4,"PROMEDIO",94;
 H4,"FECHA",108;
 H4,"DIA",124;
 H5,"*****",8;
 H5,"****",16;
 H5,"*****",54;
 H5,"*****",69;
 H5,"*****",81;
 H5,"*****",94;
 H5,"*****",112;
 H5,"*****",128;
 S,HUSOR;
 D1,CPL,6;
 D1,CCA,15;
 D1,HOM,54;
 D1,NCTA,69;
 D1,HUSOR,80;
 D1,PIFO,93;
 D1,FECHA,112;
 D1,DIA,128;
 END;

REPORT:

H1,DATE,19;
 H1,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",84;
 H1,"HOJA",110;
 H1,PAGENO,116,SPACE A1;
 H2,"E.N.E.P. ARAGON",64;
 H3,"<<UNIDAD DE PLANEACION>>",59;
 H3,"**DEPARTAMENTO DE INFORMATICA**",95,SPACE A2;
 H4,"FLANTEL",8;
 H4,"CARR",16;
 H4,"NOMBRE",41;
 H4,"# DE CTA",69;
 H4,"SORTEO",81;
 H4,"PROMEDIO",94;
 H4,"FECHA",108;
 H4,"DIA",124;
 H5,"*****",8;
 H5,"****",16;
 H5,"*****",54;
 H5,"*****",69;
 H5,"*****",81;
 H5,"*****",94;
 H5,"*****",112;
 H5,"*****",128;
 S,NON;
 D1,CPL,6;
 D1,CCA,15;
 D1,NOB,54;
 D1,ICTA,69;
 D1,HUSOR,80;
 D1,PIRO,93;
 D1,FECHA,112;
 D1,DIA,128;
 END;

REPORT:

REPORTE POR NUMERO DE SORTEO Y

H1,DATE,19;
H1,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",84;
H1,"HOJA",110;
H1,PAGENO,116,SPACE A1;
H2,"E.N.E.P. ARAGON",64;
H3,"<<UNIDAD DE PLANEACION>>",59;
H3,"**DEPARTAMENTO DE INFORMATICA**",95,SPACE A2;
H4,"PLANTEL",14;
H4,"CARR",22;
H4,"NOMBRE",47;
H4,"# DE CTA",75;
H4,"SORTEO",37;
H4,"FECHA",101;
H4,"DIA",118;
H5,"=====",14;
H5,"=====",22;
H5,"=====",60;
H5,"=====",75;
H5,"=====",37;
H5,"=====",105;
H5,"=====",122;
S1,CCA;
S,NUSOR;
D1,CPL,12;
D1,CCA,21;
D1,NOM,60;
D1,NCTA,75;
D1,NUSOR,86;
D1,FECHA,105;
D1,DIA,122;
END;

REPORT;

REPORTE POR NOMBRE FECHA Y CARRE

H1,DATE,19;
H1,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",84;
H1,"HOJA",110;
H1,PAGENO,116,SPACE 01;
H2,"E.N.E.P. APAGON",64;
H3,"COMUNIDAD DE PLANEACION>>",59;
H3,"**DEPARTAMENTO DE INFORMATICA**",30,SPACE A2;
H4,"PLANTEL",14;
H4,"CARR",22;
H4,"NOMBRE",47;
H4,"# DE CTA",75;
H4,"SORTEO",87;
H4,"FECHA",101;
H4,"DIA",118;
H5,"*****",14;
H5,"****",22;
H5,"*****",60;
H5,"*****",79;
H5,"*****",87;
H5,"*****",105;
H5,"*****",122;
S2,CCA;
S1,FECHA;
S,NOM;
D1,CPL,12;
D1,CCA,21;
D1,NOM,60;
D1,NCTA,75;
D1,USOR,86;
D1,FECHA,105;
D1,DIA,122;
END;

```

1 #FILES 0,2
2 PROGRAM DAT_SAL
3 CHARACTER TODD*59,AL*2
4 C *****PROGRAMA PARA CAMBIAR FORMATO DE DATOS*****
5 C *****EXTRAIDOS DE LA BASE DE DATOS *****
6 CHARACTER ENTRADA*20
7 CHARACTER SALIDA*20
8 CALL RCPAR(1,ENTRADA)
9 CALL RCPAR(2,SALIDA)
10 OPEN(30, FILE=ENTRADA, STATUS='OLD')
11 OPEN(40, FILE=SALIDA, STATUS='UNKNOWN')
12 K=1
13 write(1,*)'dame el numero de alumnos ==> '
14 KCON=846
15 C read(1,*)KCON
16 KCO=KCON
17 DO WHILE (K.LE.KCO)
18 C READ(30,300,END=301)AL
19 C IF(AL.NE.'1')THEN
20 C BACKSPACE(30)
21 C ENDDIF
22 C 300 FORMAT(A1)
23 READ(30,301,END=302)TODD
24 301 FORMAT(A59)
25 WRITE(40,305)KCON,TODD
26 WRITE(40,305)KCON,TODD
27 305 FORMAT(2X,I4,A59)
28 K=K+1
29 KCON=KCON-1
30 END DO
31 301 CONTINUE
32 302 CONTINUE
33 END

```

```

Modul# DAT_SAL          No errors      Prog:   S65      Blank common: none
FTH#X 5000/S61229      No warnings   Save:  none     Local EMA:  none

```

```

1 #FILES 0,2
2 C*****
3 C*****
4 C*****
5 C***** PROGRAMA PARA ENUMERAR, Y CLASIFICAR A LOS ALUMNOS DE
6 C***** TODA UN PLANTEL POR NUMERO DE SORTEO Y DIA CORRESPON-
7 C***** DIENTE DE INSCRIPCION *****
8 C*****
9 C*****
10 C*****
11 CHARACTER NON*32
12 CHARACTER CTA*8
13 CHARACTER PTL*4
14 CHARACTER NU*6
15 CHARACTER CARR*2
16 CHARACTER PROM*6
17 CHARACTER FECHA(4)*14
18 CHARACTER DIA(4)*10
19 CHARACTER ENTRADA*20
20 CHARACTER SALIDA*20
21 DATA FECHA(1) 2-MAYO-1989//
22 DATA FECHA(2) 3-MAYO-1989//
23 DATA FECHA(3) 4-MAYO-1989//
24 DATA FECHA(4) 5-MAYO-1989//
25 DATA DIA(1) MARTES//
26 DATA DIA(2) MIERCOLES//
27 DATA DIA(3) JUEVES//
28 DATA DIA(4) LUNES//
29 CALL RCPAR(1,ENTRADA)
30 CALL RCPAR(2,SALIDA)
31 OPEN(10,FILE=ENTRADA,STATUS='OLD')
32 OPEN(20,FILE=SALIDA,STATUS='UNK')
33
34 WRITE(1,*)'DIME EL NUMERO DE ALUMNOS'
35 M=3444
36
37 C READ(1,*)M
38 WRITE(1,*)'DAME # DIAS DE INSCRIPCION'
39 LDIAS=4
40 C READ(1,*)LDIAS
41 ALUM=FLOAT(M)/FLOAT(LDIAS)
42 K=M+1
43 NALUM=INT(ALUM)
44 L=0
45 I=LDIAS !POR SER DE 3er. SEMESTRE EN ADELANTE 2o. DIA
46 5 READ(10,10,END=999)HU,PTL,CARR,NON,CTA,PROM
47 10 FORMAT(A6,A4,A2,A32,A8,A6)
48 K=K-1
49 L=L+1
50 IF(I.EQ.2)THEN
51 NALUMH=INT(ALUM)+1
52 ELSE
53 NALUMH=INT(ALUM)
54 END IF
55

```

Page 2 FTH.

Opts: 77/LYI

Sat. Apr 8, 1989. 12:00 PM
/INF004/SORTEO/PON_DIA.FTH

```

56      if(L.eq.NALUM+1)then
57          i=i-1
58          L=0
59      else
60          end if
61
62
63 C      prom='
64      WRITE(20,20)NU,PTL,CARR,NOM,CTA,PROM,FECHA(I),DIA(I)
65 C      WRITE(1,20)NU,PTL,CARR,NOM,CTA,PROM,FECHA(I),DIA(I)
66      20 FORMAT(A5,A4,A2,A32,A8,A6,A12,/,A10)
67      write(1,*)k
68 C      if(k.eq.1)GOTO 999
69      GOTO 5
70 999 WRITE(1,*)'ULTIMO REGISTRADO ==> ',K
71      END

```

Module	FTH.	No errors	Prog:	770	Blank common:	none
FTHFX	5000/861229	No warnings	Save:	none	Local ENA:	none

```

1 *FILES 0,3
2 PROGRAM DAT_SAL
3 CHARACTER TODD*59,AL*2
4 CHARACTER MAT(30,30)*3
5 CHARACTER Mat(30,30)*3
6 CHARACTER MIT(30,30)*3
7
8 CHARACTER GLO(30)*2
9 CHARACTER COMP*10
10 C *****PROGRAMA PARA REALIZAR LA IMPRESION DE*****
11 C *****LAS HOJAS DE INSCRIPCION DE *****
12 C *****ALUMNOS (NOMBRE, CTR, ESCUELA) *****
13 CHARACTER ENTRADA*20
14 CHARACTER leer*140
15 character nuser**4,cpl*1,cca*2,nom*32
16 character ncta*6,pcr*8
17 character DIA*12
18 CHARACTER SALIDA*20
19 CALL PCPAR(1,ENTRADA)
20 CALL PCPAR(3,SALIDA)
21 OPEN(30, FILE=ENTRADA, STATUS='OLD')
22 OPEN(40, FILE=SALIDA, STATUS='UNKNOWN')
23 WRITE(1,*)' PANTALLA E IMPRESORA'
24 READ(1,*)IP
25
26 OPEN(UNIT=IP)
27 write(1,*)'cuantos alumnos ?'
28 read(1,*)K5
29 WRITE(1,*)'DAME LIMITE INFERIOR (NUMERO IMPAR)'
30 READ(1,*)LINI
31 C WRITE(1,*)'DAME LIMITE SUPERIOR (NUMERO PAR)'
32 C READ(1,*)LSUP
33
34 lsup=linf+k5-1
35
36
37 K=1
38 KCON=2
39 DO WHILE (K.LE.K5)
40
41 7 read(30,899,end=302)leer
42 C WRITE(1,899)leer
43 899 format(8X,a140)
44 C WRITE(1,374)LEER(1:50)
45 374 FORMAT(A50)
46 if(leer(1:4).ne.' 407')goto 7
47 backspace(30)
48
49
50 READ(30,301,END=302)cpl,cca,nom,ncta,NUSOR,FECHA,DIA
51 C WRITE(1,301)cpl,cca,nom,ncta,NUSOR,FECHA,DIA
52 301 FORMAT(8X,a4,7X,a2,7X,a32,7X,a8,7X,A4,7X,A12,8X,A12)
53 N9=N9+1
54 IF(N9.LT.LINI) GOTO 7
55 IF(N9.GT.LSUP) THEN

```

```

56          K=K5+1
57          GOTO 77
58      END IF
59      WRITE(I,*)K
60 C      WRITE(I,305)CPL,CCA,NOM,NCTA
61 C      WRITE(I,305)CPL,CCA,NOM,NCTA
62      305 FORMAT(a4,a2,a32,a6)
63      WRITE(IP,700)NUSOR,DIA,HS,NOM,NCTA
64      700 FORMAT(1H1,10//,75X,'SOR:',1X,A4,/,74X,A12,/,
65      *78X,15,/,/,19X,A32,10X,A8,7X, '89-11')
66 C700 FORMAT(1H1,10//,75X,'SOR:',1X,A4,74X,A12,/,
67      *
68
69      WRITE(IP,710)CPL,CCA
70      710 FORMAT(//,19X,'E. H. E. P. ARAGON',25X,A4,11X,A2,/)
71      WRITE(IP,720)(NCTA(I:1),I=1,8),(CPL(I:1),I=2,4),(CCA(I:1),I=1,2)
72 C      720 FORMAT(//,9X,3X,4X,A10,3X,A1,2X,4X,A10,3X,A1,2X,4X,A10,3X,A1,4X,A1,4X
73      720 FORMAT(//,13X,A1,5X,A1,3X,A1,4X,A1,3X,A1,3X,A1,4X,A1,5X,A1,2X,A1,
74      *X,A1,4X,A1,4X,A1,5X,A1)
75
76
77 C      720 FORMAT(//,7X,3X,4X,A10,2X,3X,A1,2X,4X,A10,2X,3X,A1,4X,A10)
78 C      *A1,4X,A1)
79      DO I=1,8
80          GLO(I)=NCTA(I:1)
81 C      WRITE(I,*)GLO(I)
82      END DO
83      DO J=2,4
84          GLO(J+J)=CPL(J:J)
85 C      WRITE(I,*)GLO(J+8)
86      END DO
87      DO K3=1,2
88          GLO(11+K3)=CCA(K3:K3)
89 C      WRITE(I,*)GLO(K3+12)
90      END DO
91      DO J=1,13
92          DO I=1,10
93              COMP='0123456789'
94              IF(GLO(I,J).EQ.COMP(I:1))THEN
95                  MAT(I,J)='##'
96                  MET(I,J)='**'
97                  MIT(I,J)='##'
98              ELSE
99                  MAT(I,J)=' '
100                 MET(I,J)=' '
101                 MIT(I,J)=' '
102             END IF
103         END DO
104     END DO
105     DO I=1,10
106         WRITE(IP,770)(MAT(I,J),J=1,13)
107
108
109     770 format('+',13x, a3,2x,a3,2x, a3, x , a3, 2x, a3, 2x,
110     * a3, x, a3, 2x, a3, 2x, a3, x, a3, 2x, a3, 2x,

```

```

111      *   a3,  x,  a3)
112
113      WRITE(IP,771)(NMT(1,J),J=1,13)
114
115
116      771  format('+',13x, a3, 2x, a3, 2x, a3, x , a3, 2x, a3, 2x,
117      *   a3, x, a3, 2x, a3, 2x, a3, x, a3, 2x, a3, 2x,
118      *   a3, x, a3)
119      C
120      write(ip,550)
121      772  WRITE(IP,772)(NMT(1,J),J=1,13)
122      772  FORMAT('+',13X,A3,2X,A3,2X,A3,X,A3,2X,A3,2X,A3,X,A3,2X,A3,2X,A3,
123      *   X,A3,2X,A3,2X,A3,X,A3)
124      WRITE(IP,550)
125      550  format(' ',1)
126      END DO
127      WRITE(IP,345)
128      345  FORMAT(' ',20X,'ADVERTENCIA: ',4X,'NINGUN ALUMNO PODRA SER
129      *   ',20X,' INSCRITO MAS DE DOS VECES EN UNA',23X,'MISMA ASIGNATURA
130      *   ',20X,'E. N. E. P.  ' ,23X,'
131      *   ')
132
133      MAX=1
134      ICON=ICON-1
135      77 CONTINUE
136
137      END DO
138      302 CONTINUE
139      END

```

Module DAT_SAL
FTN7X 5000/861229

No errors
No warnings

Prog: 5298
Save: none

Blank common: none
Local ENR: none

PROGRAMA 7

Files 0,2

```

C *****
C * PROGRAMA QUE QUITA LOS NUMEROS DE CUENTA REPETIDOS A UN SOLO VALOR
C *
C *****
program imprime
character vari*4, varai*2
character*1 cuenta(186), cta(93)
character archivo1*20, archivo2*20
integer buff(190)
call rcpa(1, archivo1)
call rcpa(2, archivo2)
open(4, file=archivo1, status='old', form='formatted')
open(10, file=archivo2, status='UNK', form='formatted')
do m=1, 13000
CALL LGBUF(BUFF, 190)
WRITE(1, *) ' REGISTRO NO.....', M
read(4, 10, end=40) vari, (cuenta(i), i=1, 186)
10 format(a4, 186a1)
j=0
do k=1, 185, 2
j=j+1
if(cuenta(k).ne.' ' .and. cuenta(k+1).ne.' ') then
cta(j)=cuenta(k)
endif
if(cuenta(k).ne.' ' .and. cuenta(k+1).eq.' ') then
cta(j)=cuenta(k)
endif
if(cuenta(k).eq.' ' .and. cuenta(k+1).ne.' ') then
cta(j)=cuenta(k+1)
endif
if(cuenta(k).eq.' ' .and. cuenta(k+1).eq.' ') then
cta(j)=cuenta(k)
endif
end do
varai='ai'
write(1, 70) varai, (cta(i), i=1, 93)
70 format(a2, 1x, 93a1)
call lgbuf(buff, 190)
write(10, 20) varai, (cta(i), i=1, 93)
20 format(a2, 93a1)
do l=1, 93
cta(i)=' '
end do
END DO
40 CONTINUE
end

```

```

1 #files 0,3
2
3     EMA CALU.C.D.GPU
4     INTEGER*4 CTA
5     INTEGER*4 CARRE
6     INTEGER*4 MD(10)
7     INTEGER CALU(999,99)
8     INTEGER*4 C(999,99)
9     INTEGER*4 D(999,99)
10    INTEGER*4 GPU(999,99)
11    INTEGER CLAC(12),MATE(12)
12    INTEGER B1,B
13    INTEGER B3(10)
14
15
16
17    CHARACTER ENTRADA*20
18    CHARACTER SALIDA*20
19    CHARACTER ALGO*13
20    CHARACTER ALG(12)*6
21
22
23
24
25
26    CHARACTER NT*20
27    CHARACTER CUENTA*10
28    CHARACTER MATE(12)*10
29    CHARACTER GPUPO*10
30    CHARACTER ESC*10
31    CHARACTER CA*10
32 C   CHARACTER CUENTA*10
33
34    CALL RCPAR(1,ENTRADA)
35    CALL RCPAR(2,SALIDA)
36
37    OPEN(10,FILE=ENTRADA,STATUS='UNK')
38
39 ! *** PONER ARCHIVO 20 PARA EL PROGRAMA VALIDA *****
40
41    OPEN(35,FILE=SALIDA,STATUS='UNK')
42
43    NAL=0
44    5 READ(10,10,END=978)CUENTA,ESC,<MATE(I),I=1,10>
45 C   WRITE(1,10)CUENTA,ESC,<MATE(I),I=1,10>
46    NAL=NAL+1
47    10 FORMAT(2X,A8,A5,X,10(A8,1X))
48    B=1
49    B1=1
50
51    DO I=1,8
52
53        IF(CUENTA(I:I).EQ.' ' .OR. CUENTA(I:I).EQ.'*') THEN
54
55            CUENTA='00000000'

```

```

56          WRITE(1,*)'CUENTA INCORRECTA'
57
58      END IF
59
60  END DO
61
62 C      WRITE(1,*)'CUENTA DE ALTA ',CUENTA
63 DO I=1,5
64 C      WRITE(1,*)'VALOR DE ESC ',ESC
65      IF(ESC(I):I).EQ.' ' .OR. ESC(I):I).EQ.'**') THEN
66          ESC='00000'
67 C      WRITE(1,*)'ESCUELA INCORRECTA'
68      END IF
69  END DO
70
71 C      WRITE(1,*)'ESCUELA DE ALTA ',ESC
72      B=0
73      IS=0
74      WRITE(1,*)'SÍ
75      DO I=1,10      ! PONERLE LA ETIQUETA 15 PARA EL PROGRAMA VALIDA
76          MT=MATEX(I)
77 C      WRITE(1,*)'MATERIA ',MT
78
79          DO J=1,5
80
81              IF(MT.EQ.' ' .OR. MT.EQ.'**') THEN
82                  END IF
83 C              IF(MT.EQ.' ' .OR. MT.EQ.'**') THEN
84                  IF(MT(J):J).EQ.' ' .OR. MT(J):J).EQ.'**') THEN
85                      MATEX(I)='00000000'
86 C                      WRITE(1,*)'MATERIA DE ALTA # ',I,' => ',MATEX(I)
87
88 C                      WRITE(30,10)CUENTA,ESC,(MATEX(I),I5=1,10)
89                      GOTO 15
90                  END IF
91
92              END DO
93
94 15      CONTINUE
95      END DO !FIN DEL CICLO ** I **
96      B=1
97 C      GOTO 13
98 88      IF(B.EQ.1)THEN
99          WRITE(35,10)CUENTA,ESC,(MATEX(I),I5=1,10)
100         END IF
101         GOTO 5
102
103 C      END DO      ! FIN DEL CICLO DE ** I **
104
105
106 C 13      CONTINUE
107
108
109
110

```

Page 3 FTH.

Opts: 77/LVI

Fri, Mar 17, 1989, 12:15 PM
/INF004/LISTA/ALTA.FTH

111
112
113 973 CONTINUE
114
115 END

Module FTH.
FTHM: 5900/961229

No errors
No warnings

Prog: 1106
Save: none

Blank common: none
Local EMA: 692307

```

1 #FILES 0.11
2 PROGRAM CLASIFICA_CARREPA
3 ! CLASIFICA DE ARCHIVO DE INSCRIPCION .A LOS ALUMNOS SEGUN SU
4 ! PLANTEL
5 CHARACTER ENTRADA*20
6 CHARACTER SALIDA*20
7 CHARACTER A*20
8 CHARACTER B*95
9 CHARACTER IPL*5
10 CHARACTER ICLA(20)*10
11 CHARACTER PLANTEL(12)*20
12 DATA PLANTEL(1) // INSDER : : 34 //
13 DATA PLANTEL(2) // INSING : : 34 //
14 DATA PLANTEL(3) // INSPED : : 24 //
15 DATA PLANTEL(4) // INSAFO : : 34 //
16 DATA PLANTEL(5) // INSECO : : 34 //
17 DATA PLANTEL(6) // INSDCA : : 34 //
18 DATA PLANTEL(7) // INSCP : : 34 //
19 DATA PLANTEL(8) // INSDI : : 34 //
20
21
22
23 DATA ICLA(1) // 407 //
24 DATA ICLA(2) // 411 //
25 DATA ICLA(3) // 410 //
26 DATA ICLA(4) // 401 //
27 DATA ICLA(5) // 408 //
28 DATA ICLA(6) // 428 //
29 DATA ICLA(7) // 404 //
30
31
32
33 CALL RCPRF(1,ENTRADA)
34 OPEN(10,FILE=ENTRADA,STATUS='UNK')
35 WRITE(1,*) HOLA
36 DO I=1,5
37 OPEN(I+10,FILE=PLANTEL(I),STATUS='UNK')
38 END DO
39 K=0
40 5 READ(10,100,END=999)A,IPL,B
41 C WRITE(1,*)A,' ',IPL,' ',B
42 WRITE(1,*)A
43 100 FORMAT(A10,A3,A9)
44 K=K+1
45 DO L2=1,7
46 IF(IPL.EQ.ICLA(L2))THEN
47 C IF(IPL.EQ.'401'.AND.B(1:2).EQ.'22')THEN
48 C WRITE(80,100)A,IPL,B
49 C END IF
50 WRITE(L2+10,100)A,IPL,B
51 WRITE(1,*)K,' ',IPL
52 END IF
53 END DO
54
55

```

Page 2 CLASIFICA_CARRER Opts: 77/LV:

Mon, Oct 31, 1988, 3:42 PM
/INF006/LISTA/CLAS_CA.FTN

```
56      GOTO 5  
57      *** WRITE(1,*)'FIN'  
58  
59      END
```

Module	CLASIFICA_CARRER	No errors	Prog:	2336	Blank common:	none
FTN74	5000/951229	No warnings	Save:	none	Local EMR:	none

```

1 #files 0,3
2 #EMA (NYE,0)
3 C   EMA CALU,C,D,GPU
4     INTEGER*4 CTA
5     INTEGER*4 CARRE
6     INTEGER*4 NOC(10)
7     COMMON /NYE/CALU(2998,95),C(2998,95)
8     INTEGER CALU
9     INTEGER*4 C
10 C   INTEGER*4 O
11 C   INTEGER*4 GRU(994,99)
12     INTEGER CLAK(12),MAT(12)
13     INTEGER B1,B
14     INTEGER B3(10)
15     INTEGER*4 CI,GRU1,CLA1
16
17     CHARACTER OG*2
18     CHARACTER ENTRADA*20
19     CHARACTER SALIDA*20
20     CHARACTER ALGO*13
21     CHARACTER ALGO(12)*2
22     CHARACTER GRUPOS*6
23     CHARACTER TS*10
24     CHARACTER PROF*35
25     CHARACTER MATER*35
26     CHARACTER MATI*35
27     CHARACTER FALCO*5
28
29
30     CHARACTER NT*20
31     CHARACTER CUENTA*10
32     CHARACTER MATE(12)*10
33     CHARACTER GRUPO*10
34     CHARACTER ESC*10
35     CHARACTER CA*10
36 C   CHARACTER CUENTA*10
37     CHARACTER INTER*20
38     CALL RCPAR(1,ENTRADA)
39     CALL RCPAR(2,SALIDA)
40     CALL RCPAR(3,INTER)
41
42     OPEN(10,FILE=ENTRADA,STATUS='UNK')
43
44 1 *** PONER ARCHIVO 20 PARA EL PROGRAMA VALIDA *****
45
46     OPEN(35,FILE=SALIDA,STATUS='UNK')
47     OPEN(45,FILE=INTER,STATUS='OLD')
48
49
50     NAL=0
51
52     WRITE(1,*)'DAME EL PLANTEL .....'
53     READ(1,33)IPL
54     WRITE(1,*)'DAME EL GRUPO .....'
55     READ(1,33)I11

```

```

56      33      FORMAT(I3)
57 C*****
58 C*****      inicializacion de variables de conteo por grupo      *****
59
60          IVI=1
61          IVF=9
62
63
64 C      7      CONTINUE
65 C          do i=IVI,IVF
66             do il=1,3000
67
68                 CALUK(i1,I11)=0
69
70                 write(1,*) CALUK(i1,I11)
71 cc          write(1,*) CALUK(i8,j8+S0)
72 C          end do
73          end do
74
75 C          do i11=1,10
76             7      CONTINUE
77             IL1=0
78             5      read(10,10,end=978) cuenta, esc, <mate(i), i=1,10>
79             IL1=IL1+1
80             WRITE(1,*)IL1
81 C             WRITE(1,*)CUENTA
82             if(cuenta.eq.'00000000')goto 5
83
84 C             write(1,10)cuenta,esc,<mate(i), i=1,10>
85
86             HAL=NAL+1
87             10      FORMAT(2X,A8,A5,x,10(A8,1X))
88             B=1
89             B1=1
90
91             BACKSPACE(10)
92             READ(10,27)CTA,CARRE,<CLA(K),MAT(K),K=1,10>
93 C             WRITE(1,27)CTA,CARRE,<CLA(K),MAT(K),K=1,10>
94
95
96             27      FORMAT(2X,I8,I5,x,10(2I4,1X))
97             DO K6=1,10
98                 IF<CLA(K6).GE.3000>CLA(K6)=0
99             END DO
100            BACKSPACE (10)
101            READ(10,37)ALGO,<ALG(K),MD(K),K=1,10>
102
103 C            WRITE(1,*)'***CHECA EL ULTIMO NUMERO I1***'
104 C            WRITE(1,37)ALGO,<ALG(K),MD(K),K=1,10>
105             37      FORMAT(2X,A13,x,12(A6,I2,1X))
106
107 C*****
108 C*****      ASIGNACION DE ARCHIVO DE GRUPOS POR CLAVE      *****
109 C*****
110 C*****      SE ASIGNA CTA,CARRERA,GRUPO      *****

```

```

111 C*****
112       do i3=1,10
113
114 C       WRITE(1,*)
115 C       WRITE(1,*)'VALOR DE I3 ==> ',I3
116
117       if(c1a(i3).ne.0.and.md(i3).ne.0) then
118           IF(C1A(I3).LE.3000.OR.C1A(I3).NE.0)THEN
119
120           IF(MD(I3).EQ.I11)THEN
121 C               WRITE(1,*)'IMPORTANTE ...',C1A(I3),',...',MD(I3)
122 C               WRITE(1,*)'VALOR ANTERIOR ...',CALU(C1A(I3),MD(I3))
123
124           IF(CALU(C1A(I3),MD(I3)).EQ.99)THEN
125               WRITE(1,*)'CLAVE==> ',C1A(I3)
126               WRITE(1,*)'GRUPO==> ',MD(I3)
127               WRITE(1,*)'CTA ==> ',CTA
128           ELSE
129               CALU(C1A(I3),MD(I3))=CALU(C1A(I3),MD(I3))+1
130 C               write(1,*)'CALU ...', CALU(C1A(I3),MD(I3))
131
132               C(C1A(I3),CALU(C1A(I3),MD(I3)))=CTA
133 C               WRITE(1,*)'CLA...', C(C1A(I3),CALU(C1A(I3),MD(I3)))
134
135 C               D(C1A(I3),CALU(C1A(I3),MD(I3)))=CARRE
136 C               WRITE(1,*)'CARRERA...', D(C1A(I3),CALU(C1A(I3),MD(I3)))
137
138 C               GRU(C1A(I3),CALU(C1A(I3),MD(I3)))=MAT(I3)
139 C               WRITE(1,*)'GRUPO...',GRU(C1A(I3),CALU(C1A(I3),MD(I3)))
140
141
142 C               WRITE(1,*)'# ALUMNO ==> ',HAL
143 C               WRITE(1,*)'CTA,',CTA,',CARRE,',',MAT(I3)
144           END IF
145       END IF
146       end if
147       END IF
148       end do
149       GOTO 5
150 978 CONTINUE
151       rewind(10)
152       6 READ(45,11,END=900)C1,CLA1,T5,GRU1,MAT1,PROF,SALCU
153       write(1,11)C1,CLA1,T5,GRU1,MAT1,PROF,SALCU
154
155       11 FORMAT(1X,I5,I4,A2,I2,1X,2A35,A5)
156
157       IF(GRU1.EQ.I11)THEN
158           WRITE(35,11)C1,CLA1,T5,GRU1,MAT1,PROF,SALCU
159           WRITE(1,11)C1,CLA1,T5,GRU1,MAT1,PROF,SALCU
160           IF(CALU(C1A1,I11).NE.0)THEN
161 C               ik1=gru(c1a1,1)
162 C               ik2=gru(c1a1,2)
163
164           DO I4=1,CALU(C1A1,I11)
165 C               if(gru(c1a1,i4).eq.ik1.or.gru(c1a1,i4).eq.ik2) then

```

```

166          WRITE(35,35)CLA1,I4,C(CLA1,I4),ID3,I99
167          WRITE(1,35)CLA1,I4,C(CLA1,I4),ID3,I99
168          C(CLA1,I4)=0
169 C         D(CLA1,I4)=0
170 C         GRU(CLA1,I4)=0
171 c         end if
172
173          END DO
174          CALU(CLA1,I11)=0
175          END IF
176          END IF
177
178          GOTO 6
179 900      CONTINUE
180          rewind(45)
181 C       END DO ! FIN DE 'I'
182
183          write(1,*)
184          write(1,*)
185          write(1,*)'*****      otro grupo ? *****'
186          write(1,*)'*****      otro grupo ? *****'
187 C       read(1,130)OG
188          FORMAT(A2)
189 C       130  IF(og.eq.'SI')then
190 C           write(1,*)'dame el grupo ..... '
191 C           read(1,*)ill
192 C
193 C           GOTO 7
194 C       END IF
195
196 35      FORMAT(2X,I4,1X,I3,2X,I8,1X,I5,2X,I4)
197
198
199
200      END

```

Module FTN.
FTN7X 5000/261229

No errors
No warnings

Prog: 1787
Save: none

Blank common: none
Local EMA: none

```

1 #FILES 0,5
2     PROGRAMA EJEMPLO1
3
4     character ESCU*6
5     character ESCU2*6
6     CHARACTER NOMBRE*32
7     CHARACTER CTA*8
8     CHARACTER CTA2*8
9     CHARACTER NOMBRE2*32
10    CHARACTER BAN*3
11    CHARACTER ENTRADA*20
12    CHARACTER INTER*20
13    CHARACTER SALIDA*20
14    CHARACTER PLANTEL*3
15    INTEGER*4 CTAN
16    INTEGER*4 HELEN
17    INTEGER*4 NE
18
19
20    CALL RCPAR(1,ENTRADA)
21    CALL RCPAR(2,SALIDA)
22    CALL RCPAR(3,INTER)
23
24    OPEN(UNIT=25,
25     -   FILE='SALIDA,
26     -   ACCESS='DIRECT',
27     -   STATUS='UNK',
28     -   MAXREC=20000,
29     -   RECL=50,
30     -   FORM='FORMATTED')
31
32    OPEN(10,FILE=ENTRADA,STATUS='UNK')
33    OPEN(55,FILE=INTER,STATUS='UNK')
34
35    WRITE(1,*)'PLANTEL ? '
36    READ(1,*)PLANTEL
37 C   5   READ(10,20,END=999)ESCU,NOMBRE,CTAA !PARA ALING
38 C   5   READ(10,20,END=999)ESCU,NOMBRE,CTAA !PARA ALUM_DIRE
39 C   5   READ(10,20,END=999)CTAA,NOMBRE,ESCU !PARA CINTA15
40 C   5   READ(10,20,END=999)NOMBRE,CTAA,ESCU !PARA DUMPI,MA19
41 C*****
42 C***** se lee de dump1 y de ALING *****
43 C***** por eso son dos formatos diferentes***
44 C*****
45
46 C   20  FORMAT(A32,A8,A5) !PARA DUMPI , MA19
47 C   20  FORMAT(3X,A5,A32,A8) !PARA ALING
48 C   20  FORMAT(8X,A5,3X,A32,2X,A8) !PARA ALUM_DIRE
49 C   20  FORMAT(A8,A32,A5) !PARA CINTA15
50     IF(CTAA.EQ.' ' )GOTO 5
51     DO I=1,8
52     IF(CTAA(I:I).EQ.' '.OR.CTAA(I:I).EQ.'*')GOTO 5
53     END DO
54
55     IF(ESCU(1:3).NE.PLANTEL)GOTO 5

```

```

56 C          IF(ESCU(1:3).EQ.'411'.OR.ESCU(1:3).EQ.'407')GOTO 5
57
58          BACKSPACE(10)
59
60          READ(10,30)NOMBRE,CTAA,ESCU !PARA DUMP1 Y MA19
61          WRITE(1,30)NOMBRE,CTAA,ESCU !PARA DUMP1 Y MA19
62 C          READ(10,30)CTAA,NOMBRE,ESCU !PARA CIN15
63
64 C          READ(10,30)ESCU,NOMBRE,CTAA !PARA ALING
65 C          READ(10,30)ESCU,NOMBRE,CTAA !PARA ALUM_DIRE
66 C          30  FORMAT(5X,A5,A32,I8) !PARA ALING
67 C          30  FORMAT(8X,A5,3X,A32,2X,I8) !PARA ALUM_DIRE
68          30  FORMAT(A32,I8,A5) !PARA DUMP1 Y MA19
69 C          30  FORMAT(I8,A32,A5) !PARA CINTA15
70          HELEM=MOD(CTAA,15737)
71          HE=HELEM
72          IBAN=0
73          DO WHILE( IBAN.EQ.0)
74
75          READ(25,REC=HELEM)ESCU2,NOMBRE2,CTAA2,BAN
76          IF(BAN.EQ.'APA')THEN
77              BAN=' '
78          WRITE(25,99,REC=HELEM)ESCU,NOMBRE,CTAA,BAN
79          WRITE(55,99)ESCU,NOMBRE,CTAA,BAN
80
81          99  FORMAT(A6,A32,A8,A3)
82
83          IBAN=1
84          WRITE(1,*)'REG. GRABADO !'
85          WRITE(1,*)'PRIMERA POS ...',HE
86          WRITE(1,*)'POS. ENCONTRADA ...',HELEM
87
88          ELSE
89              IF(CTAA.NE.CTAA2)THEN
90                  HELEM=HELEM+1
91                  IF(HELEM.GT.32766)THEN
92                      WRITE(1,*)'TABLA DESBORDADA'
93                      WRITE(1,*)'REGISTRO NO GRABADO'
94                      IBAN=2
95                  END IF
96              ELSE
97                  WRITE(1,*)'*** REGISTRO ANTERIORMENTE GRABADO***'
98                  IBAN=4
99              ENDIF
100             END IF
101         END DO
102         GOTO 5
103     999  CONTINUE
104
105         DO L=1,3
106         WRITE(1,*)'RUTINA DE PRUEBA'
107         WRITE(1,*)'DAME EL # DE REGISTRO...'
108         READ(1,16)I
109         16  FORMAT(I5)
110

```

```
111 READ(25,REC=1)ESCU,HOMBRE,CTAA,BAH
112 WRITE(1,*)'REGISTRO # ',I,' ',ESCU,HOMBRE,CTAA,BAH
113 END DO
114
115 END
```

Module AEJEMPL01
FTH7X 5000/861229

No errors
No warnings

Prog: 1608
Save: none

Blank common: none
Local EMA: none

```
111      READ(25,REC=1)ESCU,HOMBRE,CTAA,BAN
112      WRITE(1,*)'REGISTRO # ',I,' ',ESCU,HOMBRE,CTAA,BAN
113      END DO
114
115      END
```

Module AEJEMPL01
FTH7X 5000/861229

No errors
No warnings

Prog: 1608
Save: none

Blank common: none
Local EHA: none

```

1 #FILES 0,5
2   PROGRAMA EJEMPLO1
3
4   character ESCU*6
5   character ESCU2*6
6   CHARACTER NOMBRE*32
7   CHARACTER CTA*8
8   CHARACTER CTA2*8
9   CHARACTER NOMBRE2*32
10  CHARACTER BAN*3
11  CHARACTER ENTRADA*20
12  CHARACTER SALIDA*20
13  CHARACTER ALGO*125
14  CHARACTER DIRECTO*20
15  INTEGER*4 CTAN
16  INTEGER*4 CTAN2
17  INTEGER*4 NELEM
18  INTEGER*4 NE
19  INTEGER*4 CLAVE,HAL,CTA,CARR,GRU
20  INTEGER*4 CLAVE2,HAL2,CTA2,CARR2,GRU2
21
22  CALL FCPAR(1,ENTRADA)
23  CALL FCPAR(2,SALIDA)
24  CALL FCPAR(3,DIRECTO)
25  OPEN(UNIT=25,
26  -   FILE=DIRECTO,
27  -   ACCESS='DIRECT',
28  -   STATUS='UNK',
29  -   MAXREC=20000,
30  -   RECL=50,
31  -   FORM='FORMATTED')
32
33  OPEN(10,FILE=ENTRADA,STATUS='UNK')
34  OPEN(20,FILE=SALIDA,STATUS='UNK')
35
36  5  READ(10,28,END=999)ALGO
37  C  WRITE(1,28)ALGO
38
39  28  FORMAT(A120)
40     IF(ALGO(1:2)'.NE.' )THEN
41     WRITE(20,20)ALGO
42  20  FORMAT(A120)
43     GOTO 5
44     END IF
45
46     BACKSPACE(10)
47
48     READ(10,30)CLAVE,HAL,CTA,CARR,GRU
49     WRITE(1,30)CLAVE,HAL,CTA,CARR,GRU
50  30  FORMAT(2X,14,1X,13,2X,18,1X,15,2X,14)
51     NELEM=MOD(CTA,15737)
52     NE=NELEM
53     IBAN=0
54     ctan2=1964
55     DO WHILE( IBAN.EQ.0)

```

```

56      N2=NELEM-NE
57      if<ctan2.eq.0>then
58 C      IF<N2.EQ.52>THEN
59 C          WRITE(1,*)'REGISTRO NO ENCONTRADO'
60          WRITE(20,378)CLAVE,NAL,CTA,CARR,GRU
61      378      FORMAT(2X,I4,1X,I3,2X,I8,1X,I5,2X,I4,1X)
62          IBAN=3
63      ELSE
64
65      READ(25,48,REC=NELEM)ESCU2,NOMBRE2,CTAN2,BAN
66      48      FORMAT(A6,A32,I8,A3)
67 C      WRITE(1,*)'REGISTRO EN DIR...',NELEM
68
69 C      WRITE(1,*)'VALOR DE CTA EN DIRECTO...',CTAN2
70 C      WRITE(1,*)'VALOR DE CTA EN SECUENC...',CTA
71          IF<CTA.EQ.CTAN2>THEN
72              BAN='
73              WRITE(20,45)CLAVE,NAL,CTA,CARR,GRU,NOMBRE2
74 C              WRITE(1,45)CLAVE,NAL,CTA,CARR,GRU,NOMBRE2
75      45      FORMAT(2X,I4,1X,I3,2X,I8,1X,I5,2X,I4,1X,A35)
76              IBAN=1
77 C              WRITE(1,*)'REG. GRABADO I'
78 C              WRITE(1,*)'PRIMERA POS...',NE
79 C              WRITE(1,*)'POS. ENCONTRADA...',NELEM
80
81      ELSE
82          NELEM=NELEM+1
83          IF<NELEM.GT.32768>THEN
84 C              WRITE(1,*)'TABLA DESBORDADA'
85 C              WRITE(1,*)'REGISTRO NO GRABADO'
86              IBAN=2
87          END IF
88      END IF
89      END IF
90      END DO
91      GOTD 5
92      999 CONTINUE
93
94 C      DO L=1,5
95 C          WRITE(1,*)'ROUTINA DE PRUEBA'
96 C          WRITE(1,*)'DAME EL # DE REGISTRO...'
97 C          READ(1,16)I
98 C      16      FORMAT(I5)
99
100 C      READ(25,REC=1)ESCU,NOMBRE,CTAA,BAN
101 C          WRITE(1,*)'REGISTRO # ',I,' ',ESCU,NOMBRE,CTAA,BAN
102 C          END DO
103
104      END

```

Module AEJEMPL01
FTN7X. 5000/861229

No errors
No warnings

Prog: 1387
Save: none

Blank common: none
Local EMA: none

```

1 0 : program ordenar(input,output,file,ft);
2 0 : type
3 0 :   carrera=packed array[1..5] of char;
4 0 :   cuenta=packed array[1..8] of char;
5 0 :   dato=packed array[1..80] of char;
6 0 :   nombre=packed array[1..32] of char;
7 0 :   orden=packed array[1..130] of cuenta;
8 0 :   nombr=packed array [1..130] of nombre;
9 0 :   carr=packed array[1..130] of carrera;
10 0 :
11 0 :   VAR
12 0 :     nomb:nombr;
13 0 :     car:carr;
14 0 :     ord:orden;
15 0 :     temp_car:carrera;
16 0 :     temp1:cuenta;
17 0 :     temp2:carrera;
18 0 :     temp:nombre;
19 0 :     menor:nombre;
20 0 :     j,n,k,i:integer;
21 0 :     cta:cuenta;
22 0 :     nm:nombre;
23 0 :     entra:dato;
24 0 :     ft:text;
25 0 :     file:text;
26 0 :     q,h,rec,m:integer;
27 0 :     cl:integer;
28 0 : (*****
29 0 : (***** procedure para colocar ceros *****
30 0 : (***** en clave de grupo y materia:*****
31 0 : (*****
32 1 :   procedure zero(var entra:dato);
33 1 :     var cl:integer;
34 1 :   begin
35 1 :     for cl:=1 to 14 do
36 1 :       begin
37 1 :         if(entra[cl]=' ') then
38 1 :           begin
39 1 :             entra[cl]:='0';
40 1 :           end;
41 1 :         end;
42 1 :       end;
43 0 : (*****
44 0 : (***** inicia program *****
45 0 : (***** para ordenar datos *****
46 0 : (*****
47 0 :
48 1 :   procedure ordena_datos (var nomb:nombr;var ord:orden;
49 1 :     var car:carr;k:integer);
50 1 :   var
51 1 :     pos,var1,var2:integer;
52 1 :   begin
53 1 :     for var1:=1 to k-1 do

```

```

54 1 2          begin
55 1 2          menor:=nomb[var1];
56 1 2          pos:=var1;
57 1 2          for var2:=var1+1 to k do
58 1 3              begin
59 1 3                  if (nomb[var2]<menor) then
60 1 4                      begin
61 1 4                          menor:=nomb[var2];
62 1 4                          pos:=var2;
63 1 4                      end; ( final del if)
64 1 3                  end; ( final del ciclo interno)
65 1 2          temp:=nomb[var1]; temp1:=ord(var1); temp2:=car[var1];
66 1 2          nomb[var1]:=nomb[pos]; ord[var1]:=ord[pos]; car[var1]:=car[pos];
67 1 2          nomb[pos]:=temp;ord[pos]:=temp1; car[pos]:=temp2;
68 1 2          end (final del ciclo externo)
69 1 1          end;
70 0 : (*****
71 0 : (*****termina procedure y comienza el programa*****
72 0 : (*****
73 0 1          begin
74 0 1              reset(fle);
75 0 1              rewrite(ft);
76 0 1              rec:=0;
77 0 1              while (not eof(fle)) do
78 0 1          (*          for j:=1 to 431 do          *)
79 0 2                  begin
80 0 2                      readln(fle,entra);
81 0 2          if ((entra[1]=' ')and(entra[2]=' ')and(entra[3]<>'4')) then
82 0 3              begin
83 0 3                  for a:=1 to 8 do
84 0 4                      begin
85 0 4                          cta[a]:=entra[m+12];
86 0 4                      end; ( final del for a)
87 0 3                  for h:=1 to 32 do
88 0 4                      begin
89 0 4                          nm[h]:=entra[h+36];
90 0 4                      end;
91 0 3                  for q:=1 to 5 do
92 0 4                      begin
93 0 4                          temp_car[q]:=entra[q+21];
94 0 4                      end;
95 0 3                  k:=k+1;
96 0 3                  ord[k]:=cta;
97 0 3                  nomb[k]:=nm;
98 0 3                  car[k]:=temp_car;
99 0 3                  end else
100 0 3              end else
101 0 3                  begin
102 0 3                      rec:=rec+1;
103 0 3                      if (rec=1) then
104 0 4                          begin
105 0 4                              zero(entra);
106 0 4                              writeIn(ft,entra);

```

PROGRAMA 13-B

Pascal/1000
Ver. 2/5000Mon Dec 5, 1988 1:31 pm
ORDEMA Page 3

```

107 0 4          (****  writeln(ft);  *****)
108 0 4          end;
109 0 3          end; ( final del if)
110 0 2          if (eof(file)) then rec:=2;
111 0 2
112 0 2          if (rec=2) then
113 0 3          begin
114 0 3
115 0 3          ordna_datos(nomb,ord,car,k);
116 0 3          for n:=1 to k do
117 0 4          begin
118 0 4          writeln(ft,n:3,ord[n]:10,car[n]:6,nomb[n]:33);
119 0 4          end;
120 0 3          (****  writeln(ft);  *****)
121 0 3          if (not eof(file)) then
122 0 4          begin
123 0 4          zero(entra);
124 0 4          writeln(ft,entra);
125 0 4          end;
126 0 3          rec:=1;
127 0 3          k:=0;
128 0 3
129 0 3          end;
130 0 2          end; ( for j )
131 0 1          end.

```

```

0 Errors detected.
0 Warnings issued.
131 Source lines read.
5038 Words of program generated.

```

```

1 *FILES 0,C
2   PROGRAMA EJEMPL01
3
4   character ESCU*6
5   character ESCU2*6
6   CHARACTER NOMBRE*32
7   CHARACTER CTA*8
8   CHARACTER CTA2*8
9   CHARACTER NOMBRE2*32
10  CHARACTER BAH*3
11  CHARACTER ENTRADA*20
12  CHARACTER SALIDA*20
13  CHARACTER ALGO*125
14  INTEGER*4 CTAH
15  INTEGER*4 CTAH2
16  INTEGER*4 HELEN
17  INTEGER*4 NE
18  INTEGER*4 CLAVE,NAL,CTA,CARR,GRU
19  INTEGER*4 CLAVE2,NAL2,CTA2,CARR2,GRU2
20  CALL PCPAR(1,SALIDA)
21
22
23
24  OPEN UNIT=25,
25  -   FILE='DIRECT',
26  -   ACCESS='DIRECT',
27  -   STATUS='UNK',
28  -   MAXREC=20000,
29  -   RECL=50,
30  -   FORM='FORMATTED')
31
32  OPEN IO,FILE=SALIDA,STATUS='UNK')
33
34
35  DO L=1,20000
36
37      READ(25,REC=L)ESCU,NOMBRE,CTA,BAH
38      IF(BAH.EQ.'ARA')THEN
39          WRITE(1,*)L
40      ELSE
41          WRITE(1,*)L
42          WRITE(10,40)L,ESCU,NOMBRE,CTA
43          40  FORMAT(1X,15,2X,A6,2X,A32,2X,A8)
44      ENDIF
45  END DO
46
47  END

```

Module AEJEMPL01
FTH7X 5000/361229

No errors Prog: 1151
No warnings Save: none

Blank common: none
Local EMA: none

```

1 $FILES 0,2
2 PROGRAM FORMMIGUEL
3 CHARACTER ALGO*80
4 CHARACTER ALG2*36
5 CHARACTER ENTRADA*20
6 CHARACTER SALIDA*20
7 CHARACTER COORD*35
8 CHARACTER CCA*10
9 CHARACTER CMAT*10
10 CHARACTER CGPO*10
11 CHARACTER MATERIA*35
12 CHARACTER PROF*32
13 CHARACTER FECHA*8
14 CHARACTER PERIO*8
15 INTEGER A,B,C,D,HOJA
16 REAL C1
17 CALL RCPAR(1,ENTRADA)
18 CALL RCPAR(2,SALIDA)
19 OPEN(10,FILE=ENTRADA,STATUS='OLD')
20 OPEN(20,FILE=SALIDA,STATUS='UNK')
21 20 FORMAT(A76)
22 WRITE(1,*)'DAME LA COORDINACION'
23 READ(1,3)COORD
24 3 FORMAT(A35)
25 WRITE(1,*)'DAME LA FECHA (DD/MM/AA)'
26 READ(1,4)FECHA
27 4 FORMAT(A8)
28 WRITE(1,*)'DAME EL PERIODO (AA-I) o (AA-II)'
29 READ(1,4)PERIO
30 ICON_MAT=1
31 C1=1.
32 HOJA=1
33 IMAT=IMAT + 1
34 55 READ(10,20,END=111)ALGO
35 IF(ALGOC(1:3).EQ.' 04')THEN
36 HOJA=1
37 50 FORMAT(5X,A80)
38 BACKSPACE(10)
39 WRITE(20,60)IMAT,HOJA,FECHA,PERIO,COORD
40 WRITE(1,60)IMAT,HOJA,FECHA,PERIO,COORD
41 K=0
42 C1=0.
43 ICON_MAT=ICON_MAT+1
44 IMAT=IMAT+1
45 READ(10,30,END=111)CCA,CMAT,CGPO,MATERIA,PROF
46 30 FORMAT(2X,A5,A4,A4,1X,A35,A32)
47 WRITE(20,59)CCA,CMAT,CGPO,MATERIA,PROF
48 WRITE(1,59)CCA,CMAT,CGPO,MATERIA,PROF
49 59 FORMAT(/,4X,'CARRERA: ',A5,5X,'CLAVE: ',A4,5X,'GRUPO: ',A4
50 *,/,4X,'MATERIA: ',A35,10X,'PROFESOR: ',A32,/)
51 WRITE(20,87)
52 WRITE(1,87)
53
54 ELSE
55 K=K+1

```

PROGRAMA 15-A

Page 2 FORMMIGUEL

Opts: 77/LYI

Wed Apr 27, 1988 5:19:44 pm
/INF006/TEMPORAL/FORM_LIST

```

56      C1=C1+1.
57      format(4x,120('='),/)
58      *      ,8X,'No.',x,'# CUENTA',x,'CARRE',10X,'NOMBRE',/,
59      *      4X,120('='),/)
60
61      WRITE(20,50)ALGO
62      WRITE(1,50)ALGO
63      IF<<C1/35>>.EQ.INT<<C1/35>>THEN
64          ICON_MAT=ICON_MAT+1
65          IMAT=IMAT+1
66          HOJA=HOJA+1
67          WRITE(20,60)IMAT,HOJA,FECHA,PERIO,COORD
68          WRITE(1,60)IMAT,HOJA,FECHA,PERIO,COORD
69      60  FORMAT(1H1,2X,'PAG # ',I5,
70      *      27X,'UNIVERSIDAD NACIONAL AUTONOMA DE MEXIC
71      *      ',3X,'HOJA # ',I3,/,2X,'FECHA : ',A8,      32X,
72      *      'E. N. E. P. ARAGON',/,2X,'PERIODO : ',A6      ,32X'UNIDAD DE PLF
73      *      'EACION',/,35X,50C('*'),/,35X,'*****',6X, 'DEPARTAMENTO DE INFORME
74      *      'ICA',7X,
75
76      *****',/,35X,50C('*'),/,48X,A35,/)
77      *, 10X,'>>> IMPORTANTE : VERIFIQUESE QUE TODOS LOS ALUMNOS INSCRI
78      *DS SE ENCUENTREN EN ESTA LISTA.',/)
79      *28X,      'SI SOLO APARECE EL NUMERO DE CUENTA O NO
80      *APARECE EL ALUMNO,ESTE DEBERA ',/
81      *28X,      'PASAR A
82      *A CORRECCION',/)
83
84      WRITE(20,59)CCA,CMAT,CGPO,MATERIA,PROF
85      WRITE(1,59)CCA,CMAT,CGPO,MATERIA,PROF
86      write(20,87)
87      write(1,87)
88
89      END IF
90  ENDIF
91  GOTO 55
92  !!! CONTINUE
93
94      WRITE(1,*)'MATERIAS ATENDIDAS ==> ',ICON_MAT
95
96
97  END

```

Module FORMMIGUEL
FTN7X 2440/850304No errors
No warningsProg: 1444
Save: NoneBlank Common: None
Local Ema: None

```

files 0,2
C*****
C PROGRAMA QUE CALCULA LOS ALUMNOS INSCRITOS POR SEMESTRE Y CARRERA
C DEL ARCHIVO PRINCIPAL (CAMBIAR EL RANGO MAXIMO ACTUAL=10000)
C*****
program ESTAINSCRIP
character CARR*5, AREA*2, sem*2, ELEC
character*2 ASEM(9), CARRER(12)*5
character archivo1*20, archivo2*20
integer acont(12,9), UL, SUM(12), SUMP(15), TTOTAL, TX
DATA CARRER/'40121', '40122', '40421', '40423', '40424', '40721',
*'40821', '41025', '41121', '41126', '41132', '42021'/
DATA asem/'21', '22', '23', '24', '25', '26', '27', '28', '29'/
call rcparr(1, archivo1)
call rcparr(2, archivo2)

open(4, file=archivo1, status='old', form='formatted')
open(10, file=archivo2, status='UNK', form='formatted')

C OPEN (UNIT=6)
WRITE(1,104)
104 FORMAT(//,5X,"PROCESO ESTADISTICO DE INSCRIPCION",//,2X,
*"POR IMPRESORA.....A",//,2X,
*"POR PANTALLA.....B",//,2X,
*"SELECCIONES UNA OPCION.....")
READ(1,105)ELEC
105 FORMAT(A1)
IF (ELEC .EQ. 'A') THEN
  UL=6
ELSE
  UL=10
ENDIF
DO i=1,13000
  read(4,10, end=40)CARR, SEM
  10 format(10X, A5, 1X, 4X, A2)
  WRITE (1,*)""
  WRITREGISFRO NO", I
  DO M=1,12
    IF (CARR .EQ. CARRER(M)) THEN
      do j=1,9
        if (SEM .eq. asem(j)) then
          acont(M,j)=acont(M,j)+1
        endif
      END DO
    ENDIF
  END DO
  END DO
40 CONTINUE
WRITE(UL,131)
131 FORMAT(//,10X,"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO",//,
*" UNIDAD DE PLANEACION.....DEPARTAMENTO DE INFORMATICA ",
*///)
WRITE(UL,109)
109 FORMAT(//,1X,"ESTADISTICA DE INSCRIPCION DE LA E.H.E.P. ARAGON",
*//,1X," SEMETRES",//,26X," 1 2 3 4 5 6 7 8 ",
*" 9 TOTAL",//,25X,"
*" _____",//)
DO H=1,12
  DO G=1,9

```

PROGRAMA 16-A

```

SUM(H)=SUM(H)+ACONT(H,G)
SUMPG)=SUMPG)+ACONT(H,G)
END DO
TTOTAL=TTOTAL+SUM(H)
END DO
DO HG=1,9
TX=TX+SUMP(HG)
END DO
DO M=1,12
70 write(UL,70)CARRER(M),(ACONT(M,K),K=1,9),SUM(H)
format(2X,"CARRERA...",A5,5X,9I5,5X,I4)
END DO
110 WRITE(UL,110)
FORMAT(25X,"-----")
*" ",')
WRITE(UL,178)(SUMP(UJ),UJ=1,9),TTOTAL,TX
178 FORMAT(2X,"TOTAL....",10X,9I5,4X,I5,5X,I5)
end

```

\$files 0,2

```

C *****
C   PROGRAMA QUE CALCULA EL SOBRE CUPO DE LAS CARRERAS
C   PAS04,RES404,SALIDA
C *****
program REBASAGPO
character VAR1*94,VAR2*2
character archivo1*20, archivo2*20
integer VARR3
call rcpac(1,archivo1)
call rcpac(2,archivo2)

open(4,file=archivo1,status='old',form='formatted')
open(10,file=archivo2,status='UNK',form='formatted')
DO I=1,600
  READ(4,50,END=90)VAR1,VAR2,VAR3,VAR4
  WRITE(1,50)VAR1,VAR2,VAR3,VAR4
C   50  FORMAT(A94,A2,X,I4,X,A6)
        WRITE(1,*)' VOY EN EL REG NO..',I
        IF (VAR2.EQ.'30') THEN
          IF (VAR3.GE.'25') THEN
            WRITE(10,60)VAR1,VAR2,VAR3,VAR4
C   60  FORMAT(A94,A2,X,I4,X,A6)
          END IF
        ENDIF
        IF (VAR2.EQ.'40') THEN
          IF (VAR3.GE.'35') THEN
            WRITE(10,60)VAR1,VAR2,VAR3,VAR4
          ENDIF
        END IF
        IF (VAR2.EQ.'60') THEN
          IF (VAR3.GE.'55') THEN
            WRITE(10,60)VAR1,VAR2,VAR3,VAR4
          ENDIF
        END IF
      END DO
C   90  CONTINUE
      END

```

```

1 $FILES 0,5
2     PROGRAMA EJEMPL01
3
4     character ESCU*6
5     character ESCU2*6
6     CHARACTER NOMBRE*32
7     CHARACTER CTA*8
8     CHARACTER CTA2*8
9     CHARACTER NOMBRE2*32
10    CHARACTER BAN*3
11    CHARACTER ENTRADA*20
12    CHARACTER SALIDA*20
13    INTEGER*4 CTAN
14    INTEGER*4 HELEN
15    INTEGER*4 NF
16
17
18    CALL RCPAR(1,ENTRADA)
19 C   CALL RCPAR(2,SALIDA)
20
21    OPEN UNIT=25,
22         -   FILE='ENTRADA',
23         -   ACCESS='DIRECT',
24         -   STATUS='DISK',
25         -   MAXREC=20000,
26         -   RECL=56,
27         -   FORM='FORMATTED')
28
29         ESCU='
30         NOMBRE='
31         CTA='00000000'
32         BAN='AN'
33     DO I=1,20000
34         WRITE(25,REC=I)ESCU,NOMBRE,CTA,BAN
35     WRITE(1,*)'REGISTRO.....',I
36     END DO
37
38
39
40     END

```

Module AEJEMPL01
FTH7X 5000/861229

No errors
No warnings

Prog: 1077
Save: none

Blank common: none
Local EMA: none

```
$files 0,2
```

```
C *****
C * PROGRAMA QUE PONE EN FORMA VERTICAL LAS CLAVES Y LOS GRUPOS
C * 1ER PROGRAMA
C *
C *****
  program imprime
  character var1*8,VAR2*8,VAR3*8,VAR4*8,VAR5*8,VAR6*8
  character VAR7*8,VAR8*8,VAR9*8,VAR10*8
  CHARACTER NOMBRE*32,ESCUELA*6
  CHARACTER ARCHIVO1*20
  CHARACTER ARCHIVO2*20
  character var(10)*8
  INTEGER*4 CTA
  integer buff(500)
  call rcparr(1,archivo1)
  call rcparr(2,archivo2)
  open(4,file=archivo1,status='old',form='formatted')
  open(10,file=archivo2,status='unk',form='formatted')
  do 40 k=1,12000

  CALL LGBUF(BUFF,500)
  read(4,10,ans=67)NOMBRE,CTA,ESCUELA,(VAR(I),I=1,10)
10  FORMAT(A32,I8,A6,10(X,A8))
  WRITE(1,*)' REGISTRO #',K
  call lgbuff(buff,500)
  write(10,20)nombre,cta,escuela
20  format(a32,i8,x,a6)
  DO I=1, 10
  if (VAR(I) .NE. '00000000') then
  WRITE(10,31)VAR(I)
31  FORMAT(A8)
  ENDIF
  ENDDO
40  continue
67  continue
  end
```

#FILES 0,5

PROGRAMA EJEMPLO1

C***** DEFINICION DE VARIABLES *****

```

character ESCU*6
character ESCU2*6
CHARACTER NOMBRE*32
CHARACTER CTAA*8
CHARACTER CTAA2*8
CHARACTER NOMBRE2*32
CHARACTER BAN*3
CHARACTER ENTRADA*20
CHARACTER SALIDA*20
CHARACTER ALGO*125
CHARACTER DIRECTO*20
CHARACTER MATERIA*54
CHARACTER MATER2*36
CHARACTER RUNNAME*30, STRI1*30, STRI2*30, STRI3*30
INTEGER ERROR, PRANS(5)
INTEGER*4 CTAH
INTEGER*4 CARR
INTEGER*4 CTA
INTEGER*4 CTAN2
INTEGER*4 HELEM
INTEGER*4 NE
INTEGER*4 CLAVE2, NAL2, CTA2, CARR2, GRUZ
INTEGER BUFF(200)
DATA STRI1/'DBBLD VTIR1::26 1 ADD'/
DATA STRI2/'QUERY, VREP1::26,1,1,ECHO'/

```

```

CALL RCPAR(1, ENTRADA)
CALL RCPAR(2, SALIDA)
CALL RCPAR(3, DIRECTO)

```

C***** INICIO DE PROGRAMA *****

OPEN(UNIT=25,

```

- FILE=DIRECTO,
- ACCESS='DIRECT',
- STATUS='UNK',
- MAXREC=20000,
- RECL=50,
- FORM='FORMATTED')

```

```

OPEN(10, FILE=ENTRADA, STATUS='UNK')
OPEN(20, FILE=SALIDA, STATUS='UNK')
CALL LGBUF(BUFF, 200)
WRITE(20, 35)

```

35 FORMAT('TIRA:23:26, VRV: ',/, '#SET:REGI')

```

5 READ(10, 28, END=999)ALGO
28 FORMAT(A11)
IF(ALGO(1:2).NE.' ')THEN
WRITE(20, 20)ALGO
20 FORMAT(A11)
GOTO 5
END IF

```

BACKSPACE(10)

READ(10, 30)CTA, CARR, MATERIA, MATER2

```

C      WRITE(1,30)CTA,CARR,MATERIA,MATER2
30     FORMAT(2X,18,15,1X,A54,A36)
        HELEM=MOD(CTA,15737)
        HE=HELEM
        IBAN=0
        ctan2=1964
        DO WHILE (IBAN.EQ.0)
            H2=HELEM-HE
            IF(CTAN2.EQ.0) THEN
C      C      WRITE(1,*)'REGISTRO NO ENCONTRADO'
C      C      WRITE(20,378)CTA,CARR,MATERIA,MATER2
378     FORMAT(18,16,32X,A54,A36)
            IBAN=3
            ELSE
                READ(25,48,REC=HELEM)ESCU2,NOMBRE2,CTAN2,BAN
C      C      WRITE(1,48)ESCU2,NOMBRE2,CTAN2,BAN
48     FORMAT(A6,A32,18,A3)
            IF(CTA.EQ.CTAN2)THEN
                BAN='
                WRITE(20,45)CTA,CARR,NOMBRE2
                WRITE(20,11)MATERIA
                WRITE(20,12)MATER2
                WRITE(1,13)CTA,CARR,NOMBRE2,MATERIA,MATER2
13     FORMAT(2X,18,1X,18,1X,A32,X,A54,A36,/)
45     FORMAT(18,16,A32)
11     FORMAT(A54)
12     FORMAT(A36)
                IBAN=1
            ELSE
                HELEM=HELEM+1
                IF(HELEM.GT.32766)THEN
C      C      WRITE(1,*)'TABLA DESBORDADA'
C      C      WRITE(1,*)'REGISTRO NO GRABADO'
                IBAN=2
                END IF
            END IF
        END DO
        GOTO 5
999    CONTINUE
        write(20,19)
19     format('$END')
C***** DA LA ALTA AUTOMATICAMENTE A LA BASE DE DATOS ****
        WRITE(1,*)'
        CLOSE(25)
        CLOSE(10)
        CLOSE(20)
        WRITE(1,*)'DANDO DE ALTA LA BASE DE DATOS.....'
        ERROR=FMPRUNPROGRAM(STR1,PRAMS,RUNNAME)

        WRITE(1,*)'
        WRITE(1,*)'PURGANDO ARCHIVO SCRACHT.....'
        ERROR=FMPPURGE(VTIR1)
        ERROR=FMPRUNPROGRAM(STR2,PRAMS,RUNNAME)

```

```

$files 0,5
programa asignacion

C *****
C *ASIGNACION DE LAS MATERIAS A LAS TIRAS UTILIZANDO *
C * UTILIZANDO DIRECCIONAMIENTO DIRECTO *
C *****
  character materia*35,es*2
  character materia2+35,es2*2
  character ban*3
  character carrera*5
  character carrera2+5
  character entrada*20
  character salida*20
  character MATERIAS*20
  character directo*20
  character algo*50
  character nombre*32,cuenta*8
  character gpo*4,sem*2
  integer cvemat

  integer a(3000)

  integer cvemat2
  integer*4 neleq
  integer*4 ne
  integer*4 temporal

  call rcpa(1,MATERIAS)
  call rcpa(2,entrada)
  call rcpa(3,SALIDA)
  CALL RCPA(4,DIRECTO)
  open(unit=25,
-   file=DIRECTO,
-   access='direct',
-   status='OLD',
-   recl=48,
-   form='formatted')

  OPEN(45,FILE=MATERIAS,STATUS='OLD')

  open(10,file=entrada,status='old')
  open(20,file=salida,status='unk')
  I7=0
  i=1
  DO WHILE(I7.EQ.0)
3  READ(45,77,END=5)TEMPORAL
77  FORMAT(14)
  A(TEMPORAL)=I
  write(1,*)a(temporal)
  I=I+1

  END DO

  5  read(10,30,end=999)algo
  WRITE(1,30)ALGO
30  format(a49)
  if (algo(1:1)) .ge. '0' .and.algo(1:1) .le. '9') then
    backspace(10)

```

```
      read(10,200)cvemat,gpo
200   format(I4,a4)
      K=a(CVEMAT)
      write(1,* )k

      read(25,220,rec=k)cvemat2,MATERIA2,sem,carrera
220   format(I4,2X,A35,a2,a5)
      WRITE(1,220)CVEMAT2,MATERIA2,sem,carrera
      if (cvemat.eq.cvemat2) then
45     write(20,45)cvemat,gpo,materia2
      format(4X,I4,2X,a4,2X,a35)
      else
48     write(20,48)cvemat,gpo
      format(4X,I4,2X,a4,2X,'ERROR EN LLENADO DE HOJA')
      end if
      else
46     write(20,46)algo
      format(a50)
      end if
      GOTO 5
999   continue
      end
```

#files 0,3

```

EMA CALU,C,D,GRU
INTEGER*4 CTA
INTEGER*4 CARRE
INTEGER*4 HOC(10)
INTEGER CALU(999,99)
INTEGER*4 C(999,99)
INTEGER*4 D(999,99)
INTEGER*4 GRU(999,99)
INTEGER CLA(12),MATE(12)
INTEGER B1,B
INTEGER B3(10)

```

```

CHARACTER ENTRADA*20
CHARACTER SALIDA*20
CHARACTER ALGO*13
CHARACTER ALGO(12)*6

```

```

CHARACTER HT*20
CHARACTER CUENTA*10
CHARACTER MATE(12)*10
CHARACTER GRUPO*10
CHARACTER ESC*10
CHARACTER CA*10
C CHARACTER CUENTA*10

```

```

CALL RCPAR(1,ENTRADA)
CALL RCPAR(2,SALIDA)

```

```

OPEN(10,FILE=ENTRADA,STATUS='UNKNOWN')

```

```

! *** PONER ARCHIVO 20 PARA EL PROGRAMA VALIDA *****

```

```

OPEN(35,FILE=SALIDA,STATUS='UNKNOWN')

```

```

HAL=0

```

```

5 READ(10,10,END=999)CUENTA,ESC,(MATE(I),I=1,10)

```

```

C WRITE(1,10)CUENTA,ESC,(MATE(I),I=1,10)

```

```

HAL=HAL+1

```

```

10 FORMAT(2X,A8,A5,X,10(A8,1X))

```

```

B=1

```

```

B1=1

```

```

DO I=1,3

```

```

IF(CUENTA(I:I).EQ.' ' .OR. CUENTA(I:I).EQ.'*')THEN

```

```

CUENTA='00000000'

```

```

WRITE(1,*)'CUENTA INCORRECTA'

```

```

END IF

```

```

END DO

```

```

C      WRITE(I,*)'CUENTA DE ALTA ',Cuenta
DO I=1,3
C      WRITE(I,*)'VALOR DE ESC ',ESC
      IF(ESC(I:1).EQ.' ' .OR.ESC(I:1).EQ.'*') THEN
C          ESC='00000'
          WRITE(I,*)'ESCUELA INCORRECTA'
C      END IF
      END DO

C      WRITE(I,*)'ESCUELA DE ALTA ',ESC
      B=0
      I3=0
      write(I,*)n31
      DO I=1,10 ! PONERLE LA ETIQUETA 15 PARA EL PROGRAMA VALIDA
C          MT=MATEC(I)
          WRITE(I,*)'MAL, ',MT

C          DO J=1,9

C              IF(MT.EQ.' ' )THEN
                  END IF
C          B3(I)=1
          IF(MT(J:J).EQ.' ' .OR.MT(J:J).EQ.'*')THEN
C              MATEC(I)='00000000'
              WRITE(I,*)'MATERIA DE ALTA # ',I,' => ',MATEC(I)

C              WRITE(30,10)Cuenta,ESC,(MATEC(I5),I5=1,10)
                  GOTO 15
          END IF

      END DO

15  CONTINUE
      END DO !FIN DEL CICLO ** I **
      B=1
C      GOTO 13
88  IF(B.EQ.1)THEN
          WRITE(35,10)Cuenta,ESC,(MATEC(I5),I5=1,10)
          END IF
          GOTO 5

C      END DO ! FIN DEL CICLO DE ** I **

C 13  CONTINUE

```

978 CONTINUE

END

#FILES 0.5

PROGRAMA EJEMPLO1

```

character ESCU*6
character ESCU2*6
CHARACTER NOMBRE*32
CHARACTER CTA*8
CHARACTER CTA2*8
CHARACTER NOMBRE2*32
CHARACTER BAN*3
CHARACTER ENTRADA*20
CHARACTER SALIDA*20
INTEGER*4 CTAN
INTEGER*4 HELEN
INTEGER*4 NE

```

CALL RCPAR(1,ENTRADA)

CALL RCPAR(2,SALIDA)

OPEN(UNIT=25,

```

- FILE='ENTRADA',
- ACCESS='DIRECT',
- STATUS='UNK',
- MAXREC=20000,
- RECL=56,
- FORM='FORMATTED')

```

ESCU='

NOMBRE='

CTA='00000000'

BAN='AAA'

DO I=1,20000

WRITE(25,REC=I)ESCU,NOMBRE,CTA,BAN

WRITE(1,*)'REGISTRO.....',I

END DO

END

```

1 *FILES 0,5
2   PROGRAMA EJEMPLO1
3
4   character ESCU*6
5   character ESCU2*6
6   CHARACTER NOMBRE*32
7   CHARACTER CTA*8
8   CHARACTER CTA2*8
9   CHARACTER NOMBRE2*32
10  CHARACTER BAN*3
11  CHARACTER ENTRADA*20
12  CHARACTER SALIDA*20
13  CHARACTER error*20
14  CHARACTER ALGO*125
15  CHARACTER DIRECTO*20
16  CHARACTER mov*12
17  CHARACTER movimiento(5)*12
18  CHARACTER natepp(10)*8
19  CHARACTER nate*12
20  INTEGER*4 CTAH
21  INTEGER*4 CTAH2
22  INTEGER*4 HELEM
23  INTEGER*4 NE
24  INTEGER*4 CLAYE,HAL,CTA,CARR,GRU
25  INTEGER*4 CLAYE2,HAL2,CTA2,CARR2,GRU2
26
27 cc CALL RCPAR(1,ENTRADA)
28 c  CALL RCPAR(2,SALIDA)
29 c  CALL RCPAR(1,DIRECTO)
30 c  CALL RCPAR(4,error)
31  OPEN(UNIT=25,
32  -    FILE=DIRECTO,
33  -    ACCESS='DIRECT',
34  -    STATUS='UNK',
35  -    MAXREC=20000,
36  -    RECL=100,
37  -    FORM='FORMATTED')
38
39 c  OPEN(10,FILE=ENTRADA,STATUS='UNK')
40 c  OPEN(20,FILE=SALIDA,STATUS='UNK')
41 c  OPEN(30,FILE=error,STATUS='UNK')
42
43
44  5      write(1,*)'dame el # de cuenta '
45        read(1,111)cta
46  111   format(i5)
47
48
49
50  HELEM=MOD(CTA,15737)
51  ctan2=1964
52  do while(iban.eq.0)
53
54      H2=HELEM-NE
55      if(ctan2.eq.0)then

```

```

56
57         WRITE(1,*)'REGISTRO NO ENCONTRADO'
58
59         IBAN=3
60     ELSE
61
62         READ(25,48,REC=HELEN)ESCUZ,CTAH2,(mategpo<j>,j=1,10),nmate
63         BAN
64     48   FORMAT(a6,i8,10a8,i2,a3)
65         WRITE(1,*)'REGISTRO EN DIR...',HELEN
66
67         WRITE(1,*)'VALOR DE CTA EN DIRECTO...',CTAH2
68         WRITE(1,*)'VALOR DE CTA EN SECUENC...',CTA
69         IF(CTA.EQ.CTAH2)THEN
70             BAN=' '
71
72             IBAN=1
73             WRITE(1,*)'REG. GRABADO 1'
74         write(1,48)ESCUZ,CTAH2,(mategpo<j>,j=1,10),nmate,
75         "    BAN
76             WRITE(1,*)'POS. ENCONTRADA ...',HELEN
77
78         ELSE
79             HELEN=HELEN+1
80             IF(HELEN.GT.32766)THEN
81                 WRITE(1,*)'TABLA DESBORDADA'
82                 WRITE(1,*)'REGISTRO NO GRABADO'
83                 IBAN=2
84             END IF
85         END IF
86     END IF
87     END DO
88     GOTO 5
89 $$$ CONTINUE
90
91     END
92

```

Module AEJEMPL01
FTN7X 5000/861229

No errors
No warnings

Prog: 1512
Save: none

Blank common: none
Local EMA: none

```

1 $FILES 0,5
2   PROGRAMA EJEMPLO1
3
4   character ESCU*6
5   character ESCU2*6
6   CHARACTER NOMBRE*32
7   CHARACTER CTAA*8
8   CHARACTER CTAA2*8
9   CHARACTER NOMBRE2*32
10  CHARACTER BAN*3
11  CHARACTER ENTRADA*20
12  CHARACTER SALIDA*20
13  CHARACTER error*20
14  CHARACTER ALGO*125
15  CHARACTER DIRECTO*20
16  CHARACTER mov*12
17  CHARACTER movimiento('S')*12
18  CHARACTER mategpo('10')*8
19  CHARACTER mate*12
20  INTEGER*4 CTAN
21  INTEGER*4 CTAN2
22  INTEGER*4 HELEM
23  INTEGER*4 HE
24  INTEGER*4 CLAVE,HAL,CTA,CARR,GRU
25  INTEGER*4 CLAVE2,HAL2,CTA2,CARR2,GRU2
26
27  CALL RCPARK(1,ENTRADA)
28  CALL RCPARK(2,SALIDA)
29  CALL RCPARK(3,DIRECTO)
30  CALL RCPARK(4,error)
31  OPEK UNIT=25,
32  -   FILE=DIRECTO,
33  -   ACCESS='DIRECT',
34  -   STATUS='UNK',
35  -   MAXREC=20000,
36  -   RECL=100,
37  -   FORM='FORMATTED')
38
39  OPEK(10,FILE=ENTRADA,STATUS='UNK')
40  OPEK(20,FILE=SALIDA,STATUS='UNK')
41  OPEK(30,FILE=error,STATUS='UNK')
42
43
44
45 5  read(10,30,end=999)CTA,(movimiento(i),i=1,5)
46 30  format(2x,i8,5x,5(x,a12))
47
48  HELEM=MOD(CTA,15737)
49  ctan2=1964
50  do while(iban.eq.0)
51
52  HE2=HELEM-HE
53  if(ctan2.eq.0)then
54
55  WRITE(1,*)'REGISTRO NO ENCONTRADO'

```

PROGRAMA 25-A

page 2 REJEMPL01

Opts: 77/LYI

Wed, Mar 15, 1989, 7:50 PM
..ACTUALIZA/BUSHOM_ALTA.FTH

```

56      WRITE(30,30)cta.(movimiento(i),i=1,5)
57
58      IBAN=3
59      ELSE
60
61      READ(25,48,REC=HELEM)ESCU2,CTAH2.(mategpo(j),j=1,10),nmate,
62      *
63      *
64      *
65      *
66      *
67      *
68      *
69      *
70      *
71      *
72      *
73      *
74      *
75      C*****
76      *   rutina para determinar tipo de movimiento deseado:
77      *           alta , baja   o   cambio
78      *
79      *   y realizarlo en el archivo directo de inscripcion
80      C*****
81
82
83      do i=1,5
84      mov=movimiento(i)
85      write(1,*)'aquí ! , mov ==> ',mov
86      if (mov(1:4).ne.'0000') then
87      write(1,*)'primer if'
88
89
90      *****
91      *****          C A M B I O S          *****
92      *****
93
94
95      if(mov(5:8).ne.'****'.and.mov(5:8).ne.'0000')then
96      if(mov(9:12).ne.'****'.and.mov(9:12).ne.'0000')then
97
98
99      j1=0
100     L=1
101     do while(1.le.10.and.j1.eq.0)
102     mate=mategpo(1)
103     if(mov(1:4).eq.mate(1:4)) then
104     mate(5:8)=mov(5:8) !depende de si aparece primero
105 *           el grupo baja y luego el de alta
106     mategpo(1)=mate
107     j1=999 !solamente para cambio de edo. en la bandera
108     and if
109     l=1+l
110     end do

```

PROGRAMA 25-B

Page 3 REJEMPL01

Opts: 77/LY1

Wed, Mar 15, 1989, 7:50 PM
..ACTUALIZA/BUSNOM_ALTA.FTH

```

111
112         if(j1.ne.999) then
113             write(1,*)'error en movimiento: materia inexistente !'
114         end if
115
116     end if
117
118 *****
119 *****           A L T A S           *****
120 *****
121
122
123
124         if(mov(9:12).eq.'****') then
125             if(nmate.eq.10) then
126                 write(1,*)'ERROR : no se permiten mas de 10 materias !'
127                 write(1,*)'movimiento anulado !!!!'
128             else
129
130 *****
131 ***** A L T A S:  RUTINA PARA NO REPETICION DE MATERIAS *****
132 *****           Y MOVIMIENTO DE ALTA EN SI           *****
133 *****
134
135         k3=1
136         l=0
137         do while(k3.le.nmate.and.l.eq.0)
138             mate=matgpo(k3)
139             if (mate(1:4).ne.mov(1:4))then
140
141                 mategpo(nmate+1)=mov(1:8) ! altas
142                 write(1,*)'alta en =', matgpo(nmate+1)
143                 write(1,*)'prueba de altas'
144                 l=777
145             else
146
147                 write(1,*)'ERROR EN ALTA : ya existe inscripcion'
148                 write(1,*)'en esta materia !'
149             end if
150             k3=k3+1
151
152         end do
153
154     end if
155 end if
156
157 write(1,*)'uff , llegue!'
158
159
160 *****
161 *****           B A J A S           *****
162 *****
163
164         if(mov(5:8).eq.'****')then
165             write(1,*)'error!'

```

```

166      if{mov(9:12).ne.'***'.and.mov(9:12).ne.'0000'} then
167          write(1,*)'error 2'
168
169
170          j1=0
171          l=1
172          do while(l.le.10.and.j1.eq.0)
173              mate=mategp(1)
174              if{mov(1:4).eq.mate(1:4)}then
175                  if{mov(9:12).eq.mate(5:8)} then
176                      mategp(1)='00000000'
177                      j1=888
178                  else
179                      write(1,*)'error en la baja, <NO EXISTE GRUPO!>'
180                      end if
181                  end if
182                  write(1,*)'error3'
183                  l=l+1
184              end do
185              write(1,*)'error 4'
186
187              if{j1.ne.888} then
188                  write(1,*)'No existe inscripcion a tal materia'
189              end if
190          end if
191      end if
192  end do !***fin del ciclo de movimientos
193
194
195  ****
196  **** grabacion de las modificaciones en el mismo registro del ****
197  **** archivo directo ****
198  ****
199      ban='mar'
200      write(25,48,rec=nelem)escu2,ctan2,(mategp(j),j=1,10),nmate,ban
201
202
203  ****
204  ****
205
206
207      IBAN=1
208      WRITE(1,*)'REG. GRABADO !'
209      WRITE(1,*)'PRIMERA POS ...',NE
210      WRITE(1,*)'POS. ENCONTRADA ...',NELEM
211
212  ELSE
213      NELEM=NELEM+1
214      IF{NELEM.GT.32766}THEN
215          WRITE(1,*)'TABLA DESBORDADA'
216          WRITE(1,*)'REGISTRO NO GRABADO'
217          IBAN=2
218      END IF
219  END IF
220  END IF

```

PROGRAMA 25-D

Page 5 AEJEMPL01

Opts: 77/LYI

Wed, Mar 15, 1989, 7:50 PM
./ACTUALIZA/BUSHOM_ALTA.FTH

```

221             END DO
222             GOTO 5
223 999 CONTINUE
224
225 C           DO L=1,5
226 C           WRITE(1,*)'RUTINA DE PRUEBA'
227 C           WRITE(1,*)'DAME EL # DE REGISTRO...'
228 C           READ(1,16)I
229 C 16        FORMAT(15)
230
231 C           READ(25,REC=I)ESCU,NOMBRE,CTAA,BAN
232 C           WRITE(1,*)'REGISTRO # ',I,' ',ESCU,NOMBRE,CTAA,BAN
233 C           END DO
234
235             END

```

Module AEJEMPL01
FTN7X 5000/851229

No errors
No warnings

Prog: 2454
Save: none

Blank common: none
Local EMA: none

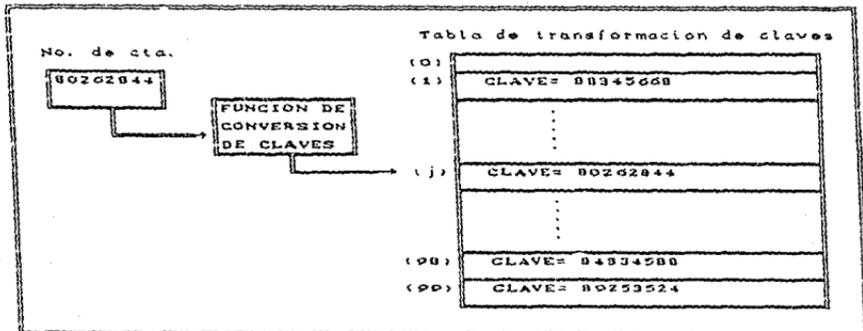
DIRECCIONAMIENTO DE ARCHIVOS

El método utilizado para lograr el direccionamiento de los archivos directos tanto para operaciones de entrada (lectura) como de salida (escritura), consiste en el intercambio de clave o mejor conocido con el término inglés "HASH". Este método tiene como fin, el aumentar la velocidad de búsqueda en el que los datos no necesitan estar ordenados.

El método de transformación de claves consiste en convertir la clave dada (numérica o alfanumérica) en una dirección (índice) dentro del arreglo. La correspondencia entre claves y la dirección en el medio de almacenamiento o en el arreglo se establece por una función de conversión (función o hash).

Así, por ejemplo, en el caso de una lista de alumnos (100), si cada uno de los 100 empleados tiene un número de identificación (clave) del 1 al 100, evidentemente puede existir una correspondencia directa entre la clave y la dirección definida en un vector o arreglo de 100 elementos.

Suponiendo que el campo clave de estos registros o elementos es un número que contenga 8 ó 9 dígitos (tal y como los contiene el número de cuenta de un alumno de la UNAM). Si se desea mantener en un arreglo todo el rango posible de valores, se necesitarán 10 elementos en la tabla de almacenamiento, cantidad difícil de tener disponibles en memoria central, aproximadamente 1,000,000,000 de registros o elementos. Si el vector o archivo solo tiene 100, 200 ó 1000 alumnos, como hacer para introducirlos en memoria por el campo clave (número de cuenta). Para hacer uso del número de cuenta como un índice de la tabla de búsquedas, se necesita un medio para convertir el campo clave en una dirección o índice más pequeño. En la figura se presenta un diagrama de como realizar la operación de conversión de una clave grande en una tabla pequeña.



Los registros o elementos del campo clave no tienen por que estar ordenados de acuerdo con los valores del campo clave, como estarían en una búsqueda binaria.

Por ejemplo, si registro del campo clave 80262844 estará almacenado en la tabla de transformación de claves (arreglo) en una posición determinada; por ejemplo, 83.

La función de transformación de clave, $H(x)$ convierte la clave (k) en una dirección (d) $H(k) \rightarrow d$ se suele expresar también como $H: X \rightarrow d$.

Si en lugar de números, las claves fueran nombres o frases de hasta 16 letras que identifican a un conjunto de un millar de personas. Existirán 26^{16} combinaciones posibles de claves que se deben transformar en 10^3 direcciones o índices posibles. La función H es, por consiguiente, evidentemente una función de paso o conversión de múltiples claves a direcciones. Dada una clave k , el primer paso en la operación de búsqueda es calcular su índice asociado $h=H(k)$ y el segundo paso -evidentemente necesario- es verificar si o no el elemento con la clave k es identificado verdaderamente por h en el arreglo T ; es decir, para verificar si la clave $T[H(k)]=X$, se deben considerar dos preguntas:

1. ¿Que clase de función H se utilizará?
2. ¿Como resolver la situación de que H no produzca la posición del elemento asociado

La respuesta a la segunda cuestión es que se debe utilizar algún método para producir una posición alternativa, es decir, el índice h' , y si esta no es aún la posición del elemento deseado, se

produce un tercer índice h'' , y así sucesivamente. El caso en el que una clave distinta de la deseada está en la posición identificada, se denomina colisión; la tarea de generación de índices alternativos se denomina tratamiento de colisiones.

Un ejemplo de colisiones puede ser:

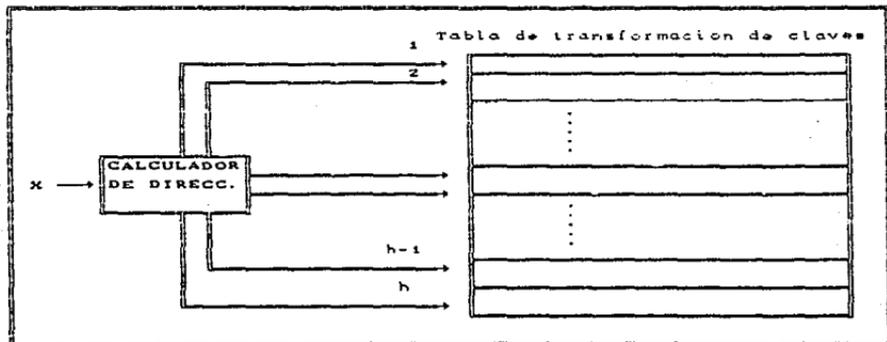
clave 84232444 función de conversión H → dirección 135
clave 82335811 función de conversión H → dirección 135

Dos claves distintas producen la misma dirección, es decir, colisiones. La elección de una buena función de conversión exige un tratamiento idóneo de colisiones, es decir, la reducción del número de colisiones.

MÉTODOS DE TRANSFORMACION DE CLAVES

Existen numerosos métodos de transformación de claves:

Todos ellos tienen en común la necesidad de convertir claves en direcciones. En esencia, la función de conversión equivale a una caja negra que podríamos llamar *calculador de direcciones*. Cuando se desea localizar un elemento de clave x , el indicador de direcciones indicará en que posición del arreglo estará situado el elemento.



TRUNCAMIENTO

Ignora parte de la clave y se utiliza la parte restante directamente como índice (considerando campos no numéricos y sus códigos numéricos). Si las claves, por ejemplo, son enteros de 8

dígitos y la tabla de transformación tiene 1000 posiciones, entonces el primero, segundo y quinto dígitos desde la derecha pueden formar la función de conversión. Por ejemplo, 72588495 se convierte en 895. El truncamiento es un método muy rápido, pero falla para distribuir las claves de modo uniforme.

PLEGAMIENTO

La técnica del plegamiento consiste en la partición de la clave en diferentes partes y la combinación de las partes en un modo conveniente (a menudo utilizando suma o multiplicación) para obtener el índice.

La clave x se divide en varias partes X_1, X_2, \dots, X_n , donde cada parte, con la única posible excepción de la última parte, tiene el mismo número de dígitos que la dirección especificada.

A continuación, se suman todas las partes:

$$h(x) = X_1 + X_2 + \dots + X_n$$

En esta operación se desprecian los dígitos más significativos que se arrastran de arrastre o acarreo.

EJEMPLO:

Un entero de 9 dígitos se puede dividir en grupos de tres, tres y dos dígitos, los grupos se suman juntos y se truncan si es necesario para que estén en el rango adecuado de índices.

Por consiguiente, si la clave es

62538194

y el número de direcciones es 100, la función de conversión será:

$$625 + 381 + 94 = 1100$$

que se truncará a 100 y que será la dirección deseada.

ARITMETICA MODULAR

Convertir la clave a un entero, dividir por el tamaño del rango del índice y tomar el resto como resultado. La función de conversión utilizada es mod (módulo o resto de la división entera)

$$h(x) = x \text{ mod } m$$

donde m es el tamaño del arreglo con índices de 0 a $m-1$. Los valores de la función -direcciones- (el resto) irán de 0 a $m-1$, ligeramente menor que el tamaño del arreglo.

La mejor elección de los módulos son los números primos. Por ejemplo, en un arreglo de 1.000 elementos se puede elegir 997 o 1009. Otros ejemplos son:

$$18 \text{ mod } 6 \quad 19 \text{ mod } 6 \quad 20 \text{ mod } 6$$

que proporcionan unos restos de 0, 1 y 2 respectivamente.

Si se desea que las direcciones vayan de 0 hasta m, la función de conversión debe ser:

$$h(x) = x \text{ mod } (m+1)$$

MITAD DEL CUADRADO

Este método consiste en calcular el cuadrado de la clave x. La función de conversión se define como

$$h(x) = c$$

donde c se obtiene eliminando dígitos a ambos extremos de x^2 para todas las claves.

COLISIONES

La función de conversión $h(x)$ no siempre proporciona valores distintos, puede suceder que para dos claves diferentes X_1 y X_2 se obtenga la misma dirección. Esta situación se denomina colisión y se deben encontrar métodos para su correcta resolución. Los ejemplos vistos, de las claves con 100 posibles direcciones. Si se considera el método del módulo en el caso de las claves y se considera el número primo 101.

123445678

123445880

proporcionarían las direcciones:

$$h(123445678) = 123445678 \text{ mod } 101 = 44$$

$$h(123445880) = 123445880 \text{ mod } 101 = 44$$

Es decir, se tienen dos elementos en la misma posición del vector o arreglo, [44]. En terminología de claves, se dice que las claves 123445678 y 123445880 han colisionado.

El único medio para evitar el problema de las colisiones totalmente es tener una posición del arreglo para cada posible número de cuenta.

En este trabajo, el tratamiento de las colisiones está explicado convenientemente en el capítulo correspondiente (capítulo II).

BIBLIOGRAFIA

FERRER ABELLO ANTONIO M.
DICCIONARIO DE TERMINOS INFORMATICOS (FASCICULO 2)
INGELEK

RAMIREZ CORTES JOSE CARLOS
MANUAL DEL CHIWRITER
ENEP-ARAGON. UNAM.

SOMMERVILLE
INGENIERIA DE SOFTWARE
SITESA

JOYANES AGUILAR LUIS
BASIC AVANZADO
(DISEÑO Y TRATAMIENTO DE ARCHIVOS)
MC GRAW-HILL

MICROSOFT
MICROSOFT MS-DOS
USER'S GUIDE
MICROSOFT

HEWLETTE PACKARD
HEWLETTE PACKARD
JOURNAL MAGAZINE
APRIL, MAY, JUNE, AUGUST.
1989

HEWLETTE PACKARD
FORTRAN 77 (MANUAL DE USUAPIO)
HEWLETTE PACKARD

HEWLETTE PACKARD
PASCAL (MANUAL)
HEWLETTE PACKARD

BIBLIOGRAFIA

HEWLETTE PACKARD
IMAGE 1000 (MANUAL)
HEWLETTE PACKARD

SENN
ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION
MC GRAW-HILL

JOYANES AGUILAR LUIS
FUNDAMENTOS DE PROGRAMACION
(ALGORITMOS Y ESTRUCTURA DE DATOS)
MC GRAW-HILL

DALE-LILLY
PASCAL (ESTRUCTURA DE DATOS)
MC GRAW-HILL

AHO-HOPCPOFT-ULLMAN
ESTRUCTURA DE DATOS Y ALGORITMOS
SITESA