



*UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO*

**ESCUELA NACIONAL DE
ESTUDIOS PROFESIONALES ACATLÁN**

**“SISTEMA AUTOMATIZADO DE CONTROL
PARA LA TITULACIÓN”**

MEMORIA DE DESEMPEÑO PROFESIONAL

Que para obtener el título de:

**LICENCIADO EN MATEMÁTICAS APLICADAS Y
COMPUTACIÓN**

presenta:

ARMANDO AMADOR MOLINA

Asesor : Act. Laura Ma. Rivera Becerra

Sta. Cruz Acatlán, México.

Febrero 1997

**TESIS CON
FALLA DE ORIGEN**

2
24.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLAN"
DIVISION DE MATEMATICAS E INGENIERIA
PROGRAMA DE MATEMATICAS APLICADAS Y
COMPUTACION

ARMANDO AMADOR MOLINA

Alumno de la carrera de Matemáticas Aplicadas y Computación

Presente.

De acuerdo a su solicitud presentada con fecha 5 de noviembre de 1996 me complace notificarle que esta Jefatura tuvo a bien asignarle el siguiente tema de el cual se desarrollará como sigue:

INTRODUCCION

I.- MARCO CONTEXTUAL E IDENTIFICACION DEL PROBLEMA.

II.- SISTEMA DE INFORMACION AUTOMATIZADO COMO PROPUESTA DE SOLUCION.

III.- DESARROLLO DEL SISTEMA, IMPLANTACION Y MANTENIMIENTO.

CONCLUSIONES
BIBLIOGRAFIA

Asimismo, fue asignado como Asesor de la Memoria de Desempeño Profesional; Act. Laura María Rivera Becerra.

Ruego a usted tomar nota que en cumplimiento de los especificado en la Ley de Profesiones, deberá presentar servicio social durante un tiempo mínimo de seis meses como requisito básico para sustentar Examen Profesional, así como de la disposición de la Coordinación de la Administración Escolar en el sentido de que se imprima en lugar visible de los ejemplares el titulo del trabajo realizado. Esta comunicación deberá imprimirse en el interior de la misma.

E.N.E.P. ACATLAN

ATENTAMENTE

"POR MI RAZA HABLARA EL ESPÍRITU"

Acatlán, Edo. de Méx. a 10 de febrero de 1997.



DEPARTAMENTO DEL PROGRAMA DE
MATEMÁTICAS E INGENIERÍAS
APLICADAS Y COMPUTACION

MAT. MAYRA ELIZONDO CORTES
Encargada del Programa de M.A.C.

cg'



DEDICATORIAS

Este trabajo y su autor están dedicados a:
Mayra Elizondo Cortés

A la mujer que simboliza la fé, la esperanza y el
amor en su máximo esplendor: MI MADRE,

A mis hermanos Octavio, Juan Carlos y Alfredo por
compartir conmigo los más intensos momentos de
mi infancia, adolescencia y juventud.
A ellos por ser sangre de mi sangre.

A Ronald, Elvira y Rubi por ya
formar parte de mi vida.

A una gran mujer, persona, amiga, jefa y asesora,
con mucho cariño. Gracias por todo Laura.

Al Dios del esfuerzo y la Diosa de la esperanza,
porque el no bajar los brazos y levantar la
cara tienen su recompensa.

A mi Escuela y a mi Universidad por
todo lo que me han dado.

A todos mis profesores y mis alumnos, por
ser un motivo más para superarme.

De todo corazón a todos aquellos que creyeron en
mi y también a los que no lo hicieron.

A.A.M.

I N D I C E

Introducción	
I. Marco contextual e identificación del problema	1
1.1 La Jefatura de Actuaria y M.A.C. dentro de la E.N.E.P. Acatlán.....	1
1.2 La Secretaría Técnica de M.A.C.....	8
1.2.1 Servicios de apoyo para la titulación de los egresados.....	9
1.2.2 Trámites necesarios para el registro de las diferentes opciones de titulación.....	11
1.3 Problemáticas identificadas y sus orígenes.....	13
1.4 Definición del problema	14
II. Sistema de información automatizado como propuesta de solución	19
2.1 Conceptos de Sistemas de Información.....	19
2.1.1 Teoría de sistemas	19
2.1.2 Clasificación de Sistemas de Información	22
2.2 Metodologías para el desarrollo de sistemas de Información	23
2.2.1 Método del ciclo de vida para el desarrollo de sistemas	23
2.2.2 Método del desarrollo de análisis estructurado	26
2.2.3 Método del prototipo de sistemas	30
2.3 Diagnóstico y propuesta de solución	30
2.3.1 Elección de un lenguaje de programación.....	32
III. Desarrollo del sistema, implantación y mantenimiento	39
3.1 Desarrollo del sistema utilizando el método de prototipos	39
3.1.1 Desarrollo del prototipo de salidas	40
3.1.2 Desarrollo del prototipo de bases de datos	44
3.2 Implantación y Mantenimiento	47
3.3 Descripción general del sistema	48
Conclusiones	77
Bibliografía	81
Apéndice	



INTRODUCCION



INTRODUCCION

Se eligió la opción de titulación: Memoria del Desempeño Profesional de entre otras cinco opciones dado que el autor de este trabajo actualmente labora como Secretario Técnico de la Licenciatura de Matemáticas Aplicadas y Computación (M.A.C.) en la E.N.E.P. Acatlán de la U.N.A.M. Precisamente en el ejercicio profesional es donde ha tenido contacto con una serie de situaciones y problemáticas reales, y ha podido poner en práctica lo aprendido en las aulas, ha obtenido experiencias y aprendizajes significativos que lo han hecho desarrollar al máximo sus capacidades y actitudes críticas y propositivas. Al realizar un trabajo escrito que resume el resultado de dicha experiencia en el proyecto de solución de un caso específico se tiene una memoria del desempeño profesional.

El presente trabajo pretende, ser un informe de todas las actividades realizadas en el desarrollo del sistema de información automatizado para el control de los trámites de titulación de los egresados de la Licenciatura en M.A.C. de la E.N.E.P. Acatlán, que tendrá como finalidad el apoyo a la toma de decisiones para la adecuada administración y planeación de los recursos en los servicios que presta la Jefatura del Programa de Actuaría y M.A.C. para la titulación. Además de describir las actividades realizadas aportará experiencias y criterios de solución a los diversos problemas que se presentaron .

Esta memoria vincula conocimientos matemáticos como la creación de modelos de análisis y diseño de algoritmos y procesos, así como de computación, que permite la utilización de lenguajes de programación para la codificación, ejecución e implementación por medio de una computadora de dichos procesos. Son precisamente estos dos tipos de conocimientos los que el egresado de la Licenciatura de Matemáticas Aplicadas y Computación puede manejar, combinar y aprovechar de manera óptima para la solución de problemas de la vida real.

En la actualidad los sistemas de información automatizados han llegado ha ser una herramienta imprescindible en casi todo ámbito de las Instituciones de servicio, administración y planeación , ya que con el crecimiento incesante de la población, el enorme número de datos por manejar hace que los sistemas de información manuales sean insuficientes y hasta obsoletos para la eficiente utilización de los recursos en bien de dicha población.

Una de las actividades como Secretario Técnico de M.A.C. es la coordinación de los trámites de registros de opciones de titulación . Esta actividad involucra el manejo de una gran cantidad de información cuyo procesamiento manual, que era el que se realizaba hasta la fecha, era tardado, tedioso y no totalmente confiable. De tal forma el desarrollo de un sistema de información automatizado

constituirá una solución viable y efectiva para el eficiente manejo de la información y de las actividades relacionadas.

Este trabajo mostrará la forma en que se aplica la metodología del desarrollo de sistemas de información en una situación real, pudiendo ser también un apoyo para los alumnos de las materias de Introducción a las Matemáticas Aplicadas y Computación, Sistemas de Información, Diseño e Implementación de Sistemas y Análisis y Diseño de Sistemas, de la carrera de M.A.C. que deseen aprender algo acerca de una aplicación real de los conocimientos teóricos que adquieren por medio de sus maestros y en los libros. Con esta finalidad se trata de redactar de manera clara, explicativa y haciendo uso de diagramas y gráficos para una mejor comprensión. Es un deseo del autor que este trabajo resulte útil a los alumnos de M.A.C. aportando experiencias y criterios en la toma de decisiones y también, de alguna manera interesante y ameno.

La memoria aquí presentada consta de los siguientes capítulos:

CAPITULO I

MARCO CONTEXTUAL E IDENTIFICACION DEL PROBLEMA

En este capítulo se describe el medio dentro del cual se presentó la problemática. Se ubicará a la Secretaría Técnica de M.A.C. dentro de la Jefatura de Programa de Actuaría y M.A.C. y ésta a su vez dentro de la E.N.E.P. Acatlán. Se definirán funciones y se mostrará el proceso por medio del cual el Secretario Técnico de M.A.C., que es quien escribe, lleva a cabo el registro de los trabajos de titulación. Sólo hasta tener totalmente claro cuál es el problema, se puede proponer alguna alternativa de solución. Así que en este capítulo también se identificará plenamente el problema así como sus orígenes y las dificultades que representa la utilización del sistema de información actual.

CAPITULO II

SISTEMA DE INFORMACION AUTOMATIZADO COMO PROPUESTA DE SOLUCION

En este capítulo se explicará por qué se propuso un sistema de información basado en la computadora como una solución viable. Se hablará, desde el punto de vista del desarrollo de sistemas de información de los conceptos necesarios para comprender sus posibilidades y limitaciones, se describirán las metodologías de desarrollo de sistemas que existen así como la que se utilizó en el desarrollo de la presente memoria (desarrollo de prototipos). Se hará también una descripción detallada de las funciones del sistema así como de algunas consideraciones para su mejor manejo.

CAPITULO III

DESARROLLO DEL SISTEMA, IMPLANTACION Y MANTENIMIENTO

En el último capítulo de este trabajo, se describirá cómo se realizó el sistema utilizando la metodología de desarrollo de prototipos.

En este capítulo también se podrán conocer las actividades por realizar para lograr una implantación óptima del sistema y cómo se realizará el mantenimiento del mismo haciendo uso de diferentes estrategias que identifican puntos tanto fuertes como débiles del sistema. Los dos tópicos analizados asegurarán un funcionamiento adecuado del sistema bajo circunstancias extremas, minimizando la posibilidad de obsolescencia y ampliando su vida útil.

Finalmente se podrán conocer las **CONCLUSIONES** a las que se llegó a lo largo de toda la memoria al trabajar en el desarrollo del sistema y los resultados que hasta ahora se han obtenido con la utilización del mismo, ya que cabe mencionar que dicho sistema opera actualmente en su totalidad, se harán también algunas reflexiones personales acerca de la elaboración del presente trabajo .



CAPITULO I

MARCO CONTEXTUAL E IDENTIFICACION DEL PROBLEMA



*"No hay enigmas, si un problema puede
plantearse, cambia puede resolverse"*
Ludwing Wittgenstein

I. MARCO CONTEXTUAL E IDENTIFICACION DEL PROBLEMA

En este capítulo se podrá conocer el marco en el cual surge la problemática, en qué consiste y por qué es necesario el desarrollo de un medio que la resuelva.

1.1 LA JEFATURA DE ACTUARIA Y M.A.C. DENTRO DE LA E.N.E.P. ACATLAN

La Escuela Nacional de Estudios Profesionales Acatlán es una Institución de educación superior descentralizada de la Universidad Nacional Autónoma de México (UNAM), una unidad multidisciplinaria que responde a las demandas educativas y al desarrollo pleno de nuestro país.

Desde su fundación, la Universidad Nacional Autónoma de México ha respondido a las exigencias educativas de nuestro país, formando profesionales, investigadores, profesores y técnicos útiles a la sociedad. Su principal propósito es estar íntegramente al servicio del país y de la humanidad con un sentido ético y de servicio social, superando constantemente cualquier interés individual.

Con el objetivo de cubrir áreas de la metrópoli densamente pobladas, se creó un programa de descentralización en la UNAM hace poco más de 20 años.

Así surgieron nuevos núcleos universitarios ubicados en la zona conurbada. Con ello surgieron las Escuelas Nacionales de Estudios Profesionales.

La historia de la ENEP Acatlán se remonta a 1974, cuando el H. Consejo Universitario aprobó la creación del Campus Acatlán. Al siguiente año, se iniciaron sus funciones con carácter de escuela independiente tanto académica como administrativamente de las demás escuelas y facultades de la Ciudad Universitaria de Coyoacán.

El 17 de marzo de 1975 la Escuela Nacional de Estudios Profesionales Acatlán fue inaugurada por el entonces rector de la UNAM, Dr. Guillermo Soberón Acevedo.

Cuando se iniciaron las funciones de la ENEP Acatlán se ofrecían 13 licenciaturas, con las que se atendían poco más de 4,000 alumnos. Las instalaciones eran reducidas, pues sólo se contaban con tres edificios y una unidad académica, donde se ubicaba la biblioteca.

Hoy la escuela atiende a una población estudiantil de casi 15,000 alumnos en 16 licenciaturas y 150 alumnos de posgrado. Cuenta con 32 edificios destinados a apoyar tanto a alumnos como a profesores en las labores de ejercicio de cátedra, apoyo académico, posgrado e investigación, así como los diferentes servicios para a la formación profesional de los alumnos.

También se tienen amplias áreas para alentar las actividades deportivas y recreativas.

Aunque la escuela es joven, durante los 20 años de su historia ha ido labrando un reconocido prestigio, el cual se manifiesta en el alto nivel académico de sus egresados, derivado de la preparación y experiencia de sus profesores.

El Campus Acatlán de la UNAM es actualmente uno de los núcleos culturales más importantes de la zona noroeste del área conurbada de la capital del país.

La ENEP Acatlán ofrece un panorama profesional versátil de carácter multidisciplinario el cual responde a demandas educativas y a las necesidades del país. Además promueve la titulación con el desarrollo e instrumentación de seis opciones de titulación además de la Tesis.

Como se mencionó anteriormente, se tienen 16 licenciaturas, lo cual hace que este campus sea, después de Ciudad Universitaria, el que ofrece más carreras.

Estas disciplinas se reparten en cinco áreas de conocimiento que son:

- Matemáticas e Ingeniería
- Diseño y Edificación
- Humanidades
- Ciencias Socioeconómicas
- Ciencias Jurídicas

A continuación, en la figura 1.1, se podrá ver el organigrama de la Secretaría General que da una idea clara de la organización a nivel académico-administrativo de la ENEP Acatlán.

ORGANIGRAMA DE LA SECRETARÍA GENERAL

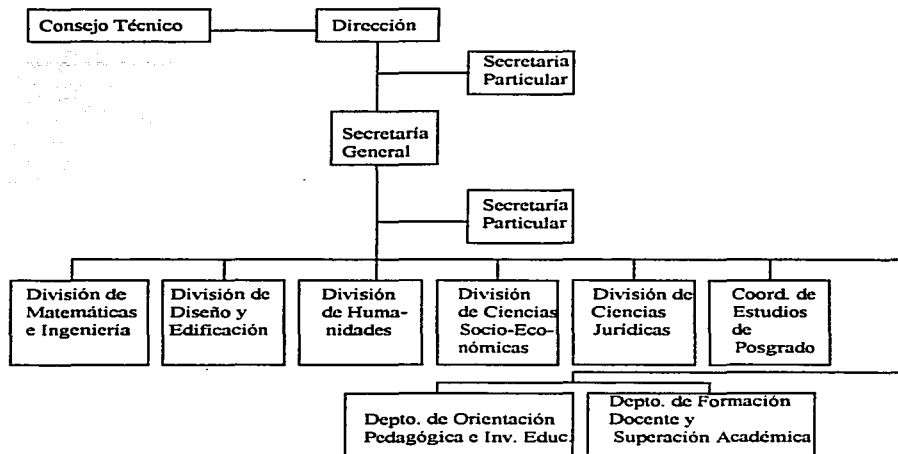


Figura 1.1

La forma en que se agrupan las licenciaturas en cada División es:

- División de Matemáticas e Ingeniería
 - Ingeniería Civil
 - Actuarial
 - Matemáticas Aplicadas y Computación
- División de Diseño y Edificación
 - Arquitectura
 - Diseño Gráfico
- División de Humanidades
 - Pedagogía
 - Periodismo y Comunicación Colectiva
 - Lengua y Literatura Hispánicas
 - Filosofía

- Historia
- Enseñanza del Inglés
- División de Ciencias Socioeconómicas
 - Sociología
 - Relaciones Internacionales
 - Ciencias Políticas y Administración Pública
 - Economía
- División de Ciencias Jurídicas
 - Derecho

En forma general, el objetivo de la Jefatura de División es definir las políticas académicas para el estudio, la docencia y la promoción de las disciplinas que integran el (los) Programa (s) de la División, de acuerdo a los lineamientos de la Secretaría General, así como coordinar y supervisar su ejecución.

A su vez, para las carreras existen Jefaturas de Programa. A continuación se presenta la figura 1.2 en la cual podrá observarse el organigrama de la División de Matemáticas e Ingeniería, que es a la que pertenece la Jefatura de Programa de Actuaría y Matemáticas Aplicadas y Computación; también podrá observarse el lugar que ocupa la Secretaría técnica para ambas carreras.

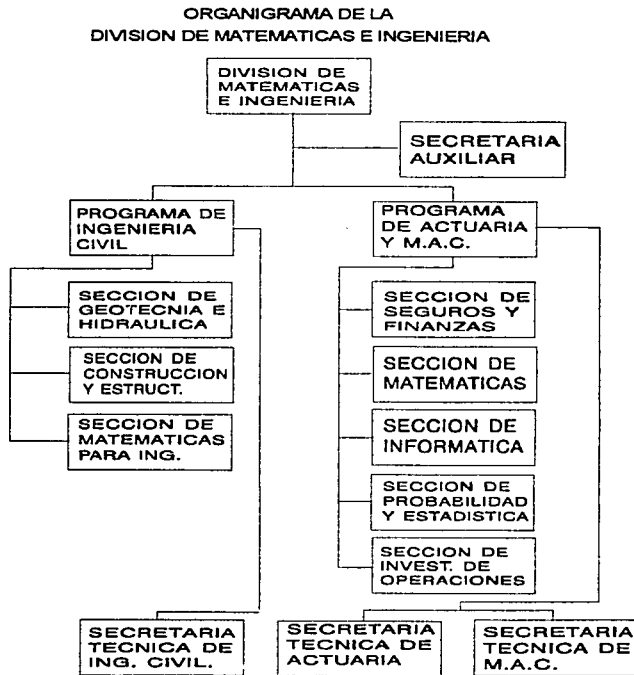


Figura 1.2

El objetivo de la Jefatura de programa es:

Definir los proyectos académicos específicos del Programa de acuerdo a las políticas establecidas por la Jefatura de la División, coordinar y supervisar su ejecución.

Enseguida se podrán observar las funciones de la Jefatura de Programa y en negritas se resaltarán específicamente aquellas relacionadas con la titulación de los egresados:

- Elaborar y presentar a la Jefatura de la División, el proyecto de actividades de los programas de docencia y estudios profesionales, y presentar la evaluación semestral correspondiente.
- Elaborar y presentar a la Jefatura de la División, el programa de trabajo semestral sobre áreas integradas por asignaturas críticas de las licenciaturas del Programa.
- Elaborar y presentar a la Jefatura de la División, el proyecto semestral de eventos académicos y actividades extracurriculares para el apoyo a las licenciaturas que integran el Programa.
- Elaborar y presentar a la Jefatura de la División, el proyecto de superación académica y formación docente del Programa.
- Analizar y presentar a la Jefatura de la División, la propuesta de planta docente del Programa.
- Proponer al Comité de Programa los horarios de clase y cupos para cada periodo lectivo.
- Establecer los horarios de los exámenes ordinarios y extraordinarios.
- Colaborar con el Jefe de la División en la organización y realización de concursos académicos de oposición.
- Analizar y presentar a la Jefatura de la División, las propuestas de requerimientos bibliográficos y material didáctico para el desarrollo del programa.
- Presentar a la Jefatura de la División los requerimientos de apoyo y de adquisición de equipo de las (s) carreras a su cargo , para los Talleres y Laboratorios.

- Remitir a la Unidad de Talleres y Laboratorios el programa académico semestral para los Talleres y Laboratorios de la (s) carrera (s) a su cargo, autorizado por la Jefatura de la División.
- Participar en la revisión e instrumentación de los planes y programas de estudio y proponer a la Jefatura de la División las modificaciones que a su juicio se requieran para su presentación al Comité de Programa.
- **Autorizar las propuestas de tesis individuales y de profesores asesores, así como avalar la terminación del trabajo.**
- Coordinar la asignación de sinodales para exámenes extraordinarios y especiales para las comisiones de revisión correspondientes.
- Presentar al H. Consejo Técnico las solicitudes de inscripción condicionada y evaluaciones especiales, para su eventual aprobación.
- **Presentar al Comité de Programa y al H. Consejo Técnico las solicitudes de registro de tesis colectivas para su eventual aprobación.**
- Promover entre los alumnos del Programa el intercambio académico con instituciones educativas que imparta (n) la (s) carrera (s) del programa u otras afines y con centros de investigación nacionales y extranjeros.
- Elaborar el dictamen académico de los procesos de convalidación y revalidación de estudios solicitados por la Unidad de Administración escolar de la Escuela.
- Designar a los profesores que participarán en las charlas de introducción a la Universidad y en las pláticas de orientación vocacional que se imparten en los planteles de educación media superior.
- Elaborar y proponer a la Jefatura de la División el programa semestral de prácticas de carrera de materia y visitas de observación, establecidas en los planes y programas de estudios de la licenciatura que integra el Programa, así como coordinar su realización.
- Formar parte del Comité de Programa.
- Presentar a la Jefatura de la División el Anteproyecto de Presupuesto anual del Programa.

- Vigilar que las actividades del programa se desarrollen conforme a lo establecido en la Legislación Universitaria.
- Llevar a cabo las actividades y comisiones que se le deleguen por acuerdo superior.

1.2 LA SECRETARÍA TÉCNICA DE M.A.C.

El objetivo de la Secretaría Técnica es:

Brindar el apoyo académico-administrativo para el desarrollo de las actividades del órgano correspondiente.

A continuación se puntualizarán sus funciones y nuevamente podrán verse denotadas con negritas aquellas que podrían referirse al objetivo del presente trabajo y a los trámites de titulación de los alumnos:

- Elaborar y tramitar las solicitudes de recursos del personal adscrito al órgano.
- Auxiliar al titular del órgano en la coordinación y supervisión de las actividades académicas y/o administrativas.
- Colaborar con las actividades de evaluación de los programas académicos y/o administrativos del órgano.
- **Sistematizar y mantener actualizada la información y documentación recibida en su área.**
- **Atender y canalizar los asuntos académicos y/o administrativos que le presente el personal del órgano correspondiente.**
- Participar en la preparación del Programa de Trabajo de cada período académico.
- Participar en la elaboración del Informe de Actividades del órgano.

- * Colaborar en la elaboración del Anteproyecto de Presupuesto.
- * **Proporcionar a los demás órganos de la Escuela la información de carácter administrativo y/o académico que éstos requieran sobre su área.**
- * Colaborar en la actualización de los instrumentos técnicos de coordinación y organización (manuales, auditorías administrativas, análisis de puestos, etc.).
- * Colaborar con el titular del órgano en el control del buen uso de los bienes materiales a cargo del área correspondiente y de su Banco de Horas.
- * Supervisar y coordinar el trabajo de oficina del órgano correspondiente.
- * **Programar y supervisar la prestación de servicios de apoyo en favor del personal académico y/o estudiantil que participa en al ámbito.**
- * Coordinar las acciones de apoyo logístico para la realización de los eventos organizados o en que participa el órgano.
- * **Desarrollar estudios, proyectos y sistemas de información para el manejo y control de los asuntos a cargo del órgano.**
- * Realizar las actividades que se le deleguen por acuerdo superior.

1.2.1 SERVICIOS DE APOYO PARA LA TITULACION DE LOS EGRESADOS

El proceso administrativo para la titulación de los egresados en la E.N.E.P. Acatlán lo desarrollan conjuntamente el Departamento de Certificación y Títulos, la Jefatura de División y la Jefatura de Programa (a través de su secretaría técnica). Las tres instancias mencionadas trabajan coordinadamente para llevar a buen fin los trámites.

La Secretaría Técnica perteneciente a la Jefatura de Programa es quien directamente realiza los servicios de apoyo en los trámites relacionados con la titulación de los egresados.

El proceso para la titulación consta de varias fases. Estas se realizan en diferentes instancias de la Universidad y también en otras de la misma E.N.E.P. Acatlán. Más adelante se describirán con detalle los procedimientos a seguir por un egresado que cumpla con los requisitos curriculares para poder obtener su título profesional.

La Jefatura de programa coordina especialmente tres de los pasos del proceso mencionado, los cuales son pre-registro del trabajo, registro y notificación de trabajo concluido. Estos pasos se detallan más adelante. Los secretarios técnicos son quienes asesoran y guían a los alumnos explicándoles cómo, cuándo y dónde se realizan.

Los alumnos acuden al secretario técnico de su carrera para obtener información acerca de las formas de titulación existentes y los trámites necesarios. Cabe señalar que desde los primeros semestres de las carreras, la Jefatura de programa pone un gran interés en que los alumnos conozcan las opciones de titulación con que cuenta la escuela, los requisitos especiales para cada una de ellas y las ventajas o desventajas con que cuentan para alguna situación específica de los alumnos.

Con tal cúmulo de información, ellos pueden dirigir su atención al desarrollo e investigación de algún tema que pueda interesarles a lo largo de la carrera en sus diferentes asignaturas; pueden pensar a futuro sobre la modalidad de titulación que más les convenga y hasta tener una idea del asesor que pudiera ser el más adecuado o que como en algunas ocasiones, se identifique con ellos para el desarrollo de su trabajo de titulación.

Los secretarios técnicos dan información verbal personalizada a cada uno de los interesados, obsequian folletos y guías informativas, también acuden a los grupos a dar información a este respecto; organizan pláticas periódicas para dar información a nivel carreras, coordinan desplegados de información masivos tanto en publicaciones a nivel interescolar en el boletín Última Hora, a nivel Universidad en la Gaceta Universitaria y hasta en periódicos de circulación en todo el país como La Jornada, El Excelsior y El Universal..

Es importante mencionar que la información dada a los alumnos no sólo queda en el ámbito administrativo, muchos de ellos llegan con su secretario técnico preguntando por temas de trabajos ya registrados y/o terminados. Esto ocurre ya que los alumnos desean saber si el tema que les interesa ya ha sido abordado por alguien más para así decidir entre continuar con el desarrollo del mismo o pensar en algún otro tema. Para este fin, se cuenta con listados de todos los temas de trabajos tanto registrados como ya terminados. Dichos listados se encuentran a disposición de los alumnos en la Jefatura de

Programa. Los encargados de su préstamo y continua actualización son los secretarios técnicos de cada carrera.

De igual manera, solicitan información de los posibles asesores para los diversos temas de su interés. Por ésto, el secretario técnico debe tener algún medio por el cual se lleven registros por temas para cada carrera de los profesores de la escuela que dominen esa área de conocimiento y que estuvieran dispuestos a asesorar trabajos de titulación. Hasta el momento se cuenta con listados actualizados que también se prestan a los alumnos.

Como puede verse para dar un buen servicio, la secretaría técnica debe mantener una gran cantidad de información que debe ser veraz y estar lo más actualizada posible.

1.2.2 TRAMITES NECESARIOS PARA EL REGISTRO DE LAS DIFERENTES OPCIONES DE TITULACION

El procedimiento a seguir de trámites de titulación en la Jefatura de Programa se realiza de la siguiente manera:

La primera parte comienza cuando los interesados en titulación acuden a la Jefatura, específicamente con el secretario técnico de su carrera y solicitan informes de titulación. Se le facilitan, entonces, folletos con toda la información necesaria. Una vez que el interesado ha seleccionado alguna de las siguientes opciones:

- ⇒ Tesis.
- ⇒ Tesina.
- ⇒ Memoria del desempeño profesional ó
- ⇒ Informe de práctica profesional de servicio a la comunidad
- ⇒ Seminario taller extracurricular

se lleva a cabo el trámite de titulación que consta de los siguientes pasos:

1. Se proporciona al interesado hoja con requisitos para realizar el pre-registro.

2. El interesado regresa con el pre-registro debidamente elaborado. Se verifica que no falte ningún dato, firma o información necesaria. En caso de que algo falte y/o existan datos mal proporcionados, se pide al alumno completar y/o corregir lo necesario y que entregue de nuevo el pre-registro correcto.
3. Una vez entregado el pre-registro sin ningún error u omisión se procede a la extensa tarea de revisar los registros existentes contra el pre-registro entregado, para lo cual se deben leer casi todos los títulos (alrededor de un 80%) para saber si alguno o algunos son semejantes al pre-registro. Es recomendable buscar registros en ambas carreras (Actuaría y Matemáticas Aplicadas y Computación) pero como en ese caso se eleva el número de comparaciones posibles en general sólo se contrasta contra registros de la misma carrera a la que pertenece el pre-registro elaborado. En muchas ocasiones la búsqueda resulta tan extensa que puede apoyarse en el criterio, la memoria y hasta la intuición para decidir su aprobación.
4. En caso de encontrar registros semejantes o muy parecidos, se solicitan correcciones en base a las analogías encontradas con registros existentes, ésto es, se puede variar desde corregir o eliminar uno o varios capítulos, hasta el rechazo del pre-registro con la sugerencia al alumno de presentar otro tema de trabajo.
5. Después de aprobado el pre-registro se facilitan al alumno los formatos de registro para trabajo de titulación. Dichos formatos deben ser llenados a máquina y contienen datos del alumno, datos del asesor y del trabajo en sí.
6. El alumno regresa a la Jefatura con sus formatos de registro llenos, se verifica que sean correctos y estén completos, si existe algún error se pide sea corregido. Se solicita la firma del Jefe de Programa autorizando el registro, se sella y se pide al alumno que entregue los formatos de la siguiente manera:
 - ◊ el formato blanco en la Jefatura para abrir el expediente correspondiente.
 - ◊ el azul a la Unidad de Administración Escolar.
 - ◊ el amarillo a su asesor, y
 - ◊ el rosa es su comprobante de registro.
7. Se abre el expediente del alumno, que contiene la siguiente información: formato blanco de registro y su pre-registro. Dicho expediente se archiva en orden alfabético por apellido paterno para su posterior consulta y trámites consecuentes.

8. El alumno acude nuevamente a la Jefatura enunciando la terminación de su trabajo. Cabe destacar que aquí se debe revisar la fecha de registro y la fecha de terminación ya que existe límite en el tiempo de culminación del trabajo para algunas opciones: el registro de la tesina tiene seis meses de caducidad y la memoria del desempeño profesional un año. En caso del que el tiempo sea excedido la política de acción de la Jefatura es hacer firmar una "carta compromiso" el egresado, la cual estipula que de no terminarse el trabajo en el tiempo establecido en dicha carta (máximo tres meses), se procederá a la baja del registro. Si el trabajo ha sido concluido en el tiempo adecuado, se le proporcionan al egresado formatos para notificación de trabajo concluido, que también se entregan en cuadruplicado blanco, azul, amarillo y rosa con la correspondiente explicación para su llenado.

El alumno entrega el formato de notificación de trabajo concluido lleno y firmado por el asesor. Se verifica que esté correcto en caso de que así sea, se sellan los formatos mencionados y se anexa la copia amarilla al expediente del alumno.

Cabe mencionar el hecho de que para la sexta opción de titulación existente, que es el Examen Global de Conocimientos con Trabajo de Aplicación, los trámites que deben realizar los alumnos no son iguales que para las otras opciones, sin embargo todo alumno que se titule por este medio debe formar, por supuesto, parte de la base de datos de titulados. Ya que la mayoría de sus datos son análogos a todas las demás opciones, como nombre, número de cuenta, opción de titulación, carrera, generación, título del trabajo, objetivo general, capítulos, resumen, sinodales, fecha de titulación, número de registro de trabajo terminado, etc.

El trámite para ésta opción consiste en inscribirse al examen de acuerdo a las fechas y requisitos que marca la convocatoria anual, presentar el examen y en caso de aprobar realizar un trabajo escrito con aplicación. Dicho trabajo es propuesto por tres de sus cinco sinodales que se le son asignados después de haber aprobado el examen, el alumno elige uno y lo desarrolla, la fecha de entrega también es contemplada por la convocatoria. Teniendo los cinco votos aprobatorios de los sinodales, el alumno está titulado quedando pendiente el trámite de cédula y título profesional.

1.3 PROBLEMAS IDENTIFICADAS Y SUS ORIGENES

El proceso de chequeo de un nuevo pre-registro contra todos los registros existentes (de las carreras de Actuaría y Matemáticas Aplicadas y Computación), es demasiado extenso y por ende consume mucho tiempo, haciendo énfasis en que son más de 1000 registros los que se tienen que revisar aumentando la posibilidad de error u omisión.

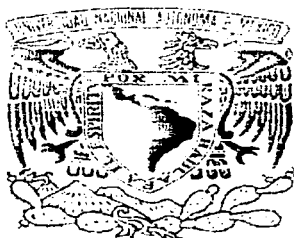
La búsqueda de datos relevantes como: el nombre del asesor, fechas de titulación, cuenta, etc., es demasiado lenta.

Estadísticas y reportes solicitados por alguna autoridad administrativa, son excesivamente lentos en su elaboración. Este tipo de información se solicita con mucha frecuencia y destacan entre otros:

- ◆ Número de titulados por carrera o Programa.
- ◆ Número de registros por carrera o Programa.
- ◆ Número de titulados por generación.
- ◆ Número de registros por generación.
- ◆ Porcentaje de titulados por generación.
- ◆ Porcentaje de titulados por cada opción de titulación.
- ◆ Número de registros por asesor.
- ◆ Número de titulados por asesor.
- ◆ Periodos de tiempo comprendidos entre registro y examen profesional.
- ◆ Número de registros por año.
- ◆ Número de registros por semestre.
- ◆ Número de registros por mes.
- ◆ Número de registros que han excedido el plazo de terminación (Memorias de desempeño profesional 12 meses, Informe de práctica profesional 12 meses y Tesina 6 meses).
- ◆ Número de veces que un profesor sido sinodal de examen profesional.
- ◆ Número de tesis cuyo tema es afín a alguno en específico.
- ◆ Títulos y/u objetivos de trabajos afines a algún tema en específico.
- ◆ Todas las combinaciones que se puedan solicitar en base a los datos existentes.

El poder obtener ésta información de una manera rápida y confiable permite la toma de decisiones encaminadas a elevar los índices de titulación ya que podrían detectarse problemáticas como:

- el tiempo excesivamente largo, en promedio que un egresado consume desde el registro de tesis hasta la realización del examen profesional,
- causas de dicha tardanza,
- causas por las cuales el proceso de titulación de un egresado es abandonado una vez hecho el registro,
- falta de seguimiento de alumnos que han registrado algún tema y abandonan el proceso, etc.



CAPITULO II

SISTEMA DE INFORMACION AUTOMATIZADO COMO PROPUESTA DE SOLUCION



Pudiendo tener toda esta información pueden realizarse acciones que conlleven a un mayor apoyo por parte del Programa a los egresados para su pronta titulación.

1.4 DEFINICION DEL PROBLEMA

Tomando en cuenta todo lo mencionado anteriormente, se requiere de un medio que pueda administrar eficazmente toda la información generada con los trámites de titulación de los egresados desde el registro de su tema de trabajo hasta su examen profesional y

- * que apoye tomas de decisión para la elevación de índices de titulación.
- * que haga más eficiente la elaboración de reportes requeridos por otras instancias de la escuela que requieren datos relativos a la titulación.
- * que mantenga segura la información que maneje.
- * que haga mucho más ágil y eficaz los trámites de titulación.
- * que sea fácil de utilizar para personal del Programa que requiera su consulta.
- * que pueda utilizarse adecuadamente con los recursos humanos y tecnológicos con que cuenta la escuela.
- * que pueda modificarse y adaptarse a cambios futuros.

FALTA PAGINA

No. 16 a la 18

"He a la mejor máquina se le puede dotar de iniciativa" Walter Lippman

II SISTEMA DE INFORMACION AUTOMATIZADO COMO PROPUESTA DE SOLUCION

En este capítulo se podrá saber por qué la opción elegida para la solución del problema es el desarrollo de un sistema de información automatizado.

2.1 CONCEPTOS DE SISTEMAS DE INFORMACION

El presente punto tiene como propósito dar información acerca de la terminología básica que se utiliza en el área del análisis y diseño de sistemas, así como introducir al lector al proceso de desarrollo de sistemas.

En una organización cualquiera que sea su ámbito de acción, el *análisis y diseño de sistemas* puede definirse como el proceso de comprender y analizar su situación actual para observar cómo trabaja y decidir si es necesario o no realizar algún cambio que mejore dicha situación. En el campo laboral la persona encargada de dichas actividades en una organización cualquiera es denominada Analista de Sistemas.

Antes de comenzar un proyecto de desarrollo de sistemas debe comprenderse el funcionamiento de la organización mediante la Teoría de Sistemas.

2.1.1 TEORIA DE SISTEMAS

Un *sistema* es un conjunto de elementos que interactúan entre sí para lograr un fin común. Desde este punto de vista, estamos rodeados de infinidad de sistemas.

Todo sistema organizacional depende, en mayor o menor medida, de una entidad abstracta llamada *sistema de información*. Este es el medio por el cual los datos fluyen de una persona o departamento hacia otros y puede ser cualquier cosa, desde la comunicación interna entre los diferentes componentes de la organización y líneas telefónicas, hasta sistemas de cómputo que generan reportes periódicos para varios usuarios. Los sistemas de información proporcionan servicio a todos los demás sistemas de una organización y enlazan todos sus componentes, de forma que éstos trabajen con eficiencia para alcanzar el mismo objetivo.

La finalidad de un sistema es la razón de que éste exista. Los sistemas interaccionan con su *medio ambiente*, esto es, todas las entidades que se encuentran fuera de la frontera del sistema. Un sistema que tiene esta característica se denomina *sistema abierto*, de otra manera se considera *sistema cerrado*.

Un elemento de suma importancia para el buen funcionamiento es el concepto de *control*. Esto implica la existencia de niveles aceptables de desempeño del sistema denominados *estándares*. Los niveles de desempeño actuales se comparan contra dichos estándares para poder corregir una desviación y asegurar de esta forma que el sistema trabaja adecuadamente. La información que se obtiene como resultado de esta comparación y el proceso de reportar las diferencias encontradas a los elementos encargados del control recibe el nombre de *retroalimentación*.

En la figura 2.1 se puede observar el diagrama del modelo básico de un sistema.

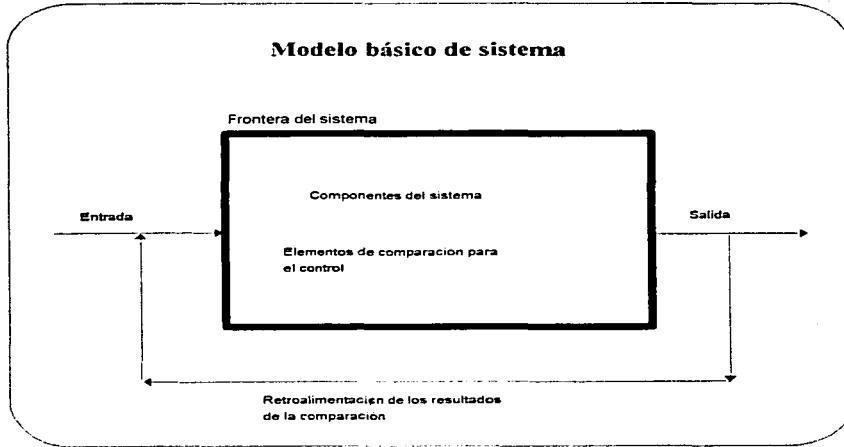


Figura 2.1

Podemos entonces decir que, todas las organizaciones son sistemas que actúan reciprocamente con su medio ambiente recibiendo entradas y produciendo salidas. Todos los sistemas pueden estar formados por otros sistemas más pequeños llamados *subsistemas*, que funcionan para alcanzar fines específicos.

Es siempre de gran importancia considerar que los objetivos de un sistema se alcanzan sólo si se mantiene bajo control. Es aquí donde la retroalimentación se convierte en un elemento esencial para el desarrollo de sistemas de buena calidad, ya sea, para la corrección y mejora de los ya existentes o para la creación de nuevos. De hecho, la autorregulación y el propio ajuste son objetivos deseables de diseño.

La teoría de sistemas es una forma de comprender mejor una situación que se presenta en la realidad, identificando y clasificando componentes y actividades de un sistema para poder evaluarlos y poder mejorarlos si es posible.

Tomando en cuenta los conceptos anteriores, podemos decir en general con respecto al sistema de información desarrollado en la presente memoria que:

a) las finalidades fueron:

- ⇒ procesar entradas tales como datos del alumno, datos del trabajo de titulación, datos del asesor, etc;
- ⇒ mantener archivos de datos relacionados con el trámite de registro de tesis y que son importantes para la Jefatura del Programa de Actuaría y M.A.C. y para otras dependencias relacionadas con este proceso.
- ⇒ producir información y reportes tales como: número de titulados de cada generación, número de titulados por cada opción de titulación, número de tesis asesoradas por cada asesor, etc.

b) formado por subsistemas que incluyen:

- ⇒ hardware, esto es, el equipo necesario para su óptimo funcionamiento;
- ⇒ software, que se refiere a los programas realizados;
- ⇒ medios de almacenamiento de datos para archivos y bases de datos, los procedimientos por medio de los cuales almacenan los datos, se relacionan, se buscan, se consultan y se presentan.

c) el sistema de información fue abierto, ya que interactúa con otros sistemas.

2.1.2 CLASIFICACION DE SISTEMAS DE INFORMACION

En general los sistemas de información pueden clasificarse en tres categorías que dependen de su finalidad:

- 1) Sistemas de procesamiento de transacciones
- 2) Sistemas de información administrativos y
- 3) Sistemas para el soporte de la toma de decisiones.

Conocer la finalidad del sistema bajo estudio esclarece muchos conceptos útiles para la comprensión del funcionamiento del sistema actual. Saber el tipo de sistema que se estudia ayuda a entenderlo mejor y a darle el enfoque y manejo adecuado.

Los sistemas de procesamiento de transacciones realizan actividades rutinarias de la organización, como ocurre por ejemplo en el manejo de nóminas e inventarios, por lo general incluyen programas de cómputo que facilitan el manejo de las transacciones controlando la entrada de datos, su procesamiento, almacenamiento, recuperación y presentación.

Los sistemas de información administrativos se orientan hacia la toma de decisiones utilizando la mayor cantidad de información de que se disponga. Ayudan a tomar decisiones en situaciones que se presentan muy frecuentemente, de tal manera que se puede estudiar todas las variables y factores de decisión con la finalidad de obtener datos que contengan la información más útil para la toma de decisiones futuras.

Los sistemas para el soporte de la toma de decisiones varían con respecto a los anteriores en que aportan información necesaria para hacer frente a problemas que se presentan de manera única. En este caso no es fácil determinar qué información debe tenerse en un momento dado. Este tipo de sistemas son adaptables a quien los usa y realizan decisiones semiestructuradas para apoyar la toma de decisiones. Los sistemas expertos podrían ser una ejemplo de tales sistemas, éstos se construyen a partir de la experiencia de quienes toman las decisiones para la solución de algún tipo particular de problema.

Basado en lo anterior se puede afirmar que el realizado es un sistema para el procesamiento de transacciones, tuvo como finalidad mejorar las actividades de registros de trabajos de titulación. Las transacciones incluían: almacenar datos de alumnos, trabajos, asesores; en una base de datos realizada manualmente, búsquedas de títulos semejantes en dicha base, mantenimiento de la base y actualización de datos, etc.

Se realizaron procedimientos para las transacciones tales como: cálculos, clasificación, ordenamiento, almacenamiento, recuperación y generación de resúmenes de información.

Todas estas actividades forman parte del nivel de operación de la Jefatura de Programa.

Otras características que existían eran:

1. Grandes volúmenes de transacciones,
2. Gran similitud entre las transacciones,
3. Los procedimientos para el procesamiento de transacciones están bien comprendidos y estos algoritmos pueden describirse a detalle
4. Existen pocas excepciones a los procedimientos normales.

El gran volumen de transacciones precisas, el nivel de operación de la Secretaría Técnica de la Jefatura de Programa condujo, como se verá más adelante, a la necesidad de un sistema de información apoyado en la computadora. En este sistema, los procedimientos que accesan datos, los procesan, almacenan, recuperan y presentan son programas de computadora codificados en algún lenguaje específico.

2.2 METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACION

Existen tres enfoques para el desarrollo de sistemas que dependen de las circunstancias específicas de necesidades, recursos con que se cuenta, tiempo límite para desarrollos, impacto del nuevo sistema en los usuarios potenciales, etc. Los tres enfoques mencionados son:

1. Método del ciclo de vida para el desarrollo de sistemas
2. Método del desarrollo de análisis estructurado
3. Método del prototipo de sistemas

A continuación se presenta el análisis de cada enfoque, sus características y las condiciones bajo las cuales se tendría el mayor beneficio para una organización.

2.2.1 METODO DEL CICLO DE VIDA PARA EL DESARROLLO DE SISTEMAS

El método del ciclo de vida para el desarrollo de sistemas es la metodología clásica a seguir, se aplica más comúnmente que cualquier otra, aunque en ocasiones no sea la óptima pero muchos analistas de sistemas desconocen las demás metodologías alternativas.

Sus características de aplicación son:

- * Requerimientos del sistema de información predecible
- * Manejable como proyecto
- * Requiere que los datos se encuentren en archivos y bases de datos
- * Gran volumen de transacciones y procesamiento
- * Requiere de la validación de los datos de entrada
- * Abarca varios departamentos
- * Tiempo de desarrollo largo
- * Desarrollo por equipos de proyecto

En el periodo de desarrollo y uso del sistema pueden identificarse varias etapas, que juntas constituyen lo que se llama el ciclo de vida de sistemas.

Probablemente Royce en 1970 propuso por primera vez el modelo inicial del ciclo de vida de sistemas, y desde entonces el modelo ha tenido muchos perfeccionamientos y variaciones. Sin embargo, un modelo general consta de las siguientes etapas:

1. *Investigación preliminar.* Debe comprenderse con total precisión lo que se desea obtener con el desarrollo del sistema. En ocasiones los usuarios del sistema no son muy claros ya que ni ellos mismos tienen cien por ciento claro lo que desean. De tal manera que el trabajo conjunto con las personas que usarán el sistema es muy importante, indispensable.
2. *Análisis y determinación de las necesidades.* Una vez acordados los servicios, restricciones y objetivos del sistema, éstos deben definirse de manera comprensible, tanto para los usuarios como para los desarrolladores. Ambos grupos citados de personas deben responder preguntas como:

- * ¿Qué es lo que se hace actualmente?
- * ¿Cómo se hace?
- * ¿Con qué frecuencia?
- * ¿De qué dimensión es el conjunto de transacciones?
- * ¿Qué tan eficientemente se realizan las tareas?
- * ¿Existe algún problema?
- * ¿Qué tan serio es?
- * ¿Cuál es su origen?

Para contestar adecuadamente estas preguntas deben utilizarse todas las fuentes de información posibles.

3. *Diseño del sistema y del software.* El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados en la fase de análisis. Se diseñan entonces procedimientos precisos de captura de datos, formatos de salidas, procesos de cálculos, estructuras de almacenamiento, clasificación y recuperación de datos, etc. Se crean documentos que contienen estas especificaciones de diseño para poder transformarlo fácilmente en uno o más programas de computación.
4. *Desarrollo del software o aplicación.* Los encargados de la programación pueden adaptar software comercial o escribir programas que cumplan con el diseño. Estas personas deben elaborar la documentación del o los programas y proporcionar una explicación del por qué los procedimientos se codificaron de cierta forma específica.
5. *Prueba del sistema.* Las unidades de programa individuales o los programas se integran y se prueban como un sistema completo para asegurar que se cubren las necesidades del software. En ocasiones las pruebas son conducidas por personas ajenas al grupo de programadores, para tratar de asegurar que sean completas e imparciales y, que así, el software sea más confiable.
6. *Implantación y mantenimiento.* La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación. Cada estrategia de implantación tiene sus méritos y características de acuerdo con la situación que se considere dentro de la empresa o institución. La actividad de mantenimiento implica corregir errores que no se descubrieron en las primeras etapas del ciclo de vida, mejorar la aplicación de las unidades del sistema y aumentar los servicios de éste a medida que se perciben nuevas necesidades.

Sin embargo, es inútil tratar de identificar las subfases dentro de cada fase del ciclo de vida de sistemas. No existe un acuerdo general sobre cuales son estas subfases, y cada proyecto específico suele dividirse de manera distinta.

Para fines de administración de actividades en el desarrollo de sistemas, es útil diferenciar las fases del ciclo de vida, pero, en la práctica, las etapas de desarrollo se superponen y proporcionan información unas a otras.

La última fase del ciclo de vida (implantación y mantenimiento) tiene una relación especial con respecto a las las fases previas del ciclo ya que la información que proporciona, es retroalimentada hacia todas las anteriores. La fase de mantenimiento puede originar cambios en las necesidades, el diseño, o puede indicar la necesidad de pruebas adicionales del sistema.

En general muchos autores explican el ciclo de vida como las diferentes etapas que deben terminarse completamente antes de dar paso a la siguiente, ésto desde un punto de vista rígido que no permite interactuar entre las fases y no da la flexibilidad para poder regresar a cualquiera de ellas desde alguna posterior. Es en la práctica donde puede verse que un manejo más real y menos severo, aunque tomando siempre en cuenta una buena administración de actividades, da como resultado sistemas más eficientes que tendrán mayor éxito.

La figura 2.2 puede dar una visión dinámica y clara de lo que debe ser el ciclo de vida de sistemas.

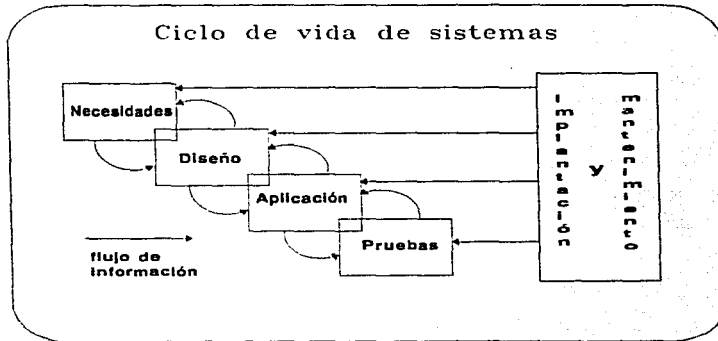


Figura 2.2

2.2.2 METODO DEL DESARROLLO DE ANALISIS ESTRUCTURADO

Generalmente es difícil comprender de manera completa sistemas grandes y complejos. El método de desarrollo de análisis estructurado tiene como finalidad superar esta problemática por medio de

- a) la división del sistema en componentes más simples y
- b) la construcción de un modelo del sistema.

En general, el análisis estructurado se basa en especificar los que se quiere que haga el sistema. No establece cómo se cumplirán los requerimientos o la forma en que se implantará la aplicación. En cambio permite que sean observados los elementos lógicos, esto es, lo que hará el sistema, separados de los componentes físicos (computadoras, terminales, etc.). Después se puede desarrollar un diseño físico para la situación en que se necesita.

Este método maneja técnicas como son símbolos gráficos, diagramas de flujo de datos y diccionario de datos.

Sin embargo la mayor utilidad de este enfoque de desarrollo es complementar a otro y no como tal.

2.2.1 METODO DEL PROTOTIPO DE SISTEMAS

Para la realización del sistema de información de la presente memoria se utilizó una herramienta que resultó sumamente útil, ésta fue el desarrollo de prototipos.

El desarrollo de prototipos es una estrategia opcional para el desarrollo de sistemas. Este método hace que el usuario participe de manera muy directa en el análisis y diseño. Es un desarrollo iterativo donde el usuario participa directamente en el proceso. En éste, el desarrollador directo del sistema iba a ser uno de los usuarios finales.

Las características de aplicación son:

- * Condiciones únicas de aplicación donde se tiene poca información.
- * Util para comprobar la factibilidad del sistema, identificar los requerimientos del usuario, evaluar el diseño de un sistema o examinar el uso de una aplicación.

El prototipo es un sistema que funciona, desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el nuevo sistema. Al igual que cualquier sistema basado en la computadora, está constituido por software que acepta entradas, realiza cálculos, produce información ya sea impresa o presentada en una pantalla, o que lleva a cabo otras actividades significativas. Es la primera versión, o iteración de un sistema de información; es el modelo original.

El usuario evalúa el diseño y la información generada por el sistema. Esto puede hacerse si los datos utilizados y las situaciones son reales. Es por lo tanto seguro que existan cambios a medida que el sistema es utilizado.

El prototipo no contiene todas las características ni lleva a cabo la totalidad de las funciones necesarias del sistema final. Más bien incluye elementos suficientes para permitir a las personas utilizar el sistema propuesto para determinar qué les gusta, qué no les gusta e identificar aquellas características que deben cambiarse o añadirse. El proceso de desarrollo y empleo de un prototipo tiene cinco características:

- ◆ El prototipo es una aplicación que funciona.
- ◆ La finalidad del prototipo es probar varias suposiciones formuladas por analistas y usuarios con respecto a las características requeridas del sistema.
- ◆ Los prototipos se crean con rapidez.
- ◆ Los prototipos evolucionan a través de un proceso iterativo.
- ◆ Los prototipos tienen un costo bajo de desarrollo.

Los prototipos, así como los dos métodos anteriores (ciclo de vida y análisis estructurado) son más eficaces en el desarrollo de sistemas cuando se aplican en circunstancias que cumplen con ciertas condiciones específicas. Cualquiera de las siguientes cinco condiciones sugieren la necesidad de utilizar prototipos:

- 1) No se conocen totalmente los requerimientos.
- 2) Se conocen los requerimientos de información pero necesitan verificarse y evaluarse.
- 3) Existen altos costos de inversión de recursos financieros y humanos así como del tiempo necesario de desarrollo.
- 4) Existe un alto riesgo de la organización al poder tener errores en la evaluación de los requerimientos o el desarrollo incorrecto del sistema.
- 5) La pretensión de instalar nueva tecnología de algún área que abra nuevas fronteras a la organización. En esencia este punto es la falta de experiencia.

El desarrollo de prototipos debe llevarse a cabo de una forma ordenada. Los pasos a seguir con el método de desarrollo de prototipos se puede ver en el siguiente diagrama, en la figura 2.3.

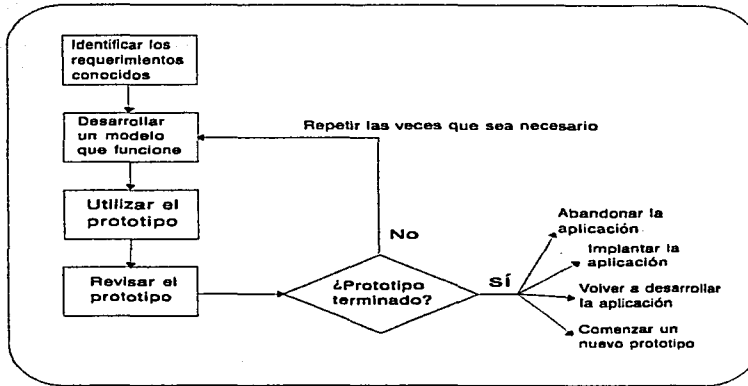


Figura 2.3

Como puede verse el desarrollo de un prototipo de aplicación sigue un proceso organizado que comienza con la identificación inicial de los requerimientos *conocidos* que deben satisfacerse. Para ello se determinan los fines para los que servirá el sistema así como el alcance de sus capacidades.

Una vez hecho ésto, se desarrolla un modelo inicial empleando herramientas tales como lenguajes de cuarta generación, generadores de aplicaciones, generadores de código, generadores de reportes, generadores de pantallas, sistemas de diccionario de datos, bibliotecas de código reutilizable. Este tipo de software se encuentra disponible en el mercado y brinda una ayuda invaluable al analista de sistemas ya que ahorra grandes cantidades de tiempo. Recordemos que los prototipos deben crearse con rapidez. Las razones por las cuales el tiempo es importante son que desarrollando prototipos rápidamente no se pierde el estado de ánimo sobre el proyecto y también que los usuarios puedan comenzar a evaluar la aplicación lo más pronto posible.

El modelo inicial se pone en uso. El usuario debe trabajar con el prototipo para evaluar sus características y operación. Manipularlo bajo condiciones reales permite obtener la familiaridad imprescindible para determinar las modificaciones, mejoras o eliminaciones necesarias.

- A medida que se evalúa el prototipo, se hacen sugerencias y cambios. Por medio de la iteración , el prototipo evoluciona hasta llegar el momento en que usuarios y analistas están de acuerdo en que el sistema incluye todas las características necesarias o es evidente que no se obtendrá mayor beneficio con una iteración adicional.

Una vez que el prototipo está terminado se debe de tomar la decisión de

a) Abandonar el proyecto

El prototipo condujo a saber que el sistema era innecesario o existía otra solución. El prototipo aportó entonces información y experiencia importantes.

b) Implantar el prototipo

El prototipo se convierte en el sistema que se necesita.

c) Redesarrollar la aplicación

La información recopilada durante el desarrollo del prototipo sugiere características que deben añadirse a la aplicación.

d) Inicio de un nuevo prototipo

Se encuentra que los requerimientos son diferentes a los anticipados. De nuevo es importante notar que el esfuerzo no se ha desperdiciado.

El desarrollo de prototipos es una técnica probada que mejora la efectividad total del esfuerzo.

2.3 DIAGNOSTICO Y PROPUESTA DE SOLUCION

Tomando en cuenta los requerimiento que debían satisfacerse , de los cuales ya se habló en el capítulo anterior , que el Programa de Actuaría y M.A.C. cuenta con instalaciones adecuadas para la instalación y manejo de una Microcomputadora PC, y que existe en la Jefatura de Actuaría y M.A.C. personal que puede aprender fácilmente la utilización de un sistema, claro está, si éste es lo suficientemente amigable para el usuario; se propuso como mejor opción el desarrollo por medio de la metodología de prototipos de un sistema de información basado en la computadora que administre la información de trámites de titulación para el Programa que sea utilizado principalmente por el Jefe de Programa así como los secretarios técnicos de M.A.C. y Actuaría en las instalaciones del Programa.

El método que se utilizó fue el de prototipos ya que aunque se conocían la mayoría de los requerimientos de información, muchos otros surgían y surgen como nuevos y el sistema debía ser capaz de atenderlos de la mejor manera posible. Dado que la información que suele requerirse se puede solicitar en cualquier momento, se necesitaba que el sistema comenzara a trabajar lo antes posible con el menor gasto de recursos posible.



CAPÍTULO III

DESARROLLO DEL SISTEMA, IMPLANTACION Y MANTENIMIENTO



El hardware con que se contaba (y se cuenta actualmente) es una Computadora Personal marca ACER con:

- Procesador 386
- 33 Mhz. de velocidad
- 4 Mb. de memoria RAM
- Disco duro de 120 Mb
- Monitor monocromático
- Drive de 3.5" (1.4 Mb)
- Drive de 5.25" (1.2 Mb)

También se cuenta con una impresora marca EPSON LQ1070 de matriz de puntos que permite formato de hoja tamaño carta, oficio y apaisada.

La base de datos se almacenaría en el disco duro de la PC así como respaldos en disco flexible.

El soporte técnico del equipo está a cargo del Departamento de Servicios de Cómputo de la Escuela. Dicho soporte consiste en mantenimiento correctivo y preventivo.

El equipo de localiza físicamente en las oficinas de la Jefatura de Actuaría y M.A.C. disponible únicamente para el personal de dicha dependencia.

El costo de desarrollo de software y adquisición de equipo es mínimo con respecto a los beneficios que se obtendrán ya que el sistema contará con:

1. Eficacia en el desempeño
 - Capacidad para realizar las tareas requeridas
 - Buen diseño en pantalla
 - Capacidad adecuada
2. Eficiencia operativa
 - Tiempos de respuesta rápidos
 - Captura eficiente
 - Salidas eficientes
 - Almacenamiento de datos eficiente
 - Respaldos eficientes
3. Facilidad de uso

Uso de interfaces satisfactorias
Disponibilidad de menús de ayuda
Interfaz flexible
Buena recuperación ante errores

4. Flexibilidad

Opciones de entrada
Opciones de salida

5. Calidad de la documentación

Buena organización
Respuestas adecuadas a las preguntas

2.3.1 ELECCION DE UN LENGUAJE DE PROGRAMACION

Una de las decisiones más importantes que se deben tomar al diseñar y construir un sistema grande de software, es la del lenguaje de programación que se va utilizar en la aplicación del sistema. La elección de un lenguaje de programación apropiado reduce al mínimo las dificultades de codificar un diseño, reduce la cantidad de pruebas necesarias, hace el programa más legible y, por lo tanto, más fácil de mantener.

El lenguaje de programación debe tener la posibilidad de utilizar nombres significativos, debe disponer de características de estructuración, como procedimientos y funciones, y el lenguaje debe tener construcciones de control adecuadas que eviten el uso de las proposiciones GOTO (Programación Estructurada).

A continuación se enlistan los criterios ideales para la selección de un lenguaje de programación :

- *Los requisitos del contratista del sistema.* El contratista de un sistema puede especificar que se use un lenguaje de programación específico, y en general, se debe respetar ese requisito. El contratista puede tener una lista de lenguajes posibles bajo los cuales se debe realizar el sistema. Esto principalmente se debe a que el personal del contratista se encargará del mantenimiento del sistema y por ende debe estar en un lenguaje que ellos entiendan.
- *Disponibilidad de compiladores del lenguaje.* Debe disponerse de un traductor del lenguaje de aplicación de aceptable eficiencia para aplicar el lenguaje.
- *Disponibilidad de instrumentos de software para apoyar el desarrollo de programas.* Los instrumentos de software, como editores de texto, procesadores de palabra, utilerías de apoyo,

depuradores de código y manejadores de compilación y ligado. Tienen una participación importante dentro del apoyo al proceso de programación.

- *El tamaño del proyecto.* Si el proyecto es excesivamente grande, puede ser apropiado diseñar y aplicar un lenguaje de programación específico para él.
- *El conocimiento del personal de programación existente.* Aunque no representa dificultad especial para programadores experimentados aprender un nuevo lenguaje, necesitan cierta práctica antes de obtener una verdadera competencia. Si no hay otros factores en contra se debe escoger un lenguaje de programación con el que esté familiarizado el personal de programación.
- *La necesidad de transportar el software.* Si el software está orientado a una sola arquitectura de hardware y su tiempo de vida es limitado, los aspectos de transporte no son importantes. Por otro lado si el sistema va a funcionar en máquinas distintas es importante seleccionar un lenguaje que permita trabajar con programas portátiles.
- *La aplicación que se esté programando.* Aunque existen los llamados lenguajes de propósito general que se pueden utilizar en cualquier aplicación, algunos son más adecuados que otros para cierto tipo de aplicaciones, por ejemplo, PASCAL y FORTRAN son bastante adecuados para aplicaciones científicas y matemáticas pero bastante inadecuados para escribir compiladores o sistemas operativos.

No hay lenguajes que sean realmente adecuados para todas las aplicaciones, y no es recomendable la estandarización de un solo lenguaje. Es mejor tener una lista de lenguajes de programación cada uno de ellos diseñado para una aplicación particular.

Por las consideraciones anteriores y los recursos con los que se cuentan, se dedujo que el lenguaje de programación debería cumplir con los siguientes requerimientos:

- La utilización de un lenguaje de programación que fuese a su vez un manejador de bases de datos.
- Que tenga la posibilidad de crear una interfaz amigable para el usuario.
- Que trabaje en cualquier computadora compatible con IBM.
- Que trabaje bajo el sistema operativo MS-DOS, versión 3.3 o superior.

- Que sea flexible y permita el fácil mantenimiento e implementación de mejoras posteriores.

Los lenguajes de programación que cumplen con la mayoría de éstas especificaciones son: DBASE III Plus, DBASE IV, Clipper 5.0, 5.1 y 5.2, Fox-pro 2.0, Foxbase+ e Informix.

En la fig. 2.4 se puede apreciar una tabla comparativa entre algunos de los lenguajes más populares de bases de datos.

Lenguajes	Velocidad	Sistema Operativo
Dbase III, IV Informix Clipper 5.1 y 5.2 Fox-base+	Lento Rápido Rápido Rápido	DOS DOS,NOVELL,UNIX DOS,NOVELL DOS,NOVEL,UNIX, XENIX. DOS.NOVELL.
Fox-Pro	Rápido	

Fig. 2.4

Aunque la mayoría de estos lenguajes están estandarizados con el formato XBASE y son rápidos en ambiente MS-DOS, se eligió el lenguaje **CLIPPER 5.2** por las siguientes razones:

- ⇒ Cumple con todos los requerimientos arriba listados.
- ⇒ Es rápido en ambiente DOS y en ambiente Multiusuario.
- ⇒ Posee un compilador y ligador integrado para creación de archivos ejecutables.
- ⇒ La existencia de software externo que genera código en lenguaje CLIPPER para pantallas de captura y formatos básicos de reportes.
- ⇒ El depurador de código (Debugger) integrado, para rastrear y corregir errores en forma rápida y eficiente.
- ⇒ Su versatilidad para migrar una aplicación monousuario en ambiente DOS a una aplicación multiusuario en ambiente Novell.
- ⇒ Sus bibliotecas integradas para enlazar código hecho en lenguaje C, Ensamblador, y Dbase.
- ⇒ Uso de Overlays dinámicos que aprovechan el límite de la memoria, es decir, no cargan todo el programa en memoria sino de acuerdo a como se va ejecutando se va asignando la memoria requerida.
- ⇒ Sus controladores de bases de datos sustituibles (RDD) que ofrecen acceso a bases de datos hechas en otros formatos, como FOX-PRO, PARADOX y DBASE IV.
- ⇒ Preprocesador del compilador potente y flexible al estilo del lenguaje C.

⇒ Posibilidad de programar con Orientación a Objetos (POO), con sus cuatro clases de objetos: GET, ERROR, TBCOLUMN, TBROWSE.

A continuación se enlistan los requerimientos de equipo para utilizar Clipper 5.2:

- * Una computadora personal IBM o compatible (100%).
- * 640 kb. de memoria RAM mínimo.
- * MS-DOS versión 3.3 o superior.
- * Una unidad de disco flexible de 3 $\frac{1}{2}$ ".
- * Disco duro con espacio para 5 megabytes.
- * Monitor monocromático (Recomendable a color.)
- * Impresora (recomendable).

Anteriormente se comentó acerca del equipo disponible para la ejecución del sistema, por tanto, se reúnen los requisitos para la utilización del paquete de Clipper y también para los del sistema terminado, de los cuales se hablará mas adelante.

El paquete completo de Clipper 5.2 contiene:

- * Editor de programa
- * Sistema de ayuda en línea (Norton Guide's)
- * Compilador
- * Generador de reportes
- * Generador de etiquetas
- * Ligador (Rlink)
- * Utilería Rmake
- * Depurador (Debugger)
- * Programas ejemplos
- * Utilerías para manejo de bases de datos (DBU)
- * Preprocesador.

FALTA PAGINA

No. 36a | la 38

"Normalmente se hacen considerables tareas que inherentemente requieren complejos algoritmos, y estos implican gran cantidad de detalles y los detalles son la punta en la cual el diseño se resaca". N. Wirth

III. DESARROLLO DEL SISTEMA, IMPLANTACION Y MANTENIMIENTO

En este capítulo se podrán conocer las acciones llevadas a cabo en el desarrollo del sistema.

3.1 DESARROLLO DEL SISTEMA UTILIZANDO EL METODO DE PROTOTIPOS

Como se vio con anterioridad, existieron varias razones para adoptar el desarrollo de prototipos como la metodología a utilizar. Algunas de estas fueron el tiempo y las necesidades. El uso de esta metodología se dio de forma natural al tener que cubrir un requerimiento de información en un tiempo tan corto como unas semanas. Se comenzó con una simple pantalla de captura que mediante sucesivas iteraciones se fue elaborando más y más hasta que finalmente llegó a un diseño sumamente manejable y eficiente. Es importante mencionar que cada mejora que se hacía al sistema era utilizado inmediatamente en una aplicación real. Esto hizo al sistema tener un alto grado de funcionalidad. Sin embargo es importante no perder la objetividad y perseguir metas alcanzables que hagan tener fin a la serie de iteraciones requeridas en la utilización de esta técnica.

Un concepto importante que debe tomarse en cuenta a lo largo del desarrollo de prototipos es la interfaz con el usuario. En forma general lo anterior se refiere a la manera en que el usuario se comunicará con el software y viceversa. La interfaz del usuario es la cara con que éste último mide la calidad del sistema. Si un sistema tiene una interfaz mal diseñada, esta expuesto al rechazo, independientemente de las posibilidades que ofrezca. Una interfaz mal diseñada puede hacer que el usuario cometa errores potencialmente catastróficos. Si la información se presenta de una manera confusa o engañosa, el usuario puede interpretar mal el significado de un elemento de información, y a partir de ahí, iniciar una secuencia de acciones peligrosas.

Para diseñar una interfaz de usuario aceptable, debe considerarse detenidamente la psicología del usuario del sistema. Algunos de los aspectos que tienen singular importancia y que deben considerarse son la sobrecarga de información, la complejidad de alguna tarea, el tiempo de respuesta del sistema y el grado de control sobre el sistema que se permite al usuario.

Como herramienta inicial se utilizó un generador de código en Clipper denominado Jennifer version 2.0, se fabricaron los prototipos de pantallas de captura y reportes de acuerdo a las problemáticas detectadas, como la limitante principal era el tiempo se identificaron los requerimientos principales los cuales fueron cubiertos por esta serie de prototipos consecutivos, de tal forma que prototipo terminado

funcionaba y era puesto en marcha. Los requerimientos secundarios fueron cubiertos sucesivamente hasta lograr una excelente presentación así como un eficiente desempeño de todas las tareas requeridas. Se tuvo especial cuidado en hacer el manejo del sistema lo más fácil posible, el tiempo de respuesta lo más reducido (dependiendo del equipo) y sobre todo que el usuario tenga un completo dominio sobre el sistema en cualquier eventualidad o posible error.

3.1.1 DESARROLLO DEL PROTOTIPO DE SALIDAS

Las salidas es la información que reciben los usuarios del sistema de información. Las salidas que dará el sistema son reportes impresos y por pantalla. Para crearlas se trabajó de cerca con los usuarios potenciales para poder hacer cualquier cambio pertinente y tratar de llegar a un resultado satisfactorio.

La salida debía satisfacer los objetivos planteados, adaptarse al usuario y sobre todo permitir la manipulación por parte del usuario para que dicho reporte se adapte a las necesidades de información que se requieren, tales necesidades son principalmente la cantidad adecuada en el momento oportuno. Se trabajó detenidamente en las salidas o reportes para obtener una estructura lo más flexible posible, se probó iterativamente hasta lograr satisfacer ésta necesidad. La posibilidad de crear una variedad de reportes de casi cualquier tipo, mediante elecciones de los campos, usando operadores relacionales, usando opciones de búsqueda y pudiendo elegir el formato final de dicho reporte, permitieron satisfacer por completo las necesidades de información que se tenían. Este procedimiento de fabricación de reportes según las necesidades del usuario se explicará a detalle más adelante. La figura 3.1 ilustra el formato de salida impresa para los datos generales de los asesores, si sólo se quieren visualizar se debe acceder a la opción "Asesor" del menú actualizar.

DISEÑO DE SALIDAS

DATOS GENERALES DEL ASESOR	
Nombre _____	Clave _____
Grado o título _____	
Dirección Domicilio _____	
Dirección Trabajo _____	
Tel. Domicilio _____	Tel. Trabajo _____

Fig. 3.1 Formato para reporte impreso sobre asesores.

Para las salidas impresas de reportes existen tres formatos distintos, la fig. 3.2 ilustra el encabezado para el reporte de datos propios del registro de trabajo de titulación, destacan el nombre del asesor, título, opción de titulación, fechas de registro, terminación y titulación, etc.

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ACATLAN JEFATURA DE ACTUARIA Y MATEMATICAS APLICADAS Y COMPUTACION										
										Pág. ____
<ENCABEZADO DE REPORTE>										Fecha ____
Nombre Alumno	Cuenta	Carrera	Gener.	Título	Opc. Tit.	Asesor	Fecha Reg.	Fecha Ter.	Fecha Tit.	Clave Reg.

Fig. 3.2 Formato para reporte impreso con datos de registro.

De la misma forma la fig. 3.3 ilustra el encabezado para el reporte con datos acerca de la titulación, entre ellos destacan opción de titulación, asesor, sinodales, fecha de titulación, etc.

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ACATLAN JEFATURA DE ACTUARIA Y MATEMATICAS APLICADAS Y COMPUTACION								
								Pág. ____
<ENCABEZADO DE REPORTE>								Fecha ____
Nombre Alumno	Carrera	Título de Trabajo	Opc. Tit.	Nombre Asesor	Fec. Reg.	Fec. Tit.	Sinodales	Clave Reg.

Fig. 3.3 Formato para reporte impreso con datos de titulación.

Para impresión de datos generales del alumno, obsérvese la fig. 3.4, aquí se imprimen conforme al encabezado presentado los datos concernientes a su nombre, cuenta, direcciones de casa y trabajo, etc. Esto es, se omiten datos de titulación y registro para concentrarse en los restantes, con éstas tres combinaciones se contemplan todos los datos existentes, con lo cual se logra una salida adecuada de la

información almacenada, sin olvidar los restantes formatos, que a continuación se explican, y las salidas por pantalla.

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ACATLAN								
JEFATURA DE ACTUARIA Y MATEMATICAS APLICADAS Y COMPUTACION								
							Pág. ____	
<ENCABEZADO DE REPORTE>							Fecha ____	
Nombre Alumno	Cuenta	Carrera	Gener.	Título de trabajo	Dir. casa	Tel. Casa	Dir. Trab.	Tel. Trabajo

Fig. 3.4 Formato para reporte impreso con datos generales.

Existe otra forma de reporte para un alumno o registro individual, cuando se realiza una búsqueda ya sea por palabra, nombre, número de cuenta o clave de registro, éste formato (Fig. 3.6) imprime TODOS los datos de el alumno seleccionado, se podría decir que es una combinación de los tres formatos de registro, titulación y datos generales pero con la diferencia de que es en forma individual.



Fig. 3.5 Reporte de información general por pantalla

En la figura 3.5 se puede apreciar el formato del diseño del reporte por pantalla de TODA la información, tener un vistazo rápido de parte de la información es útil, antes de esta pantalla siempre

hay información preliminar, para decidir entre sólo observar toda la información o la posible impresión de la misma.

DATOS GENERALES SOBRE ALUMNOS	
Nombre _____	Cuenta _____
Carrera _____	Generación _____ Clave Registro _____
Fecha Reg. _____	Fecha Term. _____ Fecha Tit. _____
Opción Titul. _____	Asesor _____
Dirección Casa _____ Tel. _____	
Dirección Trabajo _____ Tel. _____	
COAUTORES	
Autor 2 _____	
Autor 3 _____	
Autor 4 _____	
TITULO	

OBJETIVO GENERAL	

CAPITULOS	

RESUMEN	

SINODALES	

Area/Materia 1 _____	
Area/Materia 2 _____	

Fig. 3.6 Formato para reporte impreso de datos generales del alumno.

3.1.3 DESARROLLO DE PROTOTIPO DE BASES DE DATOS

Después de detallar el tipo de entradas de información, validaciones y salidas se pueden definir las bases de datos que se van a utilizar dentro del sistema, toda la información que entra y sale debe estar contemplada en la base de datos y basada en el estudio preliminar de requerimientos. Se definieron las bases de datos como a continuación se explica.

El diseño de la base de datos principal "Tesi_mac.dbf" , fig. 3.7, contiene nombre, tipo, tamaño, y descripción de cada uno de los campos que contiene la base, en esta base se almacenará toda la información que utilizará el sistema. Se crearon bases alternas que se relacionaron con la principal mediante campos índices comunes, aprovechando el fácil manejo que se logra con los campos y archivos índices con Clipper 5.2. Los campos comunes se pueden identificar fácilmente en los diagramas con un asterisco inmediatamente después del nombre.

El crear bases alternas principalmente obedeció a la necesidad de tener respaldos de información más fácilmente, es decir, si se tuviera toda la información en una sola base de datos habría más probabilidades de perder información. Además la actualización de la información se logra de manera más eficiente accediendo a una base pequeña que a una de mayor tamaño. Otra razón fundamental es la de economizar espacio en el disco donde se hará el almacenamiento, dado que trabajar con claves y campos pequeños economiza espacio y acelera la velocidad de acceso, sobre todo cuando la información es repetitiva y redundante, cosa que debemos evitar a toda costa.

La figura 3.8 muestra la estructura de la base de datos de asesores "Asesor.dbf", aquí se almacenará toda la información concerniente a los asesores desde su nombre, dirección, teléfono, etc. el campo relacionado (clv_ase) se puede identificar por el asterisco.

La figura 3.9 muestra la estructura de la base de datos de carrera "Carrera.dbf", aquí se almacenará toda la información concerniente a los carreras que se van a trabajar, el campo relacionado (clv_car) se puede identificar por el asterisco.

La figura 3.10 muestra la estructura de la base de datos de generaciones "Gener.dbf", aquí se almacenará toda la información concerniente a las generaciones de cada carrera, el campo relacionado (clv_gen) se puede identificar por el asterisco.

La figura 3.11 muestra la estructura de la base de datos de materias o áreas de conocimiento "Mater1.dbf", aquí se almacenará toda la información concerniente a las materias o áreas de conocimiento de cada carrera, el campo relacionado (clv_mat1) se puede identificar por el asterisco.

La figura 3.12 muestra la estructura de la base de datos de opciones de titulación "Opc_tit.dbf", aquí se almacenará toda la información concerniente a las opciones de titulación de cada carrera, el campo relacionado (clv_opc) se puede identificar por el asterisco.

La figura 3.13 muestra la estructura de la base de datos de autores de tesis colectivas "Colec.dbf", aquí se almacenará toda la información concerniente a los coautores de las tesis colectivas, el campo relacionado (co1) se puede identificar por el asterisco.

Nombre de campo	Tipo de campo	Tamaño de campo	Descripción
Alum	Caracter	40 espacios	Nombre del alumno
Cuen	Caracter	09 espacios	Número de cuenta del alumno
Clv_car *	Numérico	02 espacios	Clave de la carrera
Clv_gen *	Numérico	02 espacios	Clave de generación
Diret	Caracter	80 espacios	Dirección de trabajo
Dired	Caracter	80 espacios	Dirección de casa (domicilio)
Telet	Caracter	12 espacios	Teléfono de trabajo
Teled	Caracter	12 espacios	Teléfono de casa (domicilio)
Titulo	Memo	Variable	Título del trabajo
Obj_gen	Memo	Variable	Objetivo general del trabajo
Capitulo	Memo	Variable	Capítulos del trabajo
Resumen	Memo	Variable	Resumen del trabajo
Sinodal	Memo	Variable	Sinodales asignados p/revisión
Freg	Fecha	08 espacios	Fecha de registro del trabajo
Fter	Fecha	08 espacios	Fecha de terminación trabajo
Ftit	Fecha	08 espacios	Fecha de titulación del alumno
Clvr	Caracter	04 espacios	Núm. o folio del trabajo term.
Clv_opc *	Numérico	02 espacios	Clave de opción de titulación
Clv_ase *	Numérico	03 espacios	Clave de asesor
Clv_mat1*	Numérico	02 espacios	Clave de materia o área (1)
Clv_mat2*	Numérico	02 espacios	Clave de materia o área (2)

Fig. 3.7 Esquema de la base de datos Principal TESI_MAC.DBF.

Nombre de campo	Tipo de campo	Tamaño de campo	Descripción
Ases	Caracter	40 espacios	Nombre del asesor
Clv_ase*	Númérico	03 espacios	Clave de asesor
Grado	Caracter	20 espacios	Grado académico del asesor
Diret	Caracter	50 espacios	Dirección de trabajo de asesor
Dired	Caracter	50 espacios	Dirección de casa de asesor
Telet	Caracter	12 espacios	Teléfono de trabajo de asesor
Teled	Caracter	12 espacios	Teléfono de casa de asesor

Fig. 3.8 Esquema de la base de datos para Asesores (ASESOR.DBF)

Nombre de campo	Tipo de campo	Tamaño de campo	Descripción
Carr	Caracter	20 espacios	Nombre de la carrera
Clv_car*	Númérico	04 espacios	Clave de carrera

Fig. 3.9 Esquema de la base de datos para carreras (CARRERA.DBF).

Nombre de campo	Tipo de campo	Tamaño de campo	Descripción
Gene	Caracter	05 espacios	Generación (ingreso y egreso)
Clv_gen*	Númérico	02 espacios	Clave de generación
Clv_car*	Númérico	04 espacios	Clave de carrera

Fig. 3.10 Esquema de la base de datos para generación (GENER.DBF).

Nombre de campo	Tipo de campo	Tamaño de campo	Descripción
Materia	Caracter	20 espacios	Nombre del área o materia
Clv_mat*	Numérico	02 espacios	Clave de área o materia
Clv_car*	Numérico	02 espacios	Clave de carrera

Fig. 3.11 Esquema de la base de datos para materias o áreas de conocimiento (MATER.DBF).

Nombre de campo	Tipo de campo	Tamaño de campo	Descripción
Opc_tit	Caracter	45 espacios	Opción de titulación
Clv_opc*	Numérico	01 espacios	Clave de la opción de titulación

Fig. 3.12 Esquema de la base de datos para opciones de titulación (OPC_TIT.DBF).

Nombre de campo	Tipo de campo	Tamaño de campo	Descripción
Co1*	Caracter	40 espacios	Coautor 1 de tesis colectiva
Co2	Caracter	40 espacios	Coautor 2 de tesis colectiva
Co3	Caracter	40 espacios	Coautor 3 de tesis colectiva
Co4	Caracter	40 espacios	Coautor 4 de tesis colectiva
Co5	Caracter	40 espacios	Coautor 5 de tesis colectiva

Fig. 3.13 Esquema de la base de datos para Tesis colectivas (COLEC.DBF).

3.2. IMPLANTACION Y MANTENIMIENTO

La implantación es el proceso que define la operatividad del sistema de información y que permite obtener beneficios de esa operación. Dentro de las actividades que se llevaron a cabo para implantar el presente sistema destacan:

-La conversión del sistema antiguo, como se mencionó anteriormente el sistema anterior se ejecutaba en forma manual, con todas las desventajas ya explicadas, por lo que la utilización del nuevo sistema se hizo en forma de *reemplazo total*, es decir, en cuanto estuvo listo el primer prototipo de sistema, se empezó a utilizar y fue desechado el anterior.

-Captura de información, se almacenó toda la información existente en el nuevo sistema, contando con un número aproximado de 1000 registros, dicha captura se realizó en tres meses con la colaboración de capturistas prestadores de servicio social. Cabe mencionar que los trabajos ya terminados se encuentran entre todos los registros, y se distinguen por ya tener fecha de titulación y una clave para identificar físicamente su trabajo dentro de la jefatura.

-El sistema se instaló en dos computadoras dentro de la Jefatura de Actuaría y MAC, a disposición de los Secretarios Técnicos, Jefe de Programa, personal en general de la jefatura y capturistas de información.

La utilización del sistema se realiza cotidianamente, y ha dado buenos resultados tanto en el fácil acceso a los datos como en situaciones en las se ha requerido información acerca de algún aspecto relacionado con el registro de trabajos y/o titulación de los egresados.

El adiestramiento o entrenamiento ha sido sencillo ya que el sistema es muy amigable y su interfaz emite mensajes de error o ayuda para orientar y realizar una utilización del sistema óptima y libre de dudas.

Como parte del mantenimiento se hicieron pruebas de cada módulo del programa de manera independiente, se conjuntaron y se probó el sistema como una sola unidad, los resultados fueron satisfactorios. El proceso de observación y modificación para retroalimentar el sistema se ha llevado a cabo de forma iterativa, esto es, el sistema está abierto a mejoras y modificaciones, pues sabemos que no existe el software perfecto, ya sea por requerimientos no considerados en la fase de análisis o porque su ciclo de vida llega a su fin y el sistema empieza a ser obsoleto. Se consideraron estos aspectos y se trató de diseñar un sistema con una vida larga y retrasar por ende su obsolescencia. En cuanto a la programación se hizo en forma estructurada con los mismos estándares para todas las rutinas y subprogramas de que constó, además de contar con comentarios explicativos en partes confusas dentro del código (Ver apéndice).

3.3 DESCRIPCION GENERAL DEL SISTEMA

La fase de implantación se llevó a cabo en forma rápida debido a que la plataforma donde trabaja el sistema es ortodoxa y fácil de manejar, no necesita de bibliotecas externas ni software de apoyo, sólo un ambiente DOS y espacio suficiente en la PC. De hecho el proceso de pruebas, parte de la fase de mantenimiento, se realizó a la par con el proceso de creación de prototipos, ya que se creaba el siguiente prototipo en el proceso iterativo y se verificaban nuevamente los requerimientos y necesidades sobre la marcha, es decir, sobre el prototipo de sistema ya funcionando. De tal manera que el proceso de prueba realizado ya en el sistema final fue rápido, eficiente y confiable.

La capacitación se llevó a cabo con rapidez, debido principalmente a la estructura del sistema que permite ayuda en línea en cualquier parte, además de estar diseñado de manera tal que sea accesible para cualquier usuario, sin importar sus conocimientos en computación o programación desde principiante, medio y avanzado. A continuación se presenta una descripción detallada del sistema.

Para la implementación del sistema se esquematizan los módulos del programa en el diagrama descendente (Top-Down), representado por la Fig. 3.14

Desde el punto indicativo del sistema Operativo se debe teclear MAC seguido de la tecla ENTER. A continuación aparecerá un aviso parpadeante que verifica la validez y existencia de todos los archivos necesarios para la ejecución del programa.

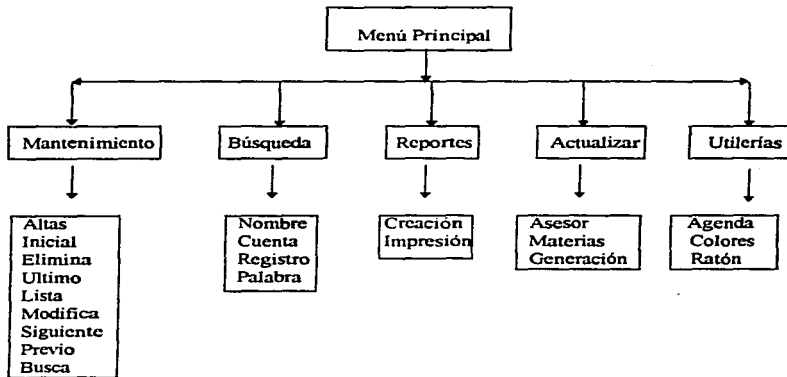


Fig. 3.14 DIAGRAMA DESCENDENTE DEL SISTEMA DE CONTROL DE TITULACION

Dentro del menú principal, Fig. 3.15 se pueden elegir las opciones de: mantenimiento, búsqueda, reportes, actualizaciones y utilerías. Para acceder a las distintas opciones del menú principal podemos:

- 1.- Navegar con flechas de cursor y teclas de movimiento de página, eligiendo la opción con la tecla [Enter].

- [→] Mueve la barra de elección una posición hacia la derecha
- [←] Mueve la barra de elección una posición hacia la izquierda
- [↑] Mueve la barra de elección una posición hacia arriba.
- [↓] Activa la opción actual, equivale a oprimir [Enter]

PgUp tiene el mismo efecto que [Enter]

PgDn tiene el mismo efecto que [Enter]

Inicio se posiciona en la primera opción del menú.

Fin se posiciona en la última opción del menú.

Nota: La barra de elección es fácilmente identificable porque está en video inverso.

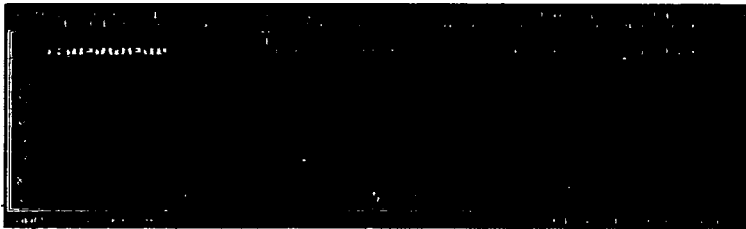


Fig 3.15 Menú Principal.

2.- Oprimiendo la primera letra de la opción deseada, es decir, si quisiéramos la opción de mantenimiento bastaría oprimir la letra "M" y automáticamente se activaría dicha opción. Dentro de la opción de MANTENIMIENTO, podemos nosotros capturar (almacenar) cualquier tipo de información para trabajo de titulación, ya sea Tesis, Tesina, Tesis Colectiva, Memoria de Desempeño Profesional, Examen Global de Conocimientos, Seminario de Titulación ó Informe de Práctica Profesional de Servicio a la Comunidad.

Dentro de la opción de BUSQUEDA, podemos realizar búsquedas de acuerdo a nuestros requerimientos; por nombre del alumno que está elaborando el trabajo, dicho nombre deberá empezar por apellido paterno, también por número de cuenta, por clave del registro del trabajo

en cuestión y una búsqueda más especializada por palabra clave principalmente para detectar trabajos parecidos o iguales.

Dentro de la opción de REPORTE, se podrán realizar reportes de acuerdo a las necesidades del usuario. El número de combinaciones para fabricar los reportes es muy extenso, sólo se eligen las condiciones apropiadas y se manda elaborar dicho reporte.

En el menú ACTUALIZAR, podremos eliminar, modificar o añadir información de los asesores, de materias o áreas de conocimiento y de generaciones.

En el menú UTILERIAS, podemos realizar consultas a la agenda personalizada así como también notas de acuerdo al plan calendarizado de actividades. También existe la posibilidad de cambiar los colores del sistema, existen varias opciones y es de acuerdo al gusto o a las características del monitor disponible. Se incluye en éste menú un pequeño Demo de cómo se implementa el ratón en este tipo de programas así como una breve explicación de su implementación y de su utilidad en otras aplicaciones.

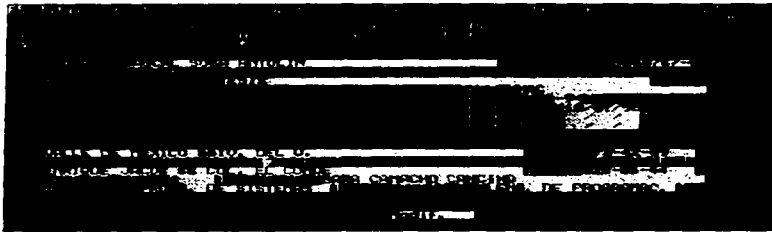


Fig. 3.16 Menú de mantenimiento

Al activar la opción de mantenimiento (Fig.3.16) tendremos acceso a las siguientes operaciones:

- 1.- Altas
- 2.- Inicial
- 3.- Elimina
- 4.- Ultimo
- 5.- Lista
- 6.- Modifica
- 7.- Siguiete
- 8.- Previo
- 9.- Busca
- 10.- Termina

Para acceder a las distintas acciones del menú tenemos dos opciones:

a).- Navegar con flechas de cursor y teclas de movimiento de página, eligiendo la opción con la tecla [Enter].

[→] Mueve la barra de elección una posición hacia la derecha

[←] Mueve la barra de elección una posición hacia la izquierda

[↑] Mueve la barra de elección una posición hacia la derecha.

PgUp tiene el mismo efecto que [Enter]

PgDn tiene el mismo efecto que [Enter]

Inicio se posiciona en la primera opción del menú.

Fin se posiciona en la última opción del menú.

Nota: La barra de elección es fácilmente identificable porque está en video inverso.

b).- Oprimiendo la primera letra de la opción deseada, es decir, si quisiéramos la opción de Altas bastaría oprimir la letra "A" y automáticamente se activaría dicha opción.

1.- Dentro de la opción de ALTAS, podemos añadir más registros a nuestra base de datos. El procedimiento consiste en varios pasos:

- Posicionarse en la opción de ALTAS y oprimir [Enter] o la letra "A". Ya dentro de Altas, el cursor aparecerá parpadeando esperando a que ingresemos el nombre del alumno empezando por el apellido paterno, apellido materno y nombre. Se oprime [Enter] para confirmar el ingreso, el sistema verifica automáticamente si no existe el nombre en la base de datos, si hay duplicación permite el alta y se visualiza el registro encontrado. Es importante, por tanto, ingresar el nombre completo sin faltas de ortografía ni omisiones de ningún tipo para que la validación de registros se realice en forma eficiente.

- Introducir la información solicitada a continuación: la cuenta del alumno, su opción de titulación, carrera, generación, fecha de registro de su trabajo, fecha de terminación, fecha de titulación, dirección y teléfono de casa, dirección y teléfono de trabajo, clave de registro, materia o área de

conocimiento 1 y materia o área de conocimiento 2. Así mismo se almacenan el título, objetivo general, capítulos, resumen, y sinodales, con la diferencia de que esta información se captura oprimiendo su correspondiente tecla de función (Fig. 3.17) como lo indica la barra de ayuda en la parte superior de la pantalla. esto es:

Tecla de activación	Recuadro de Captura
F5	Título
F6	Objetivo Gral.
F7	Capítulos
F8	Resumen
F9	Sinodales

A continuación algunas consideraciones generales acerca de la pantalla de captura, para el modo ALTAS y para Modificaciones :

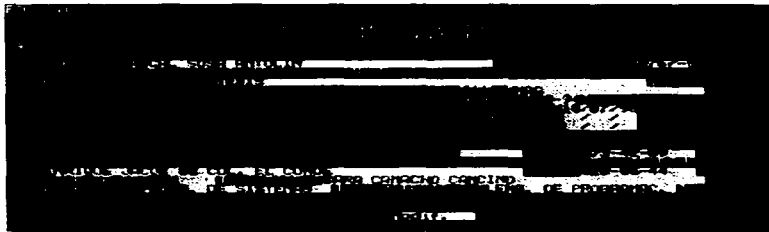


Fig. 3.17 Ejemplo de captura de Título (F5).

Mientras no estemos en posibilidad de editar nuestros datos, esto es, en modo de Altas o Modificaciones, sólo podremos visualizar la información de nuestro registro actual, tanto la información que está en la pantalla como la que podemos obtener mediante nuestras teclas de función ubicadas en la parte superior de la pantalla. Si quisieramos modificar el registro actual, tenemos que ubicarnos en la opción de modificaciones y oprimir la tecla <Enter> , a continuación ya tenemos acceso a todos nuestros campos y recuadros de pantalla con la diferencia de que además de visualizar podemos modificar y corregir la información almacenada.

Las validaciones son automáticas en cualquier tipo de datos y en cualquier campo o característica que se esté almacenando, y se realizarán como se explica a continuación:

- El nombre es clave única no se permitirá el acceso a una clave repetida.
- La cuenta sólo permitirá caracteres numéricos y también será única.
- Para la opción de titulación se desplegará una ventana con las opciones disponibles para que el usuario elija una y confirme con la tecla <Enter>.
- Para la captura de la carrera se despliega una ventana similar a la anterior con las diferentes carreras, se elige con teclas de cursor y se confirma con la tecla <Enter>.
- Para la captura de la generación se despliega una ventana similar a la anterior con las diferentes generaciones de acuerdo a la carrera elegida previamente, se elige con teclas de cursor y se confirma con la tecla <Enter>.
- Para la captura de la fecha de registro, de terminación y de titulación el formato es día/mes/año y la validación se hace en dos formas:
 - a).- la fecha de registro será anterior a la de terminación y ésta anterior a la de titulación, cualquier intento de ingresar fechas menores el sistema emitirá un mensaje y no permitirá el almacenamiento de dichos valores.
 - b).- Dependiendo de la opción de titulación se verificará sus fechas límites para terminación de trabajos, esto es, para la opción de Tesina si la diferencia entre las fechas de registro y de terminación excede los seis meses el sistema emitirá un aviso en forma acústica y visual. De la misma forma, si la opción es Memorias de Desempeño Profesional y la diferencia entre la fecha de registro y la fecha de terminación excede un año se emitirá un sonido de aviso así como un letrero parpadeante. La figura 3.18 ilustra este mensaje. Cabe señalar que la edición NO se inhabilita en este caso, sólo es un aviso al usuario para que se transmita a la autoridad competente y se tomen las medidas pertinentes.

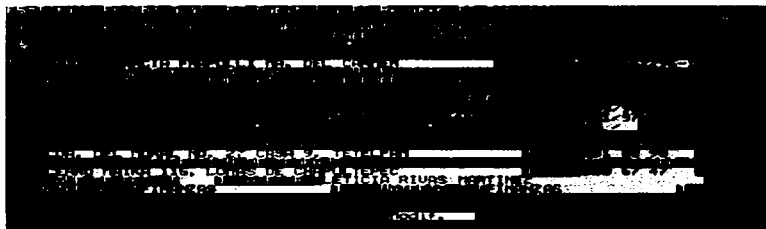


Fig. 3.18 Aviso de plazo excedido.

- Para la captura de la dirección de casa y dirección de trabajo se permite cualquier cadena de caracteres hasta un máximo de 80 espacios.
- Para los campos correspondientes a los teléfonos de casa y de trabajo se permite cualquier cadena de caracteres hasta un máximo de 12 espacios.
- Para la captura del asesor se despliega una ventana con todos los asesores disponibles, se elige uno y se confirma con la tecla <Enter> . En caso de que no exista el asesor en la ventana de despliegue se oprime la tecla <Esc> para cancelar , y posteriormente, se puede actualizar al asesor en el menú Actualizar submenú asesores como se explicará más adelante. La fig. 3.19 ilustra lo anterior.
- En las áreas de conocimiento o materias se despliega una ventana similar a la anterior con la información de todas las materias disponibles, se elige una con teclas de cursor y se confirma con la tecla <Enter>. En caso de que no exista la materia deseada o requerida se puede añadir de en el menú Actualizar submenú materias como se explicará a detalle más adelante.

Si la opción de titulación actual es la de Tesis Colectiva, aparecerá un cuadro en la parte superior de la pantalla para que se accese la información de los coautores del trabajo, la información que aparece, ilustrada en la figura 3.20, es la que diverge de los demás datos propios del trabajo, de tal manera que se capturan sólo los datos particulares, como el nombre, la cuenta, la carrera, así como sus direcciones y teléfonos de trabajo y de casa. Los demás datos se actualizan para todos los coautores automáticamente dado que son compartidos.

Terminado el proceso de edición, ya sea modificación de un registro existente o alta de un nuevo registro, aparecerá en pantalla un letrero de confirmación para almacenar en nuestra base de datos la información del registro actual que se acaba de capturar. Si contestamos negativamente a la pregunta, ningún cambio hecho se almacenará y el registro permanecerá como estaba al principio.

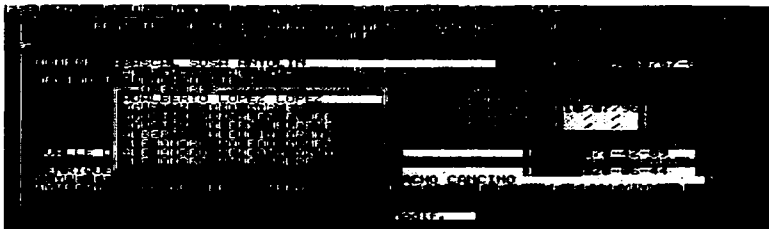


Fig. 3.19 Elección de Asesor.

Nota: Si necesitamos modificar uno o dos campos del registro actual, tal vez desearíamos evitar las cajas de despliegue de la opción de titulación, carrera, generación, asesor y materias, ya que estas cajas aparecen en forma automática al estar en modo de edición. Esto se logra eliminando dichas cajas de despliegue ANTES de entrar al modo de edición (Altas o Modificaciones), con las combinaciones de teclas que aparecen en la parte superior derecha de nuestra pantalla de captura, es decir, ALT-O habilita-inhabilita la ventana correspondiente a la Opción de titulación, ALT-C habilita-inhabilita la ventana para las Carreras, etc. Funcionan en forma de interruptor y su estado actual se puede observar en la parte inferior derecha de la pantalla: si se ve la letra 'O' está habilitada la ventana para las opciones de titulación, si se ve la letra 'C' está habilitada la ventana para las carreras, etc. Si está inhabilitada una opción se habilita oprimiendo la combinación de teclas correspondientes, si esta habilitada se inhabilita oprimiendo la combinación de teclas que corresponda, a esto se refiere cuando se habla de que funciona como interruptor.

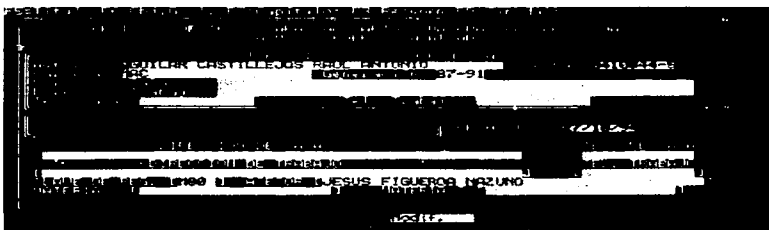


Fig. 3.20 Captura de información para Tesis Colectivas.

- 2.- Dentro de la opción INICIAL el sistema se posicionará en el primer registro de nuestra base de datos, la forma de activar dicha opción es mediante las teclas de cursor y confirmar con la tecla <ENTER> o con sólo oprimir su letra inicial "I".
- 3.- Dentro de la opción ELIMina tendremos la posibilidad de dar de baja registros, se activa dicha opción y se solicita mediante un cuadro de diálogo la confirmación de la baja.
- 4.- Para nuestra opción ULTIMO el sistema nos posiciona en el último registro de nuestra base de datos, esto nos permite, aunado a la opción INICIAL, viajar mas rápido dentro de toda nuestra información.
- 5.- La opción LISTA nos permite ver rápidamente la información relevante como nombre, número de cuenta, fecha de registro, fecha de terminación y fecha de titulación, de 20 registros a la vez, con la posibilidad de seguir "hojeando" nuestra información si confirmamos a la pregunta que aparece en la parte inferior derecha de nuestra pantalla, como lo ilustra la Fig. 3.21



Fig. 3.21 La opción LISTA para "hojear" información.

- 6.- LA opción MODIFica actualiza información existente como se explicó a detalle en la opción de ALTAS.
- 7.- La opción SIGuiente nos permite visualizar el siguiente registro de nuestra base de datos, sin olvidar que el orden se efectúa alfabéticamente por apellido paterno.
- 8.- La opción PREVIO nos permite visualizar el registro anterior de nuestra base de datos.

9.- Dentro de la opción BUSCA podemos encontrar el registro deseado tecleando sus apellidos y nombre, con la característica especial de buscar por "raíz" la palabra, es decir, si queremos buscar a "SUAREZ ANDRADE ALBERTO " sólo tendremos que teclear "SUAR" y se visualizará el primer registro que contenga las letras "SUAR". en otras palabras se posicionará en el primer "SUAREZ" que encuentre, si queremos ser más específicos tendremos que teclear mas caracteres como deseamos. Obviamente que si tecleamos apellidos y nombre(s) completos visualizaremos el registro deseado. En caso de no encontrarse el nombre buscado obtendremos un letrero de aviso, Fig 3.22, el cual nos indicará que el nombre no existe en nuestro archivo.

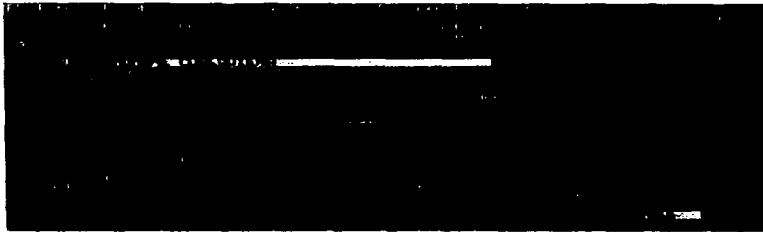


Fig. 3.22 Opción BUSCA de un nombre inexistente.

10.- Por último la opción TERMINA finaliza nuestra sesión de captura, y nos regresa al menú principal. Obtenemos el mismo efecto al oprimir la tecla <ESC>.

Desde cualquier punto podemos oprimir la tecla <F1> y obtendremos ayuda en pantalla sensible al contexto, esto es, ayuda de acuerdo a la parte del sistema que se esté ejecutando.

La segunda opción de nuestro menú principal es la de BUSQUEDA, aquí podremos realizar búsquedas por nombre del alumno, por número de cuenta, por clave del trabajo y una opción especializada para buscar palabras clave dentro de toda nuestra base de datos. Como se explicó líneas arriba resulta fácil el acceso a cualquiera de estas opciones; dentro del menú principal oprimir la letra "B" o posicionarse mediante teclas de movimiento de cursor en "BUSQUEDA" y oprimir la tecla <ENTER>. Aparecerá entonces la pantalla ilustrada en la fig. 3.23.

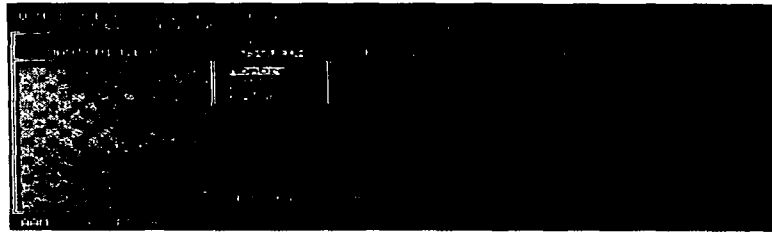


Fig. 3.23 Submenú de búsqueda

Si deseamos hacer una búsqueda por nombre, tenemos que activar la opción correspondiente oprimiendo la letra "N" u oprimiendo la tecla <ENTER>. Aparece una pantalla que nos solicita ingresemos el nombre del alumno a buscar, con el orden preestablecido de apellido paterno, materno y nombre. Si el nombre no es encontrado se despliega un mensaje de aviso y se pregunta si se quiere realizar otra búsqueda. Si el nombre se encuentra, aparece una pantalla con información básica sobre el alumno en cuestión, como su nombre, opción de titulación, carrera y el título de su trabajo, ésto se ilustra en la fig. 3.24. En la parte inferior de la pantalla aparecen teclas de activación para que se visualizen en pantalla el título=F5, el objetivo general=F6, etc.

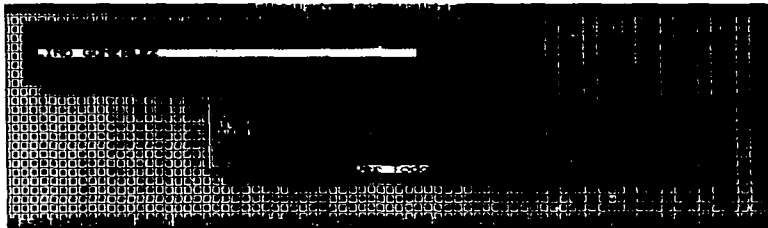


Fig. 3.24 Búsqueda por nombre.

En el recuadro donde aparece la información básica tenemos tres opciones: "Imprimir", "Ver Todo", y "Salir", las cuales siguen el mismo formato establecido para menús, esto es, si queremos activar una opción tenemos que teclear su primer letra o posicionarnos con teclas de movimiento y oprimir la tecla <ENTER> en la opción deseada.

Si deseamos imprimir la información del registro actual activamos la opción "Imprimir", el cual genera un reporte con todos los datos del alumno, el formato de dicho reporte se puede ver líneas arriba en el diseño de salidas y reportes (Figura 3.6). En caso de que la impresora no esté conectada o encendida se genera el mensaje "Impresora no lista", con la posibilidad de corregir el error y reintentar la impresión o abortar la operación para imprimir más tarde o ejecutar otra actividad.

Con la opción "Ver Todo" tenemos posibilidad de ver en pantalla TODA la información concerniente al registro actual, desde su nombre, cuenta, carrera, asesor, sinodales, direcciones, teléfonos, etc.

Por tanto, podemos buscar un registro específico y ver la información relevante en la primera pantalla, pero si lo deseamos podemos ver toda la información y mandarla a impresión o sólo visualizarla en pantalla.

La opción "Salir" nos permite realizar otra búsqueda si lo deseamos abandonando la actual o ir al submenú de búsqueda.

Para realizar una búsqueda por número de cuenta, regresamos al submenú de búsqueda y activamos la opción de "Cuenta", aparece una pantalla solicitándonos ingresar el número de cuenta. Si no se encuentra se despliega un mensaje con la opción de volver a realizar una búsqueda o regresar al menú anterior. Si se encuentra la cuenta buscada entonces aparece una pantalla con información básica del alumno (Fig. 3.25) como nombre, opción de titulación, carrera, y título de su trabajo.

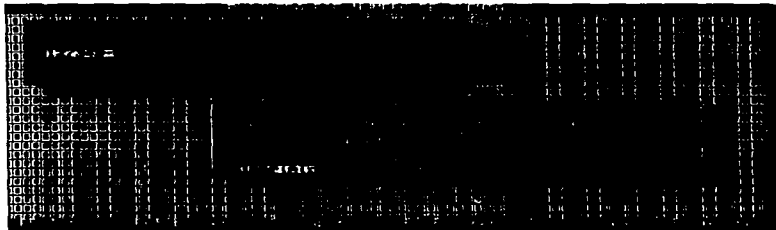


Fig. 3.25 Búsqueda por número de cuenta.

La estructura es parecida a la pantalla de búsqueda por nombre, tenemos las opciones de "Imprimir", "Ver Todo" y "Salir". Podemos visualizar toda la información por pantalla o mandarla a impresión así como regresar al menú anterior.



Fig. 3.26 Búsqueda por clave de trabajo o registro.

Para realizar una búsqueda por la clave del trabajo o clave de registro, activamos la opción "Registro" dentro del submenú "Búsqueda", aparece un recuadro para que se ingrese la clave del registro a buscar. Si no se encuentra un mensaje notifica el hecho y pregunta si se desea reintentar otra búsqueda o regresar al menú anterior. Si se encuentra el registro con la clave de trabajo accesada se despliega una pantalla análoga a las anteriores (Fig. 3.26), la cual contiene información básica y preliminar del trabajo encontrado, como nombre, opción de titulación, carrera, y título del trabajo. La estructura es parecida a la pantalla de búsqueda por número de cuenta, tenemos las opciones de "Imprimir", "Ver Todo" y "Salir". Podemos visualizar toda la información por pantalla o mandarla a impresión así como regresar al menú anterior.

La última opción de éste submenú es la búsqueda por palabra clave, accedamos oprimiendo la tecla "P" o posicionarnos en "Palabra" y confirmar con la tecla <ENTER>. Aparece entonces una pantalla en la cual podremos elegir el campo de búsqueda, como el título, objetivo general, capítulos, resumen, dirección de casa y dirección de trabajo. Esta opción es de mucha utilidad para revisar el número de trabajos que tienen palabras comunes o que pertenecen a la misma área conocimiento. El procedimiento es sencillo y se explica a continuación:

- Moverse con teclas de cursor y posicionarse en el campo deseado y confirmar con <ENTER>, aparece entonces un signo "√" el cual nos indica que la búsqueda a continuación se realizará en ese campo. Fig. 3.27, el número de elecciones es a gusto o necesidades del usuario, es decir, se puede escoger sólo el título o el resumen, o cualquier combinación entre los posibles campos.

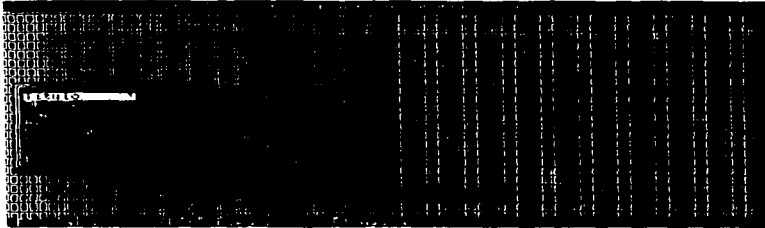


Fig. 3.27 Búsqueda en el título

- Después de definir en qué campo(s) realizaremos la búsqueda, confirmamos oprimiendo <ENTER> en "Buscar"

Aparece un cuadro solicitando las palabras claves que buscará el sistema (Fig. 3.28) en los campos que se definieron con anterioridad, se pueden escribir raíces de palabras, palabras completas o secuencia de palabras, el sistema buscará todas las ocurrencias de las palabras tecleadas y las registrará en cuadro de elección como lo ilustra la fig. 3.29. Si nosotros deseamos conocer todos los trabajos de hablan de sistemas y sistemas de información dentro del título y objetivo general, tenemos que elegir estos dos campos con la tecla <ENTER>, enseguida activar la búsqueda oprimiendo la tecla <ENTER> en "Buscar", en las palabras clave tecleamos "SISTEM", "SISTEMAS", Y "SISTEMAS DE IN". Todos los registros que el sistema encuentre con estas cadenas de caracteres en su título y su objetivo aparecen en la caja de elección arriba mencionada.

Dentro de la caja de elección aparecen los nombres de los autores que reúnen las características anteriores, ordenados en forma alfabética por apellido paterno. Para movernos dentro de ésta caja podemos elegir entre varias opciones; navegar con teclas de movimiento de cursor, con teclas de



Fig. 3.28 Buscando en título y objetivo general.

avance o retroceso de página, las combinaciones de teclas Ctrl-Inicio y Ctrl-fin nos colocan en la primera posición de la caja en pantalla y en la última posición respectivamente, las combinaciones de teclas Ctrl-RetrocesoPágina nos coloca en la última posición de los registros existentes, de la misma forma la combinación Ctrl-AvancePágina nos posiciona en el último registro existente.



Fig. 3.29 Registros que cumplen con el requisito de búsqueda

Otra manera de moverse dentro de la caja de registros es oprimir la primera letra del apellido paterno, la barra de posición se colocará en el primer registro que contenga la letra tecleada. Es decir, si se oprime la letra "B" aparece el apellido "Báez" que es el primero con la letra "B". Esto si buscamos a alguna persona en específico de otra manera se utilizan las opciones anteriormente descritas.

Después de encontrar la persona buscada, o estando en el registro actual si no se busca a alguien específico, se oprime la tecla <ENTER> para observar la información relevante (Fig. 3.30) como el nombre, opción de titulación, carrera y título del trabajo.

Tenemos en una barra de elección con tres distintas opciones para "Imprimir", "Ver Todo" o ver "Más" registros, nos movemos con teclas de cursor y elegimos con la tecla <ENTER>.

Con la opción "Imprimir" podemos mandar a impresión de acuerdo al formato establecido en el diseño de salidas y reporte, esta impresión contiene TODOS los datos del registro actual, si no se desea impresión se puede ver en pantalla TODA la información eligiendo la opción "Ver Todo" cuyo formato se puede observar en el diseño de reportes y salidas (Figura 3.5).

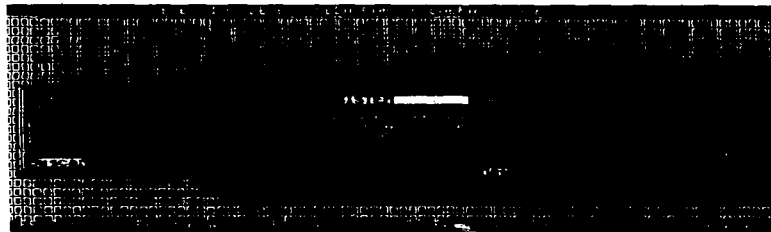


Fig. 3.30 Ver información del registro actual.

Para realizar otra combinación de búsqueda salimos oprimiendo la tecla <ESC> hasta llegar al menú para elegir campo y posteriormente las palabras clave con el mismo procedimiento arriba descrito. Dentro del menú principal tenemos el submenú para reportes, seleccionando con la tecla <ENTER> podemos escoger entre "Creación" para crear un reporte personalizado e "Impresión" para mandar imprimir el reporte previamente creado.

Primeramente debemos crear nuestro reporte, por tanto activamos la opción "Creación" y aparecerá la pantalla ilustrada en la fig. 3.31, mediante ésta estructura para crear reportes, ya sean por pantalla o impresos, se pueden crear múltiples posibilidades. Se puede elegir carrera, materia asesor, fecha de registro, fecha de terminación, fecha de titulación, generación, opción de titulación y sinodales.



Fig. 3.31 Pantalla de creación de reportes.

Elegir sólo una posibilidad, dos, tres o varias de acuerdo a las necesidades de información o impresión, una de las problemáticas detectadas y de la cual se habló en el capítulo uno es la de NO contar con información de diferentes características, como se describe a continuación:

- ◆ Número de registros por carrera o programa.
- ◆ Número de titulados por carrera o programa
- ◆ Número de titulados por generación.
- ◆ Número de registros por generación.
- ◆ Porcentaje de titulados por generación.
- ◆ Porcentaje de titulados por cada opción de titulación.
- ◆ Número de registros por asesor.
- ◆ Número de titulados por asesor.
- ◆ Periodos de tiempo comprendidos entre registro y examen profesional.
- ◆ Número de registros por año.
- ◆ Número de registros por semestre.
- ◆ Número de registros por mes.
- ◆ Número de veces que un profesor ha sido sinodal de examen profesional.
- ◆ Número de tesis cuyo tema es afín a alguno en específico.
- ◆ Títulos y/u objetivos de trabajos afines a algún tema en específico

Esto como lo más relevante solicitado y no obtenido anteriormente, ahora se pueden elaborar reportes que generen esta información y casi cualquier tipo de combinación usual o poco usual que se pueda obtener con los datos existentes. Por ejemplo, podríamos necesitar un reporte de los egresados de la

carrera de M.A.C. que registraron la opción de titulación Memorias de Desempeño Profesional y están por cumplir un año de registro, con lo cual se revisa los registros próximos a vencerse. Generar este reporte es sencillo y el procedimiento es el siguiente:

- Activamos la opción de "Carrera" oprimiendo la tecla <ENTER> aparece un recuadro en el que podemos escoger la carrera de M.A.C. utilizando teclas de cursor y confirmando con la tecla <ENTER>. En la parte derecha de la pantalla se visualiza el inicio de la construcción del reporte para mayor control del usuario.
- Nos posicionamos en la opción de "Fecha de registro" y la activamos, aparece un recuadro, ilustrado por la Fig. 3.32, en el que escogemos el tipo de conector que necesitamos, en nuestro caso necesitamos todos los alumnos que hayan registrado su trabajo entre el 1 y el 30 de septiembre de 1995, ésto con el objeto de obtener los registros de ese mes y corroborar cuales y cuantos registros van a vencer en septiembre de 1996.

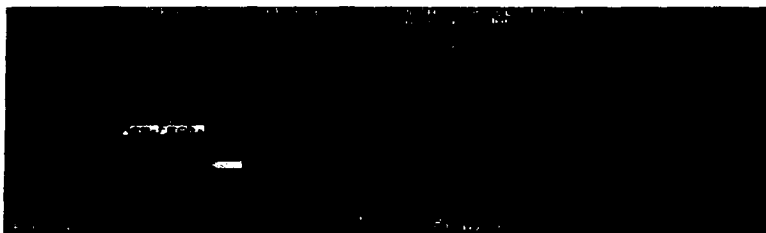


Fig. 3.32 Ejemplo para creación de reporte.

- Por último activamos la "Opción de Titulación", aparece un recuadro con las opciones de titulación vigentes, elegimos Memoria de Desempeño Profesional y de ésta forma obtendremos el "filtro" requerido, en la parte derecha de la pantalla se aprecia el reporte que a continuación crearemos de acuerdo a las condiciones ya fijadas.



Fig. 3.33 Generando el reporte creado.

- Se activa la opción de "Reporte" y aparece un letrero parpadeante (Fig. 3.33) comunicando que el reporte se está generando, cuando éste termina tenemos las opciones de ir al submenú de impresión o ver por pantalla la información del reporte generado.
- Para ver por pantalla nuestro reporte activamos la opción "Pantalla", de esta manera obtenemos la visualización del registro o registros que cumplieron con nuestra condición previamente creada, ésto se aprecia en la fig. 3.34. Podemos acceder a toda la información del registro dentro de nuestra pantalla, desde el nombre, cuenta, carrera, etc. Navegamos con teclas de cursor y se va rotando la información, en el caso del título, objetivo general, capítulos, resumen y sinodales se activa su contenido por medio de la tecla <ENTER>

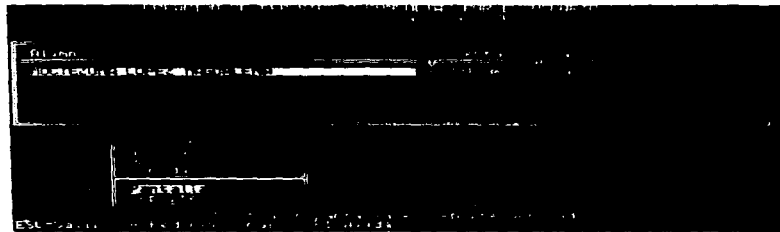


Fig. 3.34 Pantalla del reporte generado.

En la parte superior de la pantalla se aprecia el número total de registros encontrados así como el registro actual de acuerdo a la barra de selección, podemos interactuar con la pantalla de reportes y la

de visualización oprimiendo <ESC> seguido de la creación de otro reporte o simplemente añadiendo una característica más al ya existente. De esta forma se satisfacen las necesidades de información en pantalla del reporte generado, mas adelante se verá la forma de imprimir toda nuestra información.

Para mandar a impresión el reporte previamente generado activamos la opción "Impresión" del submenú "Reportes", se despliega un submenú con las distintas posibilidades de impresión como son; asesor, registro, titulación y generales (Fig. 3.35). Los formatos de dichos reportes se describen en el diseño de salidas y reportes, y es de acuerdo a las necesidades de información que se tengan para poder cambiar de tipo de reporte. Es decir, además de poder elegir entre una gran gamma de combinaciones para generar reporte, una vez generado éste, también se puede elegir el tipo de reporte impreso que se quiera o se necesite. Para "Asesor" se imprimen, a diferencia de los otros, todos los datos del asesor elegido sin posibilidad de crear reporte. Para "Registro" se imprimen predominantemente los datos propios del registro de trabajo de titulación (figura 3.2), como son; el nombre, la carrera, la opción de titulación, nombre de asesor, fechas de registro y terminación, etc. Para "Titulación" (figura 3.3) se imprimen datos propios de titulación como; nombre, fecha de registro, fecha de titulación, nombre de asesor, nombre de sinodales, clave de trabajo o registro, etc. Para "Generales" (figura 3.4) se imprimen datos como; nombre, dirección de su casa, de su trabajo, teléfonos de casa y trabajo, título, cuenta, carrera, etc.

También existe la posibilidad de escribir un encabezado acorde al tipo de reporte que se va imprimir, después de elegir reporte el sistema pregunta por el intervalo de páginas que se van a imprimir, esto permite a su vez una posibilidad más para combinar reportes, en caso de que haya necesidad, por ejemplo, de imprimir nuevamente un reporte pero sólo ciertas páginas, podemos hacerlo determinando dichas páginas. Después de especificar las páginas de impresión, el sistema pregunta por el encabezado que llevará el reporte, se puede incluir un texto de hasta 80 caracteres

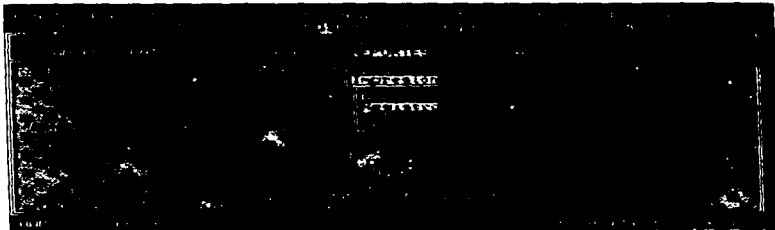


Fig 3.35 Submenú para impresión de reportes.

En caso de que la impresora no esté lista, encendida o conectada, el sistema responde con un aviso visual y acústico notificando que la impresora NO está lista, en éste punto se puede abortar la operación o corregir el error y reintentar la impresión. Si la impresora falla durante la impresión se presenta el mismo mensaje con las mismas posibilidades de reintentar o abortar.

En el menú "Actualizar" existen tres opciones: "Asesor", "Materia" y "Generación". Principalmente para dar mantenimiento y actualización a la información referente a los asesores, materias o áreas de conocimiento y a las generaciones. En la opción de "Asesor" (Fig. 3.36) se puede eliminar, agregar y modificar información de registro de asesor ya existente, la barra de estado en la parte inferior de la pantalla nos indica la forma de realizar estas operaciones.

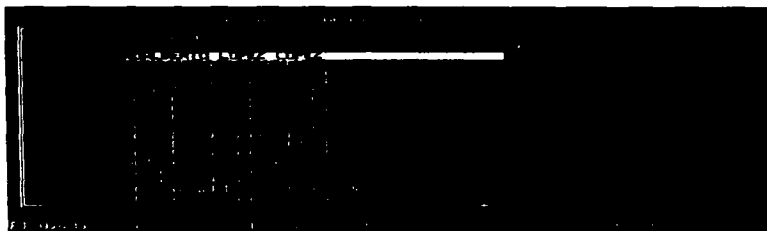


Fig. 3.36 Actualización para asesores

Dentro de la información relevante para asesores está su nombre, clave de asesor, grado o título académico, dirección de domicilio, dirección de trabajo, teléfono de casa y domicilio, y sus áreas de conocimiento. Con teclas de movimiento de cursor y desplazamiento de página se puede navegar por toda la información, también existe la posibilidad de teclear una letra, estando en el nombre de asesor, de esta forma la barra de selección se posiciona en la primera ocurrencia que encuentre con ese carácter, se pueden teclear hasta 14 caracteres para una búsqueda más flexible.

Cabe señalar que esta información es la que aparece en los recuadros de captura, creación e impresión de reportes, por lo que es importante tenerla actualizada para generar reportes y/o capturas claros, confiables y sin omisiones de ningún tipo.

Para el menú de "Materias" la estructura de la pantalla es la misma, la forma de editar, corregir y eliminar información es muy parecida, sólo que en este caso se trabaja con los datos de materias o áreas de conocimiento. Se puede visualizar y editar para las materias; su nombre, clave de materia y

carrera. De la misma forma podemos buscar la materia deseada (oprimiendo el primer caracter), navegar por la información con teclas de cursor y desplazamiento de página, etc.

Dentro de el menú de "Generación" la estructura de la pantalla es la misma, la forma de editar, corregir y eliminar información es muy parecida, pero ahora se trabaja con los datos de las generaciones de egresados. Se puede visualizar y/o editar para las generaciones; generación, clave de generación, carrera y clave de carrera. De la misma forma podemos buscar la generación deseada (oprimiendo el primer caracter), navegar por la información con teclas de cursor y desplazamiento de página, etc. Toda esta información aparece en recuadros de captura o visualización, por tanto es conveniente estar actualizando constantemente dicha información. Esto se debe hacer para asesores, materias y generaciones.

El tener actualizados estos datos proporcionará información clara, verídica y oportuna en cualquier formato de consulta, reporte o captura del sistema.

En el menú "Utilerías" existen tres opciones; agenda, colores y ratón. Para el submenú de "Agenda" se observa una pantalla (Fig. 3.37) con un calendario en la parte izquierda, aparece iluminado el día de la fecha actual. La barra de ayuda en la parte inferior de la pantalla nos indica a *grosso modo* lo que se tiene que hacer, podemos oprimir la tecla <F1> para obtener ayuda (como en cualquier parte del sistema) y poder navegar dentro el calendario. Para ésto se utilizan las teclas de cursor y desplazamiento de página con los siguientes efectos:

Septiembre, 1996						
Dom	Lun	Mar	Mie	Jue	Wie	Sáb
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Fig. 3.37 Agenda para actividades.

La tecla (←) mueve nuestra barra de selección al día anterior.

La tecla (→) mueve nuestra barra de selección al día siguiente.

La tecla (↓) mueve nuestra barra de selección a la semana siguiente.

La tecla (↑) mueve nuestra barra de selección a la semana anterior.
 La tecla PgUp mueve nuestra barra de selección al año anterior.
 La tecla PgDn mueve nuestra barra de selección al año siguiente.
 La tecla Inicio mueve nuestra barra de selección al mes anterior.
 La tecla Fin mueve nuestra barra de selección al mes siguiente.
 Asimismo se puede elegir un día cualquiera oprimiendo la tecla <F4> y accedendo el día, mes y año de la fecha deseada.
 Después de posicionar la barra de selección en el día deseado se oprime la tecla <ENTER> y aparece una ventana para editar actividades del día seleccionado (Fig. 3.38), el tamaño del texto no se limita al tamaño de la ventana, es decir, se pueden escribir las actividades que se deseen y la pantalla irá rotando para aceptar las nuevas líneas. De esta forma se tiene un control preciso de las actividades del usuario en cuanto a su vida personal, de trabajo, etc.

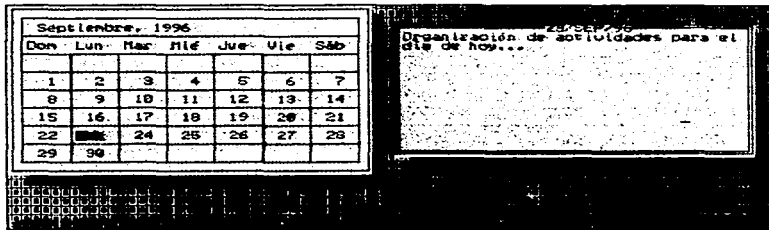


Fig. 3.38 Escribiendo actividades en la agenda.

El submenú "Colores" es una opción que permite elegir el color que el sistema desplegará, independientemente de cuestiones como el tipo de monitor que se utiliza y el gusto del usuario o capturista. Si el monitor en uso es monocromático el sistema automáticamente detecta el tipo de monitor y despliega blanco, negro y blanco resaltado. Si el monitor es a color, se despliega por omisión la combinación de azul para el fondo y blanco para las letras y otros caracteres de escritura. El formato para las combinaciones es el primer color para el fondo y el segundo para letras y caracteres de escritura. Las opciones o combinaciones de colores posibles, ilustradas en la figura 3.39, son las siguientes:



Fig. 3.39 Elección de colores.

Ngo/Bco; negro para el fondo y blanco para letras, los mensajes de advertencia y error aparecen con blanco resaltado. Esta opción está disponible aunque el monitor sea a color.

Mag/Ama; magenta para el fondo y amarillo para letras, los mensajes de advertencia y error aparecen en rojo y algunas ventanas extra en color cian o café, esta combinación es muy suave y agradable a la vista.

Caf/Ver; café para el fondo y verde para letras, los mensajes de advertencia y error aparecen en rojo y algunas ventanas extra en color cian o magenta, esta combinación es muy suave y aunque no posee mucha nitidez para la captura es una combinación poco usual.

Cyan/Azu; cian para el fondo y azul para letras, los mensajes de advertencia y error aparecen en rojo y algunas ventanas extra en color cian o café, esta combinación es muy agradable a la vista y posee bastante nitidez para la captura ya que no cansa demasiado.

Rojo/Ama; rojo para el fondo y amarillo para letras, los mensajes de advertencia y error aparecen en rojo y algunas ventanas extra en color cian o café, esta combinación es poco usual pero agradable a la vista.

Azu/Bco; azul para el fondo y blanco para letras, los mensajes de advertencia y error aparecen en rojo y algunas ventanas extra en color cian o magenta, esta combinación es muy suave y agradable a la vista, es la combinación por omisión y es ideal para la captura de acuerdo a los estándares para el uso de colores en la teoría de sistemas de información.

La última opción del submenú es "Ratón" (Fig. 3.40), aquí se implementó el uso del ratón en el menú principal con fines demostrativos y didácticos. Una de las ventajas que posee el lenguaje Clipper 5.2 es la de poder tener una interfaz con otros lenguajes de programación, de esta forma se pueden cubrir limitantes que tiene el lenguaje con bibliotecas y funciones hechas en lenguajes externos, con todo el poder del lenguaje C y el control del hardware que proporciona el lenguaje Ensamblador las características y posibilidades de Clipper crecen considerablemente. Se puede crear

interfases gráficas, agregar controladores de ratón al sistema, importar y exportar datos de hojas de cálculo como Lotus, y muchas mejoras más.

La compatibilidad en estos casos es muy importante, Clipper 5.2 fue desarrollado en Microsoft C 5.1 por lo que para lograr una interfaz completa con C, se necesita crear los archivos objeto con el compilador de Microsoft para C. Para el lenguaje Ensamblador se necesita compilar con Microsoft Assembler versión 5.0 (MASM) o posterior, ya obtenidos los archivos objeto se ligan dentro de Clipper como si se tratara de varios programas del mismo lenguaje, la diferencia estriba en que son módulos objeto de distintos lenguajes de programación.

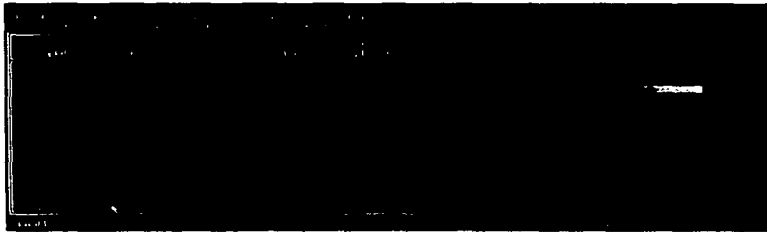


Fig. 3.40 Implementación de ratón en programas Clipper.

El procedimiento para esta implementación consistió en obtener funciones externas hechas en lenguaje Ensamblador (MASM 6.11) para tres importantes acciones:

- Detectar si el ratón está instalado.
- Verificar qué botón oprimió el usuario.
- Devolver las coordenadas (en modo texto, i.e., 24x80) en las que fue oprimida cierta tecla del ratón.

Estas funciones se obtuvieron del libro "Interfaces de Clipper" de John Mueller págs. 460-466, se compilaron en MASM 6.11 y el archivo objeto resultante se ligó con todo el sistema dentro de Clipper 5.2. Combinando el manejo de estas funciones externas así como los parámetros que

devuelve y recibe, con los objetos de Clipper Tbrowse y Tbcolumn se logró un manejo adecuado para implementar el ratón con éxito. Aunque Clipper no es puramente un lenguaje orientado a objetos se pueden mezclar características de programación estructurada con los diferentes objetos que posee como: ERROR, GET, TBROWSE y TBCOLUMN.

Se interactuó con los objetos de Clipper para crear menús y a la vez recibieran y procesaran los parámetros devueltos por las funciones de Ensamblador. De esta manera se pudo evaluar las coordenadas del ratón y el botón oprimido para ejecutar las acciones como si las órdenes fueran mandadas desde el teclado. Obviamente se trata de una implementación rudimentaria, pero es un excelente comienzo para mejoras que eleven la calidad de la interfaz entre el usuario y la computadora.

FALTA PAGINA

No. 75 a la 80

BIBLIOGRAFIA

García Badell, José Javier
Clipper 5.01
Ed. McGraw-Hill, México, 1992.

Kendall y Kendall
Análisis y diseño de sistemas
Ed. Prentice Hall Hispanoamericana, S.A., México, 1991.

Mueller, John
Interfaces de Clipper con C, Assembly y hojas de cálculo
Ed. McGraw-Hill, México, 1995

Ramalho, José A.
Clipper 5.2 avanzado
Ed. MacGraw-Hill, México, 1994

Senn, James A.
Análisis y Diseño de sistemas de información
Ed. McGraw-Hill, México, 1992

Sommerville, Ian
Ingeniería de software
Ed. Addison-Wesley Iberoamericana, México, 1988.

Straley, Stephen J.
Clipper 5.0
Ed. Grupo Noriega Editores, México, 1993

Peter, Norton
Introducción a la Computación
Ed. McGraw-Hill, México, 1995

Manual de Organización
Escuela Nacional de Estudios Profesionales Acatlán



APENDICE



```

Programa 1
/* programa para crear los menus
descendentes
para el sistema de tesis de la ENEP
Acáñán
AUTOR: ARMANDO AMADOR
MOLINA
fecha_inicio: 5-abr-1995
fecha_UK_Act: 8-jul-1996
*/
    
```

```

procedure main()
*****
#include "inkey.ch"
#include "AAtunca.CH"
#include "AAteclas.CH"
#include "AAvsior.CH"

local nOpc := 1
#xcommand READ =>
ReadModal(GetList, nT);
GetList:= { }
memvar GetList
nT = 90 //variable en segs. para
protector de pantalla !
// es en segundos == 1.5
min.
setcancol(f.)
    
```

IF ISCOLOR() //PARA DETERMINAR COLORES EN OTRO TIPO DE MONITOR

```

//variables globales para colores
cColor1="w/b+,R/w" //boo-
azul,rojo-boo (menu prin)
cColor2="b/g+g,gr/w" //cyan-
caf.,ama-boo (reportes)
cColor3="gr+/rb,R/w" //ama-
maganta,rojo-boo
cColor4="b/bg,b/w" //azul-
cyan,boo+ azul
cColor5="gr+/r,r/w" //ama-
rojo,rojo-boo
cColor6="n/g,b/w" //negro-
verde,azul-boo

//lineas
cColLin1 = "gr+/r" //ama-rojo
cColLin2 = "w+/b" //boo+
azul
cColLin3 = "w+/r" //boo+
rojo error y advertencia
cColLin4 = "r/w" //rojo /boo
    
```

```

//mensajes parpadeando
cColPar1="w/w" //boo/rojo
cColPar2="w/b+" //boo/azul
ELSE
cColor1="w/n,w/" //menú principal
cColor2="n,w,w+m"
cColor3="w/n,w/w"
cColor4="w/n,w/w"
cColor5="w/n,w/w"
cColor6="n,w,w+w/n"
cColLin1 = "w/n"
cColLin2 = "w+/u"
cColLin3 = "w+m"
cColLin4 = "w/w"
cColPar1="w/w"
cColPar2="n/w/w"
    
```

```

ENDIF
ENTRADA()
//CAJAS,AFERTURAS, ETC

!Camb_col=.f.
set message to 21 center
set wrap on
setcolor(cColor1)
set score off
set date british
cls
do part_1
set key pF1 to ayuda

while nOpc !=0 //ciclo

if !Camb_col
cls
!Camb_col:=.f.
do part_1
endif
setcolor(cColor1)
SET KEY pABAJA TO abajo
SET KEY pDERECHA TO
SET KEY pIZQUIERDA TO
@ 5,04 prompt
"MANTENIMIENTO";
message "Mantenimiento
para informacion de titulacion "
@ 5,23 prompt "BUSQUEDA";
message "Buses para
claves únicas o por palabras"
@ 5,36 prompt "REPORTES";
message "Creación y
generación de reportes"
@ 5,49 prompt "ACTUALIZAR";
message "Actualización
a informacion relevante"
@ 5,64 prompt "UTILERIAS";
message "Servicios
varios"
menu to nOpc

SET KEY pABAJA TO

cScreen := saveScreen(0,0,maxXrow(),m
axcol()) //seria igual con: save
screen to cScreen

*****
*****
if lastkey()=pESC

SETCOLOR(cColLin3)
m_sombra(11,10,13,65..L)
@ 12,11 SAY "QUIERES SALIR
DEL SISTEMA "TITULACION
ACT. MAC* 7'
OPC=2
@ 13,32 prompt "SI"
@ 13,36 prompt "NO"
MENU TO OPC
ops:=if(opc=0,2,opc)
do case
case opc=1
detpos(21,00)
close all
SET COLOR TO
CLS
devout("Todos los archivos
cerrados.. ")
INKEY(.5)
    
```

```

RETURN
case opc=2
setcolor(cColor1)
RESTORE SCREEN FROM
cScreen
nOpc=1
loop
endcase
endif

*****

SET KEY pDERECHA TO derecha
//activa set keys
SET KEY pIZQUIERDA TO
izquierda

do case
case nOpc=1
do manten
case nOpc=2
do busca
case nOpc=3
do report1
case nOpc=4
do actualiza
case nOpc=5
do utileria

endcase
restore screen from cScreen
setcolor(cColor1)

enddo
return

*****
*****
PROCEDURE DERECHA
SET KEY pDERECHA TO
//desactivacion
*Para pasar al menu de la derecha
con la flecha
keyboard
chr(pESC)+chr(pDERECHA)+CHR(
pENTER)

return

*****
*****
PROCEDURE IZQUIERDA
SET KEY pIZQUIERDA TO
//desactivacion

*Para pasar al menu de la izquierda
con la flecha
keyboard
chr(pESC)+chr(pIZQUIERDA)+CHR(
pENTER)

RETURN

*****
*****
PROCEDURE manten
local nOpc:=1
private cScreen1

setcolor(cColor1)
m_sombra(6,3,8,18,"FIE*4IE")
/n/n

while nOpc !=0
@ 7,7 prompt "Captura" message
"Mantenimiento para informacion de
titulacion "
    
```

```

menu to nOpc

cScreen1:=savescreeen(0,0
,maxxow(),maxcol())
do case
case nOpc=1
SET KEY pDERECHA
TO
SET KEY
pIZQUIERDA TO
do pant_tes //super.prg
SET KEY pDERECHA
TO derecha //activa set keys
SET KEY
pIZQUIERDA TO izquierda
//setcolor(cColor1)
restore screen from
cScreen1
otherwise
//nada
endcase
enddo
restore screen from cScreen
setcolor(cColor1)
return
*****
PROCEDURE busca
local nOpc:=1
private cScreen1
setcolor(cColor1)
m_sombra(6,21,1,1,33,ÉIE%4ÉÉ%)
//,n'n)
while nOpc !=0
@ 7,23 prompt "Nombre"
message " Busqueda por nombre en
la base de datos "
@ 8,23 prompt "Cuenta" message
" Busca por número de cuenta "
@ 9,23 prompt "Registro" message
"Busca por clave de registro"
@ 10,23 prompt "Palabra" message
"Busca por palabra clave trabajos
repetidos"
menu to nOpc

cScreen1:=savescreeen(0,0,maxxow(),
maxcol())
do case
case nOpc=1
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
do Bus_nom
SET KEY pDERECHA TO derecha
//activa set keys
SET KEY pIZQUIERDA TO
izquierda
case nOpc=2
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
do Bus_cue
SET KEY pDERECHA TO derecha
//activa set keys
SET KEY pIZQUIERDA TO
izquierda
case nOpc=3
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
do Bus_clv

```

```

SET KEY pDERECHA TO derecha
//activa set keys
SET KEY pIZQUIERDA TO
izquierda
case nOpc=4
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
do Bus_pal
SET KEY pDERECHA TO
derecha //activa set keys
SET KEY pIZQUIERDA TO
izquierda
endcase
enddo
restore screen from cScreen
setcolor(cColor1)
return
*****
PROCEDURE ABAJO
keyboard chr(pENTER)
return
*****
PROCEDURE utileria
local nOpc:=1
local ICambio
private cScreen2
setcolor(cColor1)
m_sombra(6,62,1,0,74,ÉIE%4ÉÉ%)
while nOpc !=0
@ 7,64 prompt "Agenda" message
" Activa calendario para actividades
"
@ 8,64 prompt "Colores" message
" Cambia colores del sistema "
@ 9,64 prompt "Raten" message "
Demostración de raten "
menu to nOpc
cScreen2:=savescreeen(0,0,maxxow(),
maxcol())
do case
case nOpc=1
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
agenda()
SET KEY pDERECHA TO
derecha //activa set keys
SET KEY pIZQUIERDA TO
izquierda
case nOpc=2
ICambio:=camb_col()
if ICambio
ICamb_col=.t.
return
endif
restore screen from cScreen2
setcolor(cColor1)
case nOpc=3
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
demRat()
SET KEY pDERECHA TO
derecha //activa set keys

```

```

SET KEY pIZQUIERDA TO
izquierda
endcase
enddo
restore screen from cScreen2
setcolor(cColor1)
return
*****
PROCEDURE actualiza
local nOpc:=1
private cScreen1
setcolor(cColor1)
m_sombra(6,48,10,60,ÉIE%4ÉÉ%)
while nOpc !=0
cCol_menu=3
@ 7,50 prompt "Asesor" message "
Actualiza informacion de asesores
"
@ 8,50 prompt "Materias" message
"Actualiza datos de temas/materias"
@ 9,50 prompt "Generacion"
message "Actualiza datos de
generaciones"
menu to nOpc
cScreen1:=savescreeen(0,0,maxxow(),
maxcol())
do case
case nOpc=1
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
act_ases()
SET KEY pDERECHA TO
derecha //activa set keys
SET KEY pIZQUIERDA TO
izquierda
case nOpc=2
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
act_mate()
SET KEY pDERECHA TO
derecha //activa set keys
SET KEY pIZQUIERDA TO
izquierda
case nOpc=3
SET KEY pDERECHA TO
//desactiva set keys
SET KEY pIZQUIERDA TO
act_gene()
SET KEY pDERECHA TO
derecha //activa set keys
SET KEY pIZQUIERDA TO
izquierda
endcase
enddo
restore screen from cScreen
setcolor(cColor1)
return
*****
PROCEDURE report1
local nOpc:=1
private cScreen1
setcolor(cColor1)

```

```

m_sombra(6,35,9,45,"EIE"%fE%")
cColor4="w/n,n/w"
cColor3="w/n,n/w"
cColor6="t/w+ ,w/n"
cColLin1 = "w/n"
cColLin2 = "w+/n"
cColLin3 = "w+/n"
cColLin4 = "w/w"
cColPar1="w*/n"
cColPar2="w*/w"
lCambio:=t.
return lCambio
case nOpc=2
if iscolor()
cColor1="gr+rb,R/W"
cColor2="bg+/gr,gr+lw" //cyan-
caf_ama-boo (reportes)
cColor3="gr+rb,R/W" //ama-
magenta,rojo-boo
cColor4="b/bg,b/w" //azul-
cyan,boo+ azul
cColor5="gr+r,r/w" //ama-
rojo,rojo-boo
cColor6="t/g,b/w" //negro-
verde,azul-boo
cColLin1 = "gr+/r" //ama-
rojo
cColLin2 = "w+/b" //boo+-
azul
cColLin3 = "w+/r" //boo+-
rojo error y advertencia
cColLin4 = "w/w"
cColPar1="w*/r" //boo-rojo
cColPar2="w*/b" //boo-azul
endif
lCambio:=t.
return lCambio
case nOpc=3
if iscolor()
cColor1="g/gr,R/W"
cColor2="bg+/gr,gr+lw" //cyan-
caf_ama-boo (reportes)
cColor3="gr+rb,R/W" //ama-
magenta,rojo-boo
cColor4="b/bg,b/w" //azul-
cyan,boo+ azul
cColor5="gr+r,r/w" //ama-
rojo,rojo-boo
cColor6="t/g,b/w" //negro-
verde,azul-boo
cColLin1 = "gr+/r" //ama-
rojo
cColLin2 = "w+/b" //boo+-
cyan,boo+ azul
cColor5="gr+r,r/w" //ama-
rojo,rojo-boo
cColor6="t/g,b/w" //negro-
verde,azul-boo (reportes)
cColLin3="gr+rb,R/W" //ama-
magenta,rojo-boo
cColor4="b/bg,b/w" //azul-
cyan,boo+ azul
cColor5="gr+r,r/w" //ama-
rojo,rojo-boo
cColor6="t/g,b/w" //negro-
verde,azul-boo
cColLin1 = "gr+/r" //ama-
rojo
cColLin2 = "w+/b" //boo+-
azul
cColLin3 = "w+/r" //boo+-
rojo error y advertencia
cColLin4 = "w/w"
cColPar1="w*/r" //boo-rojo
cColPar2="w*/b" //boo-azul
endif
lCambio:=t
return lCambio
case nOpc=4
if iscolor()
cColor1="b/bg,w+/b" //cyan-
caf_ama-boo (reportes)
cColor2="bg+/gr,gr+lw" //ama-
magenta,rojo-boo
cColor3="gr+rb,R/W" //ama-
magenta,rojo-boo
endif
lCambio:=t
return lCambio
case nOpc=5
if iscolor()
cColor1="gr+r,r/w" //cyan-
cColor2="bg+/gr,gr+lw" //cyan-
caf_ama-boo (reportes)
cColor3="gr+rb,R/W" //ama-
magenta,rojo-boo
cColor4="b/bg,b/w" //azul-
cyan,boo+ azul
cColor5="gr+r,r/w" //ama-
rojo,rojo-boo
cColor6="t/g,b/w" //negro-
verde,azul-boo
cColLin1 = "gr+/r" //ama-
rojo
cColLin2 = "w+/b" //boo+-
azul
cColLin3 = "w+/r" //boo+-
rojo error y advertencia
cColLin4 = "w/w"
cColPar1="w*/r" //boo-rojo
cColPar2="w*/b" //boo-azul
endif
lCambio:=t
return lCambio
case nOpc=6
if iscolor()
cColor1="w/b+ ,R/W" //cyan-
caf_ama-boo (reportes)
cColor2="bg+/gr,gr+lw" //ama-
magenta,rojo-boo
cColor3="gr+rb,R/W" //ama-
magenta,rojo-boo
cColor4="b/bg,b/w" //azul-
cyan,boo+ azul
cColor5="gr+r,r/w" //ama-
rojo,rojo-boo
cColor6="t/g,b/w" //negro-
verde,azul-boo
cColLin1 = "gr+/r" //ama-
rojo
cColLin2 = "w+/b" //boo+-
azul
cColLin3 = "w+/r" //boo+-
rojo error y advertencia
cColLin4 = "w/w"
cColPar1="w*/r" //boo-rojo
cColPar2="w*/b" //boo-azul
endif
lCambio:=t
return lCambio
case nOpc=7
if iscolor()
cColor1="w/n,n/w" //menu principal
cColor2="w/w,w+n"
cColor3="gr+rb,R/W"
cColor3="w/n,n/w"

```



```

return lCambio
endcase

enddo
restore screen from cScreen2
setcolor(cColor3)

SET KEY pDERECHA TO derecha
SET KEY plZQUIERDA TO izquierda

return lCambio
*****
procedure part_1
setpos(6,0)
for h=1 to 16 //15
? replci(chr(177),79)
next
@ 1,1 say "UNIVERSIDAD
NACIONAL AUTONOM
A DE MEXICO"
@ 2,5 say "E.N.E.P Acad'n
Jefatura de Actual;a y Materncias
Aplicadas"
diapbox(3,0,23,78,2)
@ 6,0 say
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
@ 4,24 say "MENU PRIN
CIPAL"

@ 24,1 say 'AAMF'
@ 24,7 say fin1_fecha(date())
@ 24,57 say 'F1=Ayuda'
@ 24,67 say 'ESC=Salir'

return
*****
procedure che_bases()
/* funcin para checar si existen las
basesy sus
archivos indexados
correspondientes */

if !file('tesi_mac.dbf')
aviso("No existe el archivo
tesi_mac.dbf....")
close all
cls
quit
else
if !file('tesi_mac.dta')
aviso("No existe el archivo
tesi_mac.dta....")
close all
cls
quit
endif
if !file('tesi_mac.ntx')
use tesi_mac.dbf
index on alum to tesi_mac
use
endif
if !file('cuenta.ntx')
use tesi_mac.dbf
index on cuen to cuenta
use
endif
if !file('clvr.ntx')
use tesi_mac.dbf
index on clvr to clvr
use
endif
if !file('asesor.dbf')
aviso("No existe el archivo
asesor.dbf....")
cls
quit
else
if !file('ases.ntx')
use asesor.dbf
index on ases to ases
use
endif
if !file('clv_ase.ntx')
use asesor.dbf
index on str(clv_ase,3) to clv_ase
use
endif
endif
endif
if !file('carrera.dbf')
close all
aviso("No existe el archivo
carrera.dbf....")
cls
quit
else
if !file('carr.ntx')
use carrera.dbf
index on carr to carr
use
endif
if !file('clv_car.ntx')
use carrera.dbf
index on str(clv_car,2) to clv_car
use
endif
endif
endif
if !file('gener.dbf')
close all
aviso("No existe el archivo
gener.dbf....")
cls
quit
else
if !file('gene.ntx')
use gener.dbf
index on gene to gene
use
endif
if !file('clv_gen.ntx')
use gener.dbf
index on str(clv_gen,2) to clv_gen
use
endif
endif
endif
if !file('mater.dbf')
close all
aviso("No existe el archivo
mater.dbf....")
cls
quit
else
if !file('materia.ntx')
use mater.dbf
index on materia to materia
use
endif
if !file('mat2.dbf')
close all
aviso("No existe el archivo
mat2.dbf....")
cls
quit
else
if !file('mater2.ntx')
use mater2.dbf
index on materia to mater2
use
endif
if !file('clvmat2.ntx')
use mater2.dbf
index on str(clv_mat,2) to clvmat2
use
endif
endif
endif
return
*****
procedure cajas
ZOOMIN()
@ 8,40 SAY * UN MOMENTO
POR FAVOR" colo cColPar2
@ 10,40 SAY * VERIFICANDO
LA EXISTENCIA" colo cColPar2
@ 12,40 say "DE BASES Y
ARCHIVOS INDICES" colo
cColPar2
return

```

```

.....
FUNCTION ZOOMIN(TIEMPO)
CLEAR
IF TIEMPO=>NIL
tiempo=0.1 && AUMENTAR
ESTE VALOR PARA DISMINUIR
LA VELOCIDAD
ENDIF
@ 0.0 TO 1.1
INKEY(tiempo)
@ 1.1 TO 3.3
INKEY(tiempo)
@ 3.2 TO 6.6
INKEY(tiempo)
@ 6.7 TO 9.10
INKEY(tiempo)
@ 9.9 TO 15.16
INKEY(tiempo)
@ 12.12 TO 18.27
INKEY(tiempo)
@ 15.17 TO 20.30
INKEY(tiempo)
@ 19.22 TO 24.37
INKEY(tiempo)
@ 16.27 TO 21.45
INKEY(tiempo)
@ 13.30 TO 18.55
INKEY(tiempo)
@ 5.35 TO 16.75
@ 06.36 CLEAR TO 15.74
RETURN NIL

```

```

.....
FUNCTION ZOOMOUT(TIEMPO)
IF TIEMPO=>NIL
TIEMPO=1
ENDIF
@ 4.35 CLEAR TO 16.75
INKEY(tiempo)
@ 13.30 CLEAR TO 20.55
INKEY(tiempo)
@ 16.27 CLEAR TO 21.45
INKEY(tiempo)
@ 19.22 CLEAR TO 24.37
INKEY(tiempo)
@ 15.17 CLEAR TO 22.33
INKEY(tiempo)
@ 12.12 CLEAR TO 18.27
INKEY(tiempo)
@ 9.9 CLEAR TO 15.16
INKEY(tiempo)
@ 6.7 CLEAR TO 9.10
INKEY(tiempo)
@ 3.2 CLEAR TO 6.6
INKEY(tiempo)
@ 1.1 CLEAR TO 3.3
INKEY(tiempo)
@ 0.0 CLEAR TO 1.1
RETURN NIL

```

```

.....
PROCEDURE ENTRADA
SETCURSOR(0)
cls
setcolor(cColor1)
CAJAS()
che_bases() //checa bases y sus
indexados

```

```

ponSetRelation() //apertura de todas
las bases con sus indices
ZOOMOUT(.08)
SETCURSOR(1)
RETURN

```

```

PROCEDURE AYUDA(cProc)
.....

```

```

/*trataremos con procedimientos y
variables y los mezclaremos
las cuatro primeras letras para
Procedimiento y las cuatro
últimas para la variable*/

```

```

local
cScreen:=savescree(0,0,maxrow(),m
axcol())

```

```

local Color_act:=setcolor()
local nCur := setcursor()
local cP1:=salva_mens(24)
set key 286 to agranda // ALT-A
set key pF1 to
setcolor(cColor2) //cColor1
setcursor(0)
cArchivo=cProc+'.AAM'
IF !file(cArchivo)
sonido1()
setcolor(cColLin3)
m_sombra(11,15,13,73,.L)
depos(12,16)
devoat(ERROR!!! archivo
&cArchivo de ayuda
inexistente...'.cColLin3)
inkey(2.3)
setcolor(color_act)

```

```

restscreen(0,0,maxrow(),maxcol(),cS
creeen)
setcursor(nCur)
set key pF1 to ayuda
set key 286 to //agranda //
ALT-A
return
endif

```

```

m_sombra(7,20,19,60...L)
depos(7,35)
devoat(' A Y U D A '.cColLin1)
mensaje(24, 'AAM '+
fin_fecha(date()) + space(51)
+ESC+Satur,cColor1)
depos(19,30)
devoat('ALT-A=M s
ayuda'.cColLin1)
VAR1:=
MEMMOREAD(cArchivo)
memoedit(var1,8,21,18,59,..E)
//.F.)

```

```

set key pF1 to ayuda
set key 286 to //agranda //
ALT-A
setcolor(Color_act)
setcursor(nCur)
rest_mens(24,cP1)

```

```

restscreen(0,0,maxrow(),maxcol(),cS
creeen)

```

```

return

```

```

procedure agranda
local
cPant:=savescree(0,0,maxrow(),max
col())

```

```

set key 286 to //agranda // ALT-A
m_sombra(2,5,22,75..L)
depos(2,35)
devoat(' A Y U D A '.cColLin1)
VAR1:= MEMMOREAD(cArchivo)

```

```

setcursor(1) // mientras editamos
la ayuda!!
var1=memoedit(var1,3,6,21,74..L)
//.F.) //L. mientras editamos la
ayuda!!
memoedit(cArchivo,var1)
setcursor(0) // mientras editamos la
ayuda!!

```

```

restscreen(0,0,maxrow(),maxcol(),cP
ant)
set key 286 to agranda // ALT-A
return

```

```

programa 2
*rutinas comunes para PAnt_tes.prg y
derivadas de este
*
* Aviso
* Sonido1()
* Sonido2()
* Sonido3()
* m_sombra()
* cuadro()
* mensaje()
* fin_fecha()
* trunca()
* salva_mens()
* rest_mens()

```

```

* Uh_Actualizacien -> 6-jun-96

```

```

function
aviso(texto,tiempo,cCol1,nSen)
local
cPan1:=savescreen(11,20,14,61)
local cColOrig:=setcolor()
local nCur:=setcursor()
setcursor(0)
if cCol1=nil
if iscolor()
cCol1 = w/r'
else
cCol1 = w/n'
endif
endif
if tiempo=nil
Tiempo = 1
endif
setcolor(cCol1)
if len(texto) > 39
texto = substr(texto,1,38)
endif
do case
case nSen=1
sonido1()
case nSen=2
sonido2()
case nSen=3
sonido3()
otherwise
sonido1()
endcase

tam=len(Texto)
col1 = (maxcol()-tam) / 2
m_sombra(11,20,13,60,...,t)
@ 12,col1 say texto colo cCol1 //
modificien
inkey(tiempo)
setcolor(cColOrig)
setcursor(nCur)
restscreen(11,20,14,61,cPan1)
return nil
*****
function sonido1()
tone(3000,1)
tone(1,1)
tone(3000,1)
return nil
*****
function sonido2()
tone(300,1.5)
return nil
*****
function sonido3()
tone(100,3)
return nil
*****

function
m_sombra(r1,c1,r2,c2,cCarac,lCuaed)
if lCuaed=nil
lCuaed=.f.
endif
if r1=nil
r1=1
endif
if r2=nil
r2=10
endif
if c1=nil
c1=1
endif

c1=1
endif
if c2=nil
c2=10
endif
if cCarac=nil
cCarac="E.l.v%|\$="
endif

if lCuaed
cuadro(r1,c1,r2,c2)
else
scroll(r1,c1,r2,c2)
@ r1,c1,r2,c2 box cCarac
endif

col=savescreen(r1+1,c2+1,r2+1,c2+1)
lin=savescreen(r2+1,c1+2,r2+1,c2)

for g=2 to len(col) step 2
col=stuff(col,g,1,chr(6)) //cf.
next
for g=2 to len(lin) step 2
lin=stuff(lin,g,1,chr(6)) //8
next
restscreen(r1+1,c2+1,r2+1,c2+1,col)
restscreen(r2+1,c1+2,r2+1,c2,lin)
return nil
*****
function
get_optn(r1,c1,r2,c2,cTexto,col)
local
point:=savescreen(0,0,maxrow(),maxc()
)
local cColor:=setcolor()
local tam,long,medio,col1,col2,col3
PUBLIC eleccion:=1
if col=nil
col=cColPrin
endif
if r1=nil
r1=11
endif
if r2=nil
r2=14
endif
if c1=nil
c1=20
endif
if c2=nil
c2=60
endif
tam=len(cTexto)
long=c2-c1
medio=(long/2)+c1
col1 = medio-(tam/2) //maxcol()-
tam) / 2
col2 = medio-2
col3 = medio+2
if tam > 38
cTexto=substr(cTexto,1,38)
endif
*scroll(r1,c1,r2,c2)
setcolor(cCol)
m_sombra(r1,c1,r2,c2,...,l)
@ r1+1,col1 say cTexto + '?'
@ r1+2,col2 prompt 'Si'
@ r1+2,col3 prompt 'No'
menu to eleccion
eleccion = iif(eleccion=0,2,eleccion)
restscreen(0,0,maxrow(),maxcol(),pa
nt)
setcolor(cColor)
return eleccion
*****
/* prueba para ver un real cuadro con
ilusion de crecimiento desde
su aparicion */
function cuadro(r1,c1,r2,c2)

local nCaj:=0
local aCajas:={}
local nTamRen,nTamCol,nCoc
aadd(aCajas,r1)
aadd(aCajas,c1)
aadd(aCajas,r2)
aadd(aCajas,c2)
nTamRen=r2-r1
nTamCol=c2-c1
if nTamCol<nTamRen
nCoc=int(nTamCol/nTamRen)
while r1 < 2 .and. c1 < c2
nCaj += 1
r1 += 1
r2 -= 1
c1 += nCoc
c2 -= nCoc
cAuxCaj:=str(nCaj,2)
cAuxCaj:=alltrim(cAuxCaj)
var = 'aCaj' + cAuxCaj
&var = &('{') //excelente

aadd(&var,r1)
//aCaj&cAuxCaj,r1)
aadd(&var,c1)
//j&cAuxCaj,c1)
aadd(&var,r2)
aadd(&var,c2)

enddo
for nC=nCaj to 1 step -1
cAux=str(nC,2)
cAux=alltrim(cAux)

scroll(aCaj&cAux[1],aCaj&cAux[2]
],aCaj&cAux[3],aCaj&cAux[4])
inkey(.02)

dispbox(aCaj&cAux[1],aCaj&cAux[
2],aCaj&cAux[3],aCaj&cAux[4],2
)
aCaj&cAux:= {}
next

scroll(aCajas[1],aCajas[2],aCajas[3],
aCajas[4])

dispbox(aCajas[1],aCajas[2],aCajas[
3],aCajas[4],2)

else
nCoc=int(nTamRen/nTamCol)
while c1 < c2 .and. r1 < r2

nCaj += 1
r1 += nCoc
r2 -= nCoc
c1 += 1
c2 -= 1
cAuxCaj:=str(nCaj,2)
cAuxCaj:=alltrim(cAuxCaj)
var = 'aCaj' + cAuxCaj
&var = &('{')

```

```

add(&var,r1)
add(&var,c1)
add(&var,r2)
add(&var,c2)

endifo
for nC=nCaj to 1 step -1
  cAux=str(nC,2)
  cAux=altrim(cAux)

  scroll(aCaj&cAux,{1},aCaj&cAux,{2}
],aCaj&cAux,{3},aCaj&cAux,{4})
inkey(.02)

  disbbox(aCaj&cAux,{1},aCaj&cAux
{2},aCaj&cAux,{3},aCaj&cAux,{4},2
)
  aCaj&cAux={}
  next

scroll(aCajas[1],aCajas[2],aCajas[3],
aCajas[4])

disbbox(aCajas[1],aCajas[2],aCajas[3],
aCajas[4],2)
endif
return nil

*****
procedure mensaje(lin,mens,col)
if col=nil
if iscolor()
  col="gr"/r'
else
  col="w/w"
endif
endif
if lin=nil
  lin=24
endif
if mens=nil
  mens=space(80)
endif
  @ lin,0 say space(80) colo Col
  @ lin,0 say mens colo Col

return

*****
function firm_fecha(fecha)

/*local me_s:=substr(dtoc(fecha),4,2)
local di_a:=substr(dtoc(fecha),1,2)
local an_o:=substr(dtoc(fecha),7,2)*/

local me_s:= month(fecha)
local di_a := substr(str(day(fecha)),2)
local an_o :=
substr(str(year(fecha)),4,2)
local fecha2

if empty(fecha)
  return space(8)
endif

do case
  case me_s=01
    mes="ENE"
  case me_s=02
    mes="FEB"
  case me_s=03
    mes="MAR"

  case me_s=04
    mes="ABR"
  case me_s=05
    mes="MAY"
  case me_s=06
    mes="JUN"
  case me_s=07
    mes="JUL"
  case me_s=08
    mes="AGO"
  case me_s=09
    mes="SEP"
  case me_s=10
    mes="OCT"
  case me_s=11
    mes="NOV"
  case me_s=12
    mes="DIC"
  otherwise
    mes=space(3)

  ENDCASE
  fecha2=di_a+"/"+mes+"/"+an_o
  return fecha2
*****
function salva_mens(lin)
local cPant1
if lin=nil
  lin=24
endif
cPant1:=saveScreen(lin,0,lin,maxcol()
)
return cPant1
*****
procedure rest_mens(lin,Var)
restScreen(lin,0,lin,maxcol(),Var)
return
*****
FUNCTION TERMOH(VALOR)

IF valor < 1
  SETCOLOR(cColor3)
  m_sombra(13,30,18,73,L)
  @ 14,31 SAY "  10 20 30 40 50
60 70 80 90 100 % "
  @ 15,31 SAY "
U U . . . . . L . . . . . A z "
  @ 16,31 SAY "
*****
*****
  @ 17,31 SAY "
AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAU "
  ELSE
    @ 16,33 SAY
  REPLICATE(CHR(219),valor3)
  ENDIF
  RETURN NIL

*****
function trunca()
local pasa := L
if inkey() = 27 or lastkey() = 27 or
nextkey() =27
  pasa=.f.
endif
return pasa

Programa 3

* PROGRAM: PANT_TES.PRG
* Descripcien: PANTALLA PARA
TITULACION

* Author: ARMANDO AMADOR
MOLINA with GENIFER
* Fecha: 01/03/96 Hora: 21:00

PROCEDURE PANT_TES

#include "memoedit.ch"
#include "inkey.ch"
#xcommand read =>
readmoda(GetList,nT);
GetList:={ }
local cColAux:=setcolor()
* ARCHIVOS
* BASE DE LOS REGISTROS DE
TESIS
  PonSetRelation()

* VARIABLES

* Key in database TESI_MAC
  alum = "
* Fields in database TESI_MAC
* nT=15
  cuen = "
  freg = date()
  fler = date()
  fit = date()
  direc = "
  telet = "
  dired = "
  teled = "
  clvr = "
  cOpet = "
  cCopc = 0
  cGene = "
  cCarr = "
  nCarr1 = 0
  cCo2 = "
  cCo3 = "
  cCo4 = "
  cCo5 = "
  cMat1 = "
  cMat2 = "
  cAces = "
  busca_ase = 0
  busca_car = 0
  busca_gm = 0
  busca_mat1 = 0
  busca_mat2 = 0
  busca_ope = 0
  busca_aux = "
  nCarr_aux = 0

* Flags
  abort_rec = .F.
  dupl_rec = .F.
  empty = .F.
  files_ok = .T.
  del_rec = .F.
  valid_rec = .T.
  show_all = .T.
* Otros
  key1 = 'm->alum'
  index1 = 'alum'
  null_key1 = "
  comp_key1 = 'trim(m->alum)'

list1 = 'alum,cuen,freg,fler,fit'
pak1 = .F.
choice = "
option = 0

```

```

filr_str = *
scr_num = 1
num = 1
max_screen = 1
record_no = 0
//variables memo
cMemo1 = *
cMemo2 = *
cMemo3 = *
cMemo4 = *
cMemo5 = *

* AMBIENTE
set exact off
set heading off
set bell off
set deleted on
set epoch to 1980 //excelente
set confirm on

!QuitaPon1 = .t.
!QuitaPon2 = .t.
!QuitaPon3 = .t.
!QuitaPon4 = .t.
!QuitaPon5 = .t.
* PANTALLA
do d_pantal

* APERTURA DE ARCHIVOS
* database is empty if eof() = .T.
empty = eof()
if empty
  aviso(Archivo Sin
Informacion',3,cColLin3)
endif
* PROCESO DE LOOP
do while scr_num >= 1
!Mues_mern:=.f.
do PonSet

set key 280 to ponCaja1 //ALT-O
empceemos
set key 302 to ponCaja2 //ALT-C
set key 290 to ponCaja3 //ALT-G
set key 286 to ponCaja4 //ALT-A
set key 306 to ponCaja5 //ALT-M

mensaje(0,F5=T;tulo F6=Obj.Gral.
F7=Capitulos F8=Resumen
F9=Simdiales ALT.(O,C,G,A,M)
,cColLin1)

record_no = recno()
* precéida si opcion es modo de
altas

if empty .or. (scr_num =
max_screen .and. option = 1)
option = 1
else
do carga_vl
do d_regl
option = 6
endif

setcolor(cColor4) //n/w,r/w
Set MESSAGE TO 24 CENTER
@ 24, 00 SAY Space(80)

do case

@ 23, 00 PROMPT "Altas "
MESSAGE "Adicionar nuevos
registros"
@ 23, 08 PROMPT "Inicial "
MESSAGE "Se posiciona on el
primer registro"
@ 23, 16 PROMPT "Elim. "
MESSAGE "Borra el registro actual"
@ 23, 24 PROMPT "Ultimo "
MESSAGE "Se posiciona on el
ltimo registro"
@ 23, 32 PROMPT "Lista "
MESSAGE "Lista por pantalla la inf.
almacenada"
@ 23, 40 PROMPT "Modif. "
MESSAGE "Modifica el registro
actual"
@ 23, 49 PROMPT "Sig. "
MESSAGE "Lee el siguiente
registro"
@ 23, 57 PROMPT "Previo "
MESSAGE "Lee el registro anterior"
@ 23, 65 PROMPT "Busca "
MESSAGE "Busca por clave en la
base de datos"
@ 23, 73 PROMPT "Term. "
MESSAGE "Termina Sesien De
Captura"
if !QuitaPon1
@ 24,73 say 'O' colo cColor4
else
@ 24,73 say ''
endif
if !QuitaPon2
@ 24,74 say 'C' colo cColor4
else
@ 24,74 say ''
endif
if !QuitaPon3
@ 24,75 say 'G' colo cColor4
else
@ 24,75 say ''
endif
endif
if !QuitaPon4
@ 24,76 say 'A' colo cColor4
else
@ 24,76 say ''
endif
if !QuitaPon5
@ 24,77 say 'M' colo cColor4
else
@ 24,77 say ''
endif
endif

MENU TO option
option = IIF(m->option == 0, 10, m-
>option)

do QuitaSet

set key 280 to //ALT-O
empceemos
set key 302 to //ALT-C
set key 290 to //ALT-G
set key 286 to //ALT-A
set key 306 to //ALT-M

mensaje(0,F5=T;tulo
F6=Obj.Gral. F7=Capitulos
F8=Resumen F9=Simdiales'
,cColLin1)

do case

case option = 1
valid_rec = .T.
do while valid_rec
do alta1
EndDo
* Inicio
case option = 2
go top
* Elimina
case option = 3
do eliminal
* Ultimo
case option = 4
go bottom
* Lista
case option = 5
do list_1 //with
list&num
show_all = .T.
do d_pantal1
go record_no
* Modifica
case option = 6
do modifi1
* avaNzr
case option = 7
skip
if eof()
go bottom
aviso(Ultimo Registro
',1,2,cColLin3)
endif
* Previos
case option = 8
skip -1
if bof()
go top
aviso(Primer Registro
',1,2,cColLin3)
endif
* Busca
case option = 9
do busca1.
* Salida
case option =
10
aviso('Fin de
captura...',2,cColLin3)
if del_rec
do Elim_n1
endif
Set MESSAGE TO 21 CENTER
set confirm off
setcolor(cColAct)
return
endcase
enddo

return
*****
function ponle2()
local tecla:=lastkey()
ponle()
keyboard chr(tecla) //enter
return .t.
*****

* PROCEDIMIENTOS (listados
alfabeticamente)
procedure aosp_cam1

```

```

* Get field variables
@ 6,63 get m->cuen picture '@R
99999999-9';
valid val_cuenta(m->cuen)
if !QuitaPon1
  @ 8,22 get m->cOpc picture
  '@!@kx(45); //empezemos aqui
  when
  escribe_aux('opc tit','opc',6,'seis','clv
_opc','OPCIONES TITULACION')
else
  @ 8,22 say m->cOpc colo
cColLin2
endif
if !QuitaPon2
  @ 9,33 get m->cCarr picture
  '@!@kx(20);
  when
  escribe_aux('carrera','carr',3,'tres','clv
_car','CARRERAS');
  valid ponleO
else
  //@ 9,53 say m->cCarr colo
cColLin2 //gr+lw
  @ 9,53 get m->cCarr colo
cColLin2 //gr+lw
  when ponle2O
endif
if !QuitaPon3
  @ 10,55 get m->cGene picture
  '@!@kx(5);
  when
  escribe_aux('gener','gene',4,'cuatro','c
lv_gen');
'clv_car=mCarr_aux';GENERACION
)
else
  @ 10,55 say m->cGene colo
cColLin2 //gr+lw
endif
  @ 11,58 get m->freg pict '@d'
  @ 12,58 get m->fler pict '@d';
  valid tiempoOpc(m->freg,m->fler)
  //iif(m->freg < m->fler
.OR. EMPTY (M->FTer),t...f)
  @ 13,58 get m->flit pict '@d';
  valid vid_fecha2(m->fler,m->flit)
  @ 16,4 get m->diret pict
  '@S0@!@kx(80)'
  @ 16,60 get m->teled pict
  '@!@kx(12)'
  @ 18,4 get m->diret pict
  '@S0@!@kx(80)'
  @ 18,60 get m->telet pict
  '@!@kx(12)'
  @ 19,18 get m->clvr pict
  '@!@kx(4);
  valid noDup(m->clvr)
if !QuitaPon4
  @ 19,33 get m->cAses picture
  '@!@kx(40);
  when
  escribe_aux('asesor','ases',2,'dos','clv
_ase','ASESORES')
else
  @ 19,33 say m->cAses colo
cColLin2 //gr+lw
endif
if !QuitaPon5
  @ 20,14 get m->cMat1 picture
  '@!@kx(20);
  when
  escribe_aux('mater','materia',5,'cinco'
,'clv_mat');
'clv_car=mCarr_aux';TEMAS/MATE
RIAS)
else
  @ 20,14 say m->cMat1 colo
cColLin2 //gr+lw
endif
if !QuitaPon5
  @ 20,50 get m->cMat2 picture
  '@!@kx(20);
  when
  escribe_aux('mater2','materia',8,'och
o','clv_mat');
'clv_car=mCarr_aux';TEMAS/MA
TERIAS 2)
else
  @ 20,50 say m->cMat2 colo
cColLin2 //gr+lw
endif
if m->cOpc = 3
  setcolor(cColor1); //to r/bg,w/b
//haber con esto
  @ 9,10 say "AUTORES TESIS
COLECTIVAS"
  @ 10,3 say m->cCo2
  @ 11,3 say m->cCo3
  @ 12,3 say m->cCo4
  @ 13,3 say m->cCo5
else
  setcolor(cColor1)
  scroll(9,2,14,45)
endif
return
procedure aocp_key1
* Get key variables
@ 6,11 get m->alum pict
 '@!@kx(40);
  valid iif(m->alum='',f.,t.)
  return
*****
procedure altas1
!Mluec_Mern=L
do ponSet !!!!!!!!!!!!!!!
clear gets
do ini_key1
do ini_fld1
do limp_ca1
abort_rec = F.
valid_rec = F.
do while .not. (valid_rec.or.
abort_rec)
do aocp_key1
read
m->alum = strtran(m->alum,' ')
m->alum = strtran(m->alum,' ')
m->alum = strtran(m->alum,' ')
comp_key = comp_key&num
if null_key&num = &comp_key
  * blank key
  abort_rec = T.
else
  do chk_dup1
if dupl_rec
  abort_rec = T.
else
  do aocp_cam1
  read
  valid_rec = T.
endif (dupl_rec)
endif (null_key = comp_key)
enddo (while .not. (valid_rec.or.
abort_rec))
if valid_rec
  do gra_reg1 with t.
else
  * break out of "Add" if record
  invalid
  if empty
  option = 10
else
  option = 6
  if .not. dupl_rec
  go record_no
  endif
  endif (empty)
endif (valid_rec)
do QuitaSet
return
procedure chk_dup1
* Set dupl_rec to .F. if key is a
duplicate, or to .T. otherwise
key = key&num
aam = &key
aam=strtran(aam,' ')
aam=strtran(aam,' ')
aam=strtran(aam,' ')
if .not. empty
seek trim(aam) //sin trimO aam-
>&key
endif
if option = 6
  dupl_rec = record_no <> recnoO
.and. .not. eof()
else
  dupl_rec = .not. eof()
endif
if dupl_rec
  aviso('Nombre Duplicado
!!!',cColLin3)
else
  if .not. empty
  go record_no
  endif
endif dupl_rec
return
procedure limp_ca1
* Clear screen field areas
setcolor(cColor1) // to r/bg
@ 8,22 say space (45)
@ 10,55 say space (5)
@ 9,53 say space (20)
@ 19,33 say space (40)
@ 6,63 say space (9)
@ 11,58 say space (8)
@ 12,58 say space (8)
@ 13,58 say space (8)
@ 16,4 say space (50)
@ 16,60 say space (12)
@ 18,4 say space (50)
@ 18,60 say space (12)
@ 19,18 say space (4)

```

```

@ 19,33 say space (40)
@ 20,14 say space (20)
@ 20,50 say space (20)
return

procedure elimina1
* Delete current record upon user
approval
local nReg:=recno()
get _optn('...Elimina (S/N)',cColor4)
//:SN_choice
if !election = 1
//eliminar reg. de conjuntas
if m->cCope = 3
do elim_conj
endif
go nReg //despu,s de...
delete
* reposition to the next record
skip
* if last record deleted, go to
beginning of database
if eof()
go top
endif
if eof()
* quit if last record deleted and
database becomes empty
option = 10
endif
del_rec = .T.
pat&num = .T.
endif (election = 'S')
return
.....

procedure d_reg1
* Load fields from database and
display record
do aocp_key1
do aocp_cam1
clear gets
show_all = .F.
return

procedure d_parta1
* Display stationary part of screen
setcolor(cColor1)
clear
@ 1,1 say
'O|||||||||||||||||||||||||||||'
@ 1,41 say
'-----'
@ 2,1 say ' REGISTRO DE
TESIS PARA ACTUARIA'
@ 2,42 say 'Y MATEMATICAS
APLICADAS' UN A M
@ 3,1 say '
ENEP'
@ 3,42 say 'ACATLAN
'
@ 4,1 say
'E|||||||||||||||||||||||||||||'
@ 4,41 say
'-----'
@ 5,1 say '
'
@ 5,41 say '
'
@ 6,1 say ' NOMBRE [
'
@ 6,52 say ' CUENTA [ ]
'
@ 7,1 say ' Ap.Paterno
Ap.Materno Nombre'
@ 7,41 say '(s)
'

@ 8,1 say ' OPCION
TITULACION [
'
@ 8,43 say '
'
@ 9,1 say '
'
@ 9,44 say ' CARR [
'
@ 10,1 say '
'
@ 10,47 say ' GENER. [
'
@ 11,1 say '
'
@ 11,47 say ' FECH.REG. [
'
@ 12,1 say '
'
@ 12,47 say ' FECH.TER. [
'
@ 13,1 say '
'
@ 13,47 say ' FECH.TIT. [
'
@ 14,1 say '
'
@ 14,75 say '
'
@ 15,1 say ' DIRECCION
DE CASA *
'
@ 15,60 say ' TEL. DE CASA '
'
@ 16,1 say '
'
@ 16,54 say ' ]
'
@ 17,1 say ' DIRECCION
DE TRABAJO*
'
@ 17,60 say ' TEL. TRABAJO '
'
@ 18,1 say '
'
@ 18,54 say ' [
'
@ 19,1 say ' CLAVE DE REG. [
]
@ 19,73 say ' ]
'
@ 20,1 say ' MATERIA 1 [
]
@ 20,41 say ' TERIA 2 [
]
'
@ 21,1 say '
'
@ 21,41 say '
'
@ 22,1 say '
'
'AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
'
@ 22,41 say '
'
'AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAU'
return
.....

procedure ini fld1
* Clear field variables
cuen = space(8)
freg = ctod( // )
fler = ctod( // )
flit = ctod( // )
dired = space(80)
teled = space(12)
clvr = space(4)
cOpat = space(45)
cCarr = space(20)
cGene = space(3)
cMat1 = space(20)
cMat2 = space(20)
cAses = space(40)
cMemo1 = ''
cMemo2 = ''
cMemo3 = ''
cMemo4 = ''
cMemo5 = ''

return

procedure ini_key1
* Clear key variables
alun = space(40)
return

procedure list_1
parameters list_items
* List records beginning from current
record
do while .T.
setcolor(cColor4)
clear
list subtr(uno->alum,1,30),uno->
cuen,uno->freg,uno->fler,uno->flit
next 20
if eof()
avisos('OK',3,cColIn2)
exit
endif
get _optn(20,64,23,77,'M s
(SN)',cColor4)
if !election = 2
exit
endif
enddo (while .T.)
return

* Copy fields from database record to
memory variables
alum = alun
cuen = cuen
freg = freg
fler = fler
flit = flit
dired = dired
teled = teled
dired = dired
teled = teled
clvr = clvr
cOpat = scis->opat.
cCope = scis->clv_ope
cGene = ccurro->gene
cCarr = tres->carr
nCarr1 = tres->clv_carr
cCo2 = siet->co2
cCo3 = siet->co3
cCo4 = siet->co4
cCo5 = siet->co5
cMat1 = cmo->materia
cMat2 = ocho->materia
cAses = dos->ases
busca_ase = uno->clv_ase
busca_carr = uno->clv_carr
busca_gen = uno->clv_gen
busca_ope = uno->clv_ope
busca_mat1 = uno->clv_mat1
busca_mat2 = uno->clv_mat2

cMemo1 = uno->itulo
cMemo2 = uno->obj_gen
cMemo3 = uno->capitulo
cMemo4 = uno->resumen
cMemo5 = uno->sinodal

return

procedure modifi1
* Modify current record

```

```

* Get field variables
@ 6,63 get m->cuen picture '@R
9999999-9';
valid val_cuenta(m->cuen)
if IQuitaPon1
@ 8,22 get m->cOpet picture
'@!(@kx(45)); //empezemos aqui
when
escribe_aux('ope tit','ope','6','seis','clv
_ope_', 'OPCIONES TITULACION')
else
@ 8,22 say m->cOpet color
cColLin2
endif
if IQuitaPon2
@ 9,53 get m->cCarr picture
'@!(@kx(20));
when
escribe_aux('carrera','carr','3','tres','clv
_car_', 'CARRERAS');
valid ponle()
else
//@ 9,53 say m->cCarr color
cColLin2 //gr+^w
@ 9,53 get m->cCarr color
cColLin2; //gr+^w
when ponle2()
endif
if IQuitaPon3
@ 10,55 get m->cGene picture
'@!(@kx(3));
when
escribe_aux('gener','gene','4','cuatro','c
lv_gen');
'clv_car'=nCarr_aux,'GENERACION
')
else
@ 10,55 say m->cGene color
cColLin2 //gr+^w
endif
@ 11,58 get m->freg pict '@d
@ 12,58 get m->fiter pict '@d;
valid tiempo(m->freg,m->fiter)
//if(m->freg < m->fiter
.OR. EMPTY(M->FITER),L,L)
@ 13,58 get m->fiter pict '@d;
valid vld_fecha2(m->fiter,m->fiter)
@ 16,4 get m->diret pict
'@S50!@!@kx(80)
@ 16,60 get m->telet pict
'@!@!@kx(12)
@ 18,4 get m->diret pict
'@S50!@!@kx(80)
@ 18,60 get m->telet pict
'@!@!@kx(12)
@ 19,18 get m->clvr pict
'@!@!@kx(4);
valid noDup(m->clvr)
if IQuitaPon4
@ 19,33 get m->cAses picture
'@!(@kx(40));
when
escribe_aux('asesor','ases','2','dos','clv
_ase_', 'ASESORES')
else
@ 19,33 say m->cAses color
cColLin2 //gr+^w
endif
if IQuitaPon5
@ 20,14 get m->cMat1 picture
'@!(@kx(20));
when
escribe_aux('mater','materia','5','cinco'
,'clv_mat');
'clv_car'=nCarr_aux,'TEMAS/MATE
RIAS')
else
@ 20,14 say m->cMat1 color
cColLin2 //gr+^w
endif
if IQuitaPon5
@ 20,50 get m->cMat2 picture
'@!(@kx(20));
when
escribe_aux('mater2','materia','8','och
o','clv_mat');
'clv_car'=nCarr_aux,'3TEMAS/MA
TERIAS 2')
else
@ 20,50 say m->cMat2 color
cColLin2 //gr+^w
endif
if m->cOpe = 3
setcolor(cColor1) //to r/bg,w/b
//haber con esto
dupbox(9,2,14,45,2)
@ 9,10 say "AUTORES TESIS
COLECTIVA"
@ 10,3 say m->cCo2
@ 11,3 say m->cCo3
@ 12,3 say m->cCo4
@ 13,3 say m->cCo5
else
setcolor(cColor1)
scroll(9,2,14,45)
endif
return
procedure acep_key1
* Get key variables
'@ 6,11 get m->alum pict
'@!(@kx(40));
valid if(m->alum='',f.,L)
return
*****
procedure altas1
lMues_Mem= L
do ponSet //!!!!!!
clear gets
do ini_key1
do ini fld1
do limp_cal
abort_rec = .F.
valid_rec = .F.
do while not (valid_rec.or.
abort_rec)
do acep_key1
read
m->alum = strtran(m->alum,' ','')
m->alum = strtran(m->alum,' ','')
m->alum = strtran(m->alum,' ','')
comp_key = comp_key&num
if null key&num = &comp_key
* blank key
* abort_rec = .T.
else
do chk_dup1
if dupl_rec
abort_rec = .T.
else
do acep_cal1
read
valid_rec = .T.
endif (dupl_rec)
endif (null key = comp_key)
enddo (while not (valid_rec.or.
abort_rec))
if valid_rec
do gra_reg1 with t
else
* break out of "Add" if record
invalid
if empty
option = 10
else
option = 6
if .not. dupl_rec
go record_no
endif
endif (valid_rec)
do QuitaSet
return
procedure chk_dup1
* Set dupl_rec to .F. if key is a
duplicate, or to .T. otherwise
key = key&num
aam = &key
aam=strtran(aam,' ','')
aa.m=strtran(aam,' ','')
aam=strtran(aam,' ','')
if .not. empty
seek trim(aam) //sin trim() aam-
>&key
endif
if option = 6
dupl_rec = record_no <> record()
.and .not. eof()
else
dupl_rec = .not. eof()
endif
if dupl_rec
write(Numero Duplicado
!!!!,cColLin3)
else
if .not. empty
go record_no
endif
endif dupl_rec
return
procedure limp_cal
* Clear screen field areas
setcolor(cColor1) // to r/bg
@ 8,22 say space (45)
@ 10,55 say space (5)
@ 9,53 say space (20)
@ 19,33 say space (40)
@ 6,63 say space (9)
@ 11,58 say space (8)
@ 12,58 say space (8)
@ 13,58 say space (8)
@ 16,4 say space (50)
@ 16,60 say space (12)
@ 18,4 say space (50)
@ 18,60 say space (12)
@ 19,18 say space (4)

```



```

* The following "get record" loop is
exited when either (1) a valid record
* is entered (valid_rec = .T.), or (2)
entry is aborted by a duplicate.
* key (when not allowed), or a
request from the validation
procedure.
!Mues_Mem=.t.
do ponSet ////////////////
valid_rec = .F.
abort_rec = .F.
do while .not. (valid_rec.or.
abort_rec)
do acrp_key1
read
do acrp_carr1
read
comp_key = comp_key&num
if null_key&num = &acrp_key
abort_rec = .T.
else
do chk_dup1
if dupl_rec
abort_rec = .T.
else
valid_rec = .T.
endif (dupl_rec)
endif (null_key = comp_key)
enddo (while .not. (valid_rec.or.
abort_rec))
if valid_rec
do gra_reg1 with .F.
else
if .not. dupl_rec
go record_no
endif
go record_no
endif (valid_rec)
do QuitaSet
return

procedure Elim_n1
* Pack deleted records with user
confirmation
@ 10.15 SAY "Eliminando
Registros." color cColPar1
pack
return

procedure repl_rec
* Replace database fields with
memory variables
replace uno->alum with m->alum
replace uno->cuen with m->cuen
replace uno->freg with m->freg
replace uno->flit with m->flit
replace uno->diret with m->diret
replace uno->telet with m->telet
replace uno->dired with m->dired
replace uno->teled with m->teled
replace uno->clvr with m->clvr

replace uno->titulo with upper(m-
>cMemo1)
replace uno->obj_gen with upper(m-
>cMemo2)
replace uno->capitulo with
upper(m->cMemo3)
replace uno->resumen with
upper(m->cMemo4)
replace uno->sinodal with upper(m-
>cMemo5)

replace uno->clv_ase with
busca_ase
replace uno->clv_car with
busca_car
replace uno->clv_gen with
busca_gen
replace uno->clv_opc with
busca_opc
replace uno->clv_mat1 with
busca_mat1
replace uno->clv_mat2 with
busca_mat2

COMMIT
return

procedure busca_l
* Accept key and seek record; if not
found reposition to record_no
do limp_ca1
do ini_key1
do acrp_key1
read
comp_key = comp_key&num
if null_key&num = &comp_key
* blank key
return
endif
key = key&num
seek trim(&key)
if eof()
avisos("No Existe.",2,3,cColLin3)
go record_no
endif
return

procedure gra_reg1
parameters new_rec
* If new_rec: append record currently
in memory to database;
* if .not. new_rec: replace database
record record_no with memory fields.
do QuitaSet

get_optn(,,,Grabar (S/N),cColor4)
if eleccion = 1
if new_rec
append blank
endif
do repl_rec
empty = .F.
else
if .not. empty
go record_no
endif
return
endif (choice = 'S')

if busca_opc = 3
if new_rec //aquí pondr. rutina
para leer los
tone(3000,1) // demas
autores de conj.
tone(1,1)
tone(3000,1)
do tesconj
else //modifica
sonido(1)
do testconj
do autor_cj
sele 1
endif

endif
endif

procedure skip1
* Move forward/backward several
records
skip_no = 0
@ 23,70 say " Regn get skip_no
@Z 9999"
read
skip skip_no
if eof()
go bottom
do carga_v1
do d_reg1
avisos("Ultimo Registro",2,cColLin3)
endif
if bof()
go top
do carga_v1
do d_reg1
avisos("Primer Registro
",2,cColLin3)
endif
return

* EOF PANT_TES.PRO
*****
procedure ponSetRelation
//activar set relation
close databases
sele 8
use mater2 index clvmat2,mater2
alias ocho
sele 7
use colec index col1 alias siete
sele 6
use opc_tit index clv_opc,opc1 alias
seis
sele 5
use mater index clv_mat,materia alias
cinco
sele 4
use gen index clv_gen,gene alias
cuatro
sele 3
use carrera index clv_car,carr alias
tres
sele 2
use asesor index clv_ase,ases alias
dos
sele 1
use tes_msc index
tes1_msc,cuenta,clv alias uno
set relation to str(uno->clv_ase,3)
into dos, str(uno->clv_car,2) into
tres,
str(uno->clv_gen,2) into cuatro,
str(uno->clv_mat,1,2) into cinco,
str(uno->clv_opc,1) into seis,uno-
>alum into siete,
str(uno->clv_mat,2,2)
into ocho
return
*****
FUNCTION noDup(clvReg)
par=1.
cCol_act=setcolor()
no_reg=recno()
dFec_reg=uno->freg
dFec_ter=uno->fiter

```

```

@ 9,3 SAY 'Dirección Trabajo' get
cDiret4 pict '@a50@!:@Kx(80)'
@ 10,3 SAY 'Tel. Casa' get cTeled4
pict '@!:@Kx(12)'
@ 10,35 SAY 'Tel. Trabajo' get
cTeled4 pict '@!:@Kx(12)'
read
if lastKey()=27
do grb_cjt
setcolor(cColAct)

restscreen(0,0,maxxow(),maxcol(),cP
ant3)
return
endif
if cNom4<>space(40)

RemtesConj(cNom4,cCta4,cDiret4,c
Diret4,cTeled4,cTeled4)
endif
restscreen(0,0,maxxow(),maxcol(),cP
ant3)

m_sombra(3,2,11,75,1)
devpos(5,25);devout(TE S I S C O
N J U N T A S)
@ 6,3 SAY 'Autor 5' get cNom5 pict
'@!:@Kx(40)';
valid checDup(cNom5)
@ 6,53 SAY 'Cuenta' get cCta5 pict
'@R 9999999-9';
valid val_cuenta(cCta5)
@ 7,3 SAY 'Carrera' get cCarr5 pict
'@!:@Kx(20)';
when
escribe_aux('carrera','carr','3','tres',cIv
_car,'CARRERAS');
valid ponletc()
@ 7,33 SAY 'Generación' get
cGene5 pict '@!:@Kx(5)';
when
escribe_aux('gener','gene','4','cuatro',c
lv_gen);
'clv_car=carr_aux','GENERACION
')

@ 8,3 SAY 'Dirección Casa' get
cDiret5 pict '@a50@!:@Kx(80)'
@ 9,3 SAY 'Dirección Trabajo' get
cDiret5 pict '@a50@!:@Kx(80)'
@ 10,3 SAY 'Tel. Casa' get cTeled5
pict '@!:@Kx(12)'
@ 10,35 SAY 'Tel. Trabajo' get
cTeled5 pict '@!:@Kx(12)'
read
if lastKey()=27
do grb_cjt
setcolor(cColAct)

restscreen(0,0,maxxow(),maxcol(),cP
ant3)
return
endif
if cNom5<>space(40)

RemtesConj(cNom5,cCta5,cDiret5,c
Diret5,cTeled5,cTeled5)
endif
do grb_cjt // aqui la corrección

```

```

restscreen(0,0,maxxow(),maxcol(),cP
ant3)
return
*****
*****
//ahora con la base alterna
COLEC.dbf
procedure grb_cjt

sele 7
append blank
sieto->co1:=m->alum
sieto->co2:=cNom2
sieto->co3:=cNom3
sieto->co4:=cNom4
sieto->co5:=cNom5
if empty(cNom2)
append blank
sieto->co1:=cNom2
sieto->co2:=m->alum
sieto->co3:=cNom3
sieto->co4:=cNom4
sieto->co5:=cNom5
endif
if empty(cNom3)
sieto->co1:=cNom3
sieto->co2:=m->alum
sieto->co3:=cNom2
sieto->co4:=cNom4
sieto->co5:=cNom5
endif
if empty(cNom4)
append blank
sieto->co1:=cNom4
sieto->co2:=m->alum
sieto->co3:=cNom2
sieto->co4:=cNom3
sieto->co5:=cNom5
endif
if empty(cNom5)
append blank
sieto->co1:=cNom5
sieto->co2:=m->alum
sieto->co3:=cNom2
sieto->co4:=cNom3
sieto->co5:=cNom4
endif
ponSetRelation() //aquí debe ser
return
*****
*****
function
RemtesConj(cNom,cCta,cDiret,cDire
d,cTeled,cTeled)
append blank
replace uno->alum with cNom
replace uno->cen with cCta
replace uno->freg with m->freg
replace uno->fler with m->fler
replace uno->flit with m->flit
replace uno->diret with cDiret
replace uno->cteled with cTeled
replace uno->clvr with m->clvr

replace uno->titulo with upper(m
>cMemo1)

```

```

replace uno->obj_gen with upper(m
>cMemo2)
replace uno->capitulo with
upper(m->cMemo3)
replace uno->resuman with
upper(m->cMemo4)
replace uno->sinodal with upper(m
>cMemo5)

```

```

replace uno->clv_ase with
busca_ase
replace uno->clv_car with
busca_car
replace uno->clv_gen with
busca_gen
replace uno->clv_opc with
busca_opc
replace uno->clv_mat1 with
busca_mat1
replace uno->clv_mat2 with
busca_mat2

```

```

COMMIT
return nil

```

```

*****
*****
function
busca_general(basArr,camArr,numSe
loc,numAlia,camCiv,cond,titulo)
*Exclente función
local nOpcCar:=1
local
cPanCar:=saveScreen(9,11,19,40)
local aCarr:={ }
local nReg:=reco()
local cColAct:=setcolor()
local busca
vacio=.f.
if len(titulo)>25
titulo=substr(titulo,1,24)
endif

```

```

aCarr=CAMP_ARR(basArr,camArr,
aCarr,cond)
if len(aCarr)=0
aviso('Opciones
vacias...');cColLin3)
DO CASE

```

```

case numAlia='dos'
busca_ase = 0
case numAlia='tres'
busca_car = 0
case numAlia='cuatro'
busca_gen = 0
case numAlia='cinco'
busca_mat1 = 0
case numAlia='ocho'
busca_mat2 = 0
case numAlia='seis'
busca_opc = 0
ENDCASE
ponSetRelation()
go nReg
vacio=.L
return vacio //nil
endif
ponSetRelation()
//*****
*****
setcolor(cColor3)
m_sombra(9,11,19,39,1)

```

```

@ 9,14 say Titulo //colo
nOpcCar:=ACHOICE(10,12,17,38,a
Carr,,"eleccion1)

if nOpcCar = 0
  restscreen(9,11,19,40,cPanCar)
  DO CASE
    case numAlias='dos'
      busca_asc = 0
    case numAlias='tres'
      busca_car = 0
    case numAlias='cuatro'
      busca_gen = 0
    case numAlias='cinco'
      busca_mat1 = 0
    case numAlias='ocho'
      busca_mat2 = 0
    case numAlias='scis'
      busca_opc = 0
  ENDCASE
  return .t. //vacio
endif

sele (numSelec)
set order to 2
seek (numAlias)
>(aCarr[nOpcCar])
IF FOUND()
  busca_aux=aCarr[nOpcCar]
  endif
  set order to 1
  busca=(numAlias)->&campClv
  ponSetRelation()
  go nReg //indispensable
  do case
    case numAlias='dos'
      busca_asc = busca
    case numAlias='tres'
      busca_car = busca
    case numAlias='cuatro'
      busca_gen = busca
    case numAlias='cinco'
      busca_mat1 = busca
    case numAlias='ocho'
      busca_mat2 = busca
    case numAlias='scis'
      busca_opc = busca
  endcase
  setcolor(cColAct)
  restscreen(9,11,19,40,cPanCar)
  aCarr:= {} //para liberar toda la
  memoria que se pueda
  return vacio //nil
  *****
  funcion
  escribe_aux(base,campo,sele,alias,ca
  m_clv,cond,TITULO)
  *****
  *num es para la base
  mater y para grabar clv_mat1 y
  clv_mat1.
  local nKey:=lastkey()
  local
  cPant:=savescree(0,0,maxrow(),max
  col())
  do QuitaSet
  vacio =
  busca_general(base,campo,sele,alias,
  cam_clv,cond,titulo)
  if vacio
    keyboard chr(25)+
chr(13) //nKey chr(25)
do PonSet
return .t.
endif
keyboard chr(25)+ busca_aux +
chr(13) //+ chr(nKey)

restscreen(0,0,maxrow(),maxcol(),cP
ant)
busca_aux = '' //para evitar letreros
en otros campos
do PonSet //Key 28 to editmemo1
return .t.
*****
procedure poncaja1
!QuitaPon1 = .not. (!QuitaPon1)
if !QuitaPon1
  @ 24,73 say 'O' colo cColor4 // pi A
else
  @ 24,73 say ''
endif
return
*****
procedure poncaja2
!QuitaPon2 = .not. (!QuitaPon2)
if !QuitaPon2
  @ 24,74 say 'C' colo cColor4
else
  @ 24,74 say ''
endif
return
*****
procedure poncaja3
!QuitaPon3 = .not. (!QuitaPon3)
if !QuitaPon3
  @ 24,75 say 'G' colo cColor4
else
  @ 24,75 say ''
endif
return
*****
procedure poncaja4
!QuitaPon4 = .not. (!QuitaPon4)
if !QuitaPon4
  @ 24,76 say 'A' colo cColor4
else
  @ 24,76 say ''
endif
return
*****
procedure poncaja5
!QuitaPon5 = .not. (!QuitaPon5)
if !QuitaPon5
  @ 24,77 say 'M' colo cColor4
else
  @ 24,77 say ''
endif
return
*****
procedure editMemo1
local cColAct:=setcolor()
local
part:=savescree(0,0,maxrow(),maxc
ol())
local nCursor:=setcursor()
local cP1:=salva_mens(0)
if !!Mues_mem
  setcursor(0)
  mensaje(0,ESC=Salir,cColLin1)
else
  setcursor(1)
  mensaje(0,ESC=Salir Ctrl-
W=Salvar,cColLin1)
endif
do QuitaSet
setcolor(cColor4)
m_sombra(9,2,15,45..L)
@ 9,19 say 'TITULO'
cMemo1:=memoedit(cMemo1,10,3,1
4,44,IMues_mem,"memofun",40)
setcolor(cColAct)
setcursor(nCursor)
restscreen(0,0,maxrow(),maxcol(),pa
nt)
do PonSet
rest_mens(0,cP1)
return
*****
procedure editMemo2
local cColAct:=setcolor()
local
part:=savescree(0,0,maxrow(),maxc
ol())
local nCursor:=setcursor()
local cP1:=salva_mens(0)
if !!Mues_mem
  setcursor(0)
  mensaje(0,ESC=Salir,cColLin1)
else
  setcursor(1)
  mensaje(0,ESC=Salir Ctrl-
W=Salvar,cColLin1)
endif
do QuitaSet
setcolor(cColor4)
m_sombra(9,2,15,45..L)
@ 9,14 say 'OBJETIVO GENERAL'
cMemo2:=memoedit(cMemo2,10,3,1
4,44,IMues_mem,"memofun",40)//L
setcolor(cColAct)
setcursor(nCursor)
restscreen(0,0,maxrow(),maxcol(),pa
nt)
do PonSet
rest_mens(0,cP1)
return
*****
procedure editMemo3
local cColAct:=setcolor()
local
part:=savescree(0,0,maxrow(),maxc
ol())
local nCursor:=setcursor()
local cP1:=salva_mens(0)
if !!Mues_mem
  setcursor(0)
  mensaje(0,ESC=Salir,cColLin1)
else
  setcursor(1)

```

```

mensaje(0,ESC=Salir Ctrl-
W=Salvar,cColLin1)
endif
do QuitaSet
setcolor(cColor4)
m_sombra(9,2,15,45...L)
@ 9,18 say 'CAPITULOS'
cMemo3:=memoedit(cMemo3,10,3,1
4,44,1Mues_mem,"memofun",40)
setcolor(cColAct)
setcursor(nCursor)
restscreen(0,0,maxxrow(),maxxcol(),pa
nt)
do PonSet
rest_mens(0,cP1)
return
*****
procedure editMemo4
local cColAct:=setcolor()
local
part:=saveScreen(0,0,maxxrow(),maxx
ol())
local nCursor:=setcursor()
local cP1:=salva_mens(0)
if !!Mues_mem
setcursor(0)
mensaje(0,ESC=Salir,cColLin1)
else
setcursor(1)
mensaje(0,ESC=Salir Ctrl-
W=Salvar,cColLin1)
endif
do QuitaSet
setcolor(cColor4)
m_sombra(9,2,15,45...L)
@ 9,19 say 'RESUMEN'
cMemo4:=memoedit(cMemo4,10,3,1
4,44,1Mues_mem,"memofun",40)
setcolor(cColAct)
setcursor(nCursor)
restscreen(0,0,maxxrow(),maxxcol(),pa
nt)
do PonSet
rest_mens(0,cP1)
return
*****
procedure editMemo5
local cColAct:=setcolor()
local
part:=saveScreen(0,0,maxxrow(),maxx
ol())
local nCursor:=setcursor()
local cP1:=salva_mens(0)
if !!Mues_mem
setcursor(0)
mensaje(0,ESC=Salir,cColLin1)
else
setcursor(1)
mensaje(0,ESC=Salir Ctrl-
W=Salvar,cColLin1)
endif
do QuitaSet
setcolor(cColor4)
m_sombra(9,2,15,45...L)
@ 9,19 say 'SINODALES'
cMemo5:=memoedit(cMemo5,10,3,1
4,44,1Mues_mem,"memofun",40)
setcolor(cColAct)
setcursor(nCursor)
restscreen(0,0,maxxrow(),maxxcol(),pa
nt)
do PonSet

```

```

rest_mens(0,cP1)
return
*****
function
memofun(nMode,nLine,nCol)
local nKey:=lastkey()
local resp:='N'
local nRetVal:=0 //default
do case
case nMode == 0
@@ 20,3 say "memo est. en inactivo"
case nMode == 1
@@ 20,5 say "memo est. inalterado"
case nMode == 2
@@ 20,5 say "memo est. alterado"
do case
case nKey==K_F2
nRetVal:= ME_WORDRIGHT
case nKey==K_F3
nRetVal:=
ME_BOTTTOMRIGHT
case nKey==K_F4
nRetVal:=
ME_TOGGLESCROLL //SCROLL
modo
case nKey==K_F5
nRetVal:= ME_TOGGLEWRAP
case nKey==K_ESC
get_optn(,,,Salir sin salvar
,cColor4)
do case
case eleccion=2
nRetVal := 32
case eleccion=1
return 0 //nRetVal:=default
endcase
otherwise
nRetVal=0 @@ 20,5 say "memo
est. en inactiva"
endcase
return nRetVal
*****
PROCEDURE ponSet
set key -4 to editMemo1 //15
set key -5 to editMemo2 //16
set key -6 to editMemo3 //17
set key -7 to editMemo4 //18
set key -8 to editMemo5 //19
return
*****
PROCEDURE QuitaSet
set key -4 to //15
set key -5 to //16
set key -6 to //17
set key -7 to //18
set key -8 to //19
return
*****
function checDup(civ) //para las alitas
local pasa := L
local nReg:=reano()
if empty(civ)
return .f.
endif
sele 1 //por si acaso
civ:=strtran(civ,":")
civ:=strtran(civ,":")
civ:=strtran(civ,":")

```

```

seek trim(civ)
if found() //and. nReg <=
reano()
aviso("Clave repetida.....",3,cColLin3)
pasa=.f.
else
pasa=.t.
endif
go nReg
return pasa
*****
function tiempOpc(f1,f2)
local pasa:=.f.
local val1,val2
local
fch1,fch2,mes,ano,fecha,fch_real
pasa := iff(f1 < 2 .OR.
EMPTY(f2),.t,.f)
if .not. pasa
aviso("Fecha
incorrecta",2,cColLin3,2)
return pasa
endif
do case
case "TESIN" $ M->cOpc // aux
= 2
//tesinas
fecha := dtoc(f1)
fch1 := substr(dtoc(f1),4,5)
fch2 := substr(dtoc(f1),7,8)
val1 = val(fch1)
val2 = val(fch2)
val1 = val1+6
if val1 > 12
val1 = val1 - 12
val2 = val2 + 1
endif
mes = str(val1,2)
ano = str(val2,2)
fecha = stuff(fecha,4,2,mes)
fecha = stuff(fecha,7,2,ano)
fch_real = ctoc(fecha)
if f2 > fch_real
aviso("PLAZO EXCEDIDO
!!!",0,cColPar1,3)
endif
case "MEMOR" $ M->cOpc
//nOpc_aux = 3
//memorias
fch2 := substr(dtoc(f1),7,8)
fecha := dtoc(f1)
val2 = val(fch2)
val2 = val2 + 1
ano = str(val2)
fchd_real=ctoc(fecha)
fchd_real=ctoc(fecha)
if f2 > fchd_real
aviso(" PLAZO EXCEDIDO
!!!",0,cColPar1)
endif
otherwise
pasa = .t.
endcase
return pasa
*****
function ponle()
local pasa:=.f.
nCurr_aux = busca_car
pasa=.L
return pasa

```

```

*****
function Vld_fecha2(f1,f2)
local pasa:=.f.
local pasa := iff(f1 < f2,OR.
EMPTY(f2),t...f)
if .not. pasa
aviso("Fecha incorrecta",2,2)
return pasa
endif
return pasa
*****
*
procedure autor_cjt

local nReg:=recoo()
local
cPant3:=savescreen(0,0,maxrow(),max
xool())
local cColAct:=setcolor()
local lAgrega:=.f.
setcolor(cColLin2)

do QuitaSet
cNom2:=siete->co2
cNom3:=siete->co3
cNom4:=siete->co4
cNom5:=siete->co5

cCta2:=uno->cuen
cCarr2:=tres->carr
cGene2:=cuatro->gene
cDire2:=uno->direT
cDire2:=uno->direD
cTeled2:=uno->teleT
cTeled2:=uno->teleD
busca_car = uno->clv_car
busca_gen = uno->clv_gen

sele 1
seek cNom2
if !found()
lAgrega:=.t.
endif
do miniCarga
m_sombra(5,2,11,75...t)
depos(5,25);devout("T E S I S C O
N J U N T A S")
@ 6,3 SAY 'Autor 2 ' get cNom2 pict
'@!@Kx(40);
valid checDup1(cNom2) //and.
nReg = recoo()
@ 6,3 SAY 'Cuenta ' get cCta2 pict
'@R 9999999-9'

@ 7,3 SAY 'Carrera ' get cCarr2 pict
'@!@Kx(20);
when
escribe_aux('carrera','carr','3','tres','clv
_car','CARRERAS');
valid ponle()

@ 7,33 SAY 'Generacion ' get
cGene2 pict ' @!@Kx(5);
when
escribe_aux('gener','gene','4','cuatro','c
lv_gen';

'clv_car=nCarr_aux','GENERACION
')
@ 8,3 SAY 'Dirección Casa ' get
cDire2 pict ' @s50@!@Kx(80)'
@ 9,3 SAY 'Dirección Trabajo ' get
cDire2 pict ' @s50@!@Kx(80)'
@ 10,3 SAY 'Tel. Casa ' get cTeled2
pict ' @!@Kx(12)'
@ 10,35 SAY 'Tel. Trabajo ' get
cTeled2 pict ' @!@Kx(12)'
read
if !lastkey()=27

*RempConj(cNom3,cCta2,cDire2,c
Dire2,cTeled2,cTeled2,lAgrega)
do grb_conj
setcolor(cColAct)

restscreen(0,0,maxrow(),maxxool(),cP
ant3)
go nReg
return
endif

RempConj(cNom3,cCta2,cDire2,cD
ired2,cTeled2,cTeled2,lAgrega)
lAgrega:=.f.

restscreen(0,0,maxrow(),maxxool(),cP
ant3)
sele 1
seek cNom4
if !found()
lAgrega:=.t.
endif
do miniCarga
m_sombra(5,2,11,75...t)
depos(5,25);devout("T E S I S C O
N J U N T A S")
@ 6,3 SAY 'Autor 4 ' get cNom4 pict
'@!@Kx(40);
valid checDup1(cNom4)
@ 6,33 SAY 'Cuenta ' get cCta2 pict
'@r 9999999-9'
@ 7,3 SAY 'Carrera ' get cCarr2 pict
'@!@Kx(20);
when
escribe_aux('carrera','carr','3','tres','clv
_car','CARRERAS');
valid ponle()
@ 7,33 SAY 'Generacion ' get
cGene2 pict ' @!@Kx(5);
when
escribe_aux('gener','gene','4','cuatro','c
lv_gen';
'clv_car=nCarr_aux','GENERACION
')

@ 8,3 SAY 'Dirección Casa ' get
cDire2 pict ' @s50@!@Kx(80)'
@ 9,3 SAY 'Dirección Trabajo ' get
cDire2 pict ' @s50@!@Kx(80)'
@ 10,3 SAY 'Tel. Casa ' get cTeled2
pict ' @!@Kx(12)'
@ 10,35 SAY 'Tel. Trabajo ' get
cTeled2 pict ' @!@Kx(12)'
read
if !lastkey()=27

*RempConj(cNom3,cCta2,cDire2,c
Dire2,cTeled2,cTeled2,lAgrega)
do grb_conj
setcolor(cColAct)

restscreen(0,0,maxrow(),maxxool(),cP
ant3)
sele 1
seek cNom5
if !found()

```

```

lAgrega:=t
endif
do miniCarga

m_sombra(5,2,11,75...t.)
devpos(5,25);devout(T E S I S C O
N F U N T A S)
@ 6,3 SAY 'Autor 5 ' get cNom5 pict
'@!:@Kx(40);
  valid checDup1(cNom5)
@ 6,33 SAY 'Cuenta ' get cCta2 pict
'@R 9999999-9'
@ 7,3 SAY 'Carrera ' get cCarr2 pict
'@!:@Kx(20);
  when
  escribe_aux('carrera','carr','3','tres','clv
_car','CARRERAS');
  valid ponle()
@ 7,33 SAY 'Generacion ' get
cGene2 pict ' @!:@Kx(5);
  when
  escribe_aux('gener','gene','4','cuatro','c
lv_gen';
'clv_car'=nCarr_aux,'GENERACION
')

@ 8,3 SAY 'Direccion Casa ' get
cDire2 pict ' @a50:@!:@Kx(80)
@ 9,3 SAY 'Direccion Trabajo ' get
cDire2 pict ' @a50:@!:@Kx(80)
@ 10,3 SAY 'Tel. Casa ' get cTeled2
pict ' @!:@Kx(12)
@ 10,33 SAY 'Tel. Trabajo ' get
cTeled2 pict ' @!:@Kx(12)
read
if lastkey)=27

*RempConj(cNom5,cCta2,cDire2,c
Dire2,cTeled2,cTeled2,lAgrega)
do grb_conj
setcolor(cColAct)

restscreen(0,0,maxrow(),maxcol(),cP
am3)
go nReg
return
endif

RempConj(cNom5,cCta2,cDire2,cD
ired2,cTeled2,cTeled2,lAgrega)
lAgrega:=f.
do grb_conj
setcolor(cColAct)
go nReg

restscreen(0,0,maxrow(),maxcol(),cP
am3)
return
*****
procedure miniCarga
*Carga solo unos cuantos registros

cCta2=uno->cuon
cCarr2='tres'->carr
cGene2='cuatro'->gene
cDire2=uno->direT
cDire2=uno->direD
cTeled2=uno->teleT
cTeled2=uno->teleD
busca_car = uno->clv_car
busca_gen = uno->clv_gen

return
*****
procedure grb_conj
sele 7
seek m->alum
siete->co1:=m->alum
siete->co2:=cNom2
siete->co3:=cNom3
siete->co4:=cNom4
siete->co5:=cNom5

if !empty(cNom2)
seek cNom2
if .not. found()
append blank
endif
siete->co1:=cNom2
siete->co2:=m->alum
siete->co3:=cNom3
siete->co4:=cNom4
siete->co5:=cNom5
endif

if !empty(cNom3)
seek cNom3
if .not. found()
append blank
endif
siete->co1:=cNom3
siete->co2:=m->alum
siete->co3:=cNom2
siete->co4:=cNom3
siete->co5:=cNom5
endif

if !empty(cNom4)
seek cNom4
if .not. found()
append blank
endif
siete->co1:=cNom4
siete->co2:=m->alum
siete->co3:=cNom2
siete->co4:=cNom3
siete->co5:=cNom5
endif

if !empty(cNom5)
seek cNom5
if .not. found()
append blank
endif
siete->co1:=cNom5
siete->co2:=m->alum
siete->co3:=cNom2
siete->co4:=cNom3
siete->co5:=cNom4
endif

ponStrRelation() aqu; debe ser
return
*****
function
RempConj(cNom,cCta,cDire,cDire,
cTeled,cTeled,lAgrega)
if lAgrega
append blank
endif

replace uno->alum with cNom
replace uno->cuon with cCta
replace uno->freg with m->freg
replace uno->iter with m->iter

replace uno->titulo with upper(m-
>cMemo1)
replace uno->obj_gen with upper(m-
>cMemo2)
replace uno->capitulo with
upper(m->cMemo3)
replace uno->resumen with
upper(m->cMemo4)
replace uno->sinodal with upper(m-
>cMemo5)

replace uno->clv_ase with
busca_ase
replace uno->clv_car with
busca_car
replace uno->clv_gen with
busca_gen
replace uno->clv_ope with
busca_ope
replace uno->clv_mat1 with
busca_mat1
replace uno->clv_mat2 with
busca_mat2

COMMIT
return nil
*****
function checDup1(clv)
//modificaciones
local pasa := t.
local nReg :=reco()
if empty(clv)
return f.
endif

sele 1 //por si acaso
clv=strtran(clv,' ');
clv=strtran(clv,' ');
clv=strtran(clv,' ');

seek trim(clv)
if found() .and. nReg <> reco()
aviso('Clave
repetida.....',3,cColPar1)
pasa=.f.
else
pasa=.t.
endif
go nReg
return pasa □ □ □
*****
procedure elim_conj
sele 7
seek m->alum //busca en colee->co1
if found()
delete
pack
endif
go top
while !eof()
if siete->co2 = m->alum
siete->co2 := ''
endif
if siete->co3 = m->alum

```



```

*****
procedure carrera
local nOpcCar:=1
local
cPanCar:=savescreen(13,15,18,31)
local aCarr:={}
local cColAxt:=setcolor()

nCarr=1
base='carrera'
campo='curr'
cTextCar = ''
cTextMat = ''
cTextGen = ''

cTextCarM = ''
cTextMatM = ''
cTextGenM = ''

scroll(1,40,1,maxcol())
scroll(2,40,2,maxcol()) //borra
letrero de tema
scroll(7,40,7,maxcol()) //borra
letrero de generacion
setcolor(cColor4) //Caj2

aCarr=CAMP_ARR(base,campo,aCarr)
m_sombra(13,15,17,30,...L)

nOpcCar:=ACHOICE(14,16,16,29,aCarr,'eleccion1')
if nOpcCar<=0
    CHC_NTV(base,campo)
    seek aCarr[nOpcCar]
    cTextCar:='clv_car'+str(&base->clv_car,2)
    cTextCarM:='Carrera = '+substr(&base->carr,1,18)
    nCarr:=&base->clv_car
    use
    restscreen(13,15,18,31,cPanCar)
    setcolor(cColAxt)
    @ 1,40 say cTextCarM //+
str(nCarr,2)
else
    restscreen(13,15,18,31,cPanCar)
endif
ponSetRelation()
setcolor(cColAxt)
aCarr:={} //para liberar toda la memoria que se pueda
return
*****
procedure materia
local nOpcMat:=1
local
cPanMat:=savescreen(14,15,21,36)
local aMateria:={}
local cColAxt:=setcolor()

cTextMat:=''
cTextMatM:=''
cTextMatM:=''
campo='materia'
campo='materia'
condicion='clv_car' nCarr'
scroll(2,40,2,maxcol())
setcolor(cColor4)
if cTextCar='''
    aviso('Elegir antes una carrera
    !!!2,cColLin3)
    setcolor(cColAxt)
    return
endif
aMateria=CAMP_ARR(base,campo,
aMateria,condicion)
if empty(aMateria)
    aviso('Carrera sin materias
    disponibles...2,cColLin3)
    setcolor(cColAxt)
    return
endif
m_sombra(14,15,20,35,...L)

nOpcMat:=ACHOICE(15,16,19,34,aMateria,'eleccion1')
if nOpcMat<=0
    CHC_NTV(ordem,campo)
    seek aMateria[nOpcMat]
    cTextMat:='clv_mat1'+
str(ordem->clv_mat,2)+'or,
clv_mat2'+':
str(ordem->clv_mat,2)

cTextMatM:='Materia = '+
substr(ordem->materia,1,20)
use
restscreen(14,15,21,36,cP
anMat)
setcolor(cColAxt)
@ 2,40 say cTextMatM
//+ aMateria[nOpcMat]
else
    restscreen(14,15,21,36,cP
anMat)
endif
setcolor(cColAxt)
ponSetRelation()

aMateria:={}
return
*****
procedure asesor
local nOpcAse:=1
local
cPanAse:=savescreen(14,15,23,61)
LOCAL aAsesor:={}
local cColAxt:=setcolor()

cTextAse:= ''
cTextAseM:= ''
base='asesor'
campo='ases'
scroll(3,40,3,maxcol())
setcolor(cColor4)
CAMP_ARR(base,campo,aAsesor)
m_sombra(14,15,22,60,...L)

nOpcAse:='ACHOICE(15,16,21,59,aAsesor,'eleccion1')
if nOpcAse<= 0
    CHC_NTV(base,campo)
    seek aAsesor[nOpcAse]
    cTextAse:='clv_ase'+
str(&base->clv_ase,3)
    cTextAseM:='Asesor = '+
substr(&base->ases,1,38)
    use
    restscreen(14,15,23,61,cP
anAse)
setcolor(cColAxt)
@ 3,40 say cTextAseM .
else
    restscreen(14,15,23,61,cPanAse)
endif
setcolor(cColAxt)
ponSetRelation()
return
*****
procedure fec_reg
local nOpcFre:=1
local dFecha:=dtoc(date())
local
cPanFre:=savescreen(15,20,21,26)
local cTextFre1
local cPanFre1
local cColAxt:=setcolor()
local cP1:=salva_mena(24)

if ++nVecFre>2
    cTextFre='''
cTextFreM = ''
scroll(4,39,4,maxcol())
setcolor(cColor4)
nVecFre=1
endif
mensaje(24,'ESC=Salir
F1=Ayuda','cColLin1)

aFre:={'>','<','>','<','>','<'}
setcolor(cColor4)
m_sombra(15,20,25,...L)

nOpcFre:=ACHOICE(16,21,19,24,aFre,'eleccion1')
if nOpcFre<=0
    cTextFre1:='Freg '+
aFreg[nOpcFre]
cTextFre = cTextFre +
iii(nVecFre >= 2).and. '+
cTextFre1,cTextFre1)
cTextFreM = cTextFreM +
iii(nVecFre >= 2).y'+
cTextFre1,cTextFre1)
restscreen(15,20,21,26,cPanFre)
setcolor(cColAxt)
@ 4,39 say cTextFreM
@ 22,32 say 'Fecha->' get dFecha
pic 99/99/99;
valid acomFecha(dFecha)
read
if lastkey()=27 //ESC
scroll(22,32,22,maxcol())
cTextFre = ''
cTextFreM = ''
scroll(4,39,4,maxcol())
nVecFre=0
rest_mena(24,cP1)
setcolor(cColAxt)
return
endif
scroll(22,32,22,maxcol())
restscreen(15,20,21,26,cPanFre)
cTextFre:='Freg '+
++(dFecha) + '+ ' //dtoc(dFecha)
cTextFreM:='Freg '+ dFecha
@ 4,39 say cTextFreM
else
    cTextFre = ''

```



```

cTextFteM=""
setcolor(cColAct)
scroll(4,39,4,maxcol())
nVecFreg=0
endif
setcolor(cColAct)
restscreen(15,20,21,26,cPanFte)
res_mes(24,cP1)
aFreg={ }
return
*****
procedure fce_ter
local nOpeFte:=1
local dFecha:=dtoc(date())
local
cPanFte:=savescreeen(16,20,22,26)
local cTextFte1
local cColAct:=setcolor()
local cP1:=salva_mes(24)

if ++nVecFter>2
cTextFte=""
cTextFteM=""
scroll(5,39,5,maxcol())
setcolor(cColor4)
nVecFter=1
endif
mensaje(24,'ESC=Salir
F1=Ayuda',cColLin1)
aFter:={ '>','<','>','<','>','<','>'}
setcolor(cColor4)
m_sombra(16,20,21,25,...L)

nOpeFte:=ACHOICE(17,21,20,24,a
Fte,'eleccion1')
if nOpeFte<=0
cTextFte1:=Fte+'
aFter[nOpeFte]
cTextFte=cTextFte+
iif(nVecFter >= 2,'and, '+
cTextFte1,cTextFte1)
cTextFteM=cTextFteM+
iif(nVecFter >= 2,'y '+
cTextFte1,cTextFte1)
restscreen(16,20,22,26,cPanFte)
setcolor(cColAct)
@ 5,39 say cTextFteM
@ 22,32 say 'Fecha->' get dFecha
pict '99:99:99':
valid aoomFecha(dFecha)
read
if lastkey()=27 /ESC
scroll(22,32,22,maxcol())
cTextFte=""
cTextFteM=""
scroll(5,39,5,maxcol())
nVecFter=0
res_mes(24,cP1)
setcolor(cColAct)
return
endif
scroll(22,32,22,maxcol())
restscreen(16,20,22,26,cPanFte)
cTextFte:=cTextFte+'<od
('==-(dFecha)=="' /dtoc(dFecha)
cTextFteM:=cTextFteM+dFecha
@ 5,39 say cTextFteM
else
cTextFte=""
cTextFteM=""
setcolor(cColAct)
scroll(6,39,6,maxcol())
nVecFter=0
endif
setcolor(cColAct)
restscreen(16,20,22,26,cPanFte)
res_mes(24,cP1)
aFter={ }
return
*****
procedure fce_tit
local nOpeFti:=1
local dFecha:=dtoc(date())
local
cPanFti:=savescreeen(16,20,22,26)
local cPanFti1
local cColAct:=setcolor()
local cP1:=salva_mes(24)

if ++nVecFti>2
cTextFti=""
cTextFtiM=""
scroll(6,39,6,maxcol())
setcolor(cColor4)
nVecFti=1
endif
mensaje(24,'ESC=Salir
F1=Ayuda',cColLin1)
aFti:={ '>','<','>','<','>','<','>'}
setcolor(cColor4)
m_sombra(16,20,21,25,...L)

nOpeFti:=ACHOICE(17,21,20,24,aF
ti,'eleccion1')
if nOpeFti<=0
cTextFti1:=Fti+'
aFti[nOpeFti]
cTextFti=cTextFti+
iif(nVecFti >= 2,'and, '+
cTextFti1,cTextFti1)
cTextFtiM=cTextFtiM+
iif(nVecFti >= 2,'y '+
cTextFti1,cTextFti1)
setcolor(cColAct)
@ 6,39 say cTextFtiM
@ 22,32 say 'Fecha->' get dFecha
pict '99:99:99':
valid aoomFecha(dFecha)
read
if lastkey()=27 /ESC
scroll(22,32,22,maxcol())
cTextFti=""
cTextFtiM=""
scroll(6,39,6,maxcol())
nVecFti=0
setcolor(cColAct)
return
endif
scroll(22,32,22,maxcol())
restscreen(16,20,22,26,cPanFti1)
cTextFti:=cTextFti+'<od
('==-(dFecha)=="' + dFecha
@ 6,39 say cTextFtiM
else
cTextFti=""
cTextFtiM=""
setcolor(cColAct)
scroll(6,39,6,maxcol())
nVecFti=0
endif
endif
setcolor(cColAct)
restscreen(16,20,22,26,cPanFti)
res_mes(24,cP1)
aFti:={ }
return
*****
procedure generacion
local nOpeGen:=1
local
cPanGen:=savescreeen(17,25,23,41)
local aGeneracion:={ }
local cColAct:=setcolor()

cTextGen=""
cTextGenM=""
base="gener"
campo="gene"
condicion="clv_carr=nCarr"
scroll(7,40,7,maxcol())
setcolor(cColor4)
if cTextCar=""
avisos('Elegir antes una carrera
!!!',2,cColLin3)
setcolor(cColAct)
return
endif

aGeneracion=CAMP_ARR(base,cam
po,aGeneracion,condicion)
if empty(aGeneracion)
avisos('Carrera sin
generaciones...',2,cColLin3)
setcolor(cColAct)
ponSetRelation()
return
endif

m_sombra(17,25,22,40,...t)

nOpeGen:=ACHOICE(18,26,21,39,a
Generacion,'eleccion1')
if nOpeGen<=0
CHC_NTX(orden,campo
)
seek
aGeneracion[nOpeGen]
cTextGen:="clv_gen"+
str(orden->clv_gen,2)
cTextGenM:="Generacion
" + substr(orden->gene,1,5)
use
restscreen(17,25,23,41,cP
anGen)
setcolor(cColAct)
@ 7,40 say cTextGenM
// aGeneracion[nOpeGen]
else
restscreen(17,25,23,41,cPanGen)
endif
setcolor(cColAct)
ponSetRelation()

aGeneracion:={ }
return
*****
procedure ope_tit
local nOpeTit:=1

```

```

local
cPanTit:=savescreen(12,15,17,51)
local aOpc_tit:={ }
local cColAct:=setcolor()

scroll(8,40,8,maxcol())
setcolor(cColor4)

cTextOpc:=''
cTextOpcM:=''

base='opc_tit'
campo='opct'

aOpc_tit:=CAMP_ARR(base,campo,
aOpc_tit)
m_sombra(12,15,16,50,...t)

nOpcTit:=ACHOICE(13,16,15,49,a
Opc_tit,'eleccion1')
if nOpcTit<=0
  CHC_NTX(base,campo)
  seek aOpc_Tit[nOpcTit]
  cTextOpc:='clv_opc'
  str(&base>clv_opc,1)
  cTextOpcM:='Opciones'
  substr(&base>opct,1,39)
  use
  restscreen(12,15,17,51,cPanTit)
  setcolor(cColAct)
  @ 8,40 say cTextOpcM
else
  restscreen(12,15,17,51,cPanTit)
endif
setcolor(cColAct)
ponSetRelation()
aOpc_tit:={ }
return

*****
procedure reporte
local nCur:=setcursor()
local cColOrig:=setcolor()
local
pant1:=savescreen(0,0,maxrow(),max
col())
filt_rep:=''
cnt_opc:=0
if !empty(cTextCar) // <> ''
  //si, si escogieron esta opcion que
  acumule cadena
  filt_rep = filt_rep + cTextCar
  ++ cnt_opc
endif
if !empty(cTextMat) // <> ''
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextMat,cTextMat)
endif

if !empty(cTextAse) // <> ''
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextAse,cTextAse)
endif
if !empty(cTextFre) // <> ''
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextFre,cTextFre)
endif
if !empty(cTextPti)
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextPti,cTextPti)
endif

if !empty(cTextPte)
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextPte,cTextPte)
endif
if !empty(cTextGen)
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextGen,cTextGen)
endif
if !empty(cTextOpc)
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextOpc,cTextOpc)
endif
if !empty(cTextSin)
  filt_rep = filt_rep + iif(++cnt_opc
  >= 2,.and.' + cTextSin,cTextSin)
endif

use orden alias unico
endif

if campo=nil
  campo=field(1)
endif
unico->(dbgotop())

while !eof()
  cCamp:=&campo
  add(&arreglo,cCamp)
  dbskip()
enddo
use
return arreglo*/

*****
function
camp_arr(base,campo,arreglo,cond)
close databases
if cond=nil
  use &base //alias unico
  sort on &campo to orden1
  use orden1 alias unico
else
  use &base
  copy to orden for &cond
  use orden
  sort on &campo to orden1
  use
  use orden1 alias unico
  endif

if campo=nil
  campo=field(1)
endif
unico->(dbgotop())

while !eof()
  cCamp:=&campo
  add(&arreglo,cCamp)
  dbskip()
enddo
use
return arreglo

*****
function chc_ntx(base,campo)
  use &base
  index on &campo to &base
  use
  use &base index &base
  return nil
*****
procedure pantalla
local
cPanT:=savescreen(0,0,maxrow(),max
col())
if !file('repor.dbf')
  aviso('No hay reporte
  realizado...'.3,cColIn3,3)
else
  if !BasVacía
    aviso('No hay reporte
    realizado...'.3,cColIn3,3)
    return
  endif
  close databases //necesario
  use repor alias uno
  ponSet2Relat()
  db3()

```

```

restscreen(0,0,maxrow(),maxcol(),cP
arr)
ponSetRelation()
endif
return
*****
procedur db3
/*para ubedit de consulta
nombre : DB3.prg
autor : Armando Amador
Molina
fecha : 30-marzo-1995
compila : clipper color1 /m /p /v
/w /a /n
version : clipper 5.01
linkar : RThink FI color1 PLL
Fullbase
descripcion : Ventanas desde
clipper 5.0 p.p. 429
*/

#include "SSValues.ch"
#include "SSkeys.ch"
#include "SSfunct.ch"

local iContinue:=TRUE
local pan
local cColAct:=setcolor()
local aCampRep
setcolor(cColor3)
nBottom:=20
mensaje(24,ESC=Salir
R=Redimensionar
F1=Ayuda,cColLin1)

pan:=savescreen(0,0,maxrow(),maxc
ol())
aCampRep:={'Alum','Cuen','tres
>curr','c'at'ro
>gen','dira','dired','telet';

telod,'titulo','obj_gen','capitulo','resu
men','sinodal','reg','ier'.
'rit','clvr','seis>rega','doe
>ases','cinco>materia','ocho
>materia')

aEncabRep:={'Alumno','Cuenta','Carr
era','C'omarcien','Dir'ec'ion
Trabajo','Dir'ec'ion Domicilio','Tel
Trabajo';
'Tel. Domicilio','T'itulo','Obj
Gen.','Capitulo','Resumen','Sinodales',
'Fecha Reg.','Fecha Ter.';
'Fecha Tra.','Clave Reg.','Opc'ion de
Titulacion','Asesor','Materia
1','Materia 2'}

while iContinue
begin sequenc
restscreen(0,0,maxrow(),maxcol(),pa
n)
m sombra(4,0,nBottom,maxcol()
1...t)
uno->(dbedit(5,1,nBottom
1,maxcol()
2,aCampRep,"dbkeys",aEncabRep))
iContinue:=p.FALSE
end sequenc
enddo

```

```

setcolor(cColAct)
return(nil)
*****
funcion dbkeys(nMode,nColumn)
//modo de DBedit
//posicion de
columna
local nRetvalue:=1
local nKey :=lastkey()
local cScreen
local cColAct:=setcolor()
local cP1

do case
case nMode==0
@ 1,70 say str(reco(),3) colo
cColor1 //2
@ 1,73 say '/' + str(lastreo(),3) colo
cColor1

case nMode==1 .or. nMode==2
keyboard
chr(if(nMode==2,pCTRL_PGUP,pC
TRL_PGDN))

case nMode==3
aviso('Reporte vacio/ Cualquier
teda...',0,cColLin3)
nRetvalue:=0
otherwise
do case
case nKey==pESC
nRetvalue:=0
case nKey==pENTER
if
valtype(&(field(nColumn)))==pME
MO
cScreen := savescreen(5,5,18,36)
cP1:=salva_mens(24)

mensaje(24,ESC=Salir,cColLin1)
setcolor(cColor4)
m_sombra(5,5,17,35...t)
@ 5,17 say field(nColumn)

memoedit(fieldget(nColumn),6,6,16,
34,p.FALSE,")
//mensaje(24,ESC=Salir
R=Redimensionar
F1=Ayuda,cColLin1)
rest_mens(24,cP1)
setcolor(cColAct)
restscreen(5,5,18,36,cScreen)
CLEARESC()
endif
case chr(nKey) $ 'R'
CURSOR_NORMALO //pseudoo
c'xpression definida por el usuario
@ 24,00 say space(80) colo
cColLin1
@ 24,00 say 'El nuevo tamao es '
colo cColLin1;
get nBottom pict 'a' valid
nBottom >= 8 and nBottom <= 23
read
dhtopot()
break //complemento de begin
sequenc
endcase
endcase

return(nRetValue)

```

```

*****
funcion acomFecha(dFecha)
local iPass:=f
local mes,ano,dia

dia:=val(substr(dFecha,1,2))
mes:=val(substr(dFecha,4,2))
ano:=val(substr(dFecha,7,2))

if mes > 12
aviso('Fecha no v lida!',2,cColLin3)
return .f.
endif
if dia > 31
aviso('Fecha no v lida!',2,cColLin3)
return .f.
endif

if dia <= 31 .and. (mes=0 .or. mes=1
.or. mes=3 .or. mes=5 .or.;
mes=7 .or. mes=8 .or. mes=10
.or. mes=12)
iPass=.t.
endif
if dia <= 30 .and. (mes=0 .or.
mes=2 .or. mes=4 .or. mes=6 .or.;
mes=9 .or. mes=11)
iPass=.t.
endif
if mes=2 .and. dia > 29
aviso('Fecha no v lida!',2,cColLin3)
return .f.
endif

if iPass
aviso('Fecha no v lida!',2,cColLin3)
endif

return iPass
*****
procedure limpia
iBasVaci=.t.
cTextCarM:=""
cTextMat:=""
cTextAse:=""
cTextFre:=""
cTextPi:=""
cTextFte:=""
cTextOpc:=""
cTextSin:=""

cTextCarM:=""
cTextMatM:=""
cTextAseM:=""
cTextFreM:=""
cTextPiM:=""
cTextFteM:=""
cTextOpcM:=""
cTextSinM:=""

scroll(1,40,1,maxcol())
scroll(2,40,2,maxcol())
scroll(3,40,3,maxcol())
scroll(4,40,4,maxcol())
scroll(5,40,5,maxcol())

```

```

scroll(6,40,6,maxcol())
scroll(7,40,7,maxcol())
scroll(8,40,8,maxcol())
scroll(9,40,9,maxcol())

return

*****
Procedure ponSetRelat()

close databases
sele 8
use mater2 index clvm2.mater2
alias ocho
sele 7
use colec index col alias siete
sele 6
use opc_tit index clv_opc.opc alias
seis
sele 5
use mater index clv_mat.mater1 alias
cinco
sele 4
use gener index clv_gen.gene alias
cuatro
sele 3
use carrera index clv_car.car alias
tres
sele 2
use asesor index clv_ase.ase alias
dos
sele 1
use repor alias uno
set relation to str(uno->clv_ase.3)
into dos, str(uno->clv_car.2) into
tres,
str(uno->clv_gen.2) into cuatro,
str(uno->clv_mat.1.2) into cinco,
str(uno->clv_opc.1) into seis,uno-
>alun into siete,
str(uno->clv_mat.2.2)
into ocho
return
*****
procedure sinod

local nOpcSin:=1
local
cPanSin:=saveScreen(0,0,maxrow(),
maxcol())
eSinod:=space(25)

scroll(9,40,9,maxcol())
@ 21,32 say "Sinodal a buscar->"
get cSinod pic'@!@kx25)
valid iif(cSinod="*,f..L)
read
if lastkey()=27 //ESC
scroll(21,32,21,maxcol())
cTextSim=""
cTextSimM=""
scroll(9,40,9,maxcol())
return
endif
eSinod=strtran(cSinod,'*')
/para reducir a un solo espacio
cSinod=strtran(cSinod,'*') //la
variable
cSinod=strtran(cSinod,'*')

cTextSim:=trim(cSinod) $ sinodal'
cTextSimM:='Sinodal'+ eSinod
@ 9,40 say cTextSimM
scroll(21,32,21,maxcol())

return

Programa 5

/* programa para crear los menus
descendentes
para el sistema de tesis de la ENEP
Acañ in
AUTOR: ARMANDO AMADOR
MOLINA
fecha_inicio: 6-marzo-1995
fecha_Ult_Aut: 18-mayo-1996
programa para generar un menu
sobrepuesto
*/
*****
PROCEDURE BUS_PAL
*****
*****
*****
A command READ =>
ReadModal(GetList, nT);
GetList:={}
local cColAut:=setColor()
local cPanT
local
cPanT1:=saveScreen(0,0,maxrow(),lma
xcol())
local cP1:="salva_mens(0)
local cP2:="salva_mens(24)
cls

do tapiz

cPal1=space(14)
cPal2=space(14)
cPal3=space(14)
ISalir=.t.
ponSetRelation()
aNum_tes:={ }
si="S"
aVal:={,L..t..t..t..t..L}
*do opcBusq

while .t.

mensaje(0,' ELECCION DE
BUSQUEDA PARA PALABRAS
CLAVES',oColLin1)
do opcBusq
cPanT:=saveScreen(0,0,maxrow(),max
col())

if .NOT. ISalir
ponSetRelation()
setColor(cColAut)

restScreen(0,0,maxrow(),maxcol(),cP
ant1)
return
endif

setColor(cColor4)
M_SOMBRA(7,20,20,76,..L)
@ 8,21 say "BUSQUEDA DE
TRABAJOS REPETIDOS CON
PALABRAS CLAVES"
@ 10,21 say "PALABRA 1 =>"
GET cPAL1 pic'@!@k';
valid iif(m->cPal1="*,f..L)

@ 11,21 say "PALABRA 2 =>"
GET cPAL2 pic'@!@k';
@ 12,21 say "PALABRA 3 =>"
GET cPAL3 pic'@!@k';
@ 19,21 say "ESC=Salir"
read
if lastkey()=27 //escape
ponSetRelation()
setColor(cColAut)

restScreen(0,0,maxrow(),maxcol(),cP
ant)
loop //return
endif

cPantalla:="saveScreen(0,0,maxrow(),
maxcol())
sele 1
DBGOTOPO //CON ESTO SE
ARREGLA

VALOR=0
ULTIMO=LASTREC()
TERMIOH(VALOR)
T=0
C=0

set cursor off
while eof() .AND. trunc()
//empezar bfAqueda de palabras clave
en titulo de tesis.
if aVal[1]=.f.
if altrim(m->cPal1) $
uno->titulo .or. altrim(m->cPal2) $ ;
uno->titulo .or.
altrim(m->cPal3) $ uno->titulo
aadd(aNum_tes,uno-
>alun) //
endif
endif
if aVal[2]=.f.
if altrim(m->cPal1) $
uno->obj_gen .or. altrim(m->cPal2)
$ ;
uno->obj_gen .or.
altrim(m->cPal3) $ uno->obj_gen
aadd(aNum_tes,uno-
>alun) //
endif
endif
if aVal[3]=.f.
if altrim(m->cPal1) $
uno->capitulo .or. altrim(m->cPal2)
$ ;
uno->capitulo .or.
altrim(m->cPal3) $ uno->capitulo
aadd(aNum_tes,uno-
>alun) //
endif
endif
if aVal[4]=.f.
if altrim(m->cPal1) $
uno->resumen .or. altrim(m->cPal2)
$ ;
uno->resumen .or.
altrim(m->cPal3) $ uno->resumen
aadd(aNum_tes,uno-
>alun) //
endif
endif
if aVal[5]=.f.
if altrim(m->cPal1) $
uno->direr .or. altrim(m->cPal2) $ ;

```

```

        uno->diret.or.
alltrim(m->cPal3) $ uno->diret
add(aNum_tes,uno-
>alum) //
        endif
        endif
        if aVal[6]=.f.
        if alltrim(m->cPal1) $
uno->diret.or. alltrim(m->cPal2) $ ;
        uno->diret.or.
alltrim(m->cPal3) $ uno->diret
add(aNum_tes,uno-
>alum) //
        endif
        endif
        skip
        VALOR=(++C)*100/ULTMO
TERMOH(VALOR)
devpos(18,47)
devoutpic(valor,"999.99 % ")
enddo
set cursor on

restscreen(0,0,maxxrow(),maxcol(),cP
antalla)

tam=len(aNum_tes)
if tam=0
setcolor(cColor5)
m_sombra(1,1,34,14,67...L)
@ 12,35 say "No hay registros
parceidos..." cColor1
@ 13,35 say "¡Esquede Otra vez ?
(S/N)" get si pict '!';
valid si $ 'NS'
read
if lastkey()=27
aNum_tes={}
ponSetRelation()
setcolor(cColAct)

restscreen(0,0,maxxrow(),
maxcol(),cPant1)
return // DESACTIVAR
AQUI LOS SETKEYS
endif

if si="S"
aNum_tes={}

restscreen(0,0,maxxrow(),maxcol(),cP
ant)
setcolor(cColor4)
loop
endif
aNum_tes={}
ponSetRelation()
setcolor(cColAct)

restscreen(0,0,maxxrow(),maxcol(),cP
ant1)
return // DESACTIVAR AQUI
LOS SETKEYS
endif
nOpc1=1
DO BUS_TIT

restscreen(0,0,maxxrow(),maxcol(),cP
ant)
aNum_tes={}
enddo (.L)
ponSetRelation()

```

```

setcolor(cColAct)
restscreen(0,0,maxxrow(),maxcol(),cP
ant1)

return
*****
PROCEDURE BUS_TIT
nOpc1=1
set cursor off
while nOpc1 <= 0
setcolor(cColor3)
M_SOMBRA(10,50,17,75...L)
@ 10,57 SAY ' AUTORES ' color
cColLin1
cCol_menu=5
@ 17,51 say str(tam,3) '+'
Trabajos '
nOpc1 =
ACHOICE(11,51,16,74,aNum_tes...
eleccion)
enddo
set cursor on
return
*****
*funcion para el control del menf.
FUNCTION ELECCION
local cPant2:=saveScreen(0,0,24,79)
local nLin:=mlcount(uno->titulo,48)
local cColAct:=setcolor()
local nRenMens:=24 //
set_SET_MESSAGE)
local cP1:=aaIva_mens(nRenMens)
Parameters estado,elemento,pos_rel
!Muea_mem=.f.

nOpc3=3
tipo_ret = 0 //mensaje de pacien de
la UDF -regresar cero-
if estado = 0 //SI no se hace nada en
el Achoice
tipo_ret = 2 //
else
do case
case lastkey()=13
ponset()
mensaje(nRenMens,
F5=T;tulo F6=Obj.Gen.
F7=Cap;tulos;
FR=Resumen
F9=Sinodales F1=Ayuda;cColor1)
seek
TRIM(aNum_tes|elemento)
cMemo1:=uno->titulo
cMemo2:=uno-
>obj_gen
cMemo3:=uno-
>capitulo
cMemo4:=uno-
>resumen
cMemo5:=uno-
>sinodal
setcolor(cColor1)

m_sombra(11,21,18,72...L)
@ 12,22 SAY ' Autor->
'+ substr(uno->alum,1,35) color
cColor1

```

```

@ 13,22 SAY
memoline(uno->titulo,48,1) color
cColor1
@ 14,22 SAY
memoline(uno->titulo,48,2) color
cColor1
@ 15,22 SAY
memoline(uno->titulo,48,3) color
cColor1
@ 16,22 SAY
memoline(uno->titulo,48,4) color
cColor1
@ 17,22 SAY 'Opcion-
'+ substr(seia->opc1,1,18) color
cColor1
@ 17,52 SAY 'Carrera-
'+ substr(tres-> carr,1,9) color
cColor1
por=.L
while por
@ 18,24 prompt
'Imprimir'
@ 18,36 prompt 'Ver
Todo'
@ 18,49 prompt 'M s'
menu to nOpc3
nOpc3=if(nOpc3=0,3,m-
>nOpc3)
do case
case nOpc3=1
tone(100,3)
do rpt_alum
case nOpc3=2
do Pant_vis
por=.L
case nOpc3=3
tipo_ret=2
por = .f.
endcase
enddo
setcolor(cColAct)

restscreen(0,0,24,79,cPant2)
tipo_ret=2

rest_mens(nRenMens,cP1)
quitaSet()

case (lastkey() >65 .and. lastkey()
< 122)
tipo_ret=3
case lastkey() = 27 //escape
tipo_ret=0
otherwise
tipo_ret=2
endcase
endif
return tipo_ret

*****
procedure pant_vis //pantalla total
con informacion de tesis
local cCol_act:=setcolor()
local
cPant1:=saveScreen(0,0,maxxrow(),ma
xxcol())
mensaje(24,'cCol_Act)
setcolor(cColor5)
m_sombra(0,0,23,78...L)
@ 0,40 say "N"

```

```
@ 1,1 say 'Nombre'
@ 1,9 say substr(uno->alum,1,30)
colo cColLin3
@ 1,40 say ""
@ 1,55 say 'CAPITULOS'
@ 2,1 say 'Cta.'
@ 2,6 say uno->cuen colo
cColLin3
@ 2,16 say 'Carrera'
@ 2,25 say substr(tres->carr,1,14)
colo cColLin3
@ 2,40 say ""
@ 2,41 SAY memoline(uno-
>capitulo,36,1) colo cColLin3
@ 3,1 say 'Gen.'
@ 3,6 say cuatro->gene colo
cColLin3
@ 3,18 say 'Clv.Reg.'
@ 3,28 say uno->clvr colo
cColLin3
@ 3,40 say ""
@ 3,41 SAY memoline(uno-
>capitulo,36,2) colo cColLin3
@ 4,1 say 'Fec.Reg.'
@ 4,10 say uno->freg colo
cColLin3
@ 4,20 say 'Fec.Ter.'
@ 4,30 say uno->fter colo
cColLin3
@ 4,40 say ""
@ 4,41 SAY memoline(uno-
>capitulo,36,3) colo cColLin3
@ 5,1 say 'Fec.Tit.'
@ 5,10 say uno->frit colo cColLin3
@ 5,20 say 'Opc.Tit.'
@ 5,29 say substr(seis->opt,1,10)
colo cColLin3
@ 5,40 say ""
@ 5,41 SAY memoline(uno-
>capitulo,36,4) colo cColLin3
@ 6,1 say 'Asesor'
@ 6,9 say substr(dos->ases,1,30)
colo cColLin3
@ 6,40 say ""
@ 6,41 SAY memoline(uno-
>capitulo,36,5) colo cColLin3
@ 7,40 say ""
@ 7,41 SAY memoline(uno-
>capitulo,36,6) colo cColLin3
@ 8,17 say 'TITULO'
@ 8,40 say ""
@ 8,41 SAY memoline(uno-
>capitulo,36,7) colo cColLin3
@ 9,1 SAY memoline(uno-
>itulo,38,1) colo cColLin3
@ 9,40 say ""
@ 9,55 say 'RESUMEN'
@ 10,1 SAY memoline(uno-
>itulo,38,2) colo cColLin3
@ 10,40 say ""
@ 10,41 SAY memoline(uno-
>Resuman,36,1) colo cColLin3
@ 11,1 SAY memoline(uno-
>itulo,38,3) colo cColLin3
@ 11,40 say ""
@ 11,41 SAY memoline(uno-
>Resumen,36,2) colo cColLin3
@ 12,1 SAY memoline(uno-
>itulo,38,4) colo cColLin3
@ 12,40 say ""
@ 12,41 SAY memoline(uno-
>Resuman,36,3) colo cColLin3
```

```
@ 13,1 SAY memoline(uno-
>itulo,38,5) colo cColLin3
@ 13,40 say ""
@ 13,41 SAY memoline(uno-
>Resumen,36,4) colo cColLin3
@ 14,12 say 'OBJETIVO'
GENERAL
@ 14,40 say ""
@ 14,41 SAY memoline(uno-
>Resuman,36,5) colo cColLin3
@ 15,1 SAY memoline(uno-
>obj_gen,38,1) colo cColLin3
@ 15,40 say ""
@ 15,41 SAY memoline(uno-
>Resumen,36,6) colo cColLin3
@ 16,1 SAY memoline(uno-
>obj_gen,38,2) colo cColLin3
@ 16,40 say ""
@ 16,41 SAY memoline(uno-
>Resumen,36,7) colo cColLin3
@ 17,1 SAY memoline(uno-
>obj_gen,38,3) colo cColLin3
@ 17,40 say ""
@ 17,55 say 'SINODALES'
@ 18,1 SAY memoline(uno-
>obj_gen,38,4) colo cColLin3
@ 18,40 say ""
@ 18,41 SAY memoline(uno-
>sinodal,36,1) colo cColLin3
@ 19,1 SAY memoline(uno-
>obj_gen,38,5) colo cColLin3
@ 19,40 say ""
@ 19,41 SAY memoline(uno-
>sinodal,36,2) colo cColLin3
If clv_opc=3
@ 20,15 SAY 'CO-AUTORES'
endif
@ 20,40 say ""
@ 20,41 SAY memoline(uno-
>sinodal,36,3) colo cColLin3
If clv_opc=3
@ 21,1 SAY substr(stea-
>co2,1,36) colo cColLin3
endif
@ 21,40 say ""
@ 21,41 SAY memoline(uno-
>sinodal,36,4) colo cColLin3
If clv_opc=3
@ 22,1 SAY substr(stea-
>co3,1,36) colo cColLin3
endif
@ 22,40 say ""
@ 22,41 SAY memoline(uno-
>sinodal,36,5) colo cColLin3
@ 23,40 say ""
inkey(0)
m_sombra(09,41,22,78,.,.)
@ 10,52 SAY 'Direc. Domic.'
@ 11,42 SAY substr(uno-
>direc,1,30) colo cColLin3
@ 12,42 SAY substr(uno-
>direc,31,30) colo cColLin3
@ 13,42 SAY substr(uno-
>direc,61) colo cColLin3
@ 14,52 SAY 'Direc. Trab.'
@ 15,42 SAY substr(uno-
>direc,1,30) colo cColLin3
@ 16,42 SAY substr(uno-
>direc,31,30) colo cColLin3
@ 17,42 SAY substr(uno->direc,61)
colo cColLin3
@ 18,52 SAY 'Telef. Domic.'
```

```
@ 19,42 SAY uno->teled colo
cColLin3
@ 20,52 SAY 'Telef. Trab.'
@ 21,42 SAY uno->telet colo
cColLin3
inkey(0)
setcolor(cCol_act)
restacreen(0,0,maxrow(),maxcol(),cP
ant1)
return
*****
procedure opcBusq
local
cPaint:=savescreeen(0,0,maxrow(),max
col())
local cColAct:=setcolor(0)
ponSetRelacion()
setcolor(cColor1)
pickyO
*restacreen(0,0,maxrow(),maxcol(),c
Paint)
setcolor(cColAct)
return
*****
function picky()
local
Screen.nChoice.nKey:=lastkey()
local nOpc:=1
local cColAct:=setcolor()
while nOpc <> 0
"ponSetRelacion()
mensaje(24; ESC=Salir
ENTER=Elegir F1=Ayuda
'cColLin1)
scroll(9,1,18,20)
@ 9,1 to 16,20 double
@ 16,1 say "|||||||
@ 17,1 say ""
@ 18,1 say "|||||||
@ 10,2 prompt aList[1] message
'Busca en el Titulo'
@ 11,2 prompt aList[2] message
'Busca en el Objetivo General'
@ 12,2 prompt aList[3] message
'Busca en los Capitulo'
@ 13,2 prompt aList[4] message
'Busca en el Resumen'
@ 14,2 prompt aList[5] message
'Busca en la Direccion de Trabajo'
@ 15,2 prompt aList[6] message
'Busca en la Direccion de Domicilio'
@ 17,2 prompt aList[7] message
'Realiza búsqueda'
menu to nOpc
do case
```

```

case nOpc=1
if aVal[1]
aList[1]:="Titulo 0'
aVal[1]:= .not.(aVal[1])
else
aList[1]:="Titulo
aVal[1]:= .not.(aVal[1])
endif
case nOpc=2
if aVal[2]
aList[2]:="Obj.Gen. 0'
aVal[2]:= .not.(aVal[2])
else
aList[2]:="Obj.Gen.
aVal[2]:= .not.(aVal[2])
endif
*do materia
case nOpc=3
if aVal[3]
aList[3]:="Capitulos 0'
aVal[3]:= .not.(aVal[3])
else
aList[3]:="Capitulos
aVal[3]:= .not.(aVal[3])
endif
*do asesor
case nOpc=4
if aVal[4]
aList[4]:="Resumen 0'
aVal[4]:= .not.(aVal[4])
else
aList[4]:="Resumen
aVal[4]:= .not.(aVal[4])
endif
*do fec_reg
case nOpc=5
if aVal[5]
aList[5]:="Dir.Trab. 0'
aVal[5]:= .not.(aVal[5])
else
aList[5]:="Dir.Trab.
aVal[5]:= .not.(aVal[5])
endif
*do fec_ter
case nOpc=6
if aVal[6]
aList[6]:="Dir.Dom. 0'
aVal[6]:= .not.(aVal[6])
else
aList[6]:="Dir.Dom.
aVal[6]:= .not.(aVal[6])
endif
*do fec_tit
case nOpc=7
lSalir:= .not. aVal[1] .or.(.not.
aVal[2]) .or. (.not. aVal[3]) .
.or. (.not. aVal[4]) .or.
(.not. aVal[5]) .or. (.not. aVal[6])
if .not. lSalir
aviso("Búsqueda vacía...1.5)
loop
else
return
endif
otherwise
lSalir:=.f.
endcase
enddo
return
*****
Programa 6

```

```

/* procedimiento asociado a
report1.prg para imprimir los reportes
creados con anterioridad, con la
posibilidad de mandar a impresora
directamente o a un archivo alterno
fecha inicio : 01-jun-96
fecha ult_act: 11-jun-96
autor: Armando Amador Molina
*/

procedure imp_rpt
#xcorrmand READ =>
ReadModal(GetList, nT);
GetList:={
local nOpc:=1
local cColAct:="setcolor()
local
cPantAct:=savescree(0,0,maxrow(),
maxcol())
local cPantAux
setcolor(cColor1)
m_sombra(9,36,14,47...t)

while nOpc !=0
@ 10,37 prompt "Asesor" message
" Imprime datos completos de
asesor "
@ 11,37 prompt "Registro"
message " Imprime datos sobre
registro "
@ 12,37 prompt "Titulacion"
message " Imprime datos sobre
titulacion "
@ 13,37 prompt "Generales"
message " Imprime datos generales
del alumno"
menu to nOpc
cPantAux:=savescree(0,0,maxrow(),
maxcol())

do case
case nOpc=1
do imp_asesor
case nOpc=2
do arch_reg
case nOpc=3
do arch_tit
case nOpc=4
do arch_gen
endcase
restore screen from cPantAct
setcolor(cColAct)
return
*****
procedure imp_asesor

local nOpcAse:=1
local
cPanAse:=savescree(0,0,maxrow(),
maxcol())
LOCAL aAseor:={
local cColAct :=setcolor()
local cP1:=salva_mens(24)

base="asesor"
campo="ases"
mensaje(24,ENTER="Elegir Asesor
ESC=Salir";

setcolor(cColor4)
CAMP_ARR(base,campo,aAseor)
m_sombra(12,38,19,70...t)

nOpcAse:=ACHOICE(13,39,18,69,a
Aseor,.elocion1)
if nOpcAse<= 0
CHC_NTX(base,campo)
seek aAseor(nOpcAse)
do rpt_ases
use
restscreen(0,0,maxrow(),
maxcol(),cPanAse)
setcolor(cColAct)
rest_mens(24,cP1)
else
restscreen(0,0,maxrow(),maxcol(),cP
anAse)
rest_mens(24,cP1)
endif
setcolor(cColAct)
ponSetRelation()
return
*****
Function miErrorH(objError)
break ObjError
return nil
*****
procedure Arch_Reg

local nOpc:=1
LOCAL bErrorH,bLastH,ObjErr
bErrorH:={objErr
miErrorH(ObjErr)}
bLastH:=errorblock(bErrorH)
cPant:=savescree(0,0,maxrow(),max
xcol())

cPag_ini:=1
cPag_fin:=99
cEncab:="space(80)
page_no=0
lin_no=0

if !ent_rpt()
return
endif

WHILE !
BEGIN SEQUENCE

m_sombra(18,45,20,67...t)
@ 19,48 say "Imprimiendo " colo
cColPar2
set device to print
SET PRINT on
set console off

* REPORT HEADER
@ 0,0 say chr(15) //compresion de
datos!!
@ 0,0 say chr(27)+~C'+chr(51) //con
esto le doy formato apaisado!!
@ 0,0 say chr(27)+~N'+chr(2)
do rep_hd1
go top
* REPORT BODY

c=0
valor=0
TERMOH1(VALOR)

```

```

do while .not. eof() .and. trunc()
do detalle1
skip
valor=(++c * 100)/astrec()
TERMOH1(VALOR)
enddo (while .not. eof())

RECOVER

set device to screen
set printer off
set console on
if check_imp()
loop
else
exit
endif

END //SECQUENCE
errorlock(bLastH)
exit
enddo

set device to screen
set printer off
set console on

restscreen(0,0,maxrow(),maxcol(),cP
ant)
ponSetRelation()
setcursor(1)
return
*****
****
*****
*****
* PROCEDURES

procedure adv_lin
lin_no = lin_no + 1
if lin_no > 41
do rep_hd1
endif
return

procedure detalle1
* Compute and display lin of
information
do adv_lin
nLinTit=mlcount(titulo,50)
if page_no >= cPag_ini .and.
page_no <= cPag_fin
@ lin_no, 2 say alum
@ lin_no, 44 say cuen
@ lin_no, 55 say substr(tres-
castr,1,14)
@ lin_no, 70 say caastro->gene
@ lin_no, 77 say
memoline(titulo,50,1)
@ lin_no, 130 say substr(seis-
>opt,1,15)
@ lin_no, 148 say dow->ases
@ lin_no, 189 say firm_fecha(freg)
@ lin_no, 199 say firm_fecha(ther)
@ lin_no, 209 say firm_fecha(lin)
if lin_no, 219 say civr
endif

for linea = 2 to nLinTit

if
!empty(memoline(titulo,50,linea))
lin_no=lin_no+1
if page_no >= cPag_ini
.and. page_no <= cPag_fin
@ lin_no, 77 say
memoline(titulo,50,linea)
endif
endif
next
return
*****
*****
procedure rep_hd1
do encab_gral

if page_no >= cPag_ini .and.
page_no <= cPag_fin
@ lin_no, 0 say
'*****'
@ lin_no, 40 say
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
endif
lin_no=lin_no+1
if page_no >= cPag_ini .and.
page_no <= cPag_fin
@ lin_no, 0 say * NOMBRE
DEL ALUMNO'
@ lin_no, 42 say * CUENTA '
CARRERA 'GENER'+;
* TITULO
* OPCION TTT. '+;
** ASESOR
* FECH.REG.'FECH.TER '+;
* FECH.TTT.'CLVR '+;
endif
lin_no=lin_no+1
if page_no >= cPag_ini .and.
page_no <= cPag_fin
@ lin_no, 0 say
'*****'
@ lin_no, 40 say
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'+;
endif
lin_no=lin_no+1

return
*****
*****
procedure Arch_Tit
local nOpe = 1
LOCAL bErrorH,bLastH,ObjErr
bErrorH = {objErr
nOpeErrorH(ObjErr)}
bLastH = errorlock(bErrorH)
cPAnt: = saveScreen(0,0,maxrow(),ma
xcol())
cPag_ini = 1

cPag_fin = 99
cEncab: = space(80)
page_no = 0
lin_no = 0

if !ent_rp()
return
endif

WHILE 1.
BEGIN SEQUENCE

m_sombra(18,45,20,67,1)
@ 19,48 say ' Imprimiendo ' col0
cColPar2
set device to print
SET PRINT ON
set console off

* REPORT HEADER
@ 0,0 say chr(15) //compresion de
datos!!
@ 0,0 say chr(27)-'C'+chr(51)//con
esto le doy formato apaisado!!
@ 0,0 say chr(27)-'N'+chr(2)
do rep_hd2
go top

* REPORT BODY

c=0
valor=0
TERMOH1(VALOR)

do while .not. eof() .and. trunc()
do detalle2
skip
valor=(++c * 100)/astrec()
TERMOH1(VALOR)
enddo (while .not. eof())

RECOVER

set device to screen
set printer off
set console on

if check_imp()
loop
else
exit
endif

END //SECQUENCE
errorlock(bLastH)
exit
enddo

set device to screen
set printer off
set console on

restscreen(0,0,maxrow(),maxcol(),cP
ant)
ponSetRelation()
setcursor(1)
return
*****
*****
procedure Arch_Gen
local nOpe = 1
LOCAL bErrorH,bLastH,ObjErr

```



```

if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no, 0 say "NOMBRE DEL
ALUMNO"
  @ lin_no, 41 say "CARRERA
" TITULO " "
" OPC. TIT. "
ASESOR
" FEC.REG. "FEC.TIT. "
SINODALES
"CLVR"
endif
lin_no=lin_no+1
if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no, 0 say
"#####"
  @ lin_no, 40 say
"#####"
  #####;

"#####"
#####;

"#####"
#####;

endif
lin_no=lin_no+1
return
*****

procedure adv_lin3
lin_no = lin_no + 1
if lin_no > 40
do rep_hd3
endif
return

procedure detalle3
do adv_lin3
do adv_lin3
nLinTit=mlcount(titulo,50)
if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no, 1 say alum
  @ lin_no, 42 say substr(cuen,1,9)
  @ lin_no, 52 say substr(tres-
>curr,1,15)
  @ lin_no, 68 say cuatro->gene
  @ lin_no, 75 say
memoline(titulo,50,1)
  @ lin_no,126 say
substr(dired,1,40)
  @ lin_no,167 say teled
  @ lin_no,180 say substr(dired,1,40)
  @ lin_no,221 say telct
endif

lin_no = lin_no + 1
if !empty(memoline(titulo,50,2))
if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no, 75 say
memoline(titulo,50,2)
endif
endif
if !empty(substr(dired,41))
if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no,126 say
substr(dired,41)
endif

endif
if !empty(substr(dired,41))
if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no,180 say substr(dired,41)
endif

endif
for linea=3 to nLinTit
if
!empty(memoline(titulo,50,linea))
lin_no = lin_no + 1
if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no, 75 say
memoline(titulo,50,linea)
endif
endif
next
return

procedure rep_hd3
do encab_gral

if page_no >= cPag_ini.and.
page_no <= cPag_fin
  @ lin_no, 0 say
"#####"
  @ lin_no, 40 say
"#####"
  #####;

"#####"
#####;

"#####"
#####;

endif
lin_no=lin_no+1
return
*****

function ent_rpt()
local pasa := .t.
m_sombra(17,45,20,77..t.)
@ 18,46 say 'De p g.' get cPag_ini
picr 99;
valid cPag_ini > 0
@ 18,58 say 'a'
read
if lastkey)=27

restscreen(0,0,maxxrow(),maxcol(),cP
ant)
return .f.
endif

@ 18,62 prompt ' '
@ 18,68 prompt 'Final'
menu to nOpc
do case
case nOpc=1
  @ 18,62 get cPag_fin picr 999;
  valid cPag_ini <= cPag_fin
  read
  if lastkey)=27

restscreen(0,0,maxxrow(),maxcol(),cP
ant)
return .f.
endif
case nOpc=2
  cPag_fin=200
  otherwise

restscreen(0,0,maxxrow(),maxcol(),cP
ant)

```

```

return .f.
endcase
@ 19.46 say 'Encabezado: ' get
cEncab pic@ 17@!;
valid if(cEncab=";.;.;.t.)
read
if lastkey()=27

restscreen(0,0,maxrow(),maxcol(),cP
ant)
return .f.
endif

restscreen(0,0,maxrow(),maxcol(),cP
ant)

if !checa_imp()

restscreen(0,0,maxrow(),maxcol(),cP
ant)
return .f.
endif
setcursor(0)

if !file(repor.dbf)
avisos('No existe reporte
activo...',2,cColLin3)

restscreen(0,0,maxrow(),maxcol(),cP
ant)
setcursor(1)
return .f.
else
ponRelatRpt()
endif
return pass

*****
*****
*****
FUNCTION TERMOH1(VALOR)
set device to screen
IF valor <.1
m_sombra(12,30,17,73,...)
@ 13.31 say " 10 20 30 40 50
60 70 80 90 100 %"
@ 14.31 say "
UU.....A. "
@ 15.31 say "
*****
*****
*****
@ 16.31 say "
AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA
ELSE
@ 15.33 say
REPLICATE(CHR(219),valor/3)
@ 16.46 say valor pic@ '999.99 %'
ENDIF

set device to print
RETURN NIL

*****
*****
*****
PROCEDURE RPT_ASES
LOCAL bErrorH,bLastH,ObjErr
local
cPant:=savescreeen(0,0,maxrow(),max
col())
bErrorH:={objErr}
miErrorH(ObjErr)
bLastH:=errorblock(bErrorH)
line_no=0
pag=0
//do ponRelatRpt

if !checa_imp()

restscreen(0,0,maxrow(),maxcol(),cP
ant)
return
endif
while .t.
begin sequence
m_sombra(18,45,20,67...L)
@ 19.48 say 'Imprimiendo ' colo
cColPar2
set device to print
SET PRINTER on //TO
('RPT_ASES.PRN)
set console off
do repAse1

recover
set device to screen
set printer off
set console on

if checa_imp()
loop
else
exit
endif

end sequence

errorblock(bLastH)
exit

cnddo
set device to screen
set printer off
set console on

restscreen(0,0,maxrow(),maxcol(),cP
ant)
return
*****
*****
*****
procedure av_lin1
lin_no = lin_no + 1
if lin_no > 60
do repAse1
endif
return

*****
*****
*****
procedure repAse1
* Print report header
LIN_NO=LIN_NO+1
do av_lin1
@ lin_no, 11 say 'DATOS GE
NERALES DE'
@ lin_no, 51 say 'ASESORES'
do av_lin1
do av_lin1
@ lin_no, 0 say
'AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA'
@ lin_no, 40 say
'AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA'
do av_lin1
@ lin_no, 0 say ""
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say "NOMBRE:"
@ lin_no, 9 say ases

@ lin_no, 62 say 'CLAVE:'
@ lin_no, 70 say clv_ase
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say ""
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say "GRADO O
TITULO:"
@ lin_no, 18 say grado
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say ""
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say "DIRECCION DE
CASA:"
@ lin_no, 25 say direo
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say ""
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say "DIRECCION DE
TRABAJO:"
@ lin_no, 25 say diret
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say ""
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say "TELEFONO DE
CASA:"
@ lin_no, 20 say teled
@ lin_no, 43 say TELEFONO DE
TRABAJO:"
@ lin_no, 65 say telet
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say ""
@ lin_no, 77 say ""
do av_lin1
@ lin_no, 0 say
'AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA'
@ lin_no, 40 say
'AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA'
return
*****
*****
*****
PROCEDURE RPT_ALUM
LOCAL bErrorH,bLastH,ObjErr
local
cPant:=savescreeen(0,0,maxrow(),max
col())
bErrorH:={objErr}
miErrorH(ObjErr)
bLastH:=errorblock(bErrorH)
line_no=0
pag=0
//do ponRelatRpt

if !checa_imp()

restscreen(0,0,maxrow(),maxcol(),cP
ant)
return
endif
while .t.
begin sequence
m_sombra(18,45,20,67...L)

```

@ 19,48 say 'Imprimiendo'colo

COOLPar2
 set device to print
 SET PRINTER on //TO
 (RPT_ALUM.PRN)
 set console off
 /*sele 9
 go 41 //la epscn
 @ 0,0 say nuevo-
 >p_compress
 sele 1
 *do ponRelatRpt
 go reg *//
 do repAlum
 do dots_alum
 eject
 recover
 set console on
 set device to screen
 if (cheas_imp())
 loop
 else
 exit
endif
end sequence
errorblock(bLashH)
exit
enddo
set device to screen
set console on
restscreen(0,0,maxrow(),maxcol(),Cp
ant)

return

* PROCEDURES

procedure adv_lin1
 line_no = line_no + 1
 if line_no > 66
 do repAlum
 endif
return

procedure data_alum
 * Compute and display line of
 information

local lin_aux
 line_no = line_no + 1
 @ line_no, 0 say 'NOMBRE:'
 @ line_no, 8 say substr(alum,1,41)
 @ line_no, 59 say 'CUENTA.'
 @ line_no, 67 say substr(cuen,1,9)
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say 'CARRERA:'
 @ line_no, 9 say substr(res->
 >car1,1,5)
 @ line_no, 33 say
 'GENERACION:'
 @ line_no, 45 say cuatro->gens
 @ line_no, 59 say 'CLAVE REG.'
 @ line_no, 71 say clvr
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say 'FECHA REG.'
 @ line_no, 12 say freg
 @ line_no, 28 say 'FECHA TER.'
 @ line_no, 40 say flyer

@ line_no, 56 say 'FECHA TIT.'

@ line_no, 68 say fit
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say 'OPC. TIT.'
 @ line_no, 11 say substr(seis->
 >opt,1,15)
 @ line_no, 28 say 'ASESOR:'
 @ line_no, 36 say dos->ases
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say
 'AAAAAAAAAAAAAAAAAAAAAAA'
 AAAAAAAAAAAAAAAAAAAAAAAA
 @ line_no, 40 say
 'AAAAAAAAAAAAAAAAAAAAAAA'
 AAAAAAAAAAAAAAAAAAAAAAAA
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say 'DIRECCION
 DE CASA.'
 @ line_no, 26 say substr(dired,1,50)
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say substr(dired,51)
 @ line_no, 46 say 'TELEFONO DE
 CASA.'
 @ line_no, 64 say teled
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say 'DIRECCION DE
 TRABAJO.'
 @ line_no, 26 say substr(dired,1,50)
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say substr(dired,51)
 @ line_no, 47 say TEL. DE
 TRABAJO.'
 @ line_no, 64 say telet
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say
 'AAAAAAAAAAAAAAAAAAAAAAA'
 AAAAAAAAAAAAAAAAAAAAAAAA
 @ line_no, 40 say
 'AAAAAAAAAAAAAAAAAAAAAAA'
 AAAAAAAAAAAAAAAAAAAAAAAA
 line_no = line_no + 1
 @ line_no, 28 say 'C O - A U T O
 R E S'
 if !empty(siete->co2)
 line_no = line_no + 1
 @ line_no, 0 say 'AUTOR 1.'
 @ line_no, 10 say siete->co2
 endif
 if !empty(siete->co3)
 line_no = line_no + 1
 line_no = line_no + 1
 @ line_no, 0 say 'AUTOR 3.'
 @ line_no, 10 say siete->co3
 line_no = line_no + 1
 line_no = line_no + 1
 endif
 if !empty(siete->co4)
 @ line_no, 0 say 'AUTOR 4.'
 @ line_no, 10 say siete->co4
 line_no = line_no + 1
 line_no = line_no + 1
 endif
 if !empty(siete->co5)
 @ line_no, 0 say 'AUTOR 5.'

@ line_no, 10 say siete->co5

line_no = line_no + 1
 line_no = line_no + 1
 endif
 endif //clvr_opo=3
 @ line_no, 0 say
 'AAAAAAAAAAAAAAAAAAAAAAA'
 AAAAAAAAAAAAAAAAAAAAAAAA
 @ line_no, 40 say
 'AAAAAAAAAAAAAAAAAAAAAAA'
 AAAAAAAAAAAAAAAAAAAAAAAA
 do adv_lin1
 @ line_no, 34 say 'TITULO'
 do adv_lin1
 do adv_lin1
 @ line_no, 1 say
 memoline(titulo,75,1)
 do adv_lin1
 lin_aux = 2
 while
 !empty(memoline(titulo,75,lin_aux))
 @ line_no, 1 say
 memoline(titulo,75,lin_aux)
 do adv_lin1
 lin_aux = lin_aux + 1
 enddo
 do adv_lin1
 @ line_no, 30 say 'OBJETIVO
 GENERAL'
 do adv_lin1
 do adv_lin1
 lin_aux = 1
 while
 !empty(memoline(obj_gen,75,lin_aux
 x))
 @ line_no, 1 say
 memoline(obj_gen,75,lin_aux)
 do adv_lin1
 lin_aux = lin_aux + 1
 enddo
 do adv_lin1
 do adv_lin1
 @ line_no, 33 say 'CAPITULOS'
 do adv_lin1
 do adv_lin1
 lin_aux = 1
 while
 !empty(memoline(capitulo,75,lin_aux
 x))
 @ line_no, 1 say
 memoline(capitulo,75,lin_aux)
 do adv_lin1
 lin_aux = lin_aux + 1
 enddo
 do adv_lin1
 do adv_lin1
 @ line_no, 34 say 'RESUMEN'
 do adv_lin1
 do adv_lin1
 lin_aux = 1
 while
 !empty(memoline(resumen,75,lin_aux
 x))
 @ line_no, 1 say
 memoline(resumen,75,lin_aux)
 do adv_lin1
 lin_aux = lin_aux + 1
 enddo

```

do adv_lin1
do adv_lin1
@ line_no, 33 say 'SINODALES'
do adv_lin1
do adv_lin1
lin_aux=1
while
!empty(memoline(sinodal,50,lin_aux
))
@ line_no, 1 say
memoline(sinodal,50,lin_aux)
do adv_lin1
lin_aux=lin_aux+1
enddo
do adv_lin1
do adv_lin1
@ line_no, 1 say
'AREA1/MATERIA1 '
@ line_no, 18 say cinco->materia
@ line_no, 39 say
'AREA2/MATERIA2 '
@ line_no, 56 say ocho->materia
return

procedure repAlum
line_no=1
pag=pag-1
line_no=line_no+1
@ line_no, 20 say 'DATOS
GENERALES DE ALUMNOS'
@ line_no, 58 say 'p->'
+str(pag,1)
line_no=line_no+1
@ line_no, 0 say
'AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA'
@ line_no, 40 say
'AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA'
line_no=line_no+1
return

* EOF RPT_ALUM.PRG
*****
*****

Programa 7
*****
*****

procedure demRatO
*****
*****

#include 'inkey.ch'
#include 'AAteclas.CH'
#include 'AAvalor.CH'
#include 'AAfincas.CH'
local nPosX := 0, ;
posicion X del raten
nPosY := 0, ; //posicion
Y del raten

nBotonPres := 0, ;
//Oprimida la tecla actual del raten
nEleccion := 0, ;
//seleccion del menu por el usuario
ITeclazoRat := f.

local
cPanAct:=savescree(0,0,maxxow(),
maxcol())
local cPant
local nCur:=setcursor()
public IRatPresen:=.f. //disponible
el raten?

cls
setcolor(cColor1)
setcursor(0)
do part_1
@ 5.04 say "M" color cColLin4
@ 5.05 say "ANTENIMIENTO"
@ 5.23 say "B" color cColLin4
@ 5.24 say "USQUEDA"
@ 5.36 say "R" color cColLin4
@ 5.37 say "EPORTES"
@ 5.49 say "A" color cColLin4
@ 5.50 say "CTUALIZAR"
@ 5.64 say "U" color cColLin4
@ 5.65 say "TILERIAS"
cPant:=savescree(0,0,maxxow(),max
col())

if inIRat()
IRatPresen=.t.
endif

while !(nEleccion == K_ESC)
//ciclo

ITeclazoRat:=.f.

if IRatPresen

posRaton(@nPosX,@nPosY,@nBoto
nPres)
endif

do case
case nBotonPres == 2 ; keyboard
chr(27) //ESC
case nBotonPres == 1

restscreen(0,0,maxxow(),maxcol(),cP
ant)
do case
case ((nPosX>=4 and
nPosX<=16) and (nPosY==5))
teclazoRat:=.t.
keyboard chr(77) //M
case ((nPosX>=23 and
nPosX<=30) and (nPosY==5))
teclazoRat:=.t.
keyboard chr(66) //B
case ((nPosX>=36 and
nPosX<=43) and (nPosY==5))
teclazoRat:=.t.
keyboard chr(82) //R
case ((nPosX>=49 and
nPosX<=58) and (nPosY==5))
teclazoRat:=.t.
keyboard chr(65) //A
case ((nPosX>=64 and
nPosX<=72) and (nPosY==5))
teclazoRat:=.t.
keyboard chr(85) //U
case ((nPosX>=57 and
nPosX<=64) and (nPosY==24))
teclazoRat:=.t.
keyboard chr(K_F1)
//Ayuda
case ((nPosX>=67 and
nPosX<=75) and (nPosY==24))
teclazoRat:=.t.
keyboard chr(K_ESC)
//Ayuda

otherwise
restscreen(0,0,maxxow(),maxcol(),cP
ant)
keyboard "0"

otherwise

//restscreen(0,0,maxxow(),maxcol(),c
Pant)
endif

if IRatPresen
while nBotonPres=0

posRaton(@nPosX,@nPosY,@nBoto
nPres)
enddo
endif

nBotonPres:=0

// verifica si hay una oprimida de
tecla esperando ser procesada
// si no la hay regresa arriba y vuelve
a comprobar el raten.
// si la hay procesa la tecla oprimida

if !(nextkey() == 0)
nEleccion := inkey()
else
loop
endif

do case
case nEleccion=27
ocultRat()

restscreen(0,0,maxxow(),maxcol(),cP
anAct)
setcolor(cColor1)
setcursor(nCur)
return
case nEleccion=asc('M') or.
nEleccion=asc('m')
do manten2
case nEleccion=asc('B') or.
nEleccion=asc('b')
do busca2
case nEleccion=asc('R') or.
nEleccion=asc('r')
do report2
case nEleccion=asc('A') or.
nEleccion=asc('a')

```

```

do actualiza2
case nEleccion=ase("U").or.
nEleccion=ase("u")
do utileria2
case nEleccion = K_F1
ocultRat()
ayuda("main")
iniRaton()

posRaton(@nPosX,@nPosY,@nBoto
nPres)
endcase

restscreen(0,0,maxrow(),maxcol(),cP
ant)
setcolor(cColor1)
enddo

ocultRat()
restscreen(0,0,maxrow(),maxcol(),cP
anAct)
setcolor(cColor1)
setcursor(nCur)

return
*****
PROCEDURE manten2
local
aMenu:={space(10),"Captura"}
local
cPantalla:=(0,0,maxrow(),maxcol())
local
cScreen:=saveScreen(0,0,maxrow(),m
axcol())
static nEleccion

setcolor(cColor1)

while .t.

m_sombra(6,3,8,18,"EIE"%IIE%)
//n,n)

nEleccion:=MenuTB(7,4,7,17,aMen
u)

do case
case nEleccion == 0
exit
case nEleccion=1
avisos("Opcion Captura
inhabilitada...2,cColLin3)
endcase
restore screen from cScreen
setcolor(cColor1)
enddo

return
*****
PROCEDURE busca2
local
aMenu:={space(10),"Nombre","Cuen
ta","Clave Reg.,"Palabra"}
local
cPantalla:=(0,0,maxrow(),maxcol())
static nEleccion
setcolor(cColor1)

while .t.

m_sombra(6,21,11,33,"EIE"%IIE%)
//,n,n)
nEleccion:=MenuTB(7,22,10,32,aM
enu)
do case
case nEleccion == 0
exit
case nEleccion == 1
avisos("Opcion Nombre
inhabilitada...2,cColLin3)
case nEleccion == 2
avisos("Opcion Cuenta
inhabilitada...2,cColLin3)
case nEleccion == 3
avisos("Opcion Clave Registro
inhabilitada...2,cColLin3)
case nEleccion == 4
avisos("Opcion Palabra
inhabilitada...2,cColLin3)
endcase
restscreen(0,0,maxrow(),maxcol(),cP
antalla)
setcolor(cColor1)
enddo

return
*****
PROCEDURE utileria2
local
aMenu:={space(10),"Agenda","Color
es","Raten"}
local
cScreen:=saveScreen(0,0,maxrow(),m
axcol())
static nEleccion
setcolor(cColor1)

while .t.

m_sombra(6,62,10,74,"EIE"%IIE%)

nEleccion:=MenuTB(7,63,9,73,aMe
nu)
setcolor(cColor1)

do case
case nEleccion == 0
exit
case nEleccion == 1
avisos("Opcion Agenda
inhabilitada...2,cColLin3)
case nEleccion == 2
avisos("Opcion Colores
inhabilitados...2,cColLin3)
case nEleccion == 3
avisos("Raten habilitado
!!!2,cColLin3)
endcase
enddo

restore screen from cScreen
setcolor(cColor1)

return
*****
PROCEDURE actualiza2
local
aMenu:={space(10),"Asesor","Mater
ias","Generacion"}
local
cScreen:=saveScreen(0,0,maxrow(),m
axcol())
local cScreen2
static nEleccion
setcolor(cColor1)
while .t.
m_sombra(6,48,10,59,"EIE"%IIE%)

nEleccion:=MenuTB(7,49,9,58,aMe
nu)

cScreen2:=saveScreen(0,0,maxrow(),
maxcol())
setcolor(cColor1)

do case
case nEleccion == 0
exit
case nEleccion == 1
avisos("Opcion Asesor
inhabilitada...2,cColLin3)
case nEleccion == 2
avisos("Opcion Materias
inhabilitada...2,cColLin3)
case nEleccion == 3
avisos("Opcion Generacion
inhabilitada...2,cColLin3)
endcase

restore screen from cScreen
setcolor(cColor1)
enddo

return
*****
PROCEDURE report2
local
aMenu:={space(10),"Creacion","Imp
resion"}
local
cScreen:=saveScreen(0,0,maxrow(),m
axcol())
local cScreen1
static nEleccion
setcolor(cColor1)

while .t.
m_sombra(6,35,9,45,"EIE"%IIE%)

nEleccion:=MenuTB(7,36,8,44,aMe
nu)
setcolor(cColor1)

do case
case nEleccion == 0
exit
case nEleccion == 1
avisos("Opcion Creacion
inhabilitada...2,cColLin3)
case nEleccion == 2
avisos("Opcion Impresion
inhabilitada...2,cColLin3)
endcase

restore screen from cScreen
setcolor(cColor1)

enddo
return

```

```

.....
.....
.....
.....
.....
.....
.....

```

```

//función que utiliza el objeto tBrowser
para aplicar un menú controlado
//por ratón, ya que cualquier otra
función de clipper (para manejo de
menú)
//no soporta ratón

```

```

static function
MenuTB(nRengSup,nCollzq,nRengI
nColDer,aElementos)
local nNumReng := 1
local nPosX := 0
local nPosY := 0
local nBotonPres := 0
local nCar := 0
local nl := 1
nNumA := len(aElementos)

oVisualiz :=
tBrowsenew(nRengSup,nCollzq,nRen
glInColDer)
oVisualiz.skipblock := {{nl := nl +
VerifSalto(nRengAct,aElementos)}}
oVisualiz.addcolumn(tbcolumnnew(
{{aElementos[nl]}))
//VerifClic() //rutina en ensamblador
while !(nCar == K_ESC).and.
!(nCar == K_ENTER)

```

```

while !(oVisualiz.stabilize())
enddo

```

```

if IRatPresen

```

```

PosRaton(@nPosX,@nPosY,@nBoto
nPres)
endif
//si se oprime el botón derecho se
pasa un ESC al teclado si no se pasa
ENTER
do case
case nBotonPres == 3;Keyboard
chr(27) //derecho
case (nBotonPres == 2) //and.
(VerifClic() == 1) //izquierdo
if (nPosX >= nCollzq).and.
(nPosX <= nColDer)
if (nPosY >= nRengSup).and.
(nPosY <= nRengInf)
if oVisualiz:rowPos > (nPosY -
nRengSup + 1)
while oVisualiz:rowPos >
(nPosY - nRengSup + 1)
oVisualiz:up()
while !(oVisualiz.stabilize())
enddo
nNumReng--
enddo
elseif oVisualiz:rowPos < (nPosY -
nRengSup - 1)
while oVisualiz:rowPos <
(nPosY - nRengSup + 1)
oVisualiz:down()
while !(oVisualiz.stabilize())
enddo
nNumReng++

```

```

enddo
endif
else
keyboard chr(27)
//K_ENTER //enter
endif
else
keyboard chr(27)
//K_ENTER //enter
endif
case (nBotonPres == 1) //or.
(VerifClic() > 1) //izquierdo
if (nPosX == nCollzq).and.
(nPosX <= nColDer)
if (nPosY >= nRengSup).and.
(nPosY <= nRengInf)

```

```

if oVisualiz:rowPos = (nPosY -
nRengSup + 1)
nNumReng:=oVisualiz:rowPos
keyboard chr(K_ENTER) //enter
else
TONE(900,1)
endif
else
keyboard chr(27) // _ENTER)
//enter
endif
else
keyboard chr(27)
//K_ENTER //enter
endif

```

```

endcase
if IRatPresen
while nBotonPres > 0

```

```

PosRaton(@nPosX,@nPosY,@nBoto
nPres)
enddo
endif
nBotonPres := 0
if !(inkey() == 0)
nCar := inkey()
else
loop
endif
//interpreta valor de inkey()
do case
case nCar == K_UP.or.nCar ==
K_LEFT
if oVisualiz:rowPos < 1
if !(oVisualiz:htTop)
oVisualiz:up()
nNumReng--
endif
endif
case nCar == K_DOWN.or.nCar
== K_RIGHT
if oVisualiz:rowPos < oNumA-1
//len(aElementos) 3
if (oVisualiz:htBottom)
oVisualiz:down()
nNumReng++
endif
endif

```

```

endcase
enddo

```

```

return iff(lastkey() == K_ESC,
0,nNumReng)
.....
function
VerifSalto(nRegAct,nElementos)
if nRegAct == nElementos
return 0
else
return 1
endif
return 0
.....

```