

37
24



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"CAMPUS ARAGÓN"**

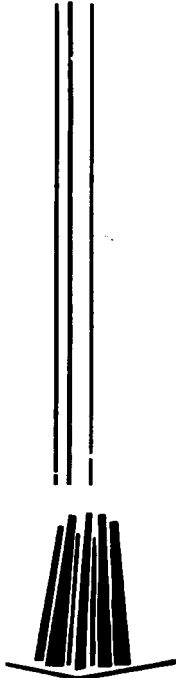
**"SISTEMA INFORMÁTICO
ADMINISTRATIVO PARA LOS CURSOS
DEL DEPARTAMENTO DE EDUCACIÓN
CONTINUA DE LA UNAM
CAMPUS ARAGÓN"**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACION**

P R E S E N T A :

CESAR MANDUJANO LUNA



MÉXICO

1997

**TESIS CON
FALLA DE ORIGEN**



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.

Dedico la presente tesis a la memoria de mi padre, el Sr. Adolfo Mandujano Tovar quien seguramente se hubiera sentido orgulloso de que su segundo hijo terminara sus estudios universitarios. También le dedico a las personas con las que convivo y disfruto diariamente de momentos bellos y gratos, y que forman parte de mi motivación diaria. Me refiero a mis hermanos Adolfo, René, Jaime, Tania y Oscar. Pero en particular, quiero dedicarla a mi madre la Sra. Maura Luna Pérez de quien admiro esa fuerza indomeñable de salir siempre avante no importando la naturaleza del problema que se le presente y de quién he aprendido que en esta vida nunca debo darme por vencido.

Doy gracias a Dios por haber concluido el presente estudio y también por haberme dado la oportunidad de pertenecer y formar parte de esta hermosa familia.

¡ GRACIAS ! ¡ MIL GRACIAS SEÑOR !

CONTENIDO

▫ INTRODUCCIÓN	VII
▫ CAPÍTULO 1	Bases de datos y sistemas de gestión de bases de datos 1
1.1	Breve reseña histórica 1
▫ CAPÍTULO 2	El departamento de educación continua de la UNAM Campus Aragón. Necesidades y problemática - 9
2.1	Análisis del problema 9
▫ CAPÍTULO 3	Alternativas de solución 27
3.1	Posibles soluciones 27
3.2	Estudio de los diferentes formatos 29
3.3	Las estructuras definitivas de las bases de datos 33
	La estructura de la base de datos de carreras. La estructura de la base de datos de expositores 34
	La estructura de la base de datos de ordenes de pago 35
	La estructura de la base de datos de cursos .. 37
3.4	Relación existente entre las bases de datos .. 39
3.5	Módulos del programa propuesto 40
	El módulo de altas 42
	El módulo de bajas 43
	El módulo de consultas 44
	El módulo de modificaciones 46
	El módulo de reportes 47

3.6	El módulo de disposición	50
	Selección del manejador de bases de datos y del lenguaje de programación	52
=CAPÍTULO 4	Programa propuesto : el "DEC" versión 1.0	53
=CAPÍTULO 5	Manual del usuario	259
5.1	Hardware necesario	259
	La memoria RAM	259
	El microprocesador y la tarjeta madre	259
	El disco duro	260
	Los manejadores de disco	260
	La impresora	261
	El No Break	261
5.2	Software necesario	261
5.3	Instalación y arranque del programa DEC	262
5.4	Los menús y pantallas del programa DEC versión 1.0	263
	La pantalla de presentación	263
	La pantalla de solicitud de clave de acceso	264
	La pantalla del menú principal	266
	El menú general de altas	269
	La pantalla de altas de cursos	270
	La pantalla de altas de ordenes de pago	276
	La pantalla de altas de expositores	283
	La pantalla de altas de carreras o profesiones	284
	El menú general de bajas	285
	La pantalla de bajas de cursos	286
	La pantalla de bajas de ordenes de pago	289
	La pantalla de bajas de expositores	290
	La pantalla de bajas de carreras o profesiones	291
	El menú general de consultas	292
	El menú de tipo de consultas	292
	La pantalla de consultas de cursos globales ..	294
	La pantalla de consultas de cursos por clave ..	295

El menú de consultas de cursos por cadena incluida	296
Los menús de consultas por cadena incluida de ordenes de pago, expositores y profesiones	299
El menú general de modificaciones	301
La pantalla de modificaciones de cursos	302
La pantalla de modificaciones de ordenes de pago	303
La pantalla de modificaciones de expositores	303
La pantalla de modificaciones de carreras o profesiones	304
El menú general de reportes	305
La pantalla de avance de cursos	306
La pantalla de cursos por vencer	310
La pantalla de lista de deudores	316
Las pantallas del registro de inscripción y reporte de recibos	316
La pantalla de lista de asistencia de cursos para profesores	324
La pantalla de reporte a la unidad de planeación	326
El menú general de disposición o configuración	328
La pantalla de selección de unidad de trabajo y/o ruta	328
El menú de tipo de impresora	329
La pantalla de datos del oficio	330
La pantalla de selección de nueva clave de acceso	331
CONCLUSIONES Y RECOMENDACIONES	334
BIBLIOGRAFÍA	336

" Todas las personas sin distinción alguna, llegamos al mundo con varias propiedades comunes, entre ellas : inteligencia, corazón y tiempo. Pero es su correcto uso el que permite marcar la diferencia entre los ganadores y los perdedores, entre aquellos que tienen éxito y aquellos que no lo poseen."

CESAR MANDUJANO LUNA

" El proceso de adquisición de conocimiento se puede ver como un arduo e interminable camino, el cual al final de cuentas conduce a enormes beneficios para aquellas personas que se atreven a tomarlo. Caso contrario, lo ofrecen aquellas personas que no lo toman y se arriesgan a viajar en el camino que conduce a la nada : el de la ignorancia. Porque si bien es cierto, es preferible y más ligero el camino del conocimiento, que el pesado camino que implica la ignorancia."

CESAR MANDUJANO LUNA

" Uno tiene que acercarse a lo que es radicalmente nuevo, con una mente en blanco, rechazando conscientemente el intento de ligar la historia con lo que ya es familiar, porque lo familiar es desesperadamente inadecuado."

EDSGER DISKSTRA

INTRODUCCIÓN

Todos los sistemas sin distinción alguna están sujetos a cambios, motivados por las nuevas demandas o requerimientos que les exige su medio ambiente. Por ello, todos los organismos con el afán de poder brindar un servicio con mayor calidad y eficiencia, deben modificar continuamente su forma de trabajar para actualizarse y no quedar rezagados en comparación con otros. Para que se cumpla lo anterior, es importante la renovación del organismo, y es gracias a la adecuación en el mismo de conceptos y tecnologías existentes e innovadoras mediante las cuales subsisten, ya que les permiten a éstos a obtener buenos resultados y por añadidura rendimientos. Al mismo tiempo, les ayuda a no convertirse en obsoletos y a mantener vigentes sus objetivos de creación y su función dentro de la sociedad.

En la actualidad el uso cada vez más frecuente de las computadoras en los organismos, se ha convertido ya no tanto en una necesidad sino en un elemento vital e indispensable, ya que les permite agilizar y simplificar sus tareas.

La presente tesis tiene como objetivo principal la instalación de un sistema informático que permita facilitar la planeación, organización y control de los cursos impartidos por el Departamento de Educación Continua de la UNAM Campus Aragón (que a futuro llamaremos DEC Aragón) y como objetivos secundarios :

1. Reducir el volumen de la información guardada en archivos convencionales de oficina, por archivos almacenados en medios de almacenamiento secundario para computadoras, como lo son : los discos duros y los discos flexibles.
2. Facilitar la búsqueda y consulta de los datos de manera veraz y oportuna.
3. Evitar la redundancia en la información.
4. Mejorar el sistema de captura y procesamiento para agilizar la entrega de formatos y reportes.
5. Evitar tener existencia de papelería empleada en los formatos, para tener la posibilidad de imprimirlos a través de una impresora conectada a la computadora.
6. Apoyar en el monitoreo de cursos, avisando al usuario con anticipación el avance de cada curso.

En el capítulo I se presenta una breve reseña histórica de las bases de datos. Su intención es plantear la evolución de las bases de datos y paralelamente introducir y dotar al lector de conceptos empleados constantemente en las mismas.

En el capítulo II se exponen las actividades que realiza el DEC Aragón, mencionando la manera en que este procede para poder constituir y llevar a cabo un curso, detallando aquellos aspectos principales de su funcionamiento.

En el capítulo III se realiza un análisis en el que se trata de entender el sistema que actualmente emplea el DEC Aragón, así como las nuevas necesidades planteadas por los usuarios y la dirección. Se identifican los problemas a los que diariamente se enfrenta y se hace un estudio de los formatos utilizados tanto de entrada como de salida y de la manera en que estos son llenados. En base a lo anterior se propone un sistema basado en computadora, principalmente integrado por en cuatro bases de datos y un programa controlador de las mismas. También se realiza el diseño de la base de datos física, en cuanto a su composición de campos, tipos y longitudes de los mismos (lo que se conoce como estructura) y de los módulos y submódulos que van a formar al sistema a realizar.

En el capítulo IV se presenta el código fuente del programa propuesto para cubrir las expectativas del DEC Aragón, el cual está escrito en CLIPPER¹. El programa se llama "DEC" y son las siglas de Departamento de Educación Continua. Este programa está dividido en varios módulos cada uno de los cuales lleva su respectiva explicación para facilitar su comprensión. El programa maneja su propio sistema de ayuda, la cual el usuario puede utilizar en cualquier momento y de persistir dudas puede consultar el manual de referencia del usuario.

En el capítulo V se presenta un manual de referencia del usuario. En él, se explican las características del equipo requerido para que el programa tenga un mejor desempeño. Al mismo tiempo, el capítulo sirve como un manual de consulta para los usuarios principiantes o para aquellos que tengan dudas sobre el uso del mismo.

¹ La denominación Clipper® es una marca registrada por Nantuket Corporation®.

CAPÍTULO 1.

BASES DE DATOS Y SISTEMAS DE GESTIÓN DE BASES DE DATOS.

En términos generales, todos hemos utilizado una base de datos. Por ejemplo, cuando empleamos nuestra agenda para consultar el teléfono y/o dirección de un familiar o amigo, o bien, cuando requerimos de los servicios de una empresa, consultamos los catálogos de especialidades y seleccionamos el más adecuado según nuestras necesidades.

En informática una base de datos se puede definir como una "colección organizada de datos de diversas índoles, que mantienen una estrecha relación entre sí y sirve para diferentes propósitos según los intereses de los usuarios que las emplean". A su vez, estas bases de datos son administradas por un programa que manipula sus datos ya sea para ordenarlos, buscarlos, reagruparlos y modificarlos. Este sistema es llamado "sistema de gestión de la base de datos" o "sistema manejador de la base de datos", que a futuro representaremos por las siglas DBMS (Database Management System.).

En el presente capítulo realizaremos un resumen sobre la historia de las bases de datos y al mismo tiempo, definiremos los conceptos que comúnmente suelen emplearse en ellas.

1.1. BREVE RESEÑA HISTÓRICA

La existencia de las bases de datos está ligada a el desarrollo de los medios de almacenamiento de información. De hecho, el medio de almacenamiento de información que permitió dicho desarrollo, fueron las tarjetas perforadas. Estas tarjetas creadas por el francés Joseph Marie Jacquard originalmente eran (y siguen siendo empleadas por algunas compañías en la actualidad) utilizadas en la industria textil, y en ellas se guardaba el programa que contenía los puntos en que se deben de disponer las agujas de un telar para confeccionar automáticamente una tela.

Pero es propiamente el uso de estas tarjetas perforadas para registrar la información de los censos de la población de los E.U. del periodo 1880 - 1890 con la que arranca la historia de las bases de datos.

En el año de 1886 el Dr. Herman Hollerit de la oficina de censos de los E.U. concluyó que sería imposible terminar el censo de trece millones de norteamericanos por los métodos de computo existentes. La realización de los censos eran, por aquellos tiempos algo complicado y porque no decirlo tedioso. En primer lugar los sitios a censar se encontraban muy dispersos y los medios de comunicación y transporte por sus características alentaban la transportación de la información, que en ocasiones, no llegaba a su destino. En segundo lugar la información al concentrarse requería de espacios grandes para depositarla debido a su volumen, lo que paralelamente entorpecía la clasificación y conteo de los datos recabados, que se realizaba por cierto manualmente, por lo que los censos no eran tan precisos que digamos, ya que siempre existía la probabilidad de incurrir en errores humanos (clasificación inadecuada, pérdida de conteo, traspapelo de datos, etc.).

Hollerit identificó dos problemas importantes a dar solución, a saber : crear un nuevo método de trabajo que permitiera acelerar la clasificación y conteo de la información y al mismo tiempo reducir el volumen de almacenamiento de la misma para terminar con el censo.

La solución que encontró, fue la adecuación de las tarjetas perforadas de los telares de Jacquard para almacenar (registrar) por grupos de características (campos) los datos (valor de los campos) de los censos, dando origen con ello a los ficheros (archivos) de tarjetas perforadas. Aquí surgen varios conceptos como : archivo, registro, campo y dato que tanto usamos en la jerga informática y en particular, en el de las bases de datos.

El sistema de ficheros de tarjetas implementado por Hollerit tuvo tanto éxito, que fue implantado en otros países. Este sistema de ficheros de tarjetas fue usado en E.U. desde su creación hasta los siguientes sesenta años.

Pero, para la década de los cuarenta el sistema se enfrentó a un nuevo cambio : "la población era cada vez más grande, por lo que los ficheros de tarjetas crecieron y dificultaron enormemente su manejo por su volumen, y además, al final de cuentas, las tarjetas perforadas no eran un medio muy adecuado para almacenar datos, esto se debe a que en ocasiones eran mal perforadas o a veces, si todo un paquete caía accidentalmente era muy difícil organizarlo de nuevo". Para mala suerte de la oficina de censos se venía en puerta el próximo censo de 1950, de tal suerte que se requería de un nuevo sistema que fuera más rápido para realizar los cómputos y no ocupara mucho espacio.

Precisamente la década de los cuarenta está caracterizada por un evento ecuménico muy importante : la segunda guerra mundial y con ella (como en todas las guerras) el advenimiento de nuevos descubrimientos científicos y avances

tecnológicos. Claro esta, no nos referimos a la invención de la bomba atómica, si no es que, es precisamente en esta década, en donde la primera generación de computadoras² alcanza su cúspide. Y es en la Universidad de Pensilvania en el año de 1946 en donde los profesores Eckert y Mauchley crean la ENIAC (Electronic Numeric Integrator And Calculator) para el departamento de la defensa de los E.U. con el fin de emplearse en cuestiones de balística. La ENIAC como fue creada para fines científicos tenía como gran ventaja su velocidad de procesamiento, aunque en contrapartida carecía de capacidad de almacenamiento. Curiosamente es por estas fechas cuando se inventa un recurso muy importante para el desarrollo de las bases de datos : la cinta magnética.

Posteriormente en 1951 Eckert y Mauchley inventan la UNIVAC-1 (Universal Automatic Computer 1), la cual empleaba cinta magnética para almacenar información de campos variables con avance tanto hacia atrás como hacia adelante. La UNIVAC-1 fue entregada a la oficina de censos de E.U. para dar abasto y terminar los censos de 1950. Este hecho permitió que se trasladaran los conceptos de archivo, registro, campo y dato a la cinta magnética.

En la década de los cincuenta surgen los analistas de sistemas y con ello los primeros programas tendientes a administrar datos. Estos programas eran por lo general, pequeños subsistemas de nómina.

Sin embargo, a pesar de que la cinta magnética revolucionó de alguna forma a los sistemas de almacenamiento, todavía presentaba una ligera desventaja : el proceso de lectura y/o escritura de la misma es de forma secuencial y por consiguiente se tiene siempre que pasar de un registro a otro lo cual involucra una pérdida de tiempo, independientemente del tamaño de la base de datos. Por ejemplo, supongamos que estamos consultando una cinta magnética con 1000 registros almacenados en ella. Imaginemos que el lector de la cinta se encuentra ubicado en el registro número 100 y requerimos la información del registro 875. ¡Tenemos que recorrer la cinta 775 registros (que de antemano ni nos interesan consultar en ese momento) y esperar hasta que llegue al registro que queremos !

² Hasta el momento se considera que han existido cuatro generaciones de computadoras : a) la primera se ubica en el periodo de 1939 a 1945 y se caracterizaba por ocupar un gran volumen, tener un funcionamiento a base de bulbos y como consecuencia disipaban una enorme cantidad de potencia, lo que ocasionaba que la vida útil del bulbo fuera corta y tuviera que ser reemplazado continuamente. Su programación era muy laboriosa ya que se realizaba por medio de interruptores. Cada interruptor se encendía o apagaba representando los números 1 y 0 respectivamente, y cierta cadena de estos podía constituir ya sean datos o instrucciones (lo que se conoce como lenguaje máquina), por lo que si existía un error en el programa el detector en que interruptor se incurrió la falla era tremendamente embarazoso; b) la segunda generación está comprendida entre 1946 a 1963 y se caracteriza por funcionar a base de transistores y en disminuir su tamaño y propagación de calor con respecto a la primera. Su programación es en base a mnemónicos que son contracciones de palabras del idioma inglés (lo que se conoce como lenguaje ensamblador) y servían para que los programadores no que tuvieran que memorizar cada una de las cadenas de unos y ceros; c) la tercera generación está ubicada entre 1964 a 1973 y se caracteriza por funcionar a base de chips que no son más que integrados capaces de contener miles de transistores lo que motivó una disminución en tamaño y difusión de calor. Su programación es en base a instrucciones escritas en su mayoría en palabras completas del idioma inglés, lo cual facilita su programación (lo que se conoce como lenguaje de alto nivel); d) la cuarta y generación abarca desde 1974 hasta la actualidad y se caracteriza por el uso de microprocesadores los cuales contienen internamente varias circuitos integrados donde cada uno realiza una actividad específica. Su tamaño es pequeño y por lo mismo su disipación de potencia. Su programación está en base a lenguajes de alto nivel y a programas de aplicación específica, esto es, programas especializados en ciertas tareas.

A mediados de los sesenta se inventa la solución de al problema de acceso secuencial : el disco magnético. Este último presenta la enorme ventaja de tener acceso directo a la información. Con este invento nace la primera generación de bases de datos, la cual se caracteriza por preocuparse sobre todo en buscar nuevos modelos de datos que optimasen la colocación y recuperación de los mismos en las memorias secundarias³ y con ello disminuir los tiempos de acceso. De antemano surgen aquí dos de los tres modelos de bases de datos existentes : el *modelo jerárquico* y el *modelo red*.

El modelo jerárquico fue desarrollado por Charles W. Bachman, a quien se le atribuye el nombre de el padre de las bases de datos. Este modelo se basa en organizar los datos de los registros en niveles de autoridad y su representación gráfica tiene la forma de un árbol invertido, ya que comienzan desde la raíz y terminan en las hojas.

Gracias al modelo jerárquico, surge el concepto de los " sistemas de gestión de información " (MIS) que no eran mas que programas vendidos por paquete para ser aplicados sobre los archivos de salida requeridos. Los MIS tuvieron una buena demanda y se realizaban integrando la estructura de los registros en el mismo programa, o sea, las características de los campos de los registros eran incluidos en el programa. Esto último originó serias consecuencias e implicaciones : " los sistemas eran muy rígidos y sólo resolvían las necesidades inmediatas y momentáneas que requerían los usuarios, pero descuidaban enormemente los posibles cambios futuros que requiriera el sistema y además existía en muchos casos duplicidad de los datos ".

Por ejemplo, supongamos un sistema de nóminas muy sencillo el cual genera el reporte mostrado en la figura 1.1.

NOMBRE DEL EMPLEADO	DÍAS TRABAJADOS	SALARIO DIARIO	IMPUESTO	SALARIO NETO
Cesar Mandujano	15	50.00	37.50	712.50
Gustavo Martínez	14	50.00	35.00	665.00
.
Itsmael Manzo	13	45.00	29.25	555.75
Adriana González	15	55.00	41.25	783.75

Figura 1.1. Reporte inicial del sistema de nóminas.

³ Suele clasificarse a las memorias de los equipos de cómputo en dos tipos : la memoria principal o primaria es la memoria de trabajo del usuario, esta tiene la característica de ser muy rápida y retiene la información almacenada en ella, siempre y cuando no se le deje de suministrar energía; y la memoria secundaria permite respaldar toda clase de archivos de importancia para el usuario de manera permanente y para usos posteriores, tal es el caso de los discos magnéticos, las cintas magnéticas, los discos ópticos, etc.

Imaginemos ahora que transcurre el tiempo y surge como nueva necesidad el reporte presentado en la figura 1.2.

NOMBRE DEL EMPLEADO	DOMICILIO	TELÉFONO
Cesar Mandujano	Matías Romero No. 50 Colonia del valle	793-6715
Gustavo Martínez	Lago Chalco No. 125 Colonia Anáhuac	552-2399
Itsmael Manzo	Norte 45 No. 39 Colonia Pantilán	798-2412
Adriana González	Río Tiber No. 245 Colonia Cuauhtémoc	625-2527

Figura 1.2. Nuevo reporte del sistema de nóminas.

Pero como podemos observar "existen dos nuevos campos (domicilio y teléfono) que no están declarados dentro del programa". Aquí existen dos posibles propuestas para "remendar" el problema.

La primera es modificar el programa declarando en el mismo los campos nuevos, adecuar los demás módulos afectados y finalmente realizar el nuevo módulo y su correspondiente base de datos para generar dicho reporte. ¡Complicado ! ¡ Verdad !

La segunda sería crear un nuevo programa que contemple y defina los tres campos requeridos y se capture la nueva base de datos para generar el reporte. Pero si hacemos esto, tenemos que duplicar los datos del campo empleado, invertir tiempo en volver a capturar esta misma información y desperdiciar espacio en el medio de almacenamiento.

Suponiendo que se haya optado por cualquiera de estas propuestas " el desarrollo de las mismas involucra tiempo y dinero ", y yendo a los extremos, que sucederías si las necesidades vuelven a cambiar, si se requiere de un nuevo y diferente reporte con diferentes campos a los ya establecidos.

Como se puede notar estos sistemas tienen muchos problemas en su etapa de mantenimiento y como vemos, lejos de solucionar el problema lo complican más y nos envuelven en un círculo vicioso, en el cuento de nunca acabar.

La figura 1.3 ilustra la manera de desarrollar los sistemas en ese tiempo. En ella se puede apreciar la dependencia directa de la forma en que se almacenan físicamente los datos y el programa de aplicación que los controla.

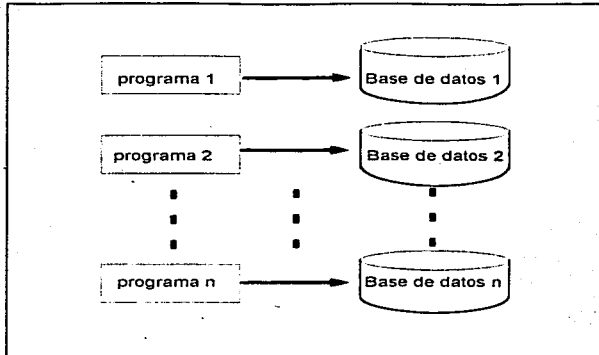


Figura 1.3. Dependencia directa en la forma de almacenar físicamente los datos y los programas.

Lo anterior permitió reconocer que las bases de datos deben de ser independientes del programa y lenguaje que las manipule.

A comienzos de los setenta se trato de encontrar nuevas alternativas para solucionar las deficiencias del modelo jerárquico. Por ello el CODASYL (Conference On Data System Languages) invirtió importantes sumas de dinero y desarrollan un nuevo modelo en 1971 : el modelo red o CODASYL.

El modelo red permite enlazar datos de una manera más eficaz que su antecesor y de esta forma reducir la redundancia de los mismos. Pero adicionalmente, este modelo lanza un concepto en demasía importante : el manejador de la base de datos (DBMS).

En la figura 1.4 se puede apreciar la relación existente entre los programas, el DBMS y las bases de datos. En ella se aprecia la independencia e integridad que existe en la base de datos.

Paralelamente al modelo red el Dr. Edgar F. Codd llamado padre de la investigación de las bases de datos presenta el nuevo e incluso vigente modelo : el *modelo relacional*.

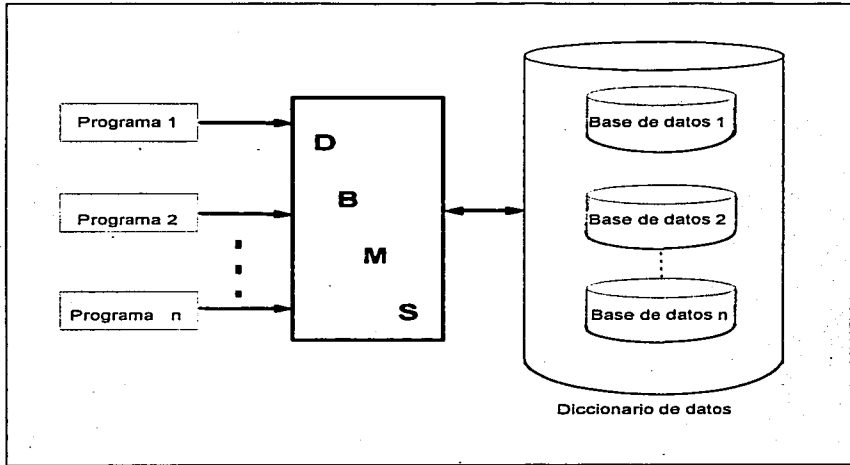


Figura 1.4. Relación existente entre los programas, los datos y el DBMS.

El modelo relacional se basa en el concepto matemático de relación. Este modelo agrupa la información en forma de tablas facilitando la extracción de datos de una tabla, modificación de los datos existentes en una tabla y agregar nuevos datos a una tabla.

Con la llegada del modelo relacional surge la segunda generación de bases de datos la cual se caracteriza por el uso de este modelo y por la creación de lenguajes de afirmación que permiten obtener los datos deseados, sin necesidad de decir como accederlos.

En los años ochenta surge una alta comercialización de DBMS basados en el modelo relacional. Pero es gracias a Waine Ratliff y a Ashton-Tate, que la concepción en cuanto a la construcción de manejadores de bases de datos cambia, al lanzar al mercado el muy popular y conocido DBASE⁴. Este manejador de hecho se convirtió en el estándar a seguir por otros manejadores. De ahí la gran similitud que tienen con el DBASE el FOXPRO⁵ y el CLIPPER, entre otros más.

Finalmente, mencionaremos que a comienzos de esta década, se comenzó a desarrollar en los centros de investigación, la tercera generación de bases de datos la cual contempla el uso de lenguajes más potentes y naturales y permite la posibilidad de deducciones y de distribución.

⁴ Ashton-Tate® era la compañía que originalmente poseía los derechos del Dbase®, pero en la actualidad estos derechos pertenecen a Borland Corporation®.

⁵ La denominación FoxPro® es una marca registrada por Microsoft Corporation®.

CAPÍTULO 2.

EL DEPARTAMENTO DE EDUCACIÓN CONTINUA DE LA UNAM CAMPUS ARAGÓN. NECESIDADES Y PROBLEMÁTICA.

El DEC Aragón, es un organismo perteneciente a la Universidad Nacional Autónoma de México, creado durante la administración del M. en I. Claudio C. Merrifield Castro en el año de 1991.

Su objetivo primordial es coadyuvar en la elevación del grado y desarrollo de la educación de la sociedad en general, a través de la impartición de diferentes clases de cursos, contribuyendo con ello al progreso económico, político, social, tecnológico y cultural del país y por ende al mejoramiento del nivel de vida de los participantes que acuden a sus cursos.

En este capítulo, presentamos un detallado análisis que permitirá conocer la forma actual de trabajar en el DEC Aragón. Esto ayudará y servirá para entender su problemática y determinar sus requerimientos, que al mismo tiempo, nos permitirán sugerir soluciones.

2.1. ANÁLISIS DEL PROBLEMA

Mes a mes el DEC Aragón elabora un calendario de cursos para invitar a la comunidad a participar en ellos. Estos cursos son muy variados en cuanto a sus contenidos y objetivos, ya que cuenta con diversos programas de capacitación, de apoyo, de actualización y culturales que pueden estar orientados a niños, jóvenes y adultos de diferentes edades. También se imparten cursos para estudiantes de diferentes niveles educativos o bien profesionistas interesados en alguna especialidad, la cual se puede ofrecer por medio de maestrías y diplomados.

Gracias al éxito obtenido por la aceptación de la comunidad, los cursos se pueden impartir tanto interna como externamente. Los cursos internos se imparten en las instalaciones de la UNAM Campus Aragón y los cursos externos se imparten en los domicilios de las instituciones o de empresas de carácter público o privado que los soliciten.

El Departamento de Educación Continua de la UJAM Campus Aragón. Necesidades y Problemática

En la figura 2.1 se presenta la evolución de los cursos impartidos en el DEC Aragón citando los hechos de principal relevancia ocurridos en el mismo en el período de 1991 hasta 1996.

FECHA	HECHOS RELEVANTES
1991	<ul style="list-style-type: none">- Se impartieron un total de 31 cursos de los cuales fueron 13 de capacitación, 08 de actualización y 15 cursos varios.- Se establecen los primeros nexos con entidades públicas y privadas para ofrecer los servicios de educación continua.
1992	<ul style="list-style-type: none">- Se impartieron un total de 93 cursos con una asistencia de 2,296 participantes. De estos cursos 59 fueron de capacitación, 29 de actualización y 05 de varios. De estos 93 cursos 88 se impartieron en las instalaciones de la UNAM Campus Aragón.- Se obtuvo la membresía de la Asociación Mexicana de Educación Continua (AMEC).
1993	<ul style="list-style-type: none">- Se incluyen los programas de apoyo a la titulación y de las materias de alto índice de reprobación.
1994	<ul style="list-style-type: none">- Se impartieron un total de 225 cursos con una asistencia de 4,800 participantes. De estos cursos 96 fueron de capacitación, 114 de actualización y 15 de varios.
1995	<ul style="list-style-type: none">- Se impartieron un total de 99 cursos con una asistencia de 904 participantes. De estos cursos 39 fueron de capacitación, 57 de actualización y 3 de varios.
1996	<ul style="list-style-type: none">- Se impartieron un total de 89 cursos con una asistencia de 1,354 participantes. De estos cursos 37 fueron de capacitación, 50 de actualización y 2 de varios.

Figura 2.1. Evolución de los cursos impartidos por el DEC Aragón.

Como se puede apreciar en la figura 2.1, la cantidad de cursos y asistentes a los mismos anualmente es considerable, y además se espera que su tendencia sea en aumento.

Es importante señalar que todas las actividades relacionadas a la administración del DEC Aragón desde la planeación y organización de cursos hasta el control de los mismos, se realizan manualmente, de tal forma que la captura de datos, los cálculos, el procesamiento, el llenado y la realización de la mayor parte de los formatos y reportes son llevados a cabo primero en un borrador a mano y luego estos son redactados en máquinas convencionales de escribir, lo que produce salidas poco profesionales. Ahora bien, dichos reportes contienen información que otros formatos también utilizan, por lo que se tiene que redactar la misma información para varios reportes, motivando una tendencia a generar error humano y también pérdida de tiempo por manejar información recurrente.

Pero mejor mencionemos el conjunto de actividades que se llevan a efecto para la programación y realización de los cursos :

a) En el caso de los cursos internos :

- 1) Primeramente se procede a la planeación del *calendario de cursos*. Para ello el DEC Aragón se auxilia de los profesores que han trabajado para él y/o estén interesados en impartir cursos. Con el fin de detectar el perfil adecuado según las características del curso que se pretende impartir, se recurre a la consulta de los currículum vitae de los mismos. De esta manera se seleccionan a los profesores y se les notifica para saber si desean impartir el curso y confirmar su participación en el mismo. En caso de aceptar el profesor, se toma en un registro de participación de profesores (que se verá mas adelante) sus datos contenidos en el currículum vitae.

OBSERVACIÓN 1. Las listas de currículum vitae, se mantienen guardados en archivos convencionales de oficina, lo cual de alguna manera involucra espacio por el volumen de papeles y dificulta la búsqueda de expedientes y consultas de los mismos.

OBSERVACIÓN 2. Los datos de los profesores se toman del sus currículum vitae y se apuntan en el registro de participación de profesores.

- 2) Ya seleccionados los cursos a impartir y los profesores, se procede a la organización y elaboración formal del calendario de cursos. Aquí se asigna a los cursos una aula, un profesor, una fecha de exposición, una duración en horas, los días de la semana en los cuales se imparta el curso, un horario, un salón si el curso se realizara en las instalaciones de la UNAM Campus Aragón, un costo y finalmente los requisitos necesarios que requieren los alumnos para integrarse al curso. Los costos dependen de la condición del participante, esto es, si los participantes pertenecen a la UNAM, si son egresados de la misma UNAM Campus Aragón o bien es público en general. Para comprobar que el alumno es universitario necesita presentar su credencial vigente, si es egresado requiere presentar su historial académico o bien si es público en general únicamente se le pide una identificación. Además, pueden existir cursos que pidan ciertos conocimientos previos como requisitos, tal es el caso de los cursos de idiomas que se imparte a diferentes niveles, así un alumno se puede inscribir al curso "Inglés avanzado módulo II" si demuestra que ya aprobó el módulo I ya sea a través de su constancia de

aprobación del curso o bien a través de la aprobación de un examen de validación del nivel de conocimientos. Lo mismo sucede en el caso de algunos cursos de computación, en donde un curso sobre cierto lenguaje de programación requiere como requisitos que el alumno sepa manejar el tipo de computadora a utilizar en el curso y/o conozca el sistema operativo o el ambiente de trabajo que emplea ese lenguaje de programación (puede ser ambiente DOS⁹ o ambiente WINDOWS⁷). Ya terminado el calendario de cursos se procede a su redacción y finalmente se realiza la propaganda necesaria para promocionarlo a través de su publicación por medio de diferentes medios como lo son carteles, "posters", folletos, etc.

OBSERVACIÓN 3. El calendario de cursos se construye de manera manual basándose en un calendario del año vigente para saber en que días se deben impartir los cursos en función de su duración. Esto puede motivar un error humano al elaborarse.

- 3) Los alumnos interesados en apuntarse en algún curso acuden a las oficinas del DEC Aragón, en donde se les piden los requisitos establecidos en el calendario de cursos y de aprobarlos se procede a la entrega de un formato llamado *orden de pago* la cual se muestra en la figura 2.2 y es pagado en la caja. El importe de este recibo depende de dos aspectos : el primero está en función como se mencionó más arriba, de que si el participante pertenece a la UNAM, o es egresado o bien es público en general; el segundo aspecto está sujeto a los posibles acuerdos a los que se puede llegar con el jefe del DEC, ya sea para conseguir descuentos, becas o el pago del curso diferido en varias exhibiciones. Como se puede ver en la figura 2.2 esta orden de pago contiene datos del alumno tales como : nombre, teléfono, organismo o carrera de procedencia y su condición y semestre en caso de ser alumno de la UNAM Campus Aragón. También contiene datos del curso tales como : número de recibo, nombre del curso, fecha de exposición, nombre del expositor, total a pagar, la fecha de pago en la caja y la fecha de entrega del recibo de pago en la oficinas del DEC Aragón. De esta manera los alumnos llevan su comprobante de pago y se les registra en la lista de alumnos inscritos para el curso deseado llamada *registro de inscripción* (ver figura 2.3). El curso se abre siempre y cuando se llene el cupo o falten pocos integrantes para completarlo. En caso de no integrar el número de alumnos para abrir el curso se les hace una llamada telefónica para ver la alternativa

⁹ DOS son las siglas en Inglés de *Disk Operating System* (Disco del sistema operativo), que es el programa administrador de los recursos de la computadora PC.

⁷ WINDOWS[®] es un programa de ambiente gráfico en el cual se ejecutan diversas utilidades gráficas, cuya realización y derechos reservados son propiedad de Microsoft Corporation[®].

El Departamento de Educación Continua de la UNAM Campus Aragón. Necesidades y Problemática

de apuntarlos a otro curso que les haya agradado o bien se procede a la devolución del importe pagado. Aquí también se le notifica al profesor la no apertura del mismo. En caso de abrirse el curso se le notifica por teléfono tanto al alumno como al profesor asignado la apertura del mismo. En este caso se solicita al profesor pase a las oficinas del DEC Aragón para llegar un acuerdo sobre la comisión que va a recibir y en caso de ser profesor por honorarios debe pasar a firmar su contrato de prestación de servicios profesionales. También se lleva acabo la inclusión de sus datos en un formato llamado registro de profesores (ver figura 2.4). Este registro contempla datos de profesor como : nombre, registro federal de contribuyentes, domicilio, teléfono título y cédula profesional. También se recaban los datos del curso que va a impartir como : nombre del curso y fecha de exposición.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGÓN DEPARTAMENTO DE EDUCACIÓN CONTINUA		
ORDEN DE PAGO		
NOMBRE : RENÉ MANDUJANO LUNA		
CARRERA : INGENIERO MECÁNICO ELÉCTRICO		SEMESTRE : 10
No. CTA. : 8756213-7		TEL. : 793-6715
PROCEDENCIA ENEP ARAGÓN (X)	UNAM : _____	OTRA : _____
NOMBRE DEL CURSO : INTRODUCCIÓN AL LENGUAJE C++		
PERIODO : 04/10/1996 - 01/11/1996		
PROFESOR : ARTURO MÉNDEZ HERNÁNDEZ		
COSTO : \$ 500.00	REC. OFICIAL : 5665544	FECHA CAJA : 02/10/1996

Figura 2.2. Orden de pago.

OBSERVACIÓN 4. Se tiene que consultar continuamente el calendario de cursos para saber los requisitos para determinado curso, lo que requiere inversión de tiempo.

OBSERVACIÓN 5. Las notificaciones por teléfono tanto del alumno como de profesores se realizan revisando los recibos de pago y los expedientes del profesor respectivamente, lo cual obstaculiza la consulta.

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGÓN
DEPARTAMENTO DE EDUCACIÓN CONTINUA

REGISTRO DE INSCRIPCIÓN

Curso o taller: **INTRODUCCIÓN AL LENGUAJE C++**

Expositor: **ARTURO MÉNDEZ HERNÁNDEZ**

Salón: **650**

Fecha del curso: **04/10/1996 - 01/11/1996** *Duración:* **50 HRS.** *UNAM:* **500.00** *EGRES:* **700.00** *PUBLICO:* **800.00**

Nombre del alumno	Teléfono	No.- Cuenta	Carrera	Semestre	Fecha	Rec. Of.	Importe	Notas
1.- RENÉ MANDUJANO LUNA	793-6715	87562137	ING MEC ELEC.	10	02/10/1996	5665344	1,000.00	
2.- TANIA CAMPOS NAVA	652-8511	00000000	NINGUNA	00	02/10/1996	5665588	1,100.00	
3.- NELLY ESCUDO PLATAS	456-5535	00000000	NINGUNA	00	01/10/1996	5660020	1,100.00	
4.- RUBEN LÓPEZ PÉREZ	789-3256	88754122	ING CIVIL	06	29/09/1996	5611212	0.00	Beca completa
5.- JAVIER CANO HIGUERA	230-6658	00000000	NINGUNA	00	28/09/1996	5611200	1,100.00	
6.- CARMEN DÍAZ DÍAZ	458-3317	89123654	ING CIVIL	06	28/09/1996	5611209	500.00	Crédito del 50%
7.- HUGO ARZATE TREVÍNO	694-1287	00000000	NINGUNA	00	27/09/1996	5569470	1,100.00	
8.- CARLOS SANTOS VEGA	345-8596	86258774	ICOM (C.U.)	04	29/09/1996	5622555	1,100.00	
9.- OSCAR LUJAN LINO	796-5214	89632857	ING MEC ELEC.	08	29/09/1996	5677777	500.00	Con descuento del 50 %
10.- AIDÉ RAZO MOLINA	658-9612	90598711	ING. COMP.	02	01/10/1996	5663784	1,000.00	

Figura 2.3. Registro de inscripción.

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGÓN
DEPARTAMENTO DE EDUCACIÓN CONTINUA**

REGISTRO DE PROFESORES

NOMBRE : ARTURO MÉNDEZ HERNÁNDEZ **RFC :** MEAHT00203TU2
DOMICILIO : DINAMARCA No. 25 COLONIA JUÁREZ
TELÉFONO : 564-96-87
TÍTULO PROFESIONAL DE : INGENIERO EN COMPUTACIÓN
NÚMERO DE CÉDULA PROFESIONAL : _____ **FECHA DE EXPEDICIÓN :** _____

NOTA . Los profesores deberán presentar, para su trámite de pago por concepto de impartición de curso o taller :
Si están adscritos a la UNAM, el último talón de pago.
Si son externos, su recibo de honorarios

Figura 2.4. Registro de profesores.

OBSERVACIÓN 6. El registro de profesores de algún modo dificulta la consulta de datos y también puede originar duplicación de información.

- 4) Antes de comenzar los cursos se procede a realizar las listas de asistencia de los mismos (ver figura 2.5). Esta lista además de auxiliar al profesor para llevar un mejor control sobre el grupo, permite determinar que alumnos cumplieron con el mínimo de asistencia requerido para hacerse acreedores a su constancia (claro está, no deben solamente cumplir con la asistencia, sino también deben de cumplir con otras actividades evaluativas enmarcadas en el curso.).

OBSERVACIÓN 7. Hay que tomar los datos del calendario de cursos y de las órdenes de pago para poder realizar las listas de profesores lo que involucra tiempo.

- 5) Todos los recibos de pago son agrupados y al mismo tiempo se realiza un informe por curso de los ingresos recabados y de los egresos por concepto de honorarios (ver figura 2.6) y también se entrega el registro de inscripción a la jefatura de presupuestos de la UNAM Campus Aragón.

OBSERVACIÓN 8. Nuevamente se incurre a la consulta y repetición de datos del calendario de cursos y las ordenes de pago para generar estos reportes.

- 6) Ya inicializados los cursos se procede a el monitoreo de los mismos, dándole un seguimiento al avance que estos tienen y también se elabora la relación de alumnos deudores. Para el caso de avance de cursos se revisan el calendario de cursos y la asistencia de los profesores. Es importante esta revisión ya que al saber cuando termina el curso se pueden programar la entrega de constancias, la liberación de las instalaciones asignadas para otro uso y la apertura de otro curso para lanzar la convocatoria necesaria. En el caso de la relación de deudores esta se realiza de manera manual calculando la diferencia que existe entre el costo del curso para ese alumno y el total pagado por el mismo. Esta lista es vital, ya que permite por una parte controlar los ingresos del departamento y por otra parte, permite el control de entrega de constancia de cursos a alumnos que hayan cubierto el importe total del costo del curso. Cabe mencionar que adicionalmente se requiere la elaboración de un informe de porcentaje de avance de cursos la unidad de planeación de la UNAM Campus Aragón el cual se muestra en la figura 2.7.

OBSERVACIÓN 9. El monitoreo de los cursos y la lista de deudores la realiza el responsable de la administración de los cursos, revisando el calendario de los mismos y las órdenes de pago. Esta misma persona se encarga de realizar el informe de ingresos y el informe de avance de cursos para la unidad de planeación. Los cálculos y formatos son realizados primero a mano para después mecanografiarlos, lo cual implica una inversión de tiempo para buscar expedientes, consultar contenidos, realizar los cálculos necesarios, pasar información a un borrador para finalmente estos últimos se redacten en máquinas de escribir. También es importante señalar que los informes siempre tienen el mismo formato.

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGÓN
DEPARTAMENTO DE EDUCACIÓN CONTINUA**

*****Of. Num.*****

****NOMBRE DEL JEFE DEL DEPARTAMENTO DE PRESUPUESTOS****

Jefe del Depto. de Presupuestos
P r e s e n t e.

Por medio del presente me permito enviar a Usted 10 recibos oficiales que amparan la cantidad de : \$ 8,500.00 (ocho mil quinientos pesos 00/100 M.N.) por concepto de ingresos recaudados en este Departamento.

Así mismo, solicito hacer el tramite para el pago de honorarios del profesor ARTURO MÉNDEZ HERNÁNDEZ, correspondientes al curso INTRODUCCIÓN AL LENGUAJE C++ que se lleva a cabo los días Miércoles, Jueves y Viernes del 4 de Octubre de 1996 al 1 de Noviembre de 1996.

Dicho honorario es por la cantidad de : \$ 4,000.00

Sin otro particular, aprovecho la ocasión para enviarle un cordial saludo.

A T E N T A M E N T E

"POR MI RAZA HABLARA EL ESPÍRITU"

San Juan de Aragón, Edo. de Méx., 20 de Agosto de 1996

EL JEFE DEL DEPARTAMENTO

****NOMBRE DEL JEFE DEL DEC ARAGÓN****

ANEXO : Recibos oficiales y la documentación correspondiente.

c.c.p. ****Nombre del director****

****Nombre del Secretario Administrativo****

****Nombre del Jefe de la Unidad de Extensión Universitaria****

****Contabilidad****

Figura 2.6. Reporte de recibos.

RELACION DE RECIBOS OFICIALES CORRESPONDIENTES AL CURSO INTRODUCCIÓN AL LENGUAJE C++ IMPARTIDO POR EL PROFESOR ARTURO LÓPEZ PÉREZ DEL 4 DE OCTUBRE DE 1996 AL 1 DE NOVIEMBRE DE 1996

1.- RENÉ MANDUJANO LUNA	5665544	02/10/1996	1,000.00	
2.- TANIA CAMPOS NAVA	5665558	02/10/1996	1,100.00	
3.- NELLY ESCUDO PLATAS	5660020	01/10/1996	1,100.00	
4.- RUBEN LÓPEZ PÉREZ	5611212	29/09/1996	0.00	Beca completa
5.- JAVIER CANO HIGUERA	5611200	28/09/1996	1,100.00	
6.- CARMEN DÍAZ DÍAZ	5611209	28/09/1996	500.00	Crédito del 50%
7.- HUGO ARZATE TREVIÑO	5569470	27/09/1996	1,100.00	
8.- CARLOS SANTOS VEGA	5622555	29/09/1996	1,100.00	
9.- OSCAR LUJAN LINO	5677777	29/09/1996	500.00	Descuento del 50 %
10.- AIDÉ RAZO MOLINA	5663784	01/10/1996	1,000.00	

TOTAL DE INGRESOS: \$ 8,500.00
HONORARIOS : \$ 4,000.00
NETO ... : \$ 4,500.00

Figura 2.6. < continuación >.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
CAMPUS ARAGÓN
SEGUIMIENTO DE AVANCE 02/02/1996 - 09/11/1997

STAFF

INSTANCIA : DEPTO. DE EDUCACIÓN CONTINUA

RESPONSABLE : ***NOMBRE DEL RESPONSABLE***

PROGRAMAS	ETAPAS DE DESARROLLO AVANCES	%	TIEMPO EMPLEADO	OBSERVACIONES
IDIOMAS	INGLES INTENSIVO COMPRENSION MODULO 1	50	3.08	
IDIOMAS	INGLES INTENSIVO COMPRENSION MODULO 2	100	2.24	
IDIOMAS	FRANCÉS COMPRENSIÓN MODULO I	100	1.05	
COMPUTACIÓN	LENGUAJE C++	70	3.03	
COMPUTACIÓN	INTRODUCCIÓN AL LENGUAJE C++	60	1.95	
COMPUTACIÓN	INTRODUCCIÓN A LA COMPUTACIÓN	55	0.99	
COMPUTACIÓN	WINDOWS 95	40	6.05	
CARRERA	TALLER DE ELECTRONICA ANALÓGICA	10	2.80	
MAESTRÍA	MAESTRÍA EN ECONOMÍA Y FINANZAS	100	5.02	
MAESTRÍA	CONSTRUCCIÓN DE BDMS RELACIONALES	100	6.02	
MAESTRÍA	MAESTRÍA EN MICROELECTRÓNICA	10	8.22	

Figura 2.7. Reporte a la unidad de planeación.

b) En el caso de cursos externos :

- 1) Estos son solicitados directamente por alguna institución o empresa al DEC Aragón para que sean impartidos en las instalaciones de los primeros.
- 2) Se llega a un acuerdo sobre características y costos del curso solicitado.
- 3) Se asigna un profesor.
- 4) Se lleva un control de avance de curso.
- 5) Finalmente se realizan los mismos reportes requeridos para el caso de cursos internos.

En resumen, podemos citar los siguientes puntos sobre las actividades de el DEC Aragón :

- 1) Es un organismo que tiene como objetivo la impartición de diversos tipos de cursos tanto internos como externos.
- 2) Gracias a su demanda se imparte un número considerable de cursos.
- 3) Se lleva un registro de cursos, profesores y alumnos en archivos de oficina, lo cual dificulta la búsqueda y consulta de contenidos de los archivos.
- 4) Se realizan diferentes formatos de manera manual y muchos de ellos requieren datos existentes en los archivos de oficina. Además se maneja información recurrente en diferentes formatos, esto es, existen formatos cuyos datos contenidos también forman parte de los contenidos de otros formatos.
- 5) Se elaboran diversos reportes cuyos cálculos se realizan de manera manual, pero a su vez , estos tienen la característica de poder ser automatizados.

Finalmente describimos los elementos que intervienen en los procesos realizados por el DEC Aragón y la relación que existe entre ellos, mediante un diagrama de flujo de datos⁹ (ver figuras 2.8 a la 2.14).

⁹ Un diagrama de flujo de datos es un método gráfico que representa a los elementos de un sistema y la relación que existe entre ellos, poniendo especial atención en la transformación que sufren los datos a través de las actividades del sistema. Los procesos que transforman los datos de entrada quedan identificados por medio de un círculo o elipse, el flujo de datos mediante flechas, el almacenamiento de información mediante un rectángulo sin un lado y los elementos origen y/o destino de la información transformada mediante un rectángulo. A diferencia de un "diagrama de flujo", el diagrama de flujo de datos no define la secuencia detallada de los procesos y las actividades llevadas a cabo por el sistema, ni tampoco define aspectos de control, solo le interesa la transformación de los datos al pasar estos por las actividades del sistema.

DIAGRAMA 1
PROCESOS DE ELABORACIÓN, PROMOCIÓN, INSCRIPCIÓN Y MONITOREO DE CURSOS

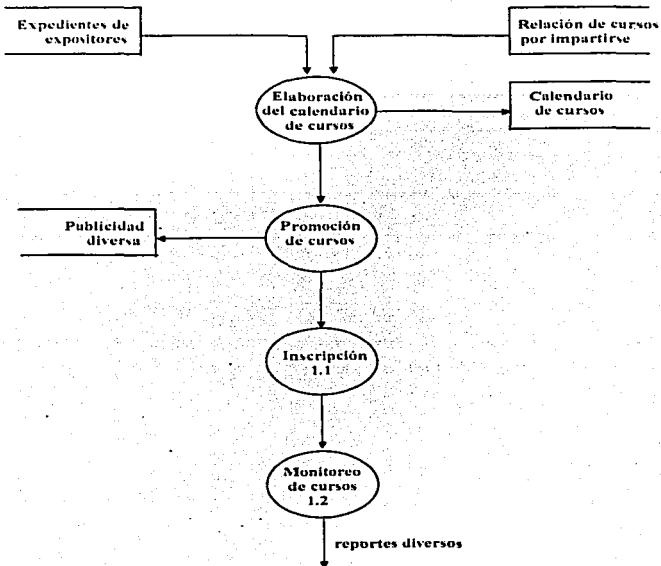


Figura 2.8.

DIAGRAMA 1.1
PROCESO INSCRIPCIÓN

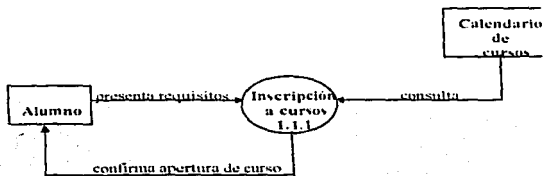


Figura 2.9.

DIAGRAMA 1.1.1
PROCESO DE INSCRIPCIÓN A CURSOS

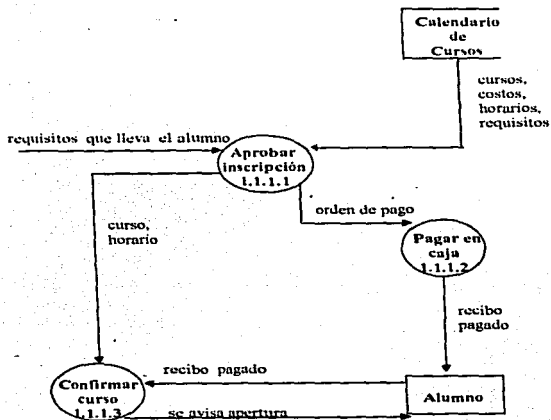


Figura 2.10.

**DIAGRAMA 1.1.1.1
PROCESO DE APROBAR INSCRIPCIÓN**

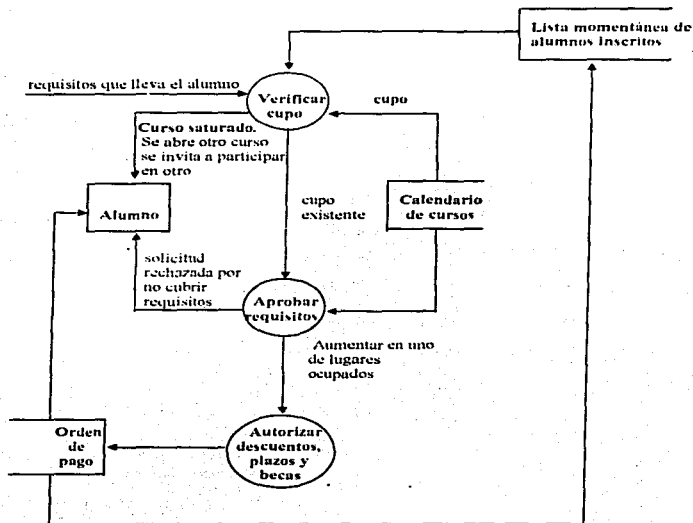


Figura 2.11.

**DIAGRAMA 1.1.1.2
PROCESO PAGAR EN CAJA**

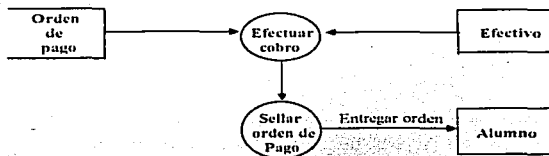


Figura 2.12.

**DIAGRAMA 1.1.1.3
PROCESO CONFIRMAR CURSO**

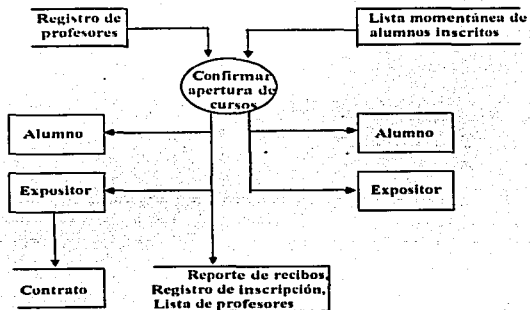


Figura 2.13.

DIAGRAMA 1.2
PROCESO DE MONITOREO DE CURSOS

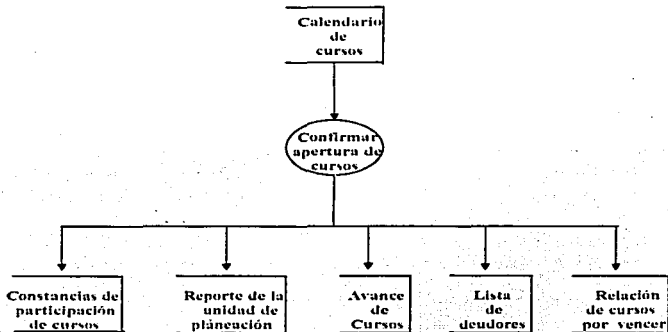


Figura 2.14.

CAPÍTULO 3.

ALTERNATIVAS DE SOLUCIÓN.

Como se pudo apreciar en el capítulo anterior, la problemática principal del DEC Aragón, es que existe dificultad para generar los diferentes reportes y formatos requeridos, ya que la mayor parte de los datos contenidos en ellos son redundantes y además se encuentran almacenados en archivos convencionales de oficina que de algún modo dificultan la consulta y computo, consumiendo tiempo del personal que ahí labora; mermando la funcionalidad del departamento y por ende indirectamente su economía.

En el presente capítulo, mencionaremos las posibles soluciones a la problemática del DEC Aragón, se determinará la más óptima de estas y se diseñará su respectivo modelo.

3.1. POSIBLES SOLUCIONES

Quizás estemos tentados a dar como primer solución, la adición de más personal que auxilie al departamento en sus actividades. Pero si hacemos esto, requeriríamos de un mayor presupuesto para pago de salarios, un mayor espacio en el departamento y la asignación de funciones y puestos para el personal nuevo.

Suponiendo en el peor de los casos que la propuesta anterior fuera aprobada, estaríamos descuidando el aspecto de que se estima que la demanda de los cursos aumente y no es posible darnos el lujo de contratar cada vez más personal.

Como vemos la propuesta de agregar más personal, lejos de ayudar complica más la situación y no resuelve en absoluto el problema de la redundancia de los datos, ni tampoco el de la economía del departamento.

Debido a que el principal problema son los datos y la manipulación de los mismos, podríamos dar como segunda solución un sistema basado en computadoras el cual contenga bases de datos y un programa que las controle. Esta solución optimiza cualquier recurso ya sea financiero, técnico, humano y el recurso no renovable tiempo. Al mismo tiempo le permite al DEC Aragón actualizarse e incorporarse al uso moderno de las computadoras para mejorar su administración.

Pero mejor retomemos las observaciones realizadas en el capítulo anterior para poder justificar la creación de dicho sistema basado en computadoras.

Para solucionar las observaciones 1, 2 y 6 que tratan sobre el curriculum vitae de los profesores, el registro de participación de profesores, el almacenamiento y espacio ocupado por ambos, la redundancia de su información y la dificultad para realizar consultas, se puede realizar un programa y una base de datos donde esta última contenga los datos más relevantes de curriculum vitae de los profesores y de esa forma facilitar la consulta y el llenado de los registros de participación de profesores. También la base de datos contribuiría a no generar duplicación de información y a disminuir el espacio que ocupan los registros de participación de profesores en archivos convencionales de oficina.

Para solucionar la observación 3 que trata sobre la utilización de un calendario anual para hacer el calendario de cursos, sobre el tiempo empleado para hacerlo y sobre el posible error humano al que se puede incurrir, se puede realizar un programa que revise las fechas, procurando que el los días capturados por el usuario sean válidos y estén comprendidos entre la fecha de inicio y la de término de un curso.

Para resolver la observación 4 que trata sobre las consultas continuas que se tienen que realizar al calendario de cursos, se puede realizar una base de datos que contenga toda la información sobre los cursos y las bases de inscripción y también un programa que la controle. De esta manera se puede obtener información inmediata y oportuna sobre los cursos.

La observación 5 que trata sobre los recibos pagados y los curriculum vitae de profesores, refuerza con mayor razón la creación de una base de datos para profesores, pero al mismo tiempo genera la necesidad de tener una base de datos para las órdenes de pago y de un programa que la controle. Con ello la búsqueda de datos y obtención de reportes se facilitaría.

Las observaciones 7 y 8 justifican nuevamente la creación de las bases de datos para profesores, cursos y órdenes de pago.

Con lo que respecta a la observación 9 que hace referencia una vez más a las consultas que se realizan a las órdenes de pago y al calendario de cursos, se podría hacer un programa que genere los diferentes reportes solicitados como los son : registros de inscripción, reportes de recibos, reportes a la unidad de planeación, vencimientos de cursos, listas de profesores, listas de deudores y avance de cursos. Este programa debe de ser capaz de extraer los datos requeridos de cada base y realizar los cálculos necesarios.

Ahora bien, si un alumno pertenece a la institución se le debe de registrar en su orden de pago la carrera a la cual pertenece. Y en el caso de los cursos, estos normalmente apoyan o van enfocados a una carrera, por lo que también es necesario

registrar la carrera a la cual va dirigido el curso. Por ende, sería necesaria una base de datos que almacene el nombre de la carrera.

Finalmente podríamos utilizar la mismas bases de datos y el mismo programa para poder controlar los cursos externos.

Así, para poder llevar a cabo esta solución, es necesario construir cuatro bases de datos : una para cursos, otra para ordenes de pago, otra para expositores y otra para las carreras existentes en la institución.

3.2. ESTUDIO DE LOS DIFERENTES FORMATOS

Como podemos apreciar, la segunda solución presentada en el apartado anterior es la más óptima. Pasemos ahora a observar detenidamente los reportes requeridos para poder determinar los campos de cada una de las bases de datos.

Examinando los datos empleados para realizar el formato de la figura 2.2, podemos clasificar la pertenencia de su información en las bases de datos como se ilustra en la figura 3.1. Lo mismo se hace con las figuras 2.3 a 2.7 (ver pertenencia de las mismas en las figuras 3.2 a 3.6 respectivamente).

DATOS DE LA ORDEN DE PAGO (FIGURA 2.2)	BASE DE DATOS ASOCIADA
Nombre del alumno	Ordenes de pago
Carrera del alumno	Carreras
Semestre	Ordenes de pago
Número de cuenta	Ordenes de pago
Teléfono del alumno	Ordenes de pago
Procedencia del alumno	Ordenes de pago
Nombre del curso	Cursos
Periodo de inicio del curso	Cursos
Periodo de termino del curso	Cursos
Nombre del expositor	Expositores
Importe del recibo pagado	Ordenes de pago
Número de recibo oficial	Ordenes de pago
Fecha de pago en caja	Ordenes de pago

Figura 3.1. Clasificación de los datos de la orden de pago en su correspondiente base de datos.

DATOS DEL REGISTRO DE INSCRIPCIÓN (FIGURA 2.3)	BASE DE DATOS ASOCIADA
Nombre del curso	Cursos
Nombre del expositor	Expositores
Salón	Cursos
Periodo de inicio del curso	Cursos
Periodo de termino del curso	Cursos
Duración en horas del curso	Cursos
Costo para universitarios	Cursos
Costo para egresados	Cursos
Costo para el público en general	Cursos
Nombre del alumno	Ordenes de pago
Teléfono del alumno	Ordenes de pago
Número de cuenta	Ordenes de pago
Carrera del alumno	Carreras
Semestre	Ordenes de pago
Fecha de entrega del recibo en el DEC	Ordenes de pago
Número de recibo oficial	Ordenes de pago
Importe del recibo pagado	Ordenes de pago
Observaciones	Ordenes de pago

Figura 3.2. Clasificación de los datos del registro de inscripción en su correspondiente base de datos.

DATOS DEL REGISTRO DE PROFESORES (FIGURA 2.4)	BASE DE DATOS ASOCIADA
Nombre del expositor	Expositores
Registro federal de contribuyentes del expositor	Expositores
Domicilio del expositor	Expositores
Teléfono del expositor	Expositores
Título profesional del expositor	Expositores
Número de cédula profesional del expositor	Expositores
Fecha de expedición de cédula profesional	Expositores

Figura 3.3. Clasificación de los datos del registro de profesores en su correspondiente base de datos.

DATOS DE LA LISTA DE ASISTENCIA (FIGURA 2.5)	BASE DE DATOS ASOCIADA
Nombre del curso	Cursos
Nombre del expositor	Expositores
Salón	Cursos
Periodo de inicio del curso	Cursos
Periodo de termino del curso	Cursos
Duración en horas del curso	Cursos
Nombre del alumno	Ordenes de pago

Figura 3.4. Clasificación de los datos de la lista de asistencia en su correspondiente base de datos.

DATOS DE LA RELACIÓN DE RECIBOS (FIGURA 2.6)	BASE DE DATOS ASOCIADA
Número de oficio	Ninguna, ya que se captura
Nombre a quien se dirige el oficio	Ninguna, ya que se captura
Cargo o puesto de quien se dirige el oficio	Ninguna, ya que se captura
Total de recibos oficiales	Ninguna, ya que se calcula
Importe total de la suma de los recibos reportados	Ninguna, ya que se calcula
Nombre del expositor	Expositores
Nombre del curso	Cursos
Días de la semana en que el curso se imparte	Cursos
Periodo de inicio del curso	Cursos
Periodo de termino del curso	Cursos
Honorarios del profesor	Ninguna, ya que se captura
Fecha de realización del oficio	Ninguna, ya que se captura
Cargo de quien envía el oficio	Ninguna, ya que se captura
Nombre de quien envía el oficio	Ninguna, ya que se captura
Con copia para jefes de qué departamentos	Ninguna, ya que se captura
Nombre del alumno	Ordenes de pago
Número de recibo oficial	Ordenes de pago
Fecha de entrega del recibo en oficinas del DEC	Ordenes de pago
Importe del recibo pagado	Ordenes de pago
Observaciones	Ninguna, ya que se captura
Total de ingreso de recibos menos honorarios	Ninguna, ya que se calcula

Figura 3.5. Clasificación de los datos de la relación de recibos en su correspondiente base de datos.

DATOS DEL REPORTE PARA LA UNIDAD PLANEACIÓN (FIGURA 2.7)	BASE DE DATOS ASOCIADA
Periodo inicial de avance	Ninguna, ya que se captura
Periodo final de avance	Ninguna, ya que se captura
Nombre del jefe del DEC	Ninguna, ya que se captura
Tipo de programa	Cursos
Etapas de desarrollo avances (nombre del curso)	Ninguna, ya que se calcula
Porcentaje de avance	Ninguna, ya que se calcula
tiempo empleado en meses	Ninguna, ya que se calcula
Observaciones	Ninguna, ya que se captura

Figura 3.6. Clasificación de los datos del reporte para la unidad de planeación en su correspondiente base de datos.

En base a las figuras 3.1, 3.2, 3.4, 3.5 y 3.6 podemos establecer qué elementos forman los campos de las cuatro bases de datos (ver figura 3.7 para la base de datos de cursos, la figura 3.8 para base de datos de ordenes de pago, la figura 3.9 para base de datos de expositores y la figura 3.10 para base de datos de carreras) y cuales serán variables o constantes dentro del programa controlador, que de hecho son aquellos elementos que tiene un valor fijo o pueden ser capturados o calculados y por consiguiente no deben de existir como elementos de la estructura de la base de datos.

CAMPOS OBTENIDOS DE LAS FIGURAS 3.1, 3.2, 3.4, 3.5 Y 3.6 PARA LA BASE DE DATOS DE CURSOS
Nombre del curso
Periodo de inicio del curso
Periodo de termino del curso
Salón
Duración en horas del curso
Costo para universitarios
Costo para egresados
Costo para el público en general
Días de la semana en que el curso se imparte
Tipo de programa

Figura 3.7. Obtención de los campos para base de datos de cursos, a partir de los reportes existentes.

CAMPOS OBTENIDOS DE LAS FIGURAS 3.1, 3.2, 3.4 Y 3.5 PARA LA BASE DE DATOS DE ORDENES DE PAGO
Nombre del alumno
Carrera del alumno
Semestre
Número de cuenta
Teléfono del alumno
Procedencia del alumno
Importe del recibo pagado
Número de recibo oficial
Fecha de pago en caja
Fecha de entrega del recibo en oficinas del DEC
Observaciones

Figura 3.8. Obtención de los campos para base de datos de ordenes de pago, a partir de los reportes existentes.

CAMPOS OBTENIDOS DE LAS FIGURAS 3.1 A LA 3.5 PARA LA BASE DE DATOS DE EXPOSITORES
Nombre del expositor
Registro federal de contribuyentes del expositor
Domicilio del expositor
Teléfono del expositor
Título profesional del expositor
Número de cédula profesional del expositor
Fecha de expedición de cédula profesional

Figura 3.9. Obtención de los campos para base de datos de expositores, a partir de los reportes existentes.

CAMPOS OBTENIDOS DE LAS FIGURAS 3.1 Y 3.2 PARA LA BASE DE DATOS DE CARRERAS
Nombre de la carrera

Figura 3.10. Obtención de los campos para base de datos de carreras a partir de los reportes existentes.

3.3. LAS ESTRUCTURAS DEFINITIVAS DE LAS BASES DE DATOS

Ahora bien, los campos obtenidos no son definitivos, ya que tenemos que depurar algunos y agregar otros. Para ello analicemos los campos obtenidos en las figuras 3.7 a la 3.10.

LA ESTRUCTURA DE LA BASE DE DATOS DE CARRERAS

A la base de datos de carreras es conveniente agregarle una *clave* para la carrera y un campo de *observaciones* de la misma.

La clave de la carrera en esta base permitiría entre otras cosas : a) ahorrar tiempo en la captura, ya que únicamente se teclearía la clave de la carrera en lugar del nombre de la misma; b) facilitaría la búsqueda de la carrera; c) conformaría la posibilidad de ordenar esta base de datos por su campo clave, y d) compartir información con otras bases de datos, ahorrando un espacio considerable de memoria.

La clave sería idéntica a la que utiliza de antemano la institución para designar a una carrera y se compone de tres dígitos.

En el caso del campo de observaciones, este serviría para registrar indicaciones relevante de esta carrera, como lo son cambios en la clave de la carrera o en el nombre de la misma.

En la figura 3.11 se ilustra la estructura o composición de campos definitiva de la base de datos de carreras.

NOMBRE DEL CAMPO	TIPO	LONGITUD	DECIMALES	EXPLICACIÓN
CVEPROFESI	CHARACTER	3	0	CLAVE DE LA CARRERA
NOMBRECARR	CHARACTER	45	0	NOMBRE DE LA CARRERA
OBSERVA	MEMO	10	0	NOTAS SOBRE LA CARRERA

Figura 3.11. Estructura definitiva de la base de datos de carreras.

LA ESTRUCTURA DE LA BASE DE DATOS DE EXPOSITORES

Como en el DEC Aragón, imparten cursos profesores de diversas instituciones, es importante indicar la procedencia de los mismos, de antemano, se lleva un control sobre el nombre de la institución a la cual pertenecen estos. También el DEC Aragón registra observaciones importantes de los curriculum vitae de los expositores. El total de expositores depende de la cantidad de cursos abiertos, pero el promedio actual del departamento es de 100 a 300 cursos impartidos al año. Partiendo del supuesto de que tengamos un expositor por curso, esto implica que tenemos un registro de 300 expositores distintos.

Debido a los motivos presentados en el párrafo anterior, es necesario agregar a la base de datos de expositores tres campos más.

Como en teoría es posible que existan expositores con nombre y apellidos iguales, necesitamos un campo *clave* que permita distinguir y hacer referencia a un expositor. Este campo facilita al mismo tiempo la ordenación y por consiguiente búsqueda de expositores. La longitud del mismo sería de 3 dígitos considerando tener los 300 expositores señalados mas arriba y a la par nos da un margen de tener dado de alta hasta 1000 expositores.

El segundo campo se refiere a la *adscripción* del expositor en el cual se especificaría el instituto u organismo al cual pertenece el expositor.

El tercer campo es el de *observaciones* en el cual se explican notas importantes sobre el expositor.

En la figura 3.12 se ilustra la estructura o composición de campos definitiva de la base de datos de expositores.

NOMBRE DEL CAMPO	TIPO	LONGITUD	DECIMALES	EXPLICACIÓN
CVEEXPO	CHARACTER	3	0	CLAVE DEL EXPOSITOR
NOMBREEXPO	CHARACTER	30	0	NOMBRE DEL EXPOSITOR
RFC	CHARACTER	13	0	REGISTRO FEDERAL DE CONTRIBUYENTES DEL EXPOSITOR
DOMICILIO	CHARACTER	40	0	DOMICILIO DEL EXPOSITOR
TELEFONO	CHARACTER	12	0	TELEFONO DEL EXPOSITOR
ADSCRIPCIO	CHARACTER	40	0	INSTITUTO U ORGANISMO AL CUAL PERTENECE EL EXPOSITOR
TITULO	CHARACTER	40	0	TITULO PROFESIONAL
CEDULA	CHARACTER	10	0	CEDULA PROFESIONAL
EXPEDICION	DATE	8	0	FECHA DE EXPEDICION DE CEDULA PROFESIONAL
OBSERVA	MEMO	10	0	NOTAS SOBRE EL EXPOSITOR

Figura 3.12. Estructura definitiva de la base de datos de expositores.

LA ESTRUCTURA DE LA BASE DE DATOS DE ORDENES DE PAGO

Por lo que se refiere a la base de datos de ordenes de pago, se requiere agregar siete nuevos campos y adecuar los datos carrera, número de recibo oficial, fecha de pago en caja, fecha de entrega de recibo de pago en el DEC y el importe del recibo de pago.

El primero de los campos es el *registro federal* de contribuyentes del alumno.

El segundo es una *clave de procedencia*, que permita seleccionar la procedencia del alumno, por ejemplo si es cero la procedencia es la ENEP Aragón, si es uno la procedencia es alguna otra dependencia de la UNAM y si es dos la procedencia es cualquier otra diferente a las anteriores.

El tercer campo es el de la *clave de la carrera* la cual permite seleccionar el número de la carrera a la cual pertenece el alumno y de esta manera el usuario no capture el nombre total de la carrera.

El cuarto campo es la *clave de la condición* del alumno, la cual determina de forma automática el costo de un curso para el interesado.

El quinto campo tiene mucho que ver con el cuarto y es el *costo del curso* que tiene o consigue el participante. Este campo debe ser lo suficientemente flexible para almacenar un número ilimitado de recibos por participante y también como para poder registrar el importe de los recibos, las fechas de pago en caja y las fechas de entrega de los mismos.

El sexto campo es la *clave del curso* al cual desea apuntarse el interesado. Con esto ya no es necesario capturar nuevamente el nombre del curso, ya que la clave del mismo es idéntica a la empleada en la base de datos de cursos. Con ella podemos tener acceso a todos los datos relacionados en el curso con igual clave. Esto nos permite ahorrar memoria debido a que se comparten datos de la base de datos de cursos con la de ordenes y de algún modo tener una mejor clasificación de la información, la cual permite búsquedas más rápidas, listados ordenados ascendentemente, etc.

El séptimo campo es el *importe del curso* para ese interesado. En teoría este campo no debería de existir, ya que un curso tiene asociado un precio dada la condición del alumno. Pero debido a que se manejan descuentos, debemos crear este campo para almacenar en él, el importe real total a pagar por el alumno sobre el curso.

En la figura 3.13 se ilustra la estructura o composición de campos definitiva de la base de datos de ordenes de pago.

NOMBRE DEL CAMPO	TIPO	LONGITUD	DECIMALES	EXPLICACIÓN
CVECURSO	CHARACTER	10	0	CLAVE DEL CURSO
NOMBREALUM	CHARACTER	40	0	NOMBRE DEL ALUMNO
TELEFONO	NUMERIC	12	0	TELEFONO DEL ALUMNO
RFC	CHARACTER	13	0	REGISTRO FEDERAL DE CONTRIBUYENTES DEL ALUMNO
CVEPROCEDE	CHARACTER	1	0	CLAVE DE LA PROCEDENCIA
PROCEDENCI	CHARACTER	40	0	PROCEDENCIA DEL ALUMNO
CVECARRERA	CHARACTER	3	0	CLAVE DE LA CARRERA DEL ALUMNO
CUENTA	CHARACTER	8	0	NUMERO DE CUENTA
CVECONDICI	CHARACTER	1	0	CLAVE DE LA CONDICIÓN
SEMESTRE	NUMERIC	2	0	SEMESTRE DEL ALUMNO
COSTOCUR	NUMERIC	8	2	COSTO DEL CURSO
RECIBOS	MEMO	10	0	PAGO DE RECIBOS
OBSERVA	MEMO	10	0	NOTAS SOBRE EL ALUMNO

Figura 3.13. Estructura definitiva de la base de datos de ordenes de pago.

LA ESTRUCTURA DE LA BASE DE DATOS DE CURSOS

En el caso de la base de datos de cursos, se tienen que agregar once nuevos campos.

El primero es el *lugar de impartición* en donde es llevado a cabo el curso.

El segundo es el *cupo* o capacidad de alumnos que puede albergar el curso.

El tercero es el *horario de inicio* con la cual comienza el curso. A su vez, por cuestiones de validación de entrada de datos, podemos dividir este campo en dos : *hora de inicio* y *minuto de inicio*.

El cuarto es el *horario de termino* con el cual finaliza el curso. De igual modo, podemos dividir este campo en dos : *hora de termino* y *minuto de termino*.

Debido a que un curso es posible que apoye a cierta carrera, es conveniente señalar a qué carrera auxilia. Por consiguiente el quinto campo es el de la *clave de carrera* que el curso apoya.

Como los costos no sólo se centralizan a los alumnos, egresados y público en general, sino de igual forma se consideran *costos* para *académicos*, para *personal administrativo* y para *personal de mantenimiento*, será necesario añadirlos, por lo que tenemos nuestros sexto, séptimo y octavo campo.

Debido a que es conveniente tener un campo en el cual se puedan realizar diversas anotaciones sobre los cursos, nuestro noveno campo es el de *observaciones*.

También se necesita agregar el campo de la *clave del profesor* asignado al curso, que es nuestro décimo campo.

Finalmente al igual que las otras bases de datos, es conveniente agregar el campo *clave del curso* para poder asociarla con los datos de un curso y facilitar la captura, búsqueda, ordenación y compartición de datos entre las bases de datos. Esta clave se compone de diez dígitos.

El primer dígito se refiere al tipo de curso, cuyo rango de valores permitidos es de 0 a 5. A continuación listamos los valores permitidos para el primer dígito :

0	=	Curso de "Varios".
1	=	Curso de "Idiomas".
2	=	Curso de "Computación".
3	=	Curso auxilia a una "Carrera".
4	=	Curso de "Maestría".
5	=	Curso de "Diplomado".

El segundo dígito se refiere a el subtipo de curso, cuyo rango de valores permitidos es de 0 a 2 donde :

0 = Curso "Normal".
 1 = Curso "Juvenil".
 2 = Curso de "Verano".

El tercer dígito se refiere a la modalidad del curso, teniendo como valores permitidos de 0 a 1 donde :

0 = Curso "Externo".
 1 = Curso "Interno".

El cuarto y quinto dígito se refieren a una clave interna de desempate para garantizar que una clave de curso no se repita.

El sexto y séptimo dígito se refieren al grupo del curso en cuestión, ya que se puede impartir un mismo curso en diferentes horarios.

NOMBRE DEL CAMPO	TIPO	LONGITUD	DECIMALES	EXPLICACIÓN
CVECURSO	CHARACTER	10	0	CLAVE DEL CURSO
NOMBRECUR	CHARACTER	40	0	NOMBRE DEL CURSO
LUGAR	CHARACTER	40	0	LUGAR DE EXPOSICION
CVEEXPO	CHARACTER	3	0	CLAVE DEL EXPOSITOR
FECHAENPO1	DATE	8	0	PERIODO DE INICIO
FECHAENPO2	DATE	8	0	PERIODO DE TERMINO
DURACION	NUMERIC	3	0	DURACION EN HORAS
HORARIO1_1	CHARACTER	2	0	HORA DE INICIO
HORARIO1_2	CHARACTER	2	0	MINUTOS DE INICIO
HORARIO2_1	CHARACTER	2	0	HORA DE TERMINO
HORARIO2_2	CHARACTER	2	0	MINUTOS DE TERMINO
SALON	CHARACTER	10	0	AULA DE IMPARTICION
COSTOALUM	NUMERIC	8	2	COSTO PARA ALUMINOS
COSTOACAD	NUMERIC	8	2	COSTO PARA ACADEMICOS
COSTOADMII	NUMERIC	8	2	COSTO PARA ADMINISTRATIVOS
COSTOMANU	NUMERIC	8	2	COSTO PARA EL PERSONAL DE MANTENIMIENTO
COSTOEGRE	NUMERIC	8	2	COSTO PARA EGRESADOS
COSTOPUBL	NUMERIC	8	2	COSTO PARA EL PUBLICO
OBSERVA	MEMO	10	0	NOTAS SOBRE EL CURSO
CUPO	NUMERIC	3	0	TOTAL DE ALUMINOS POR CURSO
DIAS	CHARACTER	7	0	DÍAS DE EXPOSICION
CVECAR	CHARACTER	3	0	CARRERA QUE APOYA EL CURSO

Figura 3.14. Estructura definitiva de la base de datos de cursos.

El octavo y noveno dígito se refieren al módulo o grado del curso, el cual siempre es mayor que cero.

El décimo dígito se refiere al nivel del curso y sólo se captura cuando el curso es un idioma, esto es de tipo 2. Los rangos permitidos son de 0 a 2, si es 0 entonces el curso no es un idioma, si es 1 el curso es un idioma y su nivel es comprensión y si es 2 el curso es un idioma y su nivel es posesión.

En la figura 3.14 se ilustra la estructura o composición de campos definitiva de la base de datos de cursos.

3.4. RELACIÓN EXISTENTE ENTRE LAS BASES DE DATOS

Como los distintos formatos aparecen ordenados de manera ascendente, podemos crear archivos indexados⁹ por los campos CVECURSO para las bases de datos de cursos y ordenes de pago y CVEEXPO para las bases de datos de ordenes de pago y expositores.

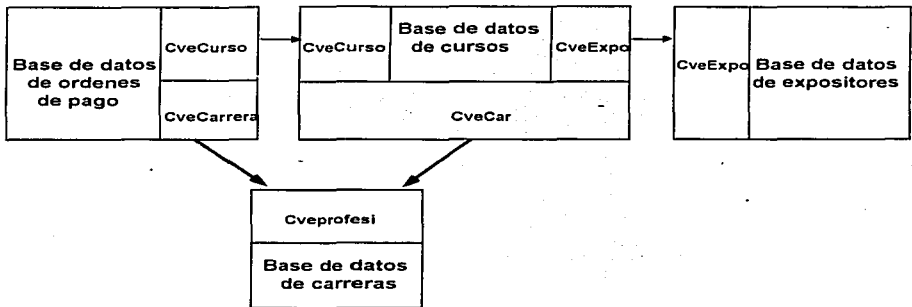


Figura 3.15. Relación existente entre las bases de datos para compartir información entre ellas.

⁹ La indexación o indización, es un método de ordenación que crea un archivo en el cual se almacena el número de los registros de una base de datos pero ya ordenados. Así, cuando requerimos que aparezca información ordenada, se consulta el orden en que esta debe de aparecer en el archivo indexado. Las ventajas principales de la indexación son que no duplican la información, si se agregan o se borran registros el archivo indexado se actualiza automáticamente garantizando mantener siempre el orden y por último, permite relacionar bases de datos lo cual permite compartir información. Como vemos no permite ahorrarnos muchos métodos que engrosarían el código fuente de un programa.

Gracias a la indexación conseguimos ordenar la información, facilitar las consultas y lo más importante, podemos compartir datos entre las bases de datos. A esto último se le llama relacionar bases de datos.

La relación existente entre las cuatro bases de datos se ilustra en la figura 3.15. Por ejemplo, la base de datos de cursos puede acceder información de la base de datos de ordenes de pago y también información de la base de datos de expositores debido a que esta está relacionada con la de ordenes de pago.

3.5. MÓDULOS DEL PROGRAMA PROPUESTO

Un programa controlador de bases de datos comúnmente está compuesto por un módulo de altas, uno de bajas, uno de consultas, uno de modificaciones y uno de reportes. Nosotros también crearemos estos módulos y agregaremos uno adicional para poder configurar el programa y el equipo con que se cuenta. A este módulo le llamaremos módulo de disposición. No obstante, haremos los módulos mencionados de acuerdo a las necesidades del DEC Aragón, a manera de cubrir los objetivos que perseguimos en el presente trabajo.

En la figura 3.16 se muestra el diagrama de módulos requeridos para realizar el programa. Para facilitar su comprensión, dividiremos y explicaremos el diagrama por secciones.

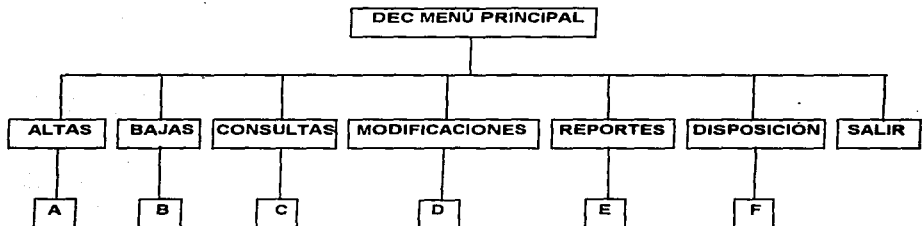


Figura 3.16. Los módulos del programa principal.

Empleando la técnica de pseudocódigo¹⁰ describiremos de manera general la composición de los módulos del menú principal.

DO WHILE la opción seleccionada por el usuario sea diferente de la opción salir

Permitir al usuario seleccionar una opción

DO CASE opción

CASE Opción = Altas

Habilitar módulo general de altas

CASE Opción = Bajas

Habilitar módulo general de bajas

CASE Opción = Consultas

Habilitar módulo general de consultas

CASE Opción = Modificaciones

Habilitar módulo general de modificaciones

CASE Opción = Reportes

Habilitar módulo general de reportes

CASE Opción = Disposición

Habilitar módulo general de disposición

ENDCASE

ENDDO

¹⁰ Un pseudocódigo es una técnica que describe los procesos existentes en un sistema por medio de lengua escrita.

EL MÓDULO DE ALTAS

Un módulo de altas tiene como fin agregar nuevos registros a una base de datos. Como tenemos cuatro bases de datos, tenemos que hacer un módulo de altas para cada una (ver figura 3.17).

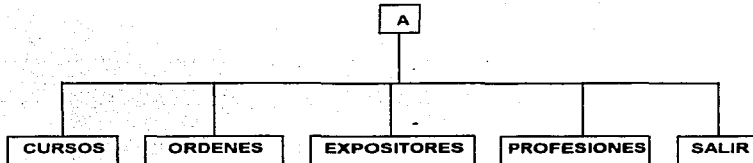


Figura 3.17. Los módulos del programa de altas.

A continuación presentamos el pseudocódigo general de la opción de altas del menú principal.

DO WHILE el usuario desee dar de alta

 Permitir la entrada de los datos del registro

 IF no existe error en los datos capturados

 Anexar el nuevo registro a la base de datos

 ELSE

 Enviar mensaje de error en la captura de datos del registro

 ENDIF

ENDDO

EL MÓDULO DE BAJAS

Un módulo de bajas tiene como objetivo eliminar registros existentes de una base de datos. Como tenemos cuatro bases de datos, tenemos que hacer un módulo de bajas para cada una (ver figura 3.18).

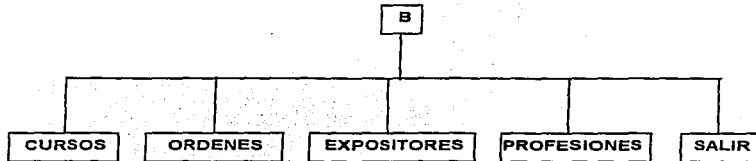


Figura 3.18. Los módulos del programa de bajas.

A continuación presentamos el pseudocódigo general de la opción de bajas del menú principal.

DO WHILE el usuario desee dar de baja

 Solicitar la clave del registro a dar de baja

IF la clave existe

 Borrar el registro de la base de datos

ELSE

 Enviar mensaje de error por inexistencia de clave

ENDIF

ENDDO

EL MÓDULO DE CONSULTAS

Un módulo de consultas tiene como fin visualizar el contenido de los registros requeridos de una base de datos.

Para agilizar las consultas y darle flexibilidad al usuario, dividiremos el módulo de consultas en tres más que son : a) consultas globales en donde se exhibe la relación total registros con el contenido de sus campos; b) consultas por clave en donde se solicita la clave del registro a consultar, y c) consultas por palabra incluida en donde se selecciona la base de datos y el campo a consultar, se captura la cadena de caracteres que se desea hallar en dicho campo y se exhiben los registros que contengan incluida esa cadena de caracteres (ver figura 3.19).

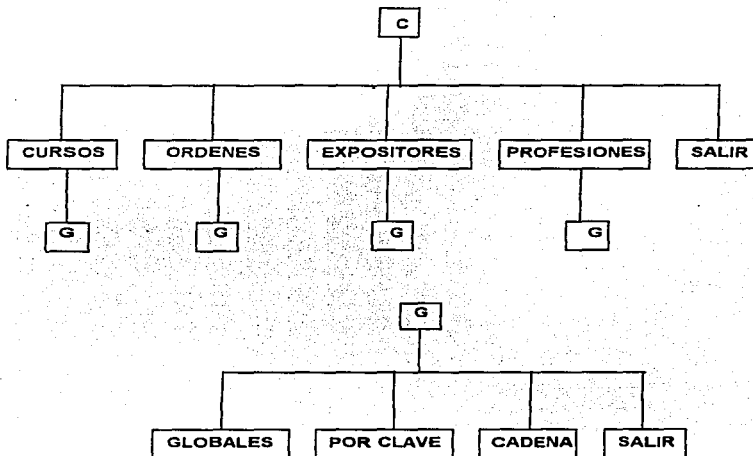


Figura 3.19. Los módulos del programa de consultas.

A continuación presentamos el pseudocódigo general de la opción de consultas del menú principal.

DO WHILE el usuario requiera consultas

Solicitar el tipo de consulta a realizar

DO CASE tipo de consulta

CASE tipo de consulta = Globales

DO WHILE el usuario requiera consultas globales

Exhibir el total de registros de la base de datos

Permitir seleccionar el registro deseado

Exhibir los datos del registro requerido

ENDDO

CASE tipo de consulta = Por clave

DO WHILE el usuario requiera consultas por clave

Solicitar la clave del registro deseado

IF la clave existe

Exhibir los datos del registro requerido

ELSE

Enviar mensaje de error por inexistencia de clave

ENDIF

ENDDO

CASE tipo de consulta = Cadena incluida

DO WHILE el usuario requiera consultas por cadena incluida

Permitir seleccionar el campo deseado

Permitir capturar la clave del registro seleccionado

Exhibir los registros que tengan la cadena incluida en el campo seleccionado

Permitir seleccionar el registro deseado

Exhibir los datos del registro requerido

ENDDO

ENDCASE

ENDDO

EL MÓDULO DE MODIFICACIONES

Un módulo de modificaciones tiene como objetivo cambiar algunos de los datos de los registros requeridos, existentes en una base de datos. Como tenemos cuatro bases de datos, tenemos que hacer un módulo de modificaciones para cada una de las bases de datos (ver figura 3.20).

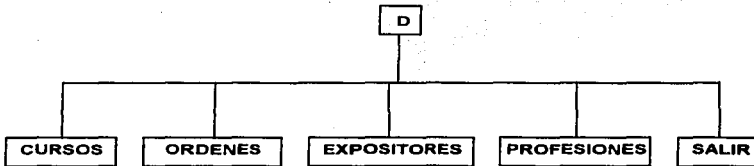


Figura 3.20. Los módulos del programa de modificaciones.

A continuación presentamos el pseudocódigo general de la opción de modificaciones del menú principal.

DO WHILE el usuario desee modificar registros

 Solicitar la clave del registro a modificar

 IF clave del registro existe

 Permitir modificación de los datos requeridos en el registro

 IF no existe error en los datos modificados

 Actualizar los datos del registro

 ELSE

 Enviar mensaje de error en la captura de datos del registro

 ENDIF

ELSE

 Enviar mensaje de error por inexistencia de clave

ENDIF

ENDDO

EL MÓDULO DE REPORTES

Quizás el módulo de reportes es el más importante de todos, ya que es el encargado de automatizar las salidas requeridas por el DEC Aragón.

Básicamente se requiere de seis reportes, a saber :

a) Porcentaje de avance de cursos. Este reporte se encargará de mostrar en pantalla el avance en porcentaje que un curso lleva hasta el momento de su consulta. También indicará el número de días que se han expuesto y los que faltan por exponer.

b) Cursos por vencer. Listará en pantalla y/o impresora y/o archivo, la relación de cursos que estén apunto de vencer según el número de días indicado por el usuario.

c) Lista de deudores. Exhibirá en pantalla y/o impresora y/o archivo, la relación de participantes que tengan débitos o saldos pendientes por liquidar para cubrir el total del costo del curso tomado.

d) Registro de inscripción y reporte de recibos. Generará en pantalla y/o impresora y/o archivo el registro de inscripción y el reporte de recibos pagados, según el número de oficio solicitado.

e) Listas de profesores. Creará la lista de inscritos a un curso seleccionado en pantalla, y/o impresora y/o archivo.

f) Informe a la unidad de planeación. Generará en pantalla y/o impresora y/o archivo el informe a la unidad de planeación en el rango de fechas requeridas por el usuario.

En la figura 3.21 se ilustra el módulo de reportes.

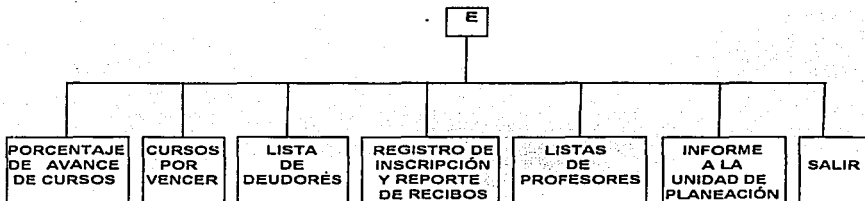


Figura 3.21. Los módulos del programa de reportes.

A continuación presentamos el pseudocódigo general de la opción de reportes del menú principal.

DO WHILE el usuario requiera reportes

 Solicitar el tipo de reporte a realizar

 DO CASE tipo de reporte

 CASE tipo de reporte = Porcentaje de avance de cursos

DO WHILE el usuario requiera porcentaje de avance de cursos

Solicitar la clave del curso

IF clave existe

Exhibir los datos del curso seleccionado, el porcentaje de avance de cursos, los días expuestos y por exponer

ELSE

Enviar mensaje de error por inexistencia de clave

ENDIF

ENDDO

CASE tipo de reporte = Lista de deudores

Exhibir la lista de deudores

CASE tipo de reporte = Registro de inscripción y reporte de recibos

DO WHILE el usuario requiera registros de inscripción

Solicitar clave del curso

IF clave existe

Presentar relación de oficios existentes

Solicitar el número de oficio

Exhibir Registro de inscripción y reporte de recibos correspondiente

ELSE

Enviar mensaje de error por inexistencia de clave

ENDIF

ENDDO

CASE tipo de reporte = Lista de profesores

DO WHILE el usuario requiera lista de profesores

Solicitar clave del curso

IF clave existe

Generar lista de asistencia

ELSE

Enviar mensaje de error por inexistencia de clave

ENDIF

ENDDO

CASE tipo de reporte = Informe a la unidad de planeación

Solicitar el periodo del cual se desea obtener el informe

Generar el informe a la unidad de planeación

ENDCASE

ENDDO

EL MÓDULO DE DISPOSICIÓN

El módulo de disposición tiene como finalidad cambiar y almacenar la configuración de la unidad de disco en donde se encuentran las bases de datos; el tipo de impresora, las partes variables de los reportes (encabezados, nombres, cargos, etc.), y la clave de acceso a los módulos de altas, bajas y modificaciones.

En la figura 3.22 se muestra el módulo de disposición.

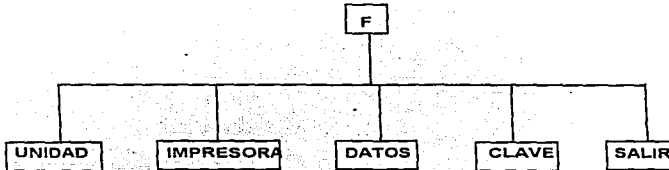


Figura 3.22. Los módulos del programa de disposición.

A continuación presentamos el pseudocódigo general de la opción de disposición del menú principal.

DO WHILE el usuario requiera cambiar los parámetros de configuración

 Solicitar el tópico de configuración a modificar

 DO CASE tópico de configuración

 CASE tópico de configuración = Unidad

 Solicitar la unidad y/o ruta en donde residen las bases de datos

 IF la unidad existe

 Cargar las bases de datos de la unidad y/o ruta especificada

 ELSE

 Enviar mensaje de error en la unidad

 ENDIF

 CASE tópico de configuración = Impresora

 Seleccionar el tipo de impresora que se desea

 Almacenar en el archivo de configuración el tipo de impresora

 CASE tópico de configuración = Datos

Permitir modificar los datos que manejan los reportes por defecto

Almacenar en el archivo de configuración los nuevos datos por defecto

CASE tópicos de configuración = Clave

Permitir modificar la clave de acceso

Almacenar en el archivo de configuración la nueva clave de acceso

ENDCASE

ENDDO

3.6. SELECCIÓN DEL MANEJADOR DE BASES DE DATOS Y DEL LENGUAJE DE PROGRAMACIÓN

Finalmente, ya que hemos encontrado y comprendido los módulos por realizar, falta por definir el manejador de las bases de datos con lenguaje de programación incluido desde el cual se elaboraremos el programa y las bases de datos.

Debido a que la mayor parte de los programas y rutinas ya vigentes en la UNAM Campus Aragón están en CLIPPER y que también a futuro se pretende enlazar estos programas a través de un sistema de redes, podemos optar por realizarlo en este lenguaje por cuestiones de compatibilidad. Aunque cabe señalar que se puede hacer en cualquier otro manejador de bases de datos con lenguaje de programación incluido, o incluso en un lenguaje de alto nivel.

CAPÍTULO 4

PROGRAMA PROPUESTO : EL "DEC" VERSIÓN 1.0

A continuación presentamos el listado completo del programa propuesto : el DEC ver. 1.0., que permitirá agilizar las actividades llevadas a cabo en el DEC Aragón.

```
/*  
* PROGRAMAMA : DEC.PRG  
* AUTOR : CESAR MANDUJANO LUNA  
* FECHA : 10/04/1996  
* COMPILACION : SIN PARAMETROS  
*/
```

```
/*  
* EXPLICACION :  
*  
* Este programa permite realizar la administración de los cursos  
* impartidos por el Departamento de Educación continua de la UNAM  
* Campus Aragón.  
*  
* El programa cuenta con un sistema de ayuda el cual facilita su  
* utilización ( ver funciones AYUDA() y AYUDAMOD() para mayor referencia ).  
*  
* NOTA : todas las funciones utilizadas en este programa son globales.  
*  
*/
```

```
-----  
* Este es el módulo principal *  
-----
```

Principal()

/*

* Principal() --> NIL

*

* Esta función es la responsable de llamar a los demás módulos.

*

*/

FUNCTION Principal

// Las constantes MARCO1, MARCO2, MARCO3, MARCO4, MARCO5, MARCO6 y
// MARCO7 contienen caracteres que permiten dibujar cajas.

#DEFINE Marco1 CHR(201)+CHR(205)+CHR(187)+CHR(186)+;
CHR(188)+CHR(205)+CHR(200)+CHR(186)

#DEFINE Marco2 CHR(218)+CHR(196)+CHR(191)+CHR(179)+;
CHR(217)+CHR(196)+CHR(192)+CHR(179)

#DEFINE Marco3 CHR(214)+CHR(196)+CHR(183)+CHR(186)+;
CHR(189)+CHR(196)+CHR(211)+CHR(186)

#DEFINE Marco4 CHR(213)+CHR(205)+CHR(184)+CHR(179)+;
CHR(190)+CHR(205)+CHR(212)+CHR(179)

#DEFINE Marco5 CHR(218)+CHR(196)+CHR(183)+CHR(186)+;
CHR(188)+CHR(205)+CHR(212)+CHR(179)

#DEFINE Marco6 CHR(220)+CHR(220)+CHR(220)+CHR(222)+;
CHR(223)+CHR(223)+CHR(223)+CHR(221)

#DEFINE Marco7 CHR(220)+CHR(220)+CHR(220)+CHR(219)+;
CHR(223)+CHR(223)+CHR(223)+CHR(219)

// Las constantes RAMAV1, RAMAV2, RAMAV3, RAMAV4, RAMAV5, RAMAV6,
// RAMAH1, RAMAH2, RAMAH3, RAMAH4, RAMAH5 y RAMAH6 tiene asignado un
// conjunto de caracteres que permiten la realización de líneas
// verticales u horizontales.

#DEFINE RamaV1 CHR(203)+CHR(186)+CHR(202)

#DEFINE RamaV2 CHR(194)+CHR(179)+CHR(193)

#DEFINE RamaV3 CHR(210)+CHR(186)+CHR(208)

```
#DEFINE RamaV4 CHR(209)+CHR(179)+CHR(207)
#DEFINE RamaV5 CHR(220)+CHR(219)+CHR(223)
#DEFINE RamaV6 CHR(223)+CHR(219)+CHR(220)
#DEFINE RamaH1 CHR(204)+CHR(205)+CHR(185)
#DEFINE RamaH2 CHR(195)+CHR(196)+CHR(180)
#DEFINE RamaH3 CHR(199)+CHR(196)+CHR(182)
#DEFINE RamaH4 CHR(198)+CHR(205)+CHR(181)
#DEFINE RamaH5 CHR(219)+CHR(223)+CHR(219)
#DEFINE RamaH6 CHR(219)+CHR(220)+CHR(219)

// La constante SALTO permite un salto de línea
#DEFINE Salto CHR(13)+CHR(10) // avance de carro + avance de línea

// La constante FLECHAV almacena los caracteres empleados para dibujar
// una barra vertical
#DEFINE FlechaV CHR(30)+CHR(176)+CHR(31)

// La constante FLECHAH almacena los caracteres empleados para dibujar
// una barra horizontal
#DEFINE FlechaH CHR(17)+CHR(176)+CHR(16)

// La constante HOY retiene la fecha del sistema
#DEFINE Hoy DATE()

// La variable ExplicaPro almacena la explicación del uso del programa
PRIVATE ExplicaPro := Salto+Salto+;
    Este programa permite administrar los cursos '+';
```



```
'impartidos por el Departamento de Educación Continua '+';
'de la UNAM Campus Aragón. En cualquier momento en caso '+';
'de que el usuario tenga alguna duda, puede presionar '+';
'las teclas < F1 > para solicitar ayuda, o bien '+';
'< ESC > para salir de la pantalla correspondiente. '+';
Salto+Salto+;
' Todos los módulos fueron programados en Clipper '+';
'Ver. 5.02 y básicamente el programa controla '+';
'altas, bajas , modificaciones, consultas y/o reportes '+';
'de cuatro bases de datos y un archivo de variables, '+';
'los cuales son respectivamente : '+';
Salto+Salto+;
' a) CURSOS.DBF que contiene toda la información '+';
'relacionada con los diversos cursos impartidos.'+';
Salto+Salto+;
' b) ORDEN.DBF que contiene los datos de los '+';
'inscritos a los cursos.'+';
Salto+Salto+;
' c) EXPOSITO.DBF que contiene los datos '+';
'relacionados a los profesores que imparten cursos.'+';
Salto+Salto+;
' d) CARRERAS.DBF que contiene la información de las '+';
'carreras o profesiones que ahí se imparten.'+';
Salto+Salto+;
' e) CONFIG.DEC que almacena la configuración del '+';
'programa.'+';
Salto+Salto+;
' Para llevar un mejor orden en las bases de datos '+';
'todas ellas ( a excepción de CONFIG.DEC ) están '+';
'indexadas en los archivos : NTXCURSO.NTX, '+';
'NTXORDEN, NTXEXPO.NTX, y NTXCARR y relacionados para '+';
'no tener información redundante.'+';
Salto+Salto+;
' Presione < ESC > para salir de esta pantalla.'
```

```
// La variable RESPONSABLE almacena el nombre del funcionario que firma
// los diversos reportes generados.
```

```
PRIVATE Responsable := "
```

```
// La variable VARAYUDA permite seleccionar el número de ayuda modular
// requerida por la función AYUDAMOD().
```

```
PRIVATE VarAyuda
```

```
// La variable Pantalla es un arreglo de 5 X 11 que permite construir
// 5 pantallas de selección usadas por el programa. Aquí PANTALLA[1],
// PANTALLA[2], PANTALLA[3], PANTALLA[4] y PANTALLA[5] tienen 11 elementos :
// los cuatro primeros hacen referencia a las coordenadas de la ventana
// principal; el quinto se refiere al tipo de marco a realizar; el sexto es
// el encabezado de la pantalla; el séptimo y octavo se refieren a las
// coordenadas iniciales de la opción; en noveno es la columna
// correspondiente donde va la explicación de la opción; el décimo es el
// nombre de la opción, y el onceavo es la explicación de la opción.
//
// Estos arreglos son empleados por las funciones : SELECCION Y
// PONPANTA.

PRIVATE Pantalla := {ARRAY(11), ARRAY(11), ARRAY(11), ARRAY(11),
ARRAY(11)}

Pantalla[1] = { 1,7,23,72,Marco7,'<< DEC MENU PRINCIPAL >>',3,10,34;;
  {' Altas      ' ' Bajas      ';;
  ' Consultas  ' ' Modificaciones ';;
  ' Reportes   ' ' Disposición   ';;
  ' Salir      ';;
  {'Permite agregar nuevos registros.';;
  'Elimina registros vigentes.';;
  'Realiza consultas diversas.';;
  'Modifica los registros vigentes.';;
  'Genera diversos tipos de reportes.';;
  'Cambia la configuración actual.';;
  'Finaliza programa DEC.' } }

Pantalla[2] = { 3,12,21,66,Marco7,',6,15,35;;
  {' Cursos    ' ' Ordenes    ';;
  ' Expositores ' ' Profesiones ';;
  ' Salir      ';;
  {'Información de los cursos.';;
  'Ordenes de pago.';;
  'Instructores de los cursos.';;
  'Carreras existentes.';;
  'Regresa al menú anterior.' } }

Pantalla[3] = { 4,11,19,68,Marco7,'<< TIPO DE CONSULTAS >>',7,14,34;;
  {' Globales  ' ' Por Clave  ';;
  ' Cadena    ' ' Salir      ';;
  {'Lista todos los registros.';;
  'Lista según la clave pedida.';;
```

```
'Lista según la cadena incluida.;;  
'Regresa al menú anterior.' } }
```

```
Pantalla[4] = { 3,10,21,68,Marco7,'<< CONFIGURACION >>',6,13,33.;;  
{ ' Unidad ';; Impresora ';;  
  ' Datos ';; Clave ';;  
  ' Salir '};;  
{ 'Selecciona la unidad de disco.;;  
'Selecciona el tipo de impresora.;;  
'Cambia los títulos por defecto.;;  
'Modifica la clave de acceso.;;  
'Regresa al menú anterior.' } }
```

```
Pantalla[5] = { 4,11,19,68,Marco7,'<< TIPO DE IMPRESORA >>',7,16,34.;;  
{ ' Matriz ';; Tinta ';;  
  ' Láser ';; Salir '};;  
{ 'Impresora de matriz de puntos.;;  
'Impresora de chorro de tinta.;;  
'Impresora láser.;;  
'Regresa al menú anterior.' } }
```

```
// Las variables M1, M2 y M3 son arreglos que contienen el nombre de  
// los campos, su encabezado y formato que debe de llevar las bases de  
// datos en las ventanas de la función estándar DBEDIT(). M1 es para  
// cursos, M2 es para ordenes de pago y M3 es para expositores.
```

```
PRIVATE m1, m2, m3
```

```
// La variable CLAVE permite capturar la clave tanto de cursos,  
// ordenes de pago, expositores y de carreras.
```

```
PRIVATE Clave
```

```
// La variable TOTALREG cuenta y almacena el total de registros que  
// cumplan con una condición especificada. Se emplea en las cuatro bases  
// de datos.
```

```
PRIVATE TotalReg
```

```
// Las variables NOPCION1, NOPCION2, NOPCION3, NOPCION4 y  
// NOPCION5 permiten controlar las alternativas de los menús.
```

```
PRIVATE nOpcion1 := nOpcion2 := nOpcion3 := nOpcion4 := nOpcion5 := 1
```

```
// Las siguientes variables realizan una copia de los campos de la  
// base de datos CURSOS.DBF
```

```
PRIVATE Alum, Acad, Admi, Mant, Egre, Publi, Nombre, Fec1, Fec2,;  
Luga, Dura, Hor1_1, Hor1_2, Hor2_1, Hor2_2, CveE, Cup, Salo,;  
CveCarr, Lu, Ma, Mi, Ju, Vi, Sa, Domingo, Memorandum
```

```
// La variable CVEAUX realiza una copia del campo CLAVE
```

```
PRIVATE CveAux
```

```
// La variable POSAUX salva la posición actual del puntero de la base  
// de datos en uso.
```

```
PRIVATE PosAux
```

```
// Las variables C1, C2, L1 y L2 almacenan las coordenadas para exhibir  
// los registros de una base de datos en consultas globales o exhibir campos  
// memorándum.
```

```
PRIVATE c1, c2, l1, l2
```

```
// Las variables G y GAUX salvan toda la pantalla actual del video.
```

```
PRIVATE G, GAux
```

```
// La variable FIN controla el bucle de la función PRINCIPAL().
```

```
PRIVATE Fin
```

```
// La variable PROSEGUIR determina si una alta, baja o modificación de  
// la base de datos activa procede.
```

```
PRIVATE Proseguir
```

```
// Las variables NUMVEN, CENTROX y CEMTROY son auxiliares empleadas
```

// por las funciones VENTANAEXPLOSIVA() y VENTANACML()

PRIVATE NumVen, CentroX, CentroY

// Las siguientes variables realizan una copia de los campos de la base
// de datos ORDENES.DBF

PRIVATE Tele, CveP, pr, Cuen, CveCond, Seme, Cost

// Las siguientes variables realizan una copia de los campos de la
// base de datos EXPOSITO.DBF

PRIVATE Registro, Dorni, Adsc, Titu, Cedu

// Las variables TEXTO y TEXTOAUX almacenan el contenido
// de los archivos de texto generados en el módulo de reportes.

PRIVATE Texto, TextoAux

// Las variables CTEXTO y CTEXTOAUX almacenan el contenido
// de los informes a generar y estos se visualiza a través de
// la función estandar MEMOEDIT().

PRIVATE cTexto, cTextoAux

// La variable NOMARCH almacena el nombre de los archivos de reportes
// a generar. Es de tipo CHARACTER.

PRIVATE NomArch := "

// La variable ENTRAR permite el acceso en altas, bajas y cambios de
// las cuatro bases de datos si la clave de acceso es correcta.

PRIVATE Entrar := .F.

// Las siguientes variables son arreglos que permiten capturar recibos
// pagados por un interesado sobre un mismo curso.

PRIVATE Recibo, Cantidad, Oficio, FechaC, FechaD

// La variable TOPE representa el último elemento en las pilas de RECIBO.
// CANTIDAD, OFICIO, FECHAC y FECHAD y la variable ACTUAL representa el
// elemento vigente.

PRIVATE Tope, Actual

// La variable CADREC almacena la cadena de caracteres a sustituir
// en el campo RECIBOS de la base de datos de ORDEN.DBF.

PRIVATE CadRec

// La variable DIASEXTRAIDOS almacena en un arreglo los números de
// días de la semana en que se imparte un curso.

PRIVATE DiasExtraidos

// La variable FECHABASE representa la fecha límite inicial en la cual
// se puede incluir cursos.

PRIVATE FechaBase

// La variable REPETIDOS almacena el número total de aquellos
// de recibos que tengan recurrencia.

PRIVATE Repetidos

// La variable RECIAUX salva el contenido del campo RECIBOS.

PRIVATE ReciaAux

// Las siguientes variables son auxiliares y permiten capturar los
// datos del registro de inscripción

PRIVATE Escuela := SPACE(50), Depto := SPACE(40), Dirigido := SPACE(40),;
Cargo1 := SPACE(40), Cargo2 := SPACE(40), Suscribe := SPACE(40),;
Fecha := Hoy, Instituci := SPACE(40), Honorario := 0,;
Ccp := "", Importe, Profe, NoOfAux

```
// La variable CVEACCESO almacena la clave para tener acceso  
// a altas, bajas y modificaciones.
```

```
PRIVATE CveAcceso := 'DEC'
```

```
// La variable LETRAS es un arreglo dinámico, el cual almacena las  
// letras iniciales de las opciones de los menús generados por la  
// variable PANTALLA, para que en la función SELECCION() sirvan como  
// teclas calientes ( Hot Keys ).
```

```
PRIVATE Letras
```

```
// Las variables FINCONSU y FINCONFIG permiten salir del módulo de  
// consultas y de configuración del sistema respectivamente.
```

```
PRIVATE FinConsu, FinConfig
```

```
// La variable PASSWORD permite digitar la cadena de la clave de acceso
```

```
PRIVATE PassWord := "
```

```
// La variable ESTRUCTURA almacena la estructura de las cuatro bases  
// de datos.
```

```
PRIVATE Estructura := { ARRAY(22), ARRAY(13), ARRAY(10), ARRAY(03) }
```

```
Estructura[1] = { { 'CVECURSO','C', 10, 0 },;  
  { 'NOMBRECUR','C', 40, 0 },;  
  { 'LUGAR','C', 40, 0 },;  
  { 'CVEEXPO','C', 3, 0 },;  
  { 'FECHAEXPO1','D', 8, 0 },;  
  { 'FECHAEXPO2','D', 8, 0 },;  
  { 'DURACION','N', 3, 0 },;  
  { 'HORARIO1_1','C', 2, 0 },;  
  { 'HORARIO1_2','C', 2, 0 },;  
  { 'HORARIO2_1','C', 2, 0 },;  
  { 'HORARIO2_2','C', 2, 0 },;  
  { 'SALON','C', 10, 0 },;  
  { 'COSTOALUM','N', 8, 2 },;  
  { 'COSTOACAD','N', 8, 2 },;  
  { 'COSTOADMIN','N', 8, 2 },;
```

```
{ 'COSTOMANU', 'N', 8, 2 },;  
{ 'COSTOEGRE', 'N', 8, 2 },;  
{ 'COSTOPUBL', 'N', 8, 2 },;  
{ 'OBSERVA', 'M', 10, 0 },;  
{ 'CUPO', 'N', 3, 0 },;  
{ 'DIAS', 'C', 7, 0 },;  
{ 'CVECAR', 'C', 3, 0 }
```

```
Estructura[2] = { { 'CVECURSO', 'C', 10, 0 },;  
  { 'NOMBREALUM', 'C', 40, 0 },;  
  { 'TELEFONO', 'N', 12, 0 },;  
  { 'RFC', 'C', 13, 0 },;  
  { 'CVEPROCEDA', 'C', 1, 0 },;  
  { 'PROCEDENCIA', 'C', 40, 0 },;  
  { 'CVECARRERA', 'C', 3, 0 },;  
  { 'CUENTA', 'C', 8, 0 },;  
  { 'CVECONDICI', 'C', 1, 0 },;  
  { 'SEMESTRE', 'N', 2, 0 },;  
  { 'COSTOCUR', 'N', 8, 2 },;  
  { 'RECIPOS', 'M', 10, 0 },;  
  { 'OBSERVA', 'M', 10, 0 } }
```

```
Estructura[3] = { { 'CVEEXPO', 'C', 3, 0 },;  
  { 'NOMBREEXPO', 'C', 30, 0 },;  
  { 'RFC', 'C', 13, 0 },;  
  { 'DOMICILIO', 'C', 40, 0 },;  
  { 'TELEFONO', 'N', 12, 0 },;  
  { 'ADSCRIPCIO', 'C', 40, 0 },;  
  { 'TITULO', 'C', 40, 0 },;  
  { 'CEDULA', 'C', 10, 0 },;  
  { 'EXPEDICION', 'D', 8, 0 },;  
  { 'OBSERVA', 'M', 10, 0 } }
```

```
Estructura[4] = { { 'CVEPROFESI', 'C', 3, 0 },;  
  { 'NOMBRECARR', 'C', 45, 0 },;  
  { 'OBSERVA', 'M', 10, 0 } }
```

```
// La variable EXPLICA, es un arreglo que almacena la explicación  
// de los elementos del arreglo ESTRUCTURA que sean de tipo caracter.  
// Al mismo tiempo delimita los campos de tipo caracter a ser  
// empleados para consultas por cadena incluida, los cuales tienen  
// un contenido diferente a NIL.
```

```
PRIVATE Explica := { ARRAY(22), ARRAY(13), ARRAY(10), ARRAY(03) }
```



```
Explica[1] = { NIL,;
'Nombre del curso.           ;;
'Lugar de exposición.        ;;
'Clave del expositor.       ;;
NIL;
NIL;
NIL;
'Hora de inicio de curso.    ;;
NIL;
'Hora de termino de curso.   ;;
NIL;
'Aula de exposición.        ;;
NIL;
NIL;
NIL;
NIL;
NIL;
NIL;
NIL;
NIL;
'Clave de la carrera que apoya. }
```

```
Explica[2] = { NIL,;
'Nombre del alumno.         ;;
NIL;
'Registro federal de contribuyentes. ;;
'Clave de la procedencia del alumno. ;;
'Instituto de adscripción del alumno. ;;
'Clave de la carrera del alumno.     ;;
'Número de cuenta del alumno.        ;;
'Clave de la condición del alumno.    ;;
NIL;
NIL;
NIL;
NIL };
```

```
Explica[3] = { NIL,;
'Nombre del expositor.      ;;
'Registro federal de contribuyentes. ;;
'Domicilio del expositor.   ;;
NIL;
'Instituto de adscripción del expositor.;;
'Título profesional del expositor.     ;;
'Cédula profesional del expositor.     ;;
NIL,;
```

```
NIL }

Explica[4] = { NIL,;
              'Nombre de la carrera.      ';;
              NIL }

// Las variables PERIODO1 y PERIODO2 almacenan la fecha inicial y la
// final respectivamente para generar el reporte a la unidad de planeación
// y para hacer respaldos.

PRIVATE Periodo1, Pericdo2

// La variable NOMPATH almacena el nombre de la última unidad y/o ruta
// seleccionada por el usuario.

PRIVATE NomPath := SPACE(70)

// Las Variables MISVAR01, MISVAR02, MISVAR03, MISVAR04, MISVAR05,
// MISVAR06, MISVAR07, MISVAR08, MISVAR09 y MISVAR10 son variables
// auxiliares que almacenan los datos de la configuración del sistema.

PRIVATE MISVAR01, MISVAR02, MISVAR03, MISVAR04, MISVAR05,;
        MISVAR06, MISVAR07, MISVAR08, MISVAR09, MISVAR10

SETCANCEL(.F.)
SETCOLOR('15/0,....')
SET CURSOR OFF
CLEAR
BienVenido() // Genera la pantalla de bienvenida al sistema.
SET TALK OFF
SET KEY 28 TO Ayuda // Se asigna la tecla F1 (28) a la función AYUDA
Fin = .F.
SET DATE FORMAT TO 'dd/mm/yyyy' // Se crea el formato de la fecha
FechaBase = CTOD('01/01/1990')
CargarArch() // Se cargan las bases de datos siempre y cuando estas existan
IF IFin
    Relaciona()
    Fondo()
    IF FILE('Config.dec') // Se carga la configuración
        RESTORE FROM Config.dec ADDITIVE
        Instituci = MisVar01
```

```
Escuela = MisVar02
Depto = MisVar03
Dirigido = MisVar04
Cargo1 = MisVar05
Suscribe = MisVar06
Cargo2 = MisVar07
Ccp = MisVar08
CveAcceso = MisVar09
nOpcion5 = MisVar10
ELSE
  MisVar01 = Instituci
  MisVar02 = Escuela
  MisVar03 = Depto
  MisVar04 = Dirigido
  MisVar05 = Cargo1
  MisVar06 = Suscribe
  MisVar07 = Cargo2
  MisVar08 = Ccp
  MisVar09 = CveAcceso
  MisVar10 = nOpcion5
ENDIF
Responsable = Suscribe
@ 24,69 SAY 'F1 = Ayuda' COLOR '0/7,....'
Acceso() // Solicita clave de acceso para entrar a altas, bajas y cambios
PonPanta(1)
DO WHILE !Fin
  /* Se escoge una opción del menú principal */
  nOpcion1 = Seleccion(1,nOpcion1)
  DO CASE
    /* Se seleccionó la opción altas */
    CASE nOpcion1 = 1
      // Si el usuario no tiene acceso
      IF !Entrar
        Error(33)
      ELSE
        /* Se escoge la base de datos a utilizar */
        Pantalla[2,6] = '<< A L T A S >>'
        PonPanta(2)
        nOpcion2 = Seleccion(2,nOpcion2)
        DO CASE
          /* Altas de cursos */
          CASE nOpcion2 = 1
            SELECT A
            AltasCursos()

          /* Altas de ordenes */
```

```
CASE nOpcion2 = 2
  AltasOrden()

/* Altas de expositores */
CASE nOpcion2 = 3
  AltasExpo()

/* Altas de Profesiones o Carreras */
CASE nOpcion2 = 4
  AltasCarr()

/* Decidió abandonar el módulo de altas */
CASE nOpcion2 = 5
  nOpcion2 = 1
ENDCASE
ENDIF

/* Se seleccionó la opción bajas */
CASE nOpcion1 = 2
  IF !Entrar
    Error(33)
  ELSE
    Pantalla[2,6] = '<< B A J A S >>'
    PonPanta(2)
    /* se escoge la base de datos a utilizar */
    nOpcion2 = Seleccion(2,nOpcion2)
    DO CASE
      /* Bajas de cursos */
      CASE nOpcion2 = 1
        BajasCursos()

      /* Bajas de ordenes de pago */
      CASE nOpcion2 = 2
        BajasOrden()

      /* Bajas de expositores */
      CASE nOpcion2 = 3
        BajasExpo()

      /* Bajas de profesiones o carreras */
      CASE nOpcion2 = 4
        BajasCarr()

      /* Decidió abandonar el menú de bajas */
      CASE nOpcion2 = 5
        nOpcion2 = 1
```

```
ENDCASE
ENDIF

/* Seleccionó la opción de consultas */
CASE nOpcion1 = 3
  FinConsu = .F.
  DO WHILE !FinConsu
    Pantalla[2,6] = '<< CONSULTAS >>'
    PonPanta(2)
    nOpcion2 = Seleccion(2,nOpcion2)
    DO CASE
      /* Consultas de cursos */
      CASE nOpcion2 = 1
        ConsuCurso()

      /* Consultas de ordenes de pago */
      CASE nOpcion2 = 2
        ConsuOrden()

      /* Consultas de expositores */
      CASE nOpcion2 = 3
        ConsuExpo()

      /* Consultas de profesiones o carreras */
      CASE nOpcion2 = 4
        ConsuCarr()

      /* Decidió salirse */
      CASE nOpcion2 = 5
        nOpcion2 = 1
        FinConsu = .T.
    ENDCASE
  ENDDO
ENDCASE

/* Seleccionó la opción de modificaciones */
CASE nOpcion1 = 4
  IF !Entrar
    Error(33)
  ELSE
    Pantalla[2,6] = '<< MODIFICACIONES >>'
    PonPanta(2)
    nOpcion2 = Seleccion(2,nOpcion2)
    DO CASE
      /* Modificaciones de cursos */
      CASE nOpcion2 = 1
        ModiCursos()
    ENDCASE
  ENDDO
ENDCASE
```

```
/* Modificaciones de ordenes de pago */  
CASE nOpcion2 = 2  
  ModiOrden()
```

```
/* Modificaciones de expositores */  
CASE nOpcion2 = 3  
  ModiExpo()
```

```
/* Modificaciones de profesiones o carreras */  
CASE nOpcion2 = 4  
  ModiCarr()
```

```
/* Decidió salirse */  
CASE nOpcion2 = 5  
  nOpcion2 = 1  
ENDCASE  
ENDIF
```

```
/* Seleccionó el módulo de reportes */  
CASE nOpcion1 = 5  
  Reportes()
```

```
/* Seleccionó el módulo de disposición o configuración */  
CASE nOpcion1 = 6  
  FinConfig = .F.  
  DO WHILE !FinConfig  
    PonPanta(4)  
    nOpcion4 = Seleccion(4,nOpcion4)  
    DO CASE  
      /* Cambia la unidad de disco */  
      CASE nOpcion4 = 1  
        Unidad()  
  
      /* Cambia el tipo de impresora */  
      CASE nOpcion4 = 2  
        Impresora()  
  
      /* Cambia los datos para los informes */  
      CASE nOpcion4 = 3  
        Datos()  
  
      /* Cambia el password */  
      CASE nOpcion4 = 4  
        ClaveDeAcc()
```

```
      /* Decidió salirse */
      CASE nOpcion4 = 5
        nOpcion4 = 1
        FinConfig = .T.
      ENDCASE
    ENDDO

    /* Decidió salir del programa */
    CASE nOpcion1 = 7
      IF Confirmar(1) = 1
        Fin = .T.
      ENDIF
    ENDCASE
    IF nOpcion1 < 7
      PonPanta(1)
    ENDIF
  ENDDO
ENDIF
SAVE ALL LIKE MisVar?? TO CONFIG.DEC
SET TALK ON
CLOSE ALL
SET KEY 28 TO
SETCANCEL(.T.)
SETCOLOR('7/0,....')
CLEAR
@ 23,00
SET CURSOR ON
SET DATE TO AMERICAN
RETURN NIL
```

```
/***
* Acceso() --> lEntrada
*
* Solicita la clave de entrada para tener acceso a los
* módulos de altas, bajas y modificaciones. De tener acceso
* <lEntrada> devuelve .T.
*
*/
FUNCTION Acceso
  LOCAL Fallas := 0

  DO WHILE Fallas < 3
    PassWord = SPACE(4)
```

```
Caja (' Teclee la clave de acceso para realizar altas,+;  
' bajas y cambios ','15/1,...')  
GAux = G  
SETCOLOR('0/0..')  
@ 12,38 GET Password PICT '@'  
READ  
RESTSCREEN(0,0,24,79,GAux)  
IF LASTKEY() <> 27  
  IF RTRIM(UPPER>Password)) == RTRIM(CveAcceso)  
    Entrar = .T.  
    Fallas = 3  
  ELSE  
    ++Fallas  
    DO CASE  
      CASE Fallas = 1 .OR. Fallas = 2  
        Error(35)  
      OTHERWISE  
        Error(36)  
        Entrar = .F.  
        Fallas = 3  
    ENDCASE  
  ENDIF  
ELSE  
  Error(36)  
  Entrar = .F.  
  Fallas = 3  
ENDIF  
ENDDO  
RETURN Entrar
```

/***

* BienVenido() --> NIL

*

* Esta función coloca la pantalla de bienvenida al programa.

*

*/

FUNCTION BienVenido

VentanaSin(Marco6, '15/7,.....')

@ 0,2,23,77 BOX Marco7 COLOR '15/7,.....'

Asteriscos(4,25,55,' ¡ Bienvenido a DEC Ver. 1.0 ! ','14/1,.....')

AnuncioDer(6,15,'Derechos reservados (C) : ','4/7,.....')

AnuncioIzq(6,43,'CESAR MANDUJANO LUNA','0/7,.....')

Ok(21, 37, '15/4,.....', '0/7,.....',4*16+15)


```
VentanaExplosiva(8,6,19,73,Marco7,)  
RamaV(10,73,17,FlechaV,'0/7,...')  
MEMOEDIT(ExplicaPro, 9,8,17,71, .F.)  
SET CURSOR OFF  
Boton(21,37,42)  
RETURN NIL
```

```
...../
```

```
.....  
* Este es el módulo de Cursos *  
...../
```

```
/***  
* ConsuCurso() --> NIL  
*  
* Permite la selección del tipo de consultas de  
* cursos, ya sean globales o por clave.  
*  
*/  
FUNCTION ConsuCurso
```

```
    PonPanta(3)  
    nOpcion3 = Seleccion(3,nOpcion3)  
    DO CASE  
        CASE nOpcion3 = 1  
            ConsuCurG()  
            FinConsu = .T.  
        CASE nOpcion3 = 2  
            ConsuCurC()  
            FinConsu = .T.  
        CASE nOpcion3 = 3  
            ConsuCurI()  
            FinConsu = .T.  
        CASE nOpcion3 = 4  
            nOpcion3 = 1  
    ENDCASE  
RETURN NIL
```

```
/*  
* AltasCursos() --> NIL  
*  
* Da de alta un nuevo curso.  
*  
*/  
FUNCTION AltasCursos  
  LOCAL Terminar  
  
  SELECT A  
  PantasBases('<< ALTAS DE CURSOS >>')  
  PonConCan()  
  IF C->(LASTREC()) = 0 // No se permiten altas de cursos ya que la base de  
  Error(07) // datos de expositores está vacía.  
  ELSE  
    Terminar = .F.  
    DO WHILE !Terminar  
      PantasCursos()  
      BlancosCur()  
      ControlAiC(20,23..T.)  
      IF Confirmar(2) = 2  
        Terminar = .T.  
      ENDIF  
    ENDDO  
  ENDIF  
  RETURN NIL
```

```
/*  
* BajasCursos() --> NIL  
*  
* Da de baja un curso existente.  
*  
*/  
FUNCTION BajasCursos  
  LOCAL Opcion, Terminar  
  
  SELECT A  
  PantasBases('<< BAJAS DE CURSOS >>')  
  PonConCan()  
  Terminar = .F.  
  DO WHILE !Terminar  
    PantasCursos()  
    IF LASTREC() = 0
```

```
Error(5)
  Terminar = .T.
ELSE
  PantaClave(1,")
  IF LASTKEY() != 27
    SEEK(Clave)
    IF FOUND()
      SKIP
      IF CveCurso = Clave
        CajaCampos()
        SETCOLOR('15/2,15/0,..')
        SET FILTER TO CveCurso = Clave
        CamposCur()
        GO TOP
        SET KEY 28 TO
        DBEDIT(11,c1,i2,c2,m1,"m3,m2,;
          CHR(220)+CHR(220).CHR(219).CHR(223)+CHR(223))
        SET KEY 28 TO AYUDA
        RESTSCREEN(0,0,24,79,G)
        IF LASTKEY() != 27
          PonInfoCur(RECNO())
          Memorandum = Observa
          ControlBaC(20,23,.F.)
        ENDIF
        SET FILTER TO
      ELSE
        SKIP -1
        PonInfoCur(RECNO())
        Memorandum = Observa
        ControlBaC(20,23,.F.)
      ENDIF
    ELSE
      Error(3)
    ENDIF
  ENDIF
  IF Confirmar(3) = 2
    Terminar = .T.
  ENDIF
ENDIF
ENDC/O
RETURN NIL
```

/*

- ControlBaC(<nRen>, <nCol>, <IVerdad>) --> NIL

*
* Controla la apertura de las opciones de la pantalla de
* bajas de cursos. Aquí <nRen> y <nCol> son las coordenadas
* en la cual está ubicada la opción "Memorándum" y <IVerdad>
* en caso de ser .F., indica que el campo memorándum no se
* puede modificar de lo contrario este si es modificable.
*

*/
FUNCTION ControlBaC(nRen, nCol, IVerdad)
LOCAL i

```
i = 2
DO WHILE i != 0
  DO CASE
    CASE i = 1
      i = 2
    CASE i = 2
      i = ControlMemo(nRen, nCol, IVerdad)
    CASE i = 3
      i = ConCan()
    CASE i = 4
      i = 3
    CASE i = 6 // Borra físicamente el curso seleccionado
      PosAux = RECNO()
      IF CncBaseOrd(45)
        Go PosAux
        DELETE
        PACK
      ENDIF
      i = 0
    ENDCASE
  ENDDO
RETURN NIL
```

```
/***
* ModiCursos() --> NIL
*
* Modifica los datos de un curso ya existente.
*
*/
```

```
FUNCTION ModiCursos
  LOCAL Opcion, Terminar
```

```
SELECT A
PantaBases('<< MODIFICACIONES DE CURSOS >>')
PonConCan()
IF LASTREC() = 0
  Error(5)
ELSE
  IF C->(LASTREC()) = 0 // No se permiten cambios de cursos ya que la
    Error(07) // base de datos de expositores está vacía.
  ELSE
    Terminar = .F.
    DO WHILE !Terminar
      PantaCursos()
      PantaClave(1,")
      IF LASTKEY() != 27
        SEEK(Clave)
        IF FOUND()
          SKIP
          IF CveCurso = Clave
            CajaCampos()
            SETCOLOR('15/2,15/0,..')
            SET FILTER TO CveCurso = Clave
            CamposCur()
            GO TOP
            SET KEY 28 TO
            DBEDIT(i1..c1,i2..c2,m1,"..m3,m2,;
              CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
            SET KEY 28 TO AYUDA
            RESTSCREEN(0,0,24,79,G)
            IF LASTKEY() != 27
              PonInfoCur(RECNO())
              LlenaCaCur()
              PosAux = RECNO()
              Memorandum = Observa
              ControlMoC(20,23,..T.)
            ENDIF
            SET FILTER TO
          ELSE
            SKIP -1
            PonInfoCur(RECNO())
            LlenaCaCur()
            PosAux = RECNO()
            Memorandum = Observa
            ControlMoC(20,23,..T.)
          ENDIF
        ELSE
          Error(3)
```

```
    ENDIF
  ENDIF
  IF Confirmar(5) = 2
    Terminar = .T.
  ENDIF
ENDDO
ENDIF
RETURN NIL
```

```
/***
```

```
* ControlaIC( <nRen>, <nCol>, <IVerdad> ) --> NIL
```

```
*
```

```
* Controla la apertura de las opciones de la pantalla de  
* altas de cursos. Aquí <nRen> y <nCol> son las coordenadas  
* en la cual está ubicada la opción "Memorándum" y <IVerdad>  
* en caso de ser .F., indica que el campo memorándum no se  
* puede modificar de lo contrario este si es modificable.
```

```
*
```

```
*/
```

```
FUNCTION ControlaIC(nRen, nCol, IVerdad)
```

```
  LOCAL i, Salte
```

```
  Salte = READEXIT(.T.)
```

```
  i = 1
```

```
  DO WHILE i != 0
```

```
    DO CASE
```

```
      CASE i = 1
```

```
        i = CapturaCur()
```

```
      CASE i = 2
```

```
        i = ControlMemo(nRen,nCol,IVerdad)
```

```
      CASE i = 3
```

```
        i = ConCan()
```

```
      CASE i = 4
```

```
        i = 1
```

```
      CASE i = 5
```

```
        i = 2
```

```
      CASE i = 6
```

```
        i = AgregaCur()
```

```
    ENDCASE
```

```
  ENDDO
```

```
  READEXIT(Salte)
```

```
  RETURN NIL
```

```
/*  
* ControlMoC( <nRen>, <nCol>, <IVerdad> ) --> NIL  
*  
* Controla la apertura de las opciones de la pantalla de  
* cambios de cursos. Aquí <nRen> y <nCol> son las coordenadas  
* en la cual está ubicada la opción "Memorándum" y <IVerdad>  
* en caso de ser .F., indica que el campo memorándum no se  
* pueda modificar de lo contrario este si es modificable.  
*  
*/  
FUNCTION ControlMoC(nRen, nCol, IVerdad)  
  LOCAL i, Salte  
  
  Salte = READEXIT(.T.)  
  i = 1  
  DO WHILE i != 0  
    DO CASE  
      CASE i = 1  
        i = CapturaCur()  
      CASE i = 2  
        i = ControlMemo(nRen, nCol, IVerdad)  
      CASE i = 3  
        i = ConCan()  
      CASE i = 4  
        i = 1  
      CASE i = 5  
        i = 2  
      CASE i = 6  
        i = CambiaCur()  
    ENDCASE  
  ENDDO  
  READEXIT(Salte)  
  RETURN NIL  
  
/*  
* BlancosCur() --> NIL  
*  
* Asigna a las variables auxiliares para dar de alta  
* los datos de un curso, su valor por defecto.  
*  
*/  
FUNCTION BlancosCur
```

```
Clave = REPLICATE('0',10)
Alum = 0
Acad = 0
Admi = 0
Mant = 0
Egre = 0
Publ = 0
Nombre = SPACE(40)
Fec1 = CTOD("")
Fec2 = C TOD("")
Luga = SPACE(40)
Dura = 0
Hor1_1 = '00'
Hor1_2 = '00'
Hor2_1 = '00'
Hor2_2 = '00'
CveE = SPACE(3)
Cup = 0
Sal = SPACE(10)
CveCarr = '000'
Lu = 'N'
Ma = 'N'
Mi = 'N'
Ju = 'N'
Vi = 'N'
Sa = 'N'
Domingo = 'N'
Memorandum = ""
RETURN NIL
```

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

```
/***
```

```
* LlanaCaCur() --> NIL
```

```
*
```

```
* Asigna los contenidos del curso seleccionado, a las variables
```

```
* auxiliares para realizar los cambios requeridos.
```

```
*
```

```
*/
```

```
FUNCTION LlanaCaCur
```

```
Clave = CveCurso
```

```
CveAux = Clave
```

```
Alum = CostoAlum
```

```
Acad = CostoAcad
```



```
Admi = CostoAdmi
Mant = CostoManu
Egre = CostoEgre
Publ = CostoPubl
Nombre = NombreCur
Fec1 = FechaExpo1
Fec2 = FechaExpo2
Luga = Lugar
Dura = Duracion
Hor1_1 = Horario1_1
Hor1_2 = Horario1_2
Hor2_1 = Horario2_1
Hor2_2 = Horario2_2
CveE = CveExpo
Cup = Cupo
Salv = Salon
CveCarr = CveCar
Lu = SUBSTR(Dias,1,1)
Ma = SUBSTR(Dias,2,1)
Mi = SUBSTR(Dias,3,1)
Ju = SUBSTR(Dias,4,1)
Vi = SUBSTR(Dias,5,1)
Sa = SUBSTR(Dias,6,1)
Domingo = SUBSTR(Dias,7,1)
Memorandum = Observa
RETURN NIL
```

```
/**
```

```
* CapturaCur() --> nAccion
```

```
*
```

```
* Permite capturar los datos para un curso nuevo o  
* ya existente. Devuelve un código numérico según la  
* tecla presionada en <nAccion>.
```

```
*
```

```
**/
```

```
FUNCTION CapturaCur  
LOCAL i, T
```

```
SETCOLOR(,0/7,,,15/0')
```

```
SET CURSOR ON
```

```
@ 3,23 GET Clave PICT '9999999999'
```

```
@ 4,23 GET Alum PICT '999,999.99'
```

```
@ 5,23 GET Acad PICT '999,999.99'
```

```
@ 6,23 GET Admi PICT '999,999.99'
```

```

@ 7.23 GET Mant PICT '999,999.99'
@ 8.23 GET Egre PICT '999,999.99'
@ 9.23 GET Publ PICT '999,999.99'
@ 10.23 GET Nombre PICT '@!'
@ 11.23 GET Fec1
@ 11.23 GET Fec2
@ 12.23 GET Luga PICT '@!'
@ 13.23 GET Dura PICT '999'
@ 14.23 GET Hor1_1 PICT '99' VALID VAL(Hor1_1) <= 23
@ 14.28 GET Hor1_2 PICT '99' VALID VAL(Hor1_2) <= 59
@ 14.31 GET Hor2_1 PICT '99' VALID VAL(Hor2_1) <= 23
@ 14.34 GET Hor2_2 PICT '99' VALID VAL(Hor2_2) <= 59
@ 15.23 GET CveE PICT '999'
@ 16.23 GET Cup PICT '999'
@ 17.23 GET Salo PICT '@!'
@ 18.23 GET CveCarr PICT '999'
@ 19.23 GET Lu PICT '@!' VALID Lu = 'S' .OR. Lu = 'N'
@ 19.31 GET Ma PICT '@!' VALID Ma = 'S' .OR. Ma = 'N'
@ 19.39 GET Mi PICT '@!' VALID Mi = 'S' .OR. Mi = 'N'
@ 19.47 GET Ju PICT '@!' VALID Ju = 'S' .OR. Ju = 'N'
@ 19.55 GET Vi PICT '@!' VALID Vi = 'S' .OR. Vi = 'N'
@ 19.63 GET Sa PICT '@!' VALID Sa = 'S' .OR. Sa = 'N'
@ 19.71 GET Domingo PICT '@!' VALID Domingo = 'S' .OR. Domingo = 'N'
READ
T = LASTKEY()
DO CASE
  CASE T = 27
    i = 0
  CASE T = 18 .OR. T = 5
    i = 3
  OTHERWISE
    i = 2
ENDCASE
SET CURSOR OFF
RETURN i

```

/***

* AgregaCur() --> <nAccion>

*

* Realiza físicamente el alta de un nuevo curso siempre

* y cuando no exista error en validación de datos. Devuelve

* un código numérico según la tecla presionada en <nAccion>.

*

*/

FUNCTION AgregaCur

```
LOCAL i

Proseguir = .T.
Clave = STRTRAN(Clave,' ',0')
SELECT A
ChcCveCu()
ValidaCve()
ValidaCveR()
IF Proseguir
  SELECT A
  DatosCurRe()
  APPEND BLANK // Realiza el alta de un nuevo curso
  GrabaCur()
  i = 0
ELSE
  i = 1
ENDIF
RETURN i
```

```
/*
* CambiaCur() --> <nAccion>
*
* Realiza físicamente cambios a los datos de un curso siempre
* y cuando no exista error en validación de los mismos. Devuelve
* un código numérico según la tecla presionada, en <nAccion>.
*/
```

```
FUNCTION CambiaCur
LOCAL i

Proseguir = .T.
Clave = STRTRAN(Clave,' ',0')
SELECT A
ChcCveMcCu()
ValidaCve()
ValidaCveR()
i = 1
IF Proseguir
  SELECT A
  DatosCurRe( )
  GO PosAux
  IF CveAux != Clave
    IF ChcBaseOrd(46)
      GO PosAux
```

```
        GrabaCur()
        i = 0
    ELSE
        Clave = CveAux
    ENDIF
ELSE
    Go PosAux
    GrabaCur()
    i = 0
ENDIF
RETURN i
```

```
/*
* ChcBaseOrd( <IError> ) --> <IContinua>
*
* Revisa la posible existencia de ordenes de pago
* asociadas a un curso que se desea dar de baja o a
* modificar, de ser así la baja o cambio correspondiente
* no procede al retornar <IContinua> como falso.
*/
FUNCTION ChcBaseOrd( QueError)
    LOCAL Conti := .T.

    SELECT B
    SEEK(CveAux)
    IF FOUND()
        Error(QueError)
        Conti = .F.
    ENDIF
    SELECT A
    RETURN Conti
```

```
/*
* ValidaCve() --> NIL
*
* Examina la correcta captura de la clave de un curso. En
* caso de existir un error, este será notificado al usuario.
*/
FUNCTION ValidaCve
```

```
IF VAL(SUBSTR(Clave,1,1)) > 5
  Error(8)
  Proseguir = .F.
ENDIF
IF VAL(SUBSTR(Clave,2,1)) > 2
  Error(9)
  Proseguir = .F.
ENDIF
IF VAL(SUBSTR(Clave,3,1)) > 1
  Error(10)
  Proseguir = .F.
ENDIF
IF SUBSTR(Clave,1,1) != '1' .AND. SUBSTR(Clave,10,1) != '0'
  Error(11)
  Proseguir = .F.
ENDIF
IF SUBSTR(Clave,8,2) = '00'
  Error(21)
  Proseguir = .F.
ENDIF
IF SUBSTR(Clave,1,1) = '1' .AND. (SUBSTR(Clave,10,1) = '0' .OR.;
  SUBSTR(Clave,10,1) > '2')
  Error(12)
  Proseguir = .F.
ENDIF
RETURN NIL
```

/***

* ValidaCveR() --> NIL

*

* Examina la correcta captura de los datos de un curso,
* con excepción de su clave. En caso de existir un error,
* este es notificado al usuario.

*

*/

FUNCTION ValidaCveR

```
IF Alum < 0 .OR. Acad < 0 .OR. Admi < 0 .OR. Mant < 0;  
.OR. Egre < 0 .OR. Publ < 0
```

```
  Error(13)
```

```
  Proseguir = .F.
```

```
ENDIF
```

```
IF Fec1 > Fec2
```

```
Error(14)
Proseguir = .F.
ELSE
  DiasDeCur(Lu+Ma+Mi+Ju+Vi+Sa+Domingo)
  IF Extrae TotD ( Fec1.Fec2 ) = 0
    Error(30)
    Proseguir = .F.
  ENDIF
  IF Fec1 < FechaBase .OR. Fec2 < FechaBase
    Error(18)
    Proseguir = .F.
  ENDIF
ENDIF
IF Dura <= 0
  Error(20)
  Proseguir = .F.
ENDIF
IF Val(Hor1_1) < 10
  Hor1_1 = '0'+STRTRAN(STR(VAL(Hor1_1)),' ','')
ENDIF
IF Val(Hor1_2) < 10
  Hor1_2 = '0'+STRTRAN(STR(VAL(Hor1_2)),' ','')
ENDIF
IF Val(Hor2_1) < 10
  Hor2_1 = '0'+STRTRAN(STR(VAL(Hor2_1)),' ','')
ENDIF
IF Val(Hor2_2) < 10
  Hor2_2 = '0'+STRTRAN(STR(VAL(Hor2_2)),' ','')
ENDIF
IF VAL(Hor1_1)*100+VAL(Hor1_2) >= VAL(Hor2_1)*100+VAL(Hor2_2)
  Error(15)
  Proseguir = .F.
ENDIF
IF Val(CveCarr) < 10
  CveCarr = '00'+STRTRAN(STR(VAL(CveCarr)),' ','')
ELSE
  IF VAL(CveCarr) >= 10 .AND. VAL(CveCarr) < 100
    CveCarr = '0'+STRTRAN(STR(VAL(CveCarr)),' ','')
  ENDIF
ENDIF
IF CveCarr != '000'
  SELECT D
  LOCATE FOR CveProfesi = CveCarr
  IF IFFOUND()
    Error(16)
    Proseguir = .F.
  
```

```
ENDIF
ENDIF
SELECT C
IF VAL(CveE) < 10
  CveE = '00'+STRTRAN(STR(VAL(CveE)),' ','')
ELSE
  IF VAL(CveE) >= 10 .AND. VAL(CveE) < 100
    CveE = '0'+STRTRAN(STR(VAL(CveE)),' ','')
  ENDIF
ENDIF
SEEK(CveE)
IF 'FOUND()'
  Error(4)
  Proseguir = .F.
ENDIF
IF Cup <= 0
  Error(19)
  Proseguir = .F.
ENDIF
IF Lu+Ma+Mi+Ju+Vi+Sa+Domingo = 'NNNNNNNN'
  Error(32)
  Proseguir = .F.
ENDIF
RETURN NIL
```

/***

* GrabaCur() --> NIL

*

* Graba físicamente en el disco de trabajo las
* modificaciones o altas realizadas de cursos.

*

*/

FUNCTION GrabaCur

```
  REPLACE CveCurso WITH Clave, NombreCur WITH Nombre, Lugar WITH Luga,;
  CveExpo WITH CveE, FechaExpo1 WITH Fec1, FechaExpo2 WITH Fec2,;
  Duracion WITH Dura, Horario1_1 WITH Hor1_1,;
  Horario1_2 WITH Hor1_2, Horario2_1 WITH Hor2_1,;
  'Horario2_2 WITH Hor2_2, Salon WITH Salo, CostoAlum WITH Alum,;
  CostoAcad WITH Acad, CostoAdmi WITH Admi, CostoManu WITH Mant,;
  CostoEgre WITH Egre, CostoPubl WITH Publ,;
  Observa WITH Memorandum, Cupo WITH Cup,;
  CveCar WITH CveCarr, Dias WITH Lu+Ma+Mi+Ju+Vi+Sa+Domingo
```

RETURN NIL

```
/*  
* ChecaCveCu() --> NIL  
*  
* Revisa en altas si la clave de un curso se repite.  
*  
*/  
FUNCTION ChecaCveCu  
  SEEK(Clave)  
  IF FOUND()  
    Error(1)  
    Proseguir = .F.  
  ENDIF  
  RETURN NIL  
  
/*  
* ChcCveMoCu --> NIL  
*  
* Revisa en modificaciones si la clave de un curso se repite.  
*  
*/  
FUNCTION ChcCveMoCu  
  SELECT A  
  SEEK(Clave)  
  IF FOUND()  
    IF CveAux != CveCurso  
      Error(1)  
      Proseguir = .F.  
      Clave = CveAux  
    ENDIF  
  ENDIF  
  RETURN NIL  
  
/*  
* ConsuCurG() --> NIL  
*  
* Consultas globales de cursos.  
*  
*/  
FUNCTION ConsuCurG  
  SELECT A
```



```
IF LASTREC() = 0
  Error(5)
ELSE
  SET FILTER TO
  PantBases('<< CONSULTAS GLOBALES DE CURSOS >>')
  HazConsuCu()
ENDIF
RETURN NIL
```

```
/*
* ConsuCurl() --> NIL
*
* Consultas por cadena incluida.
*
*/
FUNCTION ConsuCurl
```

```
  SELECT A
  IF LASTREC() = 0
    Error(5)
  ELSE
    PantBases('<< CONSULTAS DE CURSOS POR CADENA INCLUIDA >>')
    IF PalabraInc(1)
      HazConsuCu()
    SET FILTER TO
  ENDIF
ENDIF
RETURN NIL
```

```
/*
* HazConsuCu --> NIL
*
* Realiza las consultas de cursos ya
* sea por clave o por cadena incluida.
*
*/
FUNCTION HazConsuCu
  LOCAL Continuar := .T.
```

```
  CamposCur()
  OK2()
  SET KEY 28 TO
```

```
DO WHILE Continuar
  PantaCursos()
  GO TOP
  CajaCampos()
  SET COLOR('15/2.15/0,..')
  SET KEY 28 TO
  DBEDIT(1 c1.l2.c2,m1,".m3,m2,;
    CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
  SET KEY 28 TO AYUDA
  RESTSCREEN(0,0.24.79 G)
  IF LASTKEY() != 27
    PonInfoCur(RECNO())
    Memorandum = Observa
    ControlCon(20.23..F.)
  ENDIF
  IF Confirmar(4) = 2
    Continuar = .F.
  ENDIF
ENDDO
SET KEY 28 TO Ayuda
RETURN :;:L
```

/***

```
* ConsuCurC() --> NIL
*
* Realiza consultas por clave de cursos.
*
*/
```

```
FUNCTION ConsuCurC
  LOCAL Opcion, Terminar
```

```
SELECT A
PantaBases('<< CONSULTAS POR CLAVE DE CURSOS >>')
OK2()
IF LASTREC() = 0
  Error(5)
ELSE
  Terminar = .F.
  DO WHILE !Terminar
    PantaCursos()
    PantaClave(1,")
    IF LASTKEY() != 27
      SEEK(Clave)
      IF FOUND()
```

```
SKIP
IF CveCurso = Clave
  CajaCampos()
  SETCOLOR('15/2,15/0,...')
  SET FILTER TO CveCurso = Clave
  CamposCur()
  GO TOP
  SET KEY 28 TC
  DBEDIT(11,c1,l2,c2,m1," m3,m2,;
    CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
  SET KEY 28 TO AYUDA
  RESTSCREEN(0,0,24,79,G)
  IF LASTKEY() != 27
    PonInfoCur(RECNO())
    Memorandum = Observa
    ControlCon(20,23,.F.)
  ENDIF
  SET FILTER TO
ELSE
  SKIP -1
  PonInfoCur(RECNO())
  Memorandum = Observa
  ControlCon(20,23,.F.)
ENDIF
ELSE
  Error(3)
ENDIF
ENDIF
IF Confirmar(4) = 2
  Terminar = .T.
ENDIF
ENDDO
ENDIF
RETURN NIL
```

```
/***
* CamposCur() -> NIL
* Asigna los valores necesarios para la función DBEDIT para Cursos.
*/
FUNCTION CamposCur
  LOCAL i
```

```
m1 = ARRAY(FCOUNT())
m2 = ARRAY(FCOUNT())
m3 = ARRAY(FCOUNT())
FOR i = 1 TO LEN(m1)
  m1[i] = FIELDNAME(i)
NEXT

m2 = { ' Clave del curso', ' Nombre del curso', ' Lugar de exposición',;
      ' Clave del Expositor', ' Fecha de inicio', ' Fecha de término',;
      ' Duración', ' Horario inicio', ' Minutos', ' Horario término',;
      ' Minutos', ' Salón', ' Costo Alumno', ' Costo Académico',;
      ' Costo Administrativo', ' Costo Mantenimiento', ' Costo Egresado',;
      ' Costo Público', ' Observaciones', ' Cupo', ' Días',;
      ' Carrera que apoya' }

m3[13] = '$999,999.99'
m3[14] = '$999,999.99'
m3[15] = '$999,999.99'
m3[16] = '$999,999.99'
m3[17] = '$999,999.99'
m3[18] = '$999,999.99'
RETURN NIL
```

```
/*-
* PonInfoCur ( <nPosReg> ) --> NIL
*
* Coloca la información del registro <nPosReg>
* de la base de datos de cursos en la pantalla.
*/
```

```
FUNCTION PonInfoCur ( nPosReg )
LOCAL i

GO nPosReg
SETCOLOR('15/0,...')
@ 3,23 SAY CveCurso
@ 4,23 SAY CostoAlum PICT '999,999.99'
@ 5,23 SAY CostoAcad PICT '999,999.99'
@ 6,23 SAY CostoAdmi PICT '999,999.99'
@ 7,23 SAY CostoManu PICT '999,999.99'
@ 8,23 SAY CostoEgre PICT '999,999.99'
@ 9,23 SAY CostoPubl PICT '999,999.99'
@ 10,23 SAY NombreCur
@ 11,23 SAY FechaExpo1
```

```
@ 11.35 SAY FechaExpo2
@@ 12.23 SAY Lugar
@@ 13.23 SAY Duracion
@@ 14.23 SAY Horario1_1
@@ 14.26 SAY Horario1_2
@@ 14.31 SAY Horario2_1
@@ 14.31 SAY Horario2_2
@@ 15.23 SAY CveExpo
@@ 16.23 SAY Cupo
@@ 17.23 SAY Salon
@@ 18.23 SAY CveCar
CveCarr = CveCar
CveAux = CveCurso
FOR I = 1 TO 7
  @ 19.31+15 SAY SUBSTR(Dias,i,1)
NEXT
DatosCurRe()
RETURN NIL
```

```
/***
* DatosCurRe( ) --> NIL
*
* Coloca los datos de la carrera y el nombre del
* expositor según el curso capturado o seleccionado.
*
*/
```

```
FUNCTION DatosCurRe

SETCOLOR('14/0,,'')
@ 15.31 SAY C->NombreExpo
IF CveCarr = '000'
  @ 18.31 SAY 'NINGUNA'+SPACE(38)
ELSE
  SELECT D
  LOCATE FOR CveProfesi = CveCarr
  @ 18.31 SAY NombreCarr
  SELECT A
ENDIF
RETURN NIL
```

```
/*
* Pantacursos() --> NIL
* Coloca la pantalla de cursos.
*
*/
FUNCTION Pantacursos
  SETCOLOR('15/1....')
  @ 2.2,21,77 BOX Marco7+'
  @ 3,4 SAY 'Clave curso      :'
  @ 4,4 SAY 'Alumnos         :'$'
  @ 5,4 SAY 'Académicos      :'$'
  @ 6,4 SAY 'Administrativos  :'$'
  @ 7,4 SAY 'Mantenimiento   :'$'
  @ 8,4 SAY 'Egresados       :'$'
  @ 9,4 SAY 'Público general  :'$'
  @ 10,4 SAY 'Nombre curso     :'
  @ 11,4 SAY 'Periodo        : / / - / /'
  @ 12,4 SAY 'Lugar
  @ 13,4 SAY 'Duracion       HRS.'
  @ 14,4 SAY 'Horario        - : HRS.'
  @ 15,4 SAY 'Clave expositor  :'
  @ 16,4 SAY 'Cupo
  @ 17,4 SAY 'Salón
  @ 18,4 SAY 'Clave carrera  :'
  @ 19,4 SAY 'Días           : [ ] Lu [ ] Ma [ ] Mi [ ] Ju [ ] Vi [ ] +;
      ' Sa [ ] Do'
  @ 20,4 SAY 'Observaciones : Memorándum'
RETURN NIL
```

.....

.....
* Este es el módulo de ordenes de pago *
.....

```
/*  
* ConsuOrden() --> NIL  
*  
* Permite la selección del tipo de consultas de ordenes  
* de pago, ya sean globales o por clave.  
*/  
FUNCTION ConsuOrden
```

```
  PonPanta(3)  
  nOpcion3 = Seleccion(3,nOpcion3)  
  DO CASE  
    CASE nOpcion3 = 1  
      ConsuOrdG()  
      FinConsu = .T.  
    CASE nOpcion3 = 2  
      ConsuOrdC()  
      FinConsu = .T.  
    CASE nOpcion3 = 3  
      ConsuOrdI()  
      FinConsu = .T.  
    CASE nOpcion3 = 4  
      nOpcion3 = 1  
  ENDCASE  
  RETURN NIL
```

```
/*  
* AltasOrden() --> NIL  
*  
* Da de alta una nueva orden de pago.  
*/
```

```
FUNCTION AltasOrden  
  LOCAL Terminar  
  
  SELECT B  
  PantaBases('<< ALTAS DE ORDENES DE PAGO >>')  
  PonConCan()  
  IF A->(LASTREC()) = 0 // No se permiten altas ya que no existiría  
    Error(05) // ningún curso asociado.  
  ELSE  
    Terminar = .F.  
    DO WHILE !Terminar  
      PantaOrden()
```

```
BlancosOrd()
ControlAIO(18,25,.T.)
IF Confirmar(2) = 2
  Terminar = .T.
ENDIF
ENDDO
ENDIF
RETURN NIL
```

```
****
```

```
* BajasOrden() --> NIL
*
* Da de baja una orden de pago existente.
*
*/
```

```
FUNCTION BajasOrden
  LOCAL Opcion, Terminar
```

```
  SELECT B
  PantaBases('<< BAJAS DE ORDES DE ORDENES DE PAGO >>')
  PonConCan()
  Terminar = .F.
  DO WHILE !Terminar
    PantaOrden()
    IF LASTREC() = 0
      Error(6)
      Terminar = .T.
    ELSE
      PantaClave(2,")
      IF LASTKEY() != 27
        SEEK(Clave)
        IF FOUND()
          SKIP
          IF CveCurso = Clave
            CajaCampos()
            SETCOLOR('15/2,15/0,...')
            SET FILTER TO CveCurso = Clave
            CamposOrd()
            GO TOP
            SET KEY 28 TO
            DBEDIT(11,c1,l2,c2,m1,".m3,m2.;
              CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
            SET KEY 28 TO AYUDA
            SET FILTER TO
```



```
RESTSCREEN(0,0,24,79,G)
IF LASTKEY() != 27
  PonInfoOrd(RECNO())
  Memorandum = Observa
  CargarRec()
  ControlBaO(18,25,.F.)
ENDIF
ELSE
  SKIP -1
  PonInfoOrd(RECNO())
  Memorandum = Observa
  CargarRec()
  ControlBaO(18,25,.F.)
ENDIF
ELSE
  Error(3)
ENDIF
ENDIF
IF Confirmar(3) = 2
  Terminar = T
ENDIF
ENDIF
ENDDO
RETURN NIL
```

/*

* ModiOrden() --> NIL

*

* Modifica los datos de una orden de pago curso ya existente.

*

*/

FUNCTION ModiOrden

LOCAL Opcion, Terminar

SELECT B

PantaBases('<< MODIFICACIONES DE ORDENES DE PAGO >>')

PonConCan()

IF LASTREC() = 0

Error(6)

ELSE

IF A->(LASTREC()) = 0 // No se permiten cambios ya que no existiría

Error(05) // ningún curso asociado.

ELSE

Terminar = .F.

```
DO WHILE !Terminar
  PantaOrden()
  PantaClave(2,")
  IF LASTKEY() != 27
    SEEK(Clave)
    IF FOUND()
      SKIP
      IF CveCurso = Clave
        CajaCampos()
        SETCOLOR('15/2,15/0,,')
        SET FILTER TO CveCurso = Clave
        CamposOrd()
        GO TOP
        SET KEY 28 TO
        DBEDIT(11,c1,12,c2,m1,,"m3,m2,;
          CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
        SET KEY 28 TO AYUDA
        RESTSCREEN(0,0,24,79,G)
        IF LASTKEY() != 27
          PonInfoOrd(RECNO())
          LienaCaOrd()
          PosAux = RECNO()
          Memorandum = Observa
          ControlMoO(18,25,.,T.)
        ENDIF
        SET FILTER TO
      ELSE
        SKIP -1
        PonInfoOrd(RECNO())
        LienaCaOrd()
        PosAux = RECNO()
        Memorandum = Observa
        ControlMoO(18,25,.,T.)
      ENDIF
    ELSE
      Error(3)
    ENDIF
  ENDIF
  IF Confirmar(5) = 2
    Terminar = .T.
  ENDIF
ENDDO
ENDIF
ENDIF
RETURN NIL
```

```
/***  
* ControlAIO( <nRen>, <nCol>, <IVerdad> ) --> NIL  
*  
* Controla la apertura de las opciones de la pantalla de  
* altas de ordenes. Aquí <nRen> y <nCol> son las coordenadas  
* en la cual está ubicada la opción "Memorándum" y <IVerdad>  
* en caso de ser .F., indica que el campo memorándum no se  
* puede modificar de lo contrario este si es modificable.  
*  
*/
```

```
FUNCTION ControlAIO(nRen, nCol, IVerdad)
```

```
LOCAL i, Salte
```

```
Salte = READEXIT(.T.)
```

```
i = 1
```

```
DO WHILE i != 0
```

```
DO CASE
```

```
CASE i = 1
```

```
    i = CapturaOrd()
```

```
CASE i = 2
```

```
    i = ControlMemo(nRen, nCol, IVerdad)
```

```
CASE i = 3
```

```
    i = ConCan()
```

```
CASE i = 4
```

```
    i = ControlRec( .F. )
```

```
CASE i = 5
```

```
    i = CostoOrd()
```

```
CASE i = 6
```

```
    i = AgregaOrd()
```

```
ENDCASE
```

```
ENDDO
```

```
READEXIT(Salte)
```

```
RETURN NIL
```

```
/***
```

```
* CostoOrd() --> <nAccion>
```

```
*  
*
```

```
* Coloca el costo de un curso seleccionado  
* en función de su condición y al mismo tiempo  
* permite modificarlo. Devuelve en <nAccion>  
* un código numérico el cual corresponde a la  
* tecla presionada.  
*  
*/
```

```
FUNCTION CostoOrd
LOCAL T

SELECT A
SEEK(C13.7e)
IF FOUND()
  DatosCosto()
  SET CURSOR ON
  @ 15.37 GET Cost PICT '999,999.99'
  READ
  SET CURSOR OFF
  T = LASTKEY()
  DO CASE
    CASE T = 27
      i = 0
    CASE T = 5 .OR. T = 18
      i = 1
    OTHERWISE
      i = 4
  ENDCASE
ELSE
  Error(3)
  i = 1
ENDIF
SELECT B
RETURN i
```

```
/*
* DatosCosto() --> NIL
*
* Selecciona el costo de un curso que le corresponde
* a un alumno en función de su condición.
*/
```

```
FUNCTION DatosCosto

SETCOLOR('14/0,....')
DO CASE
  CASE CveCond = '0'
    Cost = A->CostoPubl
  CASE CveCond = '1'
    Cost = A->CostoAlum
  CASE CveCond = '2'
    Cost = A->CostoAcad
```

```
CASE CveCond = '3'  
  Cost = A->CostoAdmi  
CASE CveCond = '4'  
  Cost = A->CostoManu  
CASE CveCond = '5'  
  Cost = A->CostoEgre  
ENDCASE  
@ 15.25 SAY Cost PICT '999,999.99'  
RETURN NIL
```

/*

* ControlBaO(<nRen>, <nCol>, <IVerdad>) --> NIL

*/

* Controla la apertura de las opciones de la pantalla de
* bajas de ordanes. Aqui <nRen> y <nCol> son las coordenadas
* en la cual está ubicada la opción "Memorándum" y <IVerdad>
* en caso de ser .F.. indica que el campo memorándum no se
* puede modificar de lo contrario este si es modificable.

*/

```
FUNCTION ControlBaO(nRen, nCol, IVerdad)  
  LOCAL i
```

```
  i = 5
```

```
  DO WHILE i != 0
```

```
    DO CASE
```

```
      CASE i = 1
```

```
        i = 4
```

```
      CASE i = 2
```

```
        i = ControlMemo(nRen, nCol, IVerdad)
```

```
      CASE i = 3
```

```
        i = ConCan()
```

```
      CASE i = 4
```

```
        i = ControlRec(.T.)
```

```
      CASE i = 5
```

```
        i = 3
```

```
      CASE i = 6
```

```
        SELECT B // Elimina físicamente la orden de pago seleccionada
```

```
        DELETE
```

```
        PACK
```

```
        i = 0
```

```
      ENDCASE
```

```
    ENDDO
```

```
  RETURN NIL
```

```
/*  
* ControlMoO( <nRen>, <nCol>, <IVerdad> ) --> NIL  
*  
* Control la apertura de las opciones de la pantalla de  
* campos de ordenes. Aquí <nRen> y <nCol> son las coordenadas  
* en la cual está ubicada la opción "Memorándum" y <IVerdad>  
* en caso de ser .F.. indica que el campo memorándum no se  
* pueda modificar de lo contrario este si es modificable.  
*  
*/
```

```
FUNCTION ControlMoO(nRen, nCol, IVerdad)  
  LOCAL i, Salte
```

```
  Salte = READEXIT(.T.)  
  i = 1  
  DO WHILE i != 0  
    DO CASE  
      CASE i = 1  
        i = CapturaOrd()  
      CASE i = 2  
        i = ControlMemo(nRen,nCol,IVerdad)  
      CASE i = 3  
        i = ConCan()  
      CASE i = 4  
        i = ControlRec( F )  
      CASE i = 5  
        i = CostoOrd()  
      CASE i = 6  
        i = CambiaOrd()  
    ENDCASE  
  ENDDO  
  READEXIT(Salte)  
  RETURN NIL
```

```
/*  
* BlancosOrd() --> NIL  
*  
* Asigna a las variables auxiliares para dar de alta  
* los datos de ordenes de pago, su valor por defecto.  
*  
*/  
FUNCTION BlancosOrd
```

```
Clave = REPLICATE('0',10)
Nombre = SPACE(40)
Registro = SPACE(13)
Tele = 0
CveP = 0
Pr = SPACE(40)
CveCarr = '000'
Cuen = REPLICATE('0',8)
CveCond = '0'
Cost = 0
Seme = 0
ArrayVacio()
Tope = 0
Actual = 0
CadRec = CHR(0)
Memorandum = ""
RETURN NIL
```

```
/***
```

```
* LlenaCaOrd() --> NIL
```

```
*
```

```
* Asigna los contenidos de la orden de pago seleccionada, a las  
* variables auxiliares para realizar los cambios requeridos.
```

```
*
```

```
*/
```

```
FUNCTION LlenaCaOrd
```

```
Clave = CveCurso
Nombre = NombreAlum
Registro = RFC
Tele = Telefono
CveP = CveProcede
Pr = Procedenci
CveCarr = CveCarrera
Cuen = Cuenta
CveCond = CveCondici
Cost = CostoCur
Seme = Semestre
CargarRec()
Memorandum = Observa
RETURN NIL
```

```
/***
* CapturaOrd() --> nAccion
*
* Permite capturar los datos para una una orden
* de pago nueva o ya existente.
*/
FUNCTION CapturaOrd
  LOCAL i T

  SETCOLOR('0/7...15/0')
  SET CURSOR ON
  @ 4.25 GET Clave PICT '9999999999'
  @ 5.25 GET Nombre PICT '@!'
  @ 6.25 GET Registro PICT '!!!!999999!!!'
  @ 7.25 GET Tele PICT '999999999999' VALID Tele >= 0
  @ 8.25 GET CveP PICT '9'
  @ 8.35 GET pr PICT '@!'
  @ 9.25 GET CveCarr PICT '999'
  @ 10.25 GET CveCarr PICT '999999999'
  @ 11.25 GET CveCond PICT '9' VALID VAL(CveCond) >= 0 .AND. VAL(CveCond) <= 5
  @ 12.25 GET Seme PICT '09' VALID Seme >= 0
  READ
  T = LASTKEY()
  DO CASE
    CASE T = 27
      i = 0
    CASE T = 5 .OR. T = 18
      i = 3
    OTHERWISE
      i = 5
  ENDCASE
  SET CURSOR OFF
  RETURN i
```

```
/***
* AgregaOrd() --> nAccion
*
* Realiza físicamente el alta de una nueva orden de pago,
* siempre y cuando no exista error en validación de datos.
*/
FUNCTION AgregaOrd
  LOCAL i, GPanta
```


LOCAL nLongLinea := 164

Proseguir = .T.

SalvarRec()

IF validaC()=0()

 Texto = '...', ADVERTENCIA !!! SE HAN ENCONTRADO RECURRENCIAS EN LOS '+';

 'SIGUIENTES REGISTROS '+';

 Salto+Salto+Salto+;

 'RECIBO ENCONTRADO'+ '@'+;

 'CANTIDAD'+ '@'+;

 'FECHA EN DEC'+ '@'+;

 'OFICIO'+ '@'+;

 'CLAVE CURSO'+ '@'+;

 'NOMBRE DEL CURSO'+ '@'+;

 'NOMBRE DEL ALUMNO ENCONTRADO'+;

 Salto

 cTexto = cCentraTxt(1 nLongLinea,;

 '...', ADVERTENCIA !!! SE HAN ENCONTRADO '+';

 'RECURRENCIAS EN LOS SIGUIENTES REGISTROS ');;

 Salto+Salto+Salto+;

 'RECIBO ENCONTRADO'+ ' ' +;

 'CANTIDAD'+ ' ' +;

 'FECHA EN DEC'+ ' ' +;

 'OFICIO'+ ' ' +;

 'CLAVE CURSO'+ ' ' +;

 'NOMBRE DEL CURSO'+ ' ' +;

 'NOMBRE DEL ALUMNO ENCONTRADO'+ ' ' +;

 Salto+Salto

Repetidos = 0

RevisaExt(1, LASTREC())

AutoRevisa()

IF Repetidos > 0

 GPanta = SAVESCREEN(0,0,24,79)

 HazReporte('RRR',nLongLinea)

 IF Confirmar(13) = 2

 Proseguir = .F.

 ENDIF

 RESTSCREEN(0,0,24,79,GPanta)

ENDIF

IF Proseguir

 APPEND BLANK

 GrabaOrd() // realiza el alta de una nueva orden de pago

 PonInfoOrd(RECNO())

 i = 0

ELSE

 i = 1

```
ENDIF  
ELSE  
i = 1  
ENDIF  
RETURN i
```

```
/*  
 * CambiaOrd() --> <nAccion>  
 *  
 * Realiza físicamente cambios a los datos de una orden de pago  
 * siempre y cuando no exista error en validación de los mismos.  
 * Devuelve en <nAccion> un código numérico el cual corresponde a  
 * la tecla presionada.  
 *  
 */
```

```
FUNCTION CambiaOrd
```

```
LOCAL i  
LOCAL CPanta  
LOCAL nLongLinea := 164
```

```
Proseguir = .T.
```

```
Clave = STRTRAN(Clave, ' ', '0')
```

```
SELECT B
```

```
SalirRec()
```

```
IF ValidaCveO()
```

```
  Texto = '!!! ADVERTENCIA !!! SE HAN ENCONTRADO RECURRENCIAS EN LOS '+  
  'SIGUIENTES REGISTROS '+;
```

```
  Salto+Salto+Salto+;
```

```
  'RECIBO ENCONTRADO'+ '@'+;
```

```
  'CANTIDAD'+ '@'+;
```

```
  'FECHA EN DEC'+ '@'+;
```

```
  'OFICIO'+ '@'+;
```

```
  'CLAVE CURSO'+ '@'+;
```

```
  'NOMBRE DEL CURSO'+ '@'+;
```

```
  'NOMBRE DEL ALUMNO ENCONTRADO'+;
```

```
  Salto
```

```
cTexto = cCentraTxt(1, nLongLinea, '!!! ADVERTENCIA !!! SE HAN ENCONTRADO '+  
  'RECURRENCIAS EN LOS SIGUIENTES REGISTROS '+);
```

```
  Salto+Salto+Salto+;
```

```
  'RECIBO ENCONTRADO'+ ' '+;
```

```
  'CANTIDAD'+ ' '+;
```

```
  'FECHA EN DEC'+ ' '+;
```

```
  'OFICIO'+ ' '+;
```

```
  'CLAVE CURSO'+ ' '+;
```

```
'NOMBRE DEL CURSO'+ ' '+
'NOMBRE DEL ALUMNO ENCONTRADO'+ ' '+
Salto+Salto
Repetidos = 0
DO CASE
  CASE PosAux = 1
    RevisaExt(2.LASTREC())
  CASE PosAux = LASTREC()
    RevisaExt(1.LASTREC()-1)
  OTHERWISE
    RevisaExt(1.PosAux-1)
    RevisaExt(PosAux+1.LASTREC())
ENDCASE
AutoRevisa()
IF Repetidos > 0
  GPanta = SAVESCREEN(0,0,24,79)
  HazReporte('RRR',nLongLinea)
  IF Confirmar(13) = 2
    Proseguir = F
  ENDIF
  RESTSCREEN(0,0,24,79,GPanta)
ENDIF
IF Proseguir
  Go PosAux
  GrabaOrd()
  PonInfoOrd(PECNO())
  i = 0
ELSE
  i = 1
ENDIF
ELSE
  i = 1
ENDIF
RETURN i
```

```
/***
* RevisaExt( <nInicial>, <nFinal> ) --> NIL
*
* Revisa si un recibo capturado tiene igual número
* de folio que uno ya existente. Los recibos que se
* utilizan para dicho acometido son desde <nInicial>
* hasta <nFinal> Inclusive.
*/
```

```
FUNCTION RevisaExt(nInicial, nFinal)
  Local y, i, j

  FOR y = 1 TO Tope
    FOR i = nInicial TO nFinal
      GO TO
      ReciAux = Recibos
      FOR j = 1 TO ASC(SUBSTR(ReciAux,1,1))
        IF VAL(SUBSTR(ReciAux,56*j-54,12)) = Recibo[y]
          Texto = Texto+
            STR(Recibo[y])+ '@'+
            SUBSTR(ReciAux,56*j-42,12)+'@'+ // No. de recibo
            SUBSTR(ReciAux,56*j-8,10)+'@'+ // Fecha DEC
            SUBSTR(ReciAux,56*j-30,12)+'@'+ // Oficio
            A->CveCurso+'@'+
            A->NombreCur+'@'+
            NombreAlum+
            Salto
          cTexto = cTexto+
            STR(Recibo[y])+ ' '+
            SUBSTR(ReciAux,56*j-42,12)+' '+ // No. de recibo
            SUBSTR(ReciAux,56*j-8,10)+' '+ // Fecha DEC
            SUBSTR(ReciAux,56*j-30,12)+' '+ // Oficio
            A->CveCurso+' '+
            A->NombreCur+' '+
            NombreAlum+
            Salto
          ++Repetidos
        ENDIF
      NEXT
    NEXT
  NEXT
RETURN NIL
```

/***

* AutoRevisa() --> NIL

*

* Revisa si un recibo capturado tiene igual número de
* oficio que uno ya existente, para un mismo interesado
* inscrito en el mismo curso.

*

*/

```
FUNCTION AutoRevisa
  LOCAL i, j
```

```
FOR i = 1 TO Tope-1
  FOR j = i+1 TO Tope
    IF Recibo[j] = Recibo[i]
      Texto = Texto+
        STR(Recibo[j])+ '@'+
        SUBSTR(ReciAux,56*j-42,12)+'@'+ // No. de recibo
        SUBSTR(ReciAux,56*j-8,10)+'@'+ // Fecha DEC
        SUBSTR(ReciAux,56*j-30,12)+'@'+ // Oficio
        A->CveCurso+'@'+
        A->NombreCur+'@'+
        NombreAlum
      cTexto = cTexto+
        STR(Recibo[j])+ ' '+
        SUBSTR(ReciAux,56*j-42,12)+' ' + // No. de recibo
        SUBSTR(ReciAux,56*j-8,10)+' ' + // Fecha DEC
        SUBSTR(ReciAux,56*j-30,12)+' ' + // Oficio
        A->CveCurso+' '+
        A->NombreCur+' '+
        NombreAlum
      ++Repetidos
    ENDIF
  NEXT
NEXT
RETURN NIL
```

```
/*
* ValidaCveO() --> <Continua>
*
* Revisa si la clave del curso teclada para una
* orden de pago existe. Si esta existe <Continua>
* es .T. y se prosigue con la revisión de los demás
* campos capturados.
*/
FUNCTION ValidaCveO
  LOCAL TotalCupo, ContinuaRe := .T., Posicion

  Clave = STRTRAN(Clave,' ','0')
  SELECT A
  SEEK(Clave)
  IF !FOUND()
    SELECT B
    Error(3)
```

```
Proseguir = .F.  
ChcOrden()  
ELSE  
Posicion = RECNO()  
SELECT B  
COUNT FOR CveCurso = Clave TO TotalCupo  
SELECT A  
GO Posicion  
SELECT B  
IF TotalCupo+1 > A->Cupo  
IF Confirmar(16) = 2  
Proseguir = .F.  
ContinuaRe = .F.  
ENDIF  
ENDIF  
ENDIF  
ChcOrden()  
RETURN ContinuaRe
```

/***

```
* ChcOrden() --> NIL  
*  
* Revisa si los todos los datos capturados de las ordenes  
* de pago ( a excepción del de clave ) son validos.  
*  
*/
```

```
FUNCTION ChcOrden()  
LOCAL i  
  
IF Val(CveCarr) < 10  
CveCarr = '00'+STRTRAN(STR(VAL(CveCarr)),' ','')  
ELSE  
IF VAL(CveCarr) >= 10 .AND. VAL(CveCarr) < 100  
CveCarr = '0'+STRTRAN(STR(VAL(CveCarr)),' ','')  
ENDIF  
ENDIF  
IF CveCarr != '000'  
SELECT D  
LOCATE FOR CveProfesi = CveCarr  
IF !FOUND()  
Error(16)  
Proseguir = .F.  
ENDIF  
ENDIF
```

```
SELECT B
IF CveP > '2'
  Error(17)
  Proseguir = .F.
ENDIF
IF Cost < 0
  Error(13)
  Proseguir = .F.
ENDIF
IF Tope = 0
  Error(34)
  Proseguir = .F.
ELSE
  FOR i = 1 TO Tope
    IF FechaC[i] > FechaD[i]
      Error(22)
      Proseguir = .F.
    ENDIF
    IF Cantidad[i] < 0
      Error(23)
      Proseguir = .F.
    ENDIF
  NEXT
ENDIF
RETURN NIL
```

/***

* GrabaOrd() --> NIL

*

* Graba físicamente en el disco de trabajo las altas
* o modificaciones realizadas de ordenes de pago.

*

*/

FUNCTION GrabaOrd

REPLACE CveCurso WITH Clave, NombreAlum WITH Nombre, RFC WITH
Registro,;

Telefono WITH Tele, CveProcede WITH CveP, Procedenci WITH pr,;

CveCarrera WITH CveCarr, Cuenta WITH Cuen,;

CveCondi WITH CveCond, Semestre WITH Seme,;

CostoCur WITH Cost, Recibos WITH CadRec, Observa WITH Memorandum

RETURN NIL

```
/*  
* ConsuOrdG() --> NIL  
* Consultas globales de ordenes de pago.  
*/  
FUNCTION ConsuOrdG
```

```
  SELECT B  
  IF LASTREC() = 0  
    Error(6)  
  ELSE  
    PantaBases('<< CONSULTAS GLOBALES DE ORDENES DE PAGO >>')  
    HazConsuOr()  
  ENDIF  
RETURN NIL
```

```
/*  
* ConsuOrdI() --> NIL  
* Consultas por cadena incluida.  
*/  
FUNCTION ConsuOrdI
```

```
  SELECT B  
  IF LASTREC() = 0  
    Error(6)  
  ELSE  
    PantaBases('<< CONSULTAS DE ORDENES DE PAGO POR CADENA INCLUIDA >>')  
    IF PalabraInc(2)  
      HazConsuOr()  
      SET FILTER TO  
    ENDIF  
  ENDIF  
RETURN NIL
```



```
/*  
* HazConsuOr --> NIL  
*  
* Realiza las consultas de ordenes de pago  
* ya sea por clave o por cadena incluida.  
*  
*/  
FUNCTION HazConsuOr  
  LOCAL Continuar := .T.  
  
  CamposOrd()  
  OK2()  
  SET KEY 28 TO  
  Continuar = .T.  
  DO WHILE Continuar  
    PantaOrden()  
    GO TOP  
    CajaCampos()  
    SETCCLOR('15/2.15/0...')  
    SET KEY 28 TO  
    DBEDIT('1 c1.12 c2.m1 " m3.m2,;  
      CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))  
    SET KEY 28 TO AYUDA  
    RESTSCREEN(0 0.24.79 G)  
    IF LASTKEY() != 27  
      PonInfoOrd(RECNO())  
      Memorandum = Observa  
      CargarRec()  
      ContrConOr(18.25,.F.)  
    ENDIF  
    IF Confirmar(4) = 2  
      Continuar = .F.  
    ENDIF  
  ENDDO  
  SET KEY 28 TO Ayuda  
  RETURN NIL
```

```
/*  
* ConsuOrdC() --> NIL  
*  
* Realiza consultas por clave de ordenes de pago.  
*  
*/  
FUNCTION ConsuOrdC
```

LOCAL Opcion, Terminar

```
SELECT B
Pantabases('<< CONSULTAS POR CLAVE DE ORDENES DE PAGO >>')
OK2()
IF LASREC() = 0
  Error(1)
ELSE
  Terminar = F.
  DO WHILE !Terminar
    Pantorden()
    Pantclave(2,")
    IF LASTKEY() != 27
      SEEK(Clave)
      IF FOUND()
        SKIP
        IF CveCurso = Clave
          CajaCampos()
          SETCOLOR(15/2,15/0,..)
          SET FILTER TO CveCurso = Clave
          CamposOrd()
          GO TOP
          SET KEY 28 TO
          DBEDIT(1,c1,c2,m1,"m3,m2,;
            CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
          SET KEY 28 TO AYUDA
          RESTSCREEN(0,0,24,79,G)
          IF LASTKEY() != 27
            PonInfoOrd(RECNO())
            Memorandum = Observa
            CargarRec()
            ContrConOr(18,25,.F.)
          ENDIF
          SET FILTER TO
        ELSE
          SKIP -1
          PonInfoOrd(RECNO())
          Memorandum = Observa
          CargarRec()
          ContrConOr(18,25,.F.)
        ENDIF
      ELSE
        Error(3)
      ENDIF
    ENDIF
  IF Confirmar(4) = 2
```

```
Terminar = .T.  
ENDIF  
ENDDO  
ENDIF  
RETURN NIL
```

```
/*  
* CamposOrd() --> NIL  
*  
* Asigna los valores necesarios para la  
* función DBEDIT para Ordenes de pago.  
*  
*/  
FUNCTION CamposOrd  
LOCAL i  
  
m1 = ARRAY(FCOUNT())  
m2 = ARRAY(FCOUNT())  
m3 = ARRAY(FCOUNT())  
FOR i = 1 TO LEN(m1)  
m1[i] = FIELDNAME(i)  
NEXT  
m2 = { ' Clave del curso',' Nombre del alumno',' Teléfono',' RFC ';;  
' Clave de procedencia',' Procedencia',' Clave de la carrera';;  
' Número de cuenta',' Clave de la condición ',' Semestre';;  
' Costo del curso',' Recibos',' Observaciones' }  
m3[11] = '$999,999.99'  
RETURN NIL
```

```
/*  
* ContrConOr( <nRen>, <nCol>, <IVerdad> ) --> NIL  
*  
* Esta función se encarga de controlar la pantalla  
* de consultas de ordenes de pago. Aquí <nRen> y <nCol>  
* son las coordenadas en la cual está ubicada la opción  
* "Memorándum" y <IVerdad> en caso de ser .F., indica  
* que el campo memorándum no se puede modificar de lo  
* contrario este si es modificable.  
*  
*/  
FUNCTION ContrConOr(nRen, nCol, IVerdad)  
LOCAL i, Saite
```

```
Salte = READEXIT(.T.)
i = 4
DO WHILE i != 0
  DO CASE
    CASE i = 1
      i = 4
    CASE i = 2
      i = ControlMemo(nRen,nCol,IVerdad)
    CASE i = 3
      i = ControlOk()
    CASE i = 4
      i = ControlRec(.T.)
    CASE i = 5
      i = 3
  ENDCASE
ENDDO
READEXIT(Salte)
RETURN NIL
```

```
/*
* ControlRec( <IEsCon> ) -> nAccion
*
* Controla la apertura de la pantalla de recibos
* de ordenes de pago. Si <IEsCon> es .T. indica que
* la pantalla de edición de recibos a visualizar es
* la de bajas o bien la de consultas de lo contrario
* indica que la pantalla es la de altas o bien la de
* modificaciones.
*/
```

```
FUNCTION ControlRec ( IEsCon )
LOCAL Tecla, i
Mod_Color(17,25,17,31,7*16+15)
Tecla = 0
DO WHILE Tecla != 13 .AND. Tecla != 27
  Tecla = INKEY(0)
  DO CASE
    CASE Tecla = 13
      Mod_Color(17,25,17,31,15)
      IF IEsCon
        EditaRecCo()
      ELSE
        EditaRec()
    
```

```
    ENDIF
    i = 2
    CASE Tecla = 27
    i = 0
    Mod_Color(17,25,17,31,15)
    CASE Tecla = 28
    VarAyuda = 9
    AyudaMod()
    CASE Tecla = 24 .OR. Tecla = 3
    i = 2
    Mod_Color(17,25,17,31,15)
    Tecla = 27
    CASE Tecla = 5 .OR. Tecla = 18
    i = 5
    Mod_Color(17,25,17,31,15)
    Tecla = 27
  ENDCASE
  ENDDO
  RETURN i
```

```
/*
* EditaRec() --> NIL
*
* Controla las alternativas de la pantalla
* de edición de recibos en su modalidad de
* altas y modificaciones de ordenes de pago.
*
*/
```

```
FUNCTION EditaRec
  LOCAL i, Tecla

  GAux = SAVESCREEN( 0,0,24,79 )
  PanRec()
  OpcionRecA()
  SET CONFIRM ON
  i = 1
  Tecla = 0
  READEXIT(.F.)
  DO WHILE Tecla != 27
    SETCOLOR('15/2,...')
    @ 3*i+3, 02 SAY CHR(16)
    @ 3*i+3, 14 SAY CHR(17)
    Mod_Color(3*i+3, 3, 3*i+3, 13, 2*16+15)
    LimpRec()
```

```
Tecla = INKEY(0)
DO CASE
  CASE Tecla = 27
    SETCOLOR('0/2,...')
    @ 3*i+3, 02 SAY ''
    @ 3*i+3, 14 SAY ''
    Mod_Color (3*i+3, 3, 3*i+3, 13, 2*16+0)
  CASE Tecla = 28
    VarAyuda = 10
    AyudaMod()
  CASE Tecla = 13
    Boton(3*i+3, 2, 14)
    @ 3*i+3, 02 SAY ''
    @ 3*i+3, 14 SAY ''
    Mod_Color (3*i+3, 3, 3*i+3, 13, 2*16+0)
  DO CASE
    CASE Tope = 0
      DO CASE
        CASE i = 1
          Tecla = Siguiente()
        CASE i = 4
          Tecla = 27
      ENDCASE
    CASE Tope = 100
      DO CASE
        CASE i = 2
          Tecla = Anterior()
        CASE i = 3
          BorrarRec()
        CASE i = 4
          Tecla = 27
      ENDCASE
    OTHERWISE
      DO CASE
        CASE i = 1
          Tecla = Siguiente()
        CASE i = 2
          Tecla = Anterior()
        CASE i = 3
          BorrarRec()
        CASE i = 4
          Tecla = 27
      ENDCASE
  ENDCASE
CASE Tecla = 5
  SETCOLOR('0/2,...')
```

```
@ 3*i+3, 02 SAY ''
@ 3*i+3, 14 SAY ''
Mod_Color (3*i+3, 2, 3*i+3, 14, 2*16+0)
IF i > 1
  --i
ELSE
  i = 4
ENDIF
CASE Tecla = 24
  SETCOLOR('0/2,,,')
  @ 3*i+3, 02 SAY ''
  @ 3*i+3, 14 SAY ''
  Mod_Color (3*i+3, 2, 3*i+3, 14, 2*16+0)
  IF i < 4
    ++i
  ELSE
    i = 1
  ENDIF
ENDCASE
ENDDO
IF Tope > 0
  Actual = 1
ENDIF
RESTSCREEN( 0,0,24,79, GAux)
SET CONFIRM OFF
READEXIT(.T.)
RETURN NIL
```

/***

* EditaRecCo() --> NIL

*

* Controla las alternativas de la pantalla
* de edición de recibos en su modalidad de
* bajas y consultas de ordenes de pago.

*

*/

FUNCTION EditaRecCo

LOCAL i, Tecla

GAux = SAVESCREEN(0,0,24,79)

PanRec()

OpcionRecC()

SET CONFIRM ON

i = 1

```
Tecla = 0
READEXIT(.F.)
DO WHILE Tecla != 27
  SETCOLOR('15/2,...')
  @ 4*i+2, 02 SAY CHR(16)
  @ 4*i+2, 14 SAY CHR(17)
  Mod_Color(4*i+2, 3, 4*i+2, 13, 2*16+15)
  LimpRec()
  Tecla = INKEY(0)
DO CASE
  CASE Tecla = 27
    SETCOLOR('0/2,...')
    @ 4*i+2, 02 SAY ''
    @ 4*i+2, 14 SAY ''
    Mod_Color(4*i+2, 3, 4*i+2, 13, 2*16+0)
  CASE Tecla = 28
    VarAyuda = 12
    AyudaMod()
  CASE Tecla = 13
    Boton(4*i+2, 2, 14)
    @ 4*i+2, 02 SAY ''
    @ 4*i+2, 14 SAY ''
    Mod_Color(4*i+2, 3, 4*i+2, 13, 2*16+0)
  DO CASE
    CASE Tope = 0
      IF i = 3
        Tecla = 27
      ENDIF
    CASE Tope = 1
      DO CASE
        CASE i = 1
          SiguienteC()
        CASE i = 3
          Tecla = 27
        ENDCASE
    CASE Tope = 100
      DO CASE
        CASE i = 2
          AnteriorC()
        CASE i = 3
          Tecla = 27
        ENDCASE
    OTHERWISE
      DO CASE
        CASE i = 1
          SiguienteC()
```



```
        CASE i = 2
            AnteriorC()
        CASE i = 3
            Tecla = 27
        ENDCASE
    ENDCASE
CASE Tecla = 5
    SETCOLOR('0/2,...')
    @ 4*i+2, 02 SAY ''
    @ 4*i+2, 14 SAY ''
    Mod_Color (4*i+2, 2, 4*i+2, 14, 2*16+0)
    IF i > 1
        --i
    ELSE
        i = 3
    ENDIF
CASE Tecla = 24
    SETCOLOR('0/2,...')
    @ 4*i+2, 02 SAY ''
    @ 4*i+2, 14 SAY ''
    Mod_Color (4*i+2, 2, 4*i+2, 14, 2*16+0)
    IF i < 3
        ++i
    ELSE
        i = 1
    ENDIF
ENDCASE
ENDDO
RESTSCREEN( 0,0,24,79, GAux)
IF Tope > 0
    Actual = 1
ENDIF
SET CONFIRM OFF
READEXIT(.T.)
RETURN NIL
```

```
/***
* PanRec() --> NIL
*
* Coloca la pantalla para edición de
* ordenes de pago pero sin opciones.
*
*/
FUNCTION PanRec
```

```
SETCOLOR('15/3,...')
@ 2,0,20,77 BOX Marco7+' '
RamaH(4,0,77,RamaH6)
SombraMarco(2,0,20,77)
@ 5,18,17,75 BOX Marco7
RamaH(12,18,75,RamaH6)
SETCOLOR('1/3,...')
@ 06,32 SAY 'Datos para el recibo : '
@ 08,21 SAY 'Número de recibo          : '
@ 09,21 SAY 'Importe del recibo         : $'
@ 10,21 SAY 'Fecha de pago en caja     : '
@ 11,21 SAY 'Fecha de entrega en DEC   : '
@ 13,21 SAY 'Costo del curso para este alumno : $'
@ 13,61 SAY Cost PICT '999,999,999,99'
@ 14,21 SAY 'Suma parcial de los recibos pagados : $'
@ 15,21 SAY 'Saldo ( + = Adeudo y - = A favor ) : $'
@ 16,21 SAY 'Número de oficio         : '
SETCOLOR('0/3,...')
@ 3,22 SAY CAPTURA DE RECIBOS'
@ 18,4 SAY CHR(17)+CHR(196)+CHR(196)+CHR(217)+' = Entrar a captura'+;
' Pg/Dn y Pg/Up = Salir de captura ESC = Salir'
RETURN NIL
```

```
/***
* OpcionRecA() --> NIL
*
* Coloca las opciones para del menú de edición
* de recibos, pero en su modalidad de altas y
* modificaciones.
*
*/
FUNCTION OpcionRecA
LOCAL i

SETCOLOR('0/2,...')
@ 06,02 SAY ' Siguiente '
@ 09,02 SAY ' Anterior '
@ 12,02 SAY ' Borrar '
@ 15,02 SAY ' Salir '
FOR i = 1 TO 4
  SombraMensaje(3*i+3,02,14,'0/3,...')
NEXT
RETURN NIL
```

```
/*  
* OpcionRecC() --> NIL  
*  
* Coloca las opciones para del menú de edición  
* de recibos, pero en su modalidad de bajas y  
* consultas.  
*  
*/  
FUNCTION OpcionRecC  
  LOCAL i  
  
  SETCOLOR('0/2,...')  
  @ 06,02 SAY ' Siguiete '  
  @ 10,02 SAY ' Anterior '  
  @ 14,02 SAY ' Salir '  
  FOR i = 1 TO 3  
    SombraMensaje(4*i+2,02,14,'0/3,...')  
  NEXT  
RETURN NIL
```

```
/*  
* Siguiete() --> NIL  
*  
* Permite editar los recibos del registro vigente y  
* sólo opera en altas y modificaciones de ordenes de  
* pago. Esta edición es siempre hacia adelante. Se  
* permiten hasta 100 registros inclusive.  
*  
*/  
FUNCTION Siguiete  
  LOCAL Dato1, Dato2, Fecha1, Fecha2  
  
  SET CURSOR ON  
  IF Actual < 100  
    IF Tope = Actual  
      ++Tope  
      ++Actual  
      NoDeRec()  
      Dato1 = 0  
      Dato2 = 0  
      Fecha1 = CTOD("")  
      Fecha2 = CTOD("")  
      @ 08,61 GET Dato1 PICT '999999999999'  
      @ 09,61 GET Dato2 PICT '999,999,999.99'
```

```
@ 10,61 GET Fecha1
@ 11,61 GET Fecha2
READ
AADD(Recibo, Dato1)
AADD(Cantidad, Dato2)
AADD(FechaC, Fecha1)
AADD(FechaD, Fecha2)
AADD(Oficio, SPACE(12))
ELSE
  ++Actual
  NoDeRec()
  @ 08,61 GET Recibo[Actual] PICT '999999999999'
  @ 09,61 GET Cantidad[Actual] PICT '999,999,999.99'
  @ 10,61 GET FechaC[Actual]
  @ 11,61 GET FechaD[Actual]
  READ
ENDIF
ENDIF
SET CURSOR OFF
RETURN LASTKEY()
```

/****

- * Anterior() -> NIL
- *
- * Permite editar los recibos del registro vigente y
- * sólo opera en altas y modificaciones de ordenes de
- * pago. Esta edición es siempre hacia atrás.
- *
- * /

FUNCTION Anterior

```
IF Actual > 1
  --Actual
ENDIF
NoDeRec()
SET CURSOR ON
@ 08,61 GET Recibo[Actual] PICT '999999999999'
@ 09,61 GET Cantidad[Actual] PICT '999,999,999.99'
@ 10,61 GET FechaC[Actual]
@ 11,61 GET FechaD[Actual]
READ
SET CURSOR OFF
RETURN LASTKEY()
```

```
/***
* SiguienteC() --> NIL
*
* Permite editar los recibos del registro vigente y
* sólo opera en bajas y consultas de ordenes de pago.
* Esta edición es siempre hacia adelante.
*
*/
FUNCTION SiguienteC

  IF Actual < Tope
    ++Actual
    NoDeRec()
    @ 08,61 SAY Recibo[Actual] PICT '99999999999999'
    @ 09,61 SAY Cantidad[Actual] PICT '999,999,999.99'
    @ 10,61 SAY FechaC[Actual]
    @ 11,61 SAY FechaD[Actual]
  ENDIF
RETURN NIL

/***
* AnteriorC() --> NIL
*
* Permite editar los recibos del registro vigente y
* sólo opera en bajas y consultas de ordenes de pago.
* Esta edición es siempre hacia atrás.
*
*/
FUNCTION AnteriorC

  IF Actual > 1
    --Actual
    NoDeRec()
    @ 08,61 SAY Recibo[Actual] PICT '99999999999999'
    @ 09,61 SAY Cantidad[Actual] PICT '999,999,999.99'
    @ 10,61 SAY FechaC[Actual]
    @ 11,61 SAY FechaD[Actual]
  ENDIF
RETURN NIL
```

```
/***  
* SalvarRec() --> NIL  
*  
* Salva los contenidos de los recibos  
* capturados de las ordenes de pago del  
* registro actual en la variable <CadRec>.  
*  
*/  
FUNCTION SalvarRec  
  LOCAL i, VAux  
  CadRec = CHR(Tope)  
  FOR i = 1 TO Tope  
    VAux = LTRIM(STR(Recibo[i]))  
    VAux = REPLICATE(' ',12-LEN(VAux))+VAux  
    CadRec = CadRec+VAux  
    IF Cantidad[i] = 0  
      CadRec = CadRec+'    0.00'  
    ELSE  
      VAux = LTRIM(STR(Cantidad[i]))  
      VAux = REPLICATE(' ',12-LEN(VAux))+VAux  
      CadRec = CadRec+VAux  
    ENDIF  
    CadRec = CadRec+Oficio[i]+DTC(FechaC[i])+DTC(FechaD[i])  
  NEXT  
RETURN NIL
```

```
/***  
* BorrarRec() --> NIL  
*  
* Elimina un recibo del registro de ordenes  
* de pago vigente.  
*  
*/  
FUNCTION BorrarRec  
  ADEL(Recibo, Actual)  
  ADEL(Cantidad, Actual)  
  ADEL(FechaC, Actual)  
  ADEL(FechaD, Actual)  
  ADEL(Oficio, Actual)  
  IF Tope = Actual  
    --Actual  
  ENDIF  
  --Tope  
  ASIZE(Recibo, Tope)
```

```
ASIZE(Cantidad, Tope)
ASIZE(FechaC, Tope)
ASIZE(FechaD, Tope)
ASIZE(Oficio, Tope)
RETURN NIL
```

```
/*
* LimpRec() --> NIL
*
* Limpia el área de edición de los recibos
* de pago del registro de ordenes de pago
* vigente.
*/
FUNCTION LimpRec
  LOCAL ColoresViejos := SETCOLOR()

  SETCOLOR('15/0,...')
  IF Tope = 0
    @ 08,61 SAY SPACE(12)
    @ 09,61 SAY SPACE(14)
    @ 10,61 SAY CTOD("")
    @ 11,61 SAY CTOD("")
    @ 16,61 SAY SPACE(12)
  ELSE
    @ 08,61 SAY Recibo[Actual] PICT '999999999999'
    @ 09,61 SAY Cantidad[Actual] PICT '999,999,999.99'
    @ 10,61 SAY FechaC[Actual]
    @ 11,61 SAY FechaD[Actual]
    @ 16,61 SAY Oficio[Actual]
  ENDIF
  NoDeRec()
  SETCOLOR(ColoresViejos)
  Adicional()
RETURN NIL
```

```
/*
* Adicional() --> NIL
*
* Realiza la suma de los recibos existentes
* en la orden de pago del registro vigente.
*/
```

```
FUNCTION Adicional
  LOCAL i, ColoresViejos := SETCOLOR(), Suma

  SETCOLOR('15/0,....')
  Suma = 0
  FOR i = 1 TO Tope
    Suma = Suma + Cantidad[i]
  NEXT
  @ 14,61 SAY Suma PICT '999,999,999.99'
  @ 15,61 SAY Cost-Suma PICT '999,999,999.99'
  SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/**
 * NoDeRec() --> NIL
 *
 * Coloca el número de recibo vigente y el total
 * de recibos existentes en el registro de ordenes
 * de pago actual.
 */
FUNCTION NoDeRec
  LOCAL ColoresViejos := SETCOLOR()

  SETCOLOR('15/0,....')
  @ 6,56 SAY RIGHT(STR(Actual),3)+'/'+RIGHT(STR(Tope),3)
  SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/**
 * PonInfoOrd ( <nPosReg> ) --> NIL
 *
 * Coloca la información del registro <nPosReg> de la
 * base de datos de ordenes de pago, en la pantalla.
 */
FUNCTION PonInfoOrd ( nPosReg )

  GO nPosReg
  SETCOLOR('15/0,....')
  @ 4,25 SAY CveCurso
  @ 5,25 SAY NombreAlum
```



```
@ 6,25 SAY RFC
@ 7,25 SAY Telefono
@ 8,25 SAY CveProcede
@ 8,36 SAY Procedenci
@ 9,25 SAY CveCarrera
@ 10,25 SAY Cuenta
@ 11,25 SAY CveCondici
@ 12,25 SAY Semestre
@ 16,25 SAY CostoCur PICT '9999999.99'
Cost = CostoCur
CveCarr = CveCarrera
CveCond = CveCondici
DatosOrdRe()
RETURN NIL
```

```
/*
* DatosOrdRe() --> NIL
*
* Coloca el periodo de exposición, nombre del
* expositor, salón, nombre del curso, nombre de
* la carrera y nombre de la condición que
* corresponden a los datos del registro actual
* de la orden de pago vigente.
*
*/
FUNCTION DatosOrdRe
```

```
SETCOLOR('14/0,...')
@ 13,25 SAY A->FechaExpo1
@ 13,38 SAY A->FechaExpo2
@ 14,25 SAY C->NombreExpo
@ 15,25 SAY A->Salon
@ 4,36 SAY A->NombreCur
IF CveCarr = '000'
  @ 9,31 SAY 'NINGUNA'+SPACE(38)
ELSE
  SELECT D
  LOCATE FOR CveProfes = CveCarr
  @ 9,31 SAY NombreCarr
  SELECT B
ENDIF
DO CASE
  CASE CveCond = '0'
    @ 11,36 SAY 'PUBLICO EN GENERAL'
```

```
CASE CveCond = '1'  
  @ 11,36 SAY 'ALUMNO'  
CASE CveCond = '2'  
  @ 11,36 SAY 'ACADEMICO'  
CASE CveCond = '3'  
  @ 11,36 SAY 'ADMINISTRATIVO'  
CASE CveCond = '4'  
  @ 11,36 SAY 'MANTENIMIENTO'  
CASE CveCond = '5'  
  @ 11,36 SAY 'EGRESADO'  
ENDCASE  
RETURN NIL
```

```
/***  
* PantaOrden() -> NIL  
*  
* Coloca la pantalla de ordenes de pago.  
*  
*/  
FUNCTION PantaOrden
```

```
SETCOLOR('15/1,...')  
@ 2,2,21,77 BOX Marco7+ '  
@ 4,4 SAY 'Clave del curso : '  
@ 5,4 SAY 'Nombre del alumno : '  
@ 6,4 SAY 'RFC : '  
@ 7,4 SAY 'Teléfono : '  
@ 8,4 SAY 'Clave procedencia : '  
@ 9,4 SAY 'Clave carrera : '  
@ 10,4 SAY 'Número de cuenta : '  
@ 11,4 SAY 'Condición : '  
@ 12,4 SAY 'Semestre : '  
@ 13,4 SAY 'Periodo : / / - / / '  
@ 14,4 SAY 'Expositor : '  
@ 15,4 SAY 'Salón : '  
@ 16,4 SAY 'Costo del curso : $ '  
@ 17,4 SAY 'Número de recibo : Recibos '  
@ 18,4 SAY 'Observaciones : Memorándum '  
RETURN NIL
```

```
-----/
```

```
*****  
* Este es el módulo de expositores *  
*****
```

```
/**  
* ConsuExpo() --> NIL  
*  
* Permite la selección del tipo de consultas de  
* expositores, ya sean globales o por clave.  
**/
```

```
FUNCTION ConsuExpo
```

```
  PonPanta(3)  
  nOpcion3 = Seleccion(3,nOpcion3)  
  DO CASE  
    CASE nOpcion3 = 1  
      ConsuExpG()  
      FinConsu = .T.  
    CASE nOpcion3 = 2  
      ConsuExpC()  
      FinConsu = .T.  
    CASE nOpcion3 = 3  
      ConsuExpI()  
      FinConsu = .T.  
    CASE nOpcion3 = 4  
      nOpcion3 = 1  
  ENDCASE  
  RETURN NIL
```

```
/**  
* AltasExpo() --> NIL  
*  
* Da de alta un nuevo expositor.  
**/  
FUNCTION AltasExpo  
  LOCAL Terminar  
  
  SELECT C  
  PantaBases('<< ALTAS DE EXPOSITORES >>')
```

```
PonConCan()
PantaExpo()
Terminar = .F.
DO WHILE !Terminar
  BlancosExp()
  ControlAIE(20,31,.T.)
  IF Confirmar(2) = 2
    Terminar = .T.
  ENDIF
ENDDO
RETURN NIL
```

```
/*
* BajasExpo() --> NIL
*
* Da de baja un expositor existente.
*
*/
FUNCTION BajasExpo
  LOCAL Opcion, Terminar

  SELECT C
  PantaBases('<< BAJAS DE EXPOSITORES >>')
  PonConCan()
  Terminar = .F.
  DO WHILE !Terminar
    PantaExpo()
    IF LASTREC() = 0
      Error(7)
      Terminar = .T.
    ELSE
      PantaClave(3,")
      IF LASTKEY() != 27
        SEEK(Clave)
        IF FOUND()
          SKIP
          IF CveExpo = Clave
            CajaCampos()
            SETCOLOR('15/2,15/0,,,')
            SET FILTER TO CveExpo = Clave
            CamposExp()
            GO TOP
            SET KEY 28 TO
            DBEDIT(I1,c1,I2,c2,m1,,m2,;
```

```
        CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
SET KEY 28 TO AYUDA
RESTSCREEN(0,0,24,79,G)
IF LASTKEY() != 27
    PonInfoExp(RECNO())
    Memorandum = Observa
    ControlBaE(20,31,.F.)
ENDIF
SET FILTER TO
ELSE
SKIP -1
PonInfoExp(RECNO())
Memorandum = Observa
ControlBaE(20,31,.F.)
ENDIF
ELSE
Error(4)
ENDIF
ENDIF
IF Confirmar(3) = 2
    Terminar = .T.
ENDIF
ENDDO
RETURN NIL
```

/***

- * ControlBaE(<nRen>, <nCol>, <IVerdad>) --> NIL
- *
- * Controla la apertura de las opciones de la pantalla de
- * bajas de expositores. Aquí <nRen> y <nCol> son las coordenadas
- * en la cual está ubicada la opción "Memorándum" y <IVerdad>
- * en caso de ser .F., indica que el campo memorándum no se
- * puede modificar de lo contrario este si es modificable.
- *

*/

```
FUNCTION ControlBaE(nRen, nCol, IVerdad)
LOCAL i

i = 2
DO WHILE i != 0
DO CASE
CASE i = 1
i = 2
```

```
CASE i = 2
  i = ControlMemo(nRen,nCol,IVerdad)
CASE i = 3
  i = ConCan()
CASE i = 4
  i = 2
CASE i = 5
  i = 3
CASE i = 6
  PosAux = RECNO()
  IF ChcBaseCur(47)
    Go PosAux
    DELETE // Elimina físicamente el registro de un expositor
    PACK
  ENDIF
  i = 0
ENDCASE
ENDDO
RETURN NIL
```

/*

```
* ModiExpo() --> NIL
*
* Modifica los datos de un expositor ya existente
*
*/
```

```
FUNCTION ModiExpo
  LOCAL Opcion, Terminar

  SELECT C
  PantaBases('<< MODIFICACIONES DE EXPOSITORES >>')
  PonConCan()
  IF LASTREC() = 0
    Error(7)
  ELSE
    Terminar = .F.
    DO WHILE !Terminar
      PantaExpo()
      PantaClave(3,")
      IF LASTKEY() != 27
        SEEK(Clave)
        IF FOUND()
          SKIP
          IF CveExpo = Clave
```

```
CajaCampos()
SETCOLOR("15/2,15/0,..")
SET FILTER TO CveExpo = Clave
CamposExp()
GO TOP
SET KEY 28 TO
DBEDIT(11,c1,l2,c2,m1,"..m2,;
      CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
SET KEY 28 TO AYUDA
RESTSCREEN(0,0,24,79,G)
IF LASTKEY() != 27
  PonInfoExp(RECNO())
  LlenaCaExp()
  PosAux = RECNO()
  Memorandum = Observa
  ControlMoE(20,31,..T.)
ENDIF
SET FILTER TO
ELSE
  SKIP -1
  PonInfoExp(RECNO())
  LlenaCaExp()
  PosAux = RECNO()
  Memorandum = Observa
  ControlMoE(20,31,..T.)
ENDIF
ELSE
  Error(4)
ENDIF
ENDIF
IF Confirmar(5) = 2
  Terminar = .T.
ENDIF
ENDDO
ENDIF
RETURN NIL
```

```
/***
* ControlAIE( <nRen>, <nCol>, <IVerdad> ) --> NIL
*
* Controla la apertura de las opciones de la pantalla de
* altas de expositores. Aquí <nRen> y <nCol> son las coordenadas
* en la cual está ubicada la opción "Memorándum" y <IVerdad>
* en caso de ser .F., indica que el campo memorándum no se
* puede modificar de lo contrario este si es modificable.
*
*/
FUNCTION ControlAIE(nRen, nCol, IVerdad)
  LOCAL i, Salte

  Salte = READEXIT(.T.)
  i = 1
  DO WHILE i != 0
    DO CASE
      CASE i = 1
        i = CapturaExp()
      CASE i = 2
        i = ControlMemo(nRen,nCol,IVerdad)
      CASE i = 3
        i = ConCan()
      CASE i = 4
        i = 1
      CASE i = 5
        i = 2
      CASE i = 6
        i = AgregaExp()
    ENDCASE
  ENDDO
  READEXIT(Salte)
RETURN NIL

/***
* ControlMoE( <nRen>, <nCol>, <IVerdad> ) --> NIL
*
* Controla la apertura de las opciones de la pantalla de
* cambios de expositores. Aquí <nRen> y <nCol> son las coordenadas
* en la cual está ubicada la opción "Memorándum" y <IVerdad>
* en caso de ser .F., indica que el campo memorándum no se
* puede modificar de lo contrario este si es modificable.
*
*/
```



```
FUNCTION ControlMoE(nRen, nCol, IVerdad)
  LOCAL i, Salte

  Salte = READEXIT(.T.)
  i = 1
  DO WHILE i != 0
    DO CASE
      CASE i = 1
        i = CapturaExp()
      CASE i = 2
        i = ControlMemo(nRen,nCol,IVerdad)
      CASE i = 3
        i = ConCan()
      CASE i = 4
        i = 1
      CASE i = 5
        i = 2
      CASE i = 6
        i = CambiaExp()
    ENDCASE
  ENDDO
  READEXIT(Salte)
RETURN NIL
```

```
/***
* BlancosExp() --> NIL
*
* Asigna a las variables auxiliares para dar de alta
* los datos de un expositor, su valor por defecto.
*
*/
FUNCTION BlancosExp

  Clave = '000'
  Nombre = SPACE(40)
  Registro = SPACE(13)
  Domi = SPACE(40)
  Tele = 0
  Adsc = SPACE(40)
  Titu = SPACE(40)
  Cedu = SPACE(10)
  Fec1 = CTOD("")
  Memorandum = ""
RETURN NIL
```

```
/*  
* LlanaCaExp() --> NIL  
*  
* Asigna los contenidos del expositor seleccionado, a las variables  
* auxiliares para realizar los cambios requeridos.  
*  
*/
```

```
FUNCTION LlanaCaExp
```

```
    Clave = CveExpo  
    CveAux = Clave  
    Nombre = NombreExpo  
    Registro = RFC  
    Domi = Domicilio  
    Tele = Telefono  
    Adsc = Adscripcio  
    Titu = Titulo  
    Cedu = Cedula  
    Fec1 = Expedicion  
    Memorandum = Observa  
RETURN NIL
```

```
/*  
* CapturaExp() --> <nAccion>  
*  
* Permite capturar los datos para un expositor nuevo  
* o ya existente. Devuelve en <nAccion> el número de  
* código correspondiente según la tecla presionada.  
*  
*/
```

```
FUNCTION CapturaExp
```

```
    LOCAL i
```

```
    SETCOLOR('07...15/0')  
    SET CURSOR ON  
    @ 2,31 GET Clave PICT '9999999999'  
    @ 4,31 GET Nombre PICT '@!'  
    @ 6,31 GET Registro PICT '!!!!999999!!!'  
    @ 8,31 GET Domi PICT '@!'  
    @ 10,31 GET Tele PICT '999999999999' VALID Tele >= 0  
    @ 12,31 GET Adsc PICT '@!'  
    @ 14,31 GET Titu PICT '@!'  
    @ 16,31 GET Cedu PICT '@!'  
    @ 18,31 GET Fec1
```

```
READ
T = LASTKEY()
DO CASE
  CASE T = 27
    i = 0
  CASE T = 18 .OR. T = 5
    i = 3
  OTHERWISE
    i = 2
ENDCASE
SET CURSOR OFF
RETURN i
```

```
/*
* AgregaExp() --> <nAccion>
*
* Realiza físicamente el alta de un nuevo expositor
* siempre y cuando no exista error en validación de
* datos. Devuelve en <nAccion> el número de código
* correspondiente según la tecla presionada.
*/
```

```
FUNCTION AgregaExp
  LOCAL i

  Proseguir = .T.
  Clave = STRTRAN(Clave,' ','0')
  ChecaCveEx()
  IF Proseguir
    APPEND BLANK
    GrabaExp() // Realiza el alta de un nuevo expositor.
    i = 0
  ELSE
    i = 1
  ENDIF
RETURN i
```

```
/*  
* CambiaExp() --> nAccion  
*  
* Realiza físicamente cambios a los datos de un  
* expositor siempre y cuando no exista error en la  
* validación de los mismos. Devuelve en <nAccion>  
* el número de código correspondiente según la  
* tecla presionada.  
*/
```

```
FUNCTION CambiaExp  
  LOCAL i
```

```
  Proseguir = .T.  
  Clave = STRTRAN(Clave,',';0')  
  ChcCveMoEx()  
  i = 1  
  IF Proseguir  
    Go PosAux  
    IF Clave != CveAux  
      IF ChcBaseCur(48)  
        Go PosAux  
        GrabaExp()  
        i = 0  
      ELSE  
        Clave = CveAux  
      ENDIF  
    ELSE  
      Go PosAux  
      GrabaExp()  
      i = 0  
    ENDIF  
  ENDIF  
  RETURN i
```

```
/*  
* ChcBaseCur( <IError> ) -> <IContinua>  
*  
* Revisa la posible existencia de cursos  
* asociados a un expositor que se desea dar  
* de baja o a modificar, de ser así, la baja  
* o cambio correspondiente no procede al  
* retornar <IContinua> como falso.  
*/
```

```
FUNCTION ChcBaseCur( QueError)  
LOCAL Conti := .T.
```

```
SELECT A  
LOCATE FOR CveExpo = CveAux  
IF FOUND()  
    Error(QueError)  
    Conti = .F.  
ENDIF  
SELECT C  
RETURN Conti
```

```
/*  
* ChecaCveEx() -> NIL  
*  
* Revisa si una clave de profesor se repiete  
* al dar un nuevo registro de alta.  
*  
*/
```

```
FUNCTION ChecaCveEx
```

```
    SEEK(Clave)  
    IF FOUND()  
        Error(2)  
        Proseguir = .F.  
    ENDIF  
RETURN NIL
```

```
/*  
* ChcCveMoEx() -> NIL  
*  
* Revisa si una clave de profesor se repiete  
* al modificar un registro.  
*  
*/
```

```
FUNCTION ChcCveMoEx
```

```
    SELECT C  
    SEEK(Clave)  
    IF FOUND()  
        IF CveAux != CveExpo  
            Error(2)
```

```
Proseguir = .F.  
Clave = CveAux  
ENDIF  
ENDIF  
RETURN NIL
```

```
/*  
* GrabaExp() --> NIL  
*  
* Graba físicamente en el disco de trabajo las  
* modificaciones o altas realizadas de expositores.  
*  
*/  
FUNCTION GrabaExp  
    REPLACE CveExpo WITH Clave, NombreExpo WITH Nombre, RFC WITH Registro,;  
    Domicilio WITH Domi, Telefono WITH Tele, Adscripcio WITH Adsc,;  
    Titulo WITH Titu, Cedula WITH Cedu, Expedicion WITH Fec1,;  
    Observa WITH Memorandum  
RETURN NIL
```

```
/*  
* ConsuExpG() --> NIL  
*  
* Consultas globales de expositores.  
*  
*/  
FUNCTION ConsuExpG  
    SELECT C  
    IF LASTREC() = 0  
        Error(7)  
    ELSE  
        PantaBases('<< CONSULTAS GLOBALES DE EXPOSITORES >>')  
        HazConsuEx()  
    ENDIF  
RETURN NIL
```

```
/*  
* ConsuExpl() -> NIL  
*  
* Consultas por cadena incluida.  
*  
*/  
FUNCTION ConsuExpl  
  
    SELECT C  
    IF LASTREC() = 0  
        Error(7)  
    ELSE  
        PantaBases('<< CONSULTAS DE EXPOSITORES POR CADENA INCLUIDA >>')  
        IF PalabraInc(3)  
            HazConsuEx()  
        SET FILTER TO  
    ENDIF  
    ENDIF  
RETURN NIL
```

```
/*  
* HazConsuEx -> NIL  
*  
* Realiza las consultas de expositores ya  
* sea por clave o por cadena incluida.  
*  
*/  
FUNCTION HazConsuEx  
    LOCAL Continuar := .T.  
  
    CamposExp()  
    OK2()  
    SET KEY 28 TO  
    Continuar = .T.  
    DO WHILE Continuar  
        PantaExpo()  
        GO TOP  
        CajaCampos()  
        SETCOLOR('15/2,15/0,,')  
        SET KEY 28 TO  
        DBEDIT(11,c1,,12,c2,m1,,m2,;  
            CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))  
        SET KEY 28 TO AYUDA  
        RESTSCREEN(0,0,24,79,G)
```

```
IF LASTKEY() != 27
  PonInfoExp(RECNO())
  Memorandum = Observa
  ControlCon(20,31,.F.)
ENDIF
IF Confirmar(4) = 2
  Continuar = .F.
ENDIF
ENDDO
SET KEY 28 TO Ayuda
RETURN NIL
```

```
/*
```

```
* ConsuExpC() --> NIL
*
* Realiza consultas por clave de expositores.
*
*/
```

```
FUNCTION ConsuExpC
  LOCAL Opcion, Terminar

  SELECT C
  PantaBases('<< CONSULTAS POR CLAVE DE EXPOSITORES >>')
  OK2()
  IF LASTREC() = 0
    Error(7)
  ELSE
    Terminar = .F.
    DO WHILE !Terminar
      PantaExpo()
      PantaClave(3,")
      IF LASTKEY() != 27
        SEEK(Clave)
        IF FOUND()
          SKIP
          IF CveExpo = Clave
            CajaCampos()
            SETCOLOR("15/2,15/0,,")
            SET FILTER TO CveExpo = Clave
            CamposExp()
            GO TOP
            SET KEY 28 TO
            DBEDIT(11,c1,12,c2,m1,"",m2,;
              CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
```



```
SET KEY 28 TO AYUDA
RESTSCREEN(0,0,24,79,G)
IF LASTKEY() != 27
  PonInfoExp(RECNO())
  Memorandum = Observa
  ControlCon(20,31,..F.)
ENDIF
SET FILTER TO
ELSE
SKIP -1
PonInfoExp(RECNO())
Memorandum = Observa
ControlCon(20,31,..F.)
ENDIF
ELSE
Error(4)
ENDIF
ENDIF
IF Confirmar(4) = 2
  Terminar = .T.
ENDIF
ENDDO
ENDIF
RETURN NIL
```

/***

* CamposExp() -> NIL

*

* Asigna los valores necesarios para la función DBEDIT para Expositores.

*

*/

FUNCTION CamposExp

LOCAL i

m1 = ARRAY(FCOUNT())

m2 = ARRAY(FCOUNT())

FOR i = 1 TO LEN(m1)

 m1[i] = FIELDNAME(i)

NEXT

m2 = { ' Clave del expositor', ' Nombre del expositor', ' R.F.C.',;

 ' Domicilio', ' Teléfono', ' Adscripción', ' Título', ' Cédula',;

 ' Fecha de expedición', ' Observaciones' }

RETURN NIL

```
/***  
* PonInfoExp ( <nPosReg> ) --> NIL  
*  
* Coloca la información del registro < nPosReg > de  
* la base de datos de expositores en la pantalla.  
*  
*/  
FUNCTION PonInfoExp ( nPosReg )  
  
    GO nPosReg  
    SETCOLOR('15/0,...')  
    @ 2,31 SAY CveExpo  
    @ 4,31 SAY NombreExpo  
    @ 6,31 SAY RFC  
    @ 8,31 SAY Domicilio  
    @ 10,31 SAY Telefono  
    @ 12,31 SAY Adscripcio  
    @ 14,31 SAY Titulo  
    @ 16,31 SAY Cedula  
    @ 18,31 SAY Expedicion  
    CveAux = CveExpo  
RETURN NIL
```

```
/***  
* PantaExpo() --> NIL  
*  
* Coloca la pantalla de expositores.  
*  
*/  
FUNCTION PantaExpo  
  
    SETCOLOR('15/1,...')  
    @ 1,6,21,72 BOX Marco7+''  
  
    @ 2,8 SAY 'Clave del expositor :'  
    @ 4,8 SAY 'Nombre del expositor :'  
    @ 6,8 SAY 'R.F.C.  
    @ 8,8 SAY 'Domicilio  
    @ 10,8 SAY 'Teléfono  
    @ 12,8 SAY 'Adscripción  
    @ 14,8 SAY 'Título  
    @ 16,8 SAY 'Cédula
```

```
@ 18,8 SAY 'Fecha de expedición : / /'  
@ 20,8 SAY 'Observaciones      : Memorándum'  
RETURN NIL
```

```
-----/
```

```
-----  
*      Este es el módulo de Carreras o Profesiones  
-----/
```

```
/***  
* ConsuCarr() --> NIL  
*  
* Permite la selección del tipo de consultas de  
* Carreras o profesiones, ya sean globales o por clave.  
*  
*/  
FUNCTION ConsuCarr  
  
  PonPanta(3)  
  nOpcion3 = Seleccion(3,nOpcion3)  
  DO CASE  
    CASE nOpcion3 = 1  
      ConsuCarrG()  
      FinConsu = .T.  
    CASE nOpcion3 = 2  
      ConsuCarrC()  
      FinConsu = .T.  
    CASE nOpcion3 = 3  
      ConsuCarrI()  
      FinConsu = .T.  
    CASE nOpcion3 = 4  
      nOpcion3 = 1  
  ENDCASE  
RETURN NIL
```

```
/***  
* AltasCarr() --> NIL  
*  
* Da de alta una nueva carrera.  
*  
*/  
FUNCTION AltasCarr  
  LOCAL Terminar  
  
  SELECT D  
  PantaBases('<< ALTAS DE CARRERAS O PROFESIONES >>')  
  PonConCan()  
  PantaCarr()  
  Terminar = .F.  
  DO WHILE !Terminar  
    BlancosCarr()  
    ControlAIP(16,29,.T.)  
    IF Confirmar(2) = 2  
      Terminar = .T.  
    ENDIF  
  ENDDO  
  RETURN NIL
```

```
/***  
* BajasCarr() --> NIL  
*  
* Da de baja una carrera o profesión existente.  
*  
*/  
FUNCTION BajasCarr  
  LOCAL Opcion, Terminar  
  
  SELECT D  
  PantaBases('<< BAJAS DE CARRERAS O PROFESIONES >>')  
  PonConCan()  
  Terminar = .F.  
  DO WHILE !Terminar  
    PantaCarr()  
    IF LASTREC() = 0  
      Error(37)  
      Terminar = .T.  
    ELSE  
      PantaClave(4,"")  
      IF LASTKEY() != 27
```

```
SEEK(Clave)
IF FOUND()
  SKIP
  IF CveProfesi = Clave
    CajaCampos()
    SETCOLOR('15/2,15/0,,')
    SET FILTER TO CveProfesi = Clave
    CamposCar()
    GO TOP
    SET KEY 28 TO
    DBEDIT(11,c1,l2,c2,m1,,m2,;
      CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
    SET KEY 28 TO AYUDA
    RESTSCREEN(0,0,24,79,G)
    IF LASTKEY() != 27
      PonInfoCarr(RECNO())
      Memorandum = Observa
      ControlBaP(16,29,.F.)
    ENDIF
    SET FILTER TO
  ELSE
    SKIP -1
    PonInfoCarr(RECNO())
    Memorandum = Observa
    ControlBaP(16,29,.F.)
  ENDIF
ELSE
  Error(38)
ENDIF
ENDIF
IF Confirmar(3) = 2
  Terminar = .T.
ENDIF
ENDIF
ENDDO
RETURN NIL
```

```
/*  
* ControlBaP( <nRen>, <nCol>, <IVerdad> ) --> NIL  
*  
* Controla la apertura de las opciones de la pantalla de  
* bajas de carreras. Aquí <nRen> y <nCol> son las coordenadas  
* en la cual está ubicada la opción "Memorándum" y <IVerdad>  
* en caso de ser .F., indica que el campo memorándum no se  
* puede modificar de lo contrario este si es modificable.  
*  
*/  
FUNCTION ControlBaP(nRen, nCol, IVerdad)  
  LOCAL i  
  
  i = 2  
  DO WHILE i != 0  
    DO CASE  
      CASE i = 1  
        i = 2  
      CASE i = 2  
        i = ControlMemo(nRen,nCol,IVerdad)  
      CASE i = 3  
        i = ConCan()  
      CASE i = 4  
        i = 2  
      CASE i = 5  
        i = 3  
      CASE i = 6  
        PosAux = RECNO()  
        IF ChcBaseCar(49,51)  
          Go PosAux  
          DELETE // Elimina físicamente el registro de un carrera  
          PACK  
          ENDIF  
          i = 0  
        ENDCASE  
      ENDDO  
    RETURN NIL  
  
/*  
* ModiCarr() --> NIL  
*  
* Modifica los datos de una carrera o profesión ya existente  
*  
*/
```

```
FUNCTION ModiCarr
  LOCAL Opcion, Terminar

  SELECT D
  PantaBases('<< MODIFICACIONES DE CARRERAS O PROFESIONES >>')
  PonConCan()
  IF LASTREC() = 0
    Error(37)
  ELSE
    Terminar = .F.
    DO WHILE !Terminar
      PantaCarr()
      PantaClave(4,")
      IF LASTKEY() != 27
        SEEK(Clave)
        IF FOUND()
          SKIP
          IF CveProfesi = Clave
            CajaCampos()
            SETCOLOR('15/2,15/0,,')
            SET FILTER TO CveProfesi = Clave
            CamposCar()
            GO TOP
            SET KEY 28 TO
            DBEDIT(I1,c1,I2,c2,m1,"",m2;
              CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
            SET KEY 28 TO AYUDA
            RESTSCREEN(0,0,24,79,G)
            IF LASTKEY() != 27
              PonInfoCarr(RECNO())
              LlenaCaCarr()
              PosAux = RECNO()
              Memorandum = Observa
              ControlMoP(16,29,.T.)
            ENDIF
            SET FILTER TO
          ELSE
            SKIP -1
            PonInfoCarr(RECNO())
            LlenaCaCarr()
            PosAux = RECNO()
            Memorandum = Observa
            ControlMoP(16,29,.T.)
          ENDIF
        ELSE
          Error(38)
        ENDIF
      ENDIF
    ENDWHILE
  ENDIF

```

```
    ENDF
  ENDF
  IF Confirmar(5) = 2
    Terminar = .T.
  ENDF
ENDDO
ENDF
RETURN NIL
```

```
/***
```

```
* ControlAIP( <nRen>, <nCol>, <IVerdad> ) --> NIL
```

```
*
```

- * Controla la apertura de las opciones de la pantalla de
- * altas de carreras. Aquí <nRen> y <nCol> son las coordenadas
- * en la cual está ubicada la opción "Memorándum" y <IVerdad>
- * en caso de ser .F., indica que el campo memorándum no se
- * puede modificar de lo contrario este si es modificable.
- *

```
*/
```

```
FUNCTION ControlAIP(nRen, nCol, IVerdad)
```

```
  LOCAL i, Salte
```

```
  Salte = READEXIT(.T.)
```

```
  i = 1
```

```
  DO WHILE i != 0
```

```
    DO CASE
```

```
      CASE i = 1
```

```
        i = CapturaCarr()
```

```
      CASE i = 2
```

```
        i = ControlMemo(nRen, nCol, IVerdad)
```

```
      CASE i = 3
```

```
        i = ConCan()
```

```
      CASE i = 4
```

```
        i = 1
```

```
      CASE i = 5
```

```
        i = 2
```

```
      CASE i = 6
```

```
        i = AgregaCarr()
```

```
    ENDCASE
```

```
  ENDDO
```

```
  READEXIT(Salte)
```

```
RETURN NIL
```



```
/*  
* ControlMoP( <nRen>, <nCol>, <IVerdad> ) --> NIL  
*  
* Controla la apertura de las opciones de la pantalla de  
* cambios de carreras. Aquí <nRen> y <nCol> son las coordenadas  
* en la cual está ubicada la opción "Memorándum" y <IVerdad>  
* en caso de ser .F., indica que el campo memorándum no se  
* puede modificar de lo contrario este si es modificable.  
*  
*/  
FUNCTION ControlMoP(nRen, nCol, IVerdad)  
  LOCAL i, Salte  
  
  Salte = READEXIT(.T.)  
  i = 1  
  DO WHILE i != 0  
    DO CASE  
      CASE i = 1  
        i = CapturaCarr()  
      CASE i = 2  
        i = ControlMemo(nRen,nCol,IVerdad)  
      CASE i = 3  
        i = ConCan()  
      CASE i = 4  
        i = 1  
      CASE i = 5  
        i = 2  
      CASE i = 6  
        i = CambiaCarr()  
    ENDCASE  
  ENDDO  
  READEXIT(Salte)  
RETURN NIL
```

```
/*  
* BlancosCarr() --> NIL  
*  
* Asigna a las variables auxiliares para dar de alta  
* los datos de una carrera su valor por defecto.  
*  
*/  
FUNCTION BlancosCarr
```

```
Clave = '000'  
Nombre = SPACE(45)  
Memorandum = "  
RETURN NIL
```

```
/***
```

```
* LlenaCaCarr() --> NIL
```

```
*
```

```
* Asigna los contenidos de la carrera  
* seleccionada, a las variables auxiliares  
* para realizar los cambios requeridos.  
*
```

```
*/
```

```
FUNCTION LlenaCaCarr
```

```
Clave = CveProfesi  
CveAux = Clave  
Nombre = NombreCarr  
Memorandum = Observa
```

```
RETURN NIL
```

```
/***
```

```
* CapturaCarr() --> <nAccion>
```

```
*
```

```
* Permite capturar los datos para una  
* carrera nueva o ya existente. Devuelve  
* en <nAccion> el número de código  
* correspondiente según la tecla presionada.  
*
```

```
*/
```

```
FUNCTION CapturaCarr
```

```
LOCAL i
```

```
SETCOLOR('0/7,,,15/0')
```

```
SET CURSOR ON
```

```
@ 6,29 GET Clave PICT '999'
```

```
@ 11,29 GET Nombre PICT '@!'
```

```
READ
```

```
T = LASTKEY()
```

```
DO CASE
```

```
  CASE T = 27
```

```
    i = 0
```

```
CASE T = 18 .OR. T = 5
  i = 3
  OTHERWISE
  i = 2
ENDCASE
SET CURSOR OFF
RETURN i
```

```
/***
```

```
* AgregaCarr() --> nAccion
*
* Realiza físicamente el alta de una nueva
* carrera siempre y cuando no exista error en
* validación de datos. Devuelve en <nAccion>
* el número de código correspondiente según
* la tecla presionada.
```

```
*
```

```
*/
```

```
FUNCTION AgregaCarr
  LOCAL i
```

```
  Proseguir = .T.
```

```
  Clave = STRTRAN(Clave, ',','0')
```

```
  ChecaCveCa()
```

```
  IF Proseguir
```

```
    APPEND BLANK
```

```
    GrabaCarr() // Realiza el alta de una nueva carrera.
```

```
    i = 0
```

```
  ELSE
```

```
    i = 1
```

```
  ENDIF
```

```
RETURN i
```

```
/***
```

```
* CambiaCarr() --> nAccion
*
* Realiza físicamente cambios a los datos
* de una carrera siempre y cuando no exista
* error en validación de los mismos. Devuelve
* en <nAccion> el número de código
* correspondiente según la tecla presionada.
```

```
*
```

```
*/
```

```
FUNCTION CambiaCarr
  LOCAL i

  Proseguir = .T.
  Clave = STRTRAN(Clave,'','0')
  ChcCveMoCa()
  i = 1
  IF Proseguir
    Go PosAux
    IF Clave != CveAux
      IF ChcBaseCar(50,52)
        Go PosAux
        GrabaCarr()
        i = 0
      ELSE
        Clave = CveAux
      ENDIF
    ELSE
      Go PosAux
      GrabaCarr()
      i = 0
    ENDIF
  ENDIF
  RETURN i
```

/*

```
* ChcBaseCar( <IError1>, <IError1> ) --> IContinua
*
* Revisa la posible existencia de cursos y ordenes de pago
* asociadas a una carrera que se desea dar de baja o a modificar,
* de ser así, la baja o cambio correspondiente no procede al
* retornar IContinua como falso.
*
```

*/

```
FUNCTION ChcBaseCar( QueError1, QueError2 )
  LOCAL Conti := .T.

  SELECT A
  LOCATE FOR CveCar = CveAux
  IF FOUND()
    Error(QueError1)
    Conti = .F.
  ENDIF
  SELECT B
```

```
LOCATE FOR CveCarrera = CveAux
IF FOUND()
  Error(QueError2)
  Conti = .F.
ENDIF
SELECT D
RETURN Conti
```

```
/**
 * ChecaCveCa()
 *
 * Revisa si una clave de carrera se repiete
 * al dar un nuevo registro de alta.
 *
```

```
*/
FUNCTION ChecaCveCa
```

```
  IF Clave = '000'
    Error(44)
    Proseguir = .F.
  ELSE
    SEEK(Clave)
    IF FOUND()
      Error(39)
      Proseguir = .F.
    ENDIF
  ENDIF
RETURN NIL
```

```
/**
 * ChcCveMoCa() --> NIL
 *
 * Revisa si una clave de carrera se repiete
 * al modificar un registro.
 *
```

```
*/
FUNCTION ChcCveMoCa
```

```
  IF Clave = '000'
    Error(44)
    Proseguir = .F.
    Clave = CveAux
```

```
ELSE
  SELECT D
  SEEK(Clave)
  IF FOUND()
    IF CveAux != CveProfesi
      Error(39)
      Proseguir = .F.
      Clave = CveAux
    ENDIF
  ENDIF
ENDIF
RETURN NIL
```

```
/***
 * GrabaCarr() --> NIL
 *
 * Graba físicamente en el disco de trabajo las
 * modificaciones o altas realizadas de carreras.
 *
 */
```

```
FUNCTION GrabaCarr
```

```
  REPLACE CveProfesio WITH Clave, NombreCarr WITH Nombre,;
  Observa WITH Memorandum
RETURN NIL
```

```
/***
 * ConsuCarrG() --> NIL
 *
 * Realiza consultas globales de carreras.
 *
 */
```

```
FUNCTION ConsuCarrG
```

```
  SELECT D
  IF LASTREC() = 0
    Error(37)
  ELSE
    PantaBases('<< CONSULTAS GLOBALES DE CARRERAS O PROFESIONES >>')
    HazConsuCa()
  ENDIF
RETURN NIL
```

```
/*  
* ConsuCarrl() --> NIL  
*  
* Consultas de carreras por cadena incluida.  
*  
*/  
FUNCTION ConsuCarrl  
  
  SELECT D  
  IF LASTREC() = 0  
    Error(37)  
  ELSE  
    PantaBases('<< CONSULTAS DE CARRERAS O PROFESIONES '+  
      'POR CADENA INCLUIDA >>')  
    IF PalabraInc(4)  
      HazConsuCa()  
      SET FILTER TO  
    ENDIF  
  ENDIF  
RETURN NIL
```

```
/*  
* HazConsuCa --> NIL  
*  
* Realiza las consultas de carreras o profesiones, ya  
* sea por clave o por cadena incluida.  
*  
*/  
FUNCTION HazConsuCa  
  LOCAL Continuar := .T.  
  
  CamposCar()  
  OK2()  
  SET KEY 28 TO  
  Continuar = .T.  
  DO WHILE Continuar  
    PantaCarr()  
    GO TOP  
    CajaCampos()  
    SETCOLOR('15/2,15/0,..')  
    SET KEY 28 TO  
    DBEDIT(11,c1,l2,c2,m1,"",m2,;  
      CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
```

```
SET KEY 28 TO AYUDA
RESTSCREEN(0,0,24,79,G)
IF LASTKEY() != 27
  PonInfoCarr(RECNO())
  Memorandum = Observa
  ControlCon(16,29,.F.)
ENDIF
IF Confirmar(4) = 2
  Continuar = .F.
ENDIF
ENDDO
SET KEY 28 TO Ayuda
RETURN NIL
```

/***

```
* ConsuCarrC() --> NIL
*
* Realiza consultas por clave de carreras.
*
*/
```

```
FUNCTION ConsuCarrC
  LOCAL Opcion, Terminar
```

```
  SELECT D
  PantaBases('<< CONSULTAS POR CLAVE DE CARRERAS O PROFESIONES >>')
  OK(2)
  IF LASTREC() = 0
    Error(37)
  ELSE
    Terminar = .F.
    DO WHILE !Terminar
      PantaCarr()
      PantaClave(4,"")
      IF LASTKEY() != 27
        SEEK(Clave)
        IF FOUND()
          SKIP
          IF CveProfesi = Clave
            CajaCampos()
            SETCOLOR('15/2,15/0,,')
            SET FILTER TO CveProfesi = Clave
            CamposCar()
            GO TOP
            SET KEY 28 TO
```



```
DBEDIT(I1,c1,I2,c2,m1,"",m2,;
      CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
SET KEY 28 TO AYUDA
RESTSCREEN(0,0,24,79,G)
IF LASTKEY() != 27
  PonInfoCarr(RECNO())
  Memorandum = Observa
  ControlCon(16,29,.F.)
ENDIF
SET FILTER TO
ELSE
SKIP -1
PonInfoCarr(RECNO())
Memorandum = Observa
ControlCon(16,29,.F.)
ENDIF
ELSE
Error(38)
ENDIF
ENDIF
IF Confirmar(4) = 2
  Terminar = .T.
ENDIF
ENDDO
ENDIF
RETURN NIL
```

/***

* CamposCar() --> NIL

*

* Asigna los valores necesarios para la función DBEDIT para Carreras.

*

*/

FUNCTION CamposCar

LOCAL i

m1 = ARRAY(FCOUNT())

m2 = ARRAY(FCOUNT())

FOR i = 1 TO LEN(m1)

 m1[i] = FIELDNAME(i)

NEXT

m2 = { ' Clave de la carrera', ' Nombre de la carrera', ' Observaciones' }

RETURN NIL

```
/*  
* PonInfoCarr ( <nPosReg> ) --> NIL  
*  
* Coloca la información del registro <nPosReg> de  
* la base de datos de carreras en la pantalla.  
*  
*/
```

```
FUNCTION PonInfoCarr ( nPosReg )
```

```
    GO nPosReg  
    SETCOLOR('15/0,...')  
    @ 06,29 SAY CveProfesi  
    @ 11,29 SAY NombreCarr  
    CveAux = CveProfesi  
RETURN NIL
```

```
/*  
* PantaCarr() --> NIL  
*  
* Coloca la pantalla de carreras.  
*  
*/
```

```
FUNCTION PantaCarr
```

```
    SETCOLOR('15/1,...')  
    @ 3,4,19,75 BOX Marco7+' '  
    @ 06,6 SAY 'Clave de la carrera :'  
    @ 11,6 SAY 'Nombre de la carrera :'  
    @ 16,6 SAY 'Observaciones      : Memorándum'  
RETURN NIL
```

```
...../
```

```
.....  
*           Este es el módulo de reportes  
...../
```

/***

* Reportes() --> NIL

*

* Controla el menú principal del módulo de reportes.

*/

FUNCTION Reportes

LOCAL Opcion

PRIVATE ContAlum // Cuenta el total de alumnos en un curso a ser
// notificados en los reportes.

SET WRAP ON

Opcion = 1

DO WHILE Opcion != 7 .AND. LASTKEY() != 27

PantaBases('<< MODULO DE REPORTES >>')

SETCOLOR('15/2,15/0,...')

@ 7,15,15,65 BOX Marco7+''

SombraMarco(7,15,15,65)

@ 8,16 PROMPT ' 1. PORCENTAJE DE AVANCE DE CURSOS

@ 9,16 PROMPT ' 2. CURSOS POR VENCERSE

@ 10,16 PROMPT ' 3. LISTA DE DEUDORES

@ 11,16 PROMPT ' 4. REGISTRO DE INSCRIPCION Y REPORTE DE RECIBOS '

@ 12,16 PROMPT ' 5. LISTA DE PROFESORES

@ 13,16 PROMPT ' 6. INFORME A LA UNIDAD DE PLANEACION

@ 14,16 PROMPT ' 7. SALIR

MENU TO Opcion

DO CASE

CASE Opcion = 1

RepAvance()

CASE Opcion = 2

CursosPorV()

CASE Opcion = 3

ListaDeud()

CASE Opcion = 4

RegInscrip()

CASE Opcion = 5

ListaProf()

CASE Opcion = 6

RepPlanea()

ENDCASE

ENDDO

SET WRAP OFF

RETURN NIL

/***

- * RepAvance() --> NIL
- *
- * Exhibe en la pantalla el porcentaje de avance
- * de un curso seleccionado por su clave hasta la
- * fecha del sistema.
- *

*/

FUNCTION RepAvance

LOCAL Opcion, Terminar

SELECT A

PantaBases('<< AVANCE DE CURSO HASTA EL : '+UPPER(ExtraerFecha(Hoy))+
'>>')

OK2()

IF LASTREC() = 0

Error(5)

ELSE

Terminar = .F.

DO WHILE !Terminar

PantaAvance()

SETCOLOR('15/1,...')

@ 18, 21 SAY REPLICATE(CHR(176),40)

@ 19, 31 SAY STR(0)

PantaClave(1,"

IF LASTKEY() != 27

SEEK(Clave)

IF FOUND()

SKIP

IF CveCurso = Clave

CajaCampos()

SETCOLOR('15/2,15/0,...')

SET FILTER TO CveCurso = Clave

CamposCur()

GO TOP

SET KEY 28 TO

DBEDIT(11,c1,l2,c2,m1,"m3,m2,;

CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))

SET KEY 28 TO AYUDA

RESTSCREEN(0,0,24,79,G)

IF LASTKEY() != 27

PonInfoAva(RECNO())

Porcentaje()

ControlOk()

ENDIF

SET FILTER TO

```
ELSE
  SKIP -1
  PonInfoAva(RECNO())
  Porcentaje()
  ControlOk()
ENDIF
ELSE
  Error(3)
ENDIF
ENDIF
IF Confirmar(6) = 2
  Terminar = .T.
ENDIF
ENDDO
ENDIF
RETURN NIL
```

```
/*
* Porcentaje() --> NIL
*
* Calcula el porcentaje de avance de un curso.
*
*/
```

```
FUNCTION Porcentaje
  LOCAL TotalDias, DiasAvance, i

  DiasDeCur(Dias)
  TotalDias = ExtraeTotD(FechaExpo1, FechaExpo2)
  SETCOLOR('15/0,....')
  @ 13,23 SAY STR(0)
  @ 14,23 SAY STR(TotalDias)
  IF Hoy < FechaExpo1
    Caja(' Porcentaje de avance = 0%. El curso todavía no empieza. ');
    '15/3,....')
    Ok( 12, 37, '15/7,....', '0/3,....', 7*16+4)
    INKEY(10)
    Boton(12,37,42)
    RESTSCREEN(0,0,24,79,G)
  ELSE
    IF Hoy > FechaExpo2
      DiasDeCur(Dias)
      DiasAvance = ExtraeTotD(FechaExpo1, FechaExpo2)
      SETCOLOR('15/0,....')
      @ 13,23 SAY STR(DiasAvance)
```

```
@ 14,23 SAY STR(0)
SETCOLOR('14/1,....')
@ 18,21 SAY REPLICATE(CHR(219),40)
@ 19,31 SAY STR(100)
Caja(' Porcentaje de avance = 100%. El curso ya terminó ','15/3,....')
Ok( 12, 37,'15/7,....','0/3,....',7*16+4)
INKEY(10)
Boton(12,37,42)
RESTSCREEN(0,0,24,79,G)
ELSE
  DiasDeCur(Dias)
  TotalDias = ExtraeTotD(FechaExpo1,FechaExpo2)
  DiasAvance = ExtraeTotD(FechaExpo1,Hoy)
  SETCOLOR('15/0,....')
  @ 13,23 SAY STR(DiasAvance)
  @ 14,23 SAY STR(TotalDias-DiasAvance)
  SETCOLOR('14/1,....')
  FOR i = 1 TO INT(DiasAvance*40/TotalDias)
    @ 18,20+i SAY CHR(219)
    @ 19,31 SAY STR(INT(i*2.5))
    DELAY(0.0001)
  NEXT
  @ 19,31 SAY STR(INT(DiasAvance*100/TotalDias))+ '%'
ENDIF
ENDIF
RETURN NIL
```

```
/*
```

```
* RepPlanea() --> NIL
```

```
*
```

```
* Genera el reporte a entregar a la unidad de planeación.
```

```
*
```

```
*/
```

```
FUNCTION RepPlanea
```

```
  LOCAL i
```

```
  LOCAL nLongLinea := 120
```

```
  SELECT A
```

```
  SET FILTER TO
```

```
  SET KEY 28 TO
```

```
  PantaBases('<<< REPORTE A LA UNIDAD DE PLANEACION >>')
```

```
  IF LASTREC() = 0
```

```
    Error(5)
```

```
  ELSE
```

```
EscogerPer()
IF LASTKEY() != 27
  IF Periodo1 > Periodo2
    Error(14)
  ELSE
    IF Periodo1 < FechaBase .OR. Periodo2 < FechaBase
      Error(18)
    ELSE
      Texto = ALLTRIM(Instituci)+;
      Salto+;
      ALLTRIM(Escuela)+;
      Salto+;
      'SEGUIMIENTO DE AVANCE '+;
      DTOC(Periodo1)+' - '+DTOC(Periodo2)+;
      Salto+Salto+;
      'STAFF'+;
      Salto+Salto+;
      'INSTANCIA : '+ALLTRIM(Depto)+' '+';
      'RESPONSABLE : '+Responsable+;
      Salto+Salto+Salto+;
      'PROGRAMAS'+ '@'+;
      'ETAPAS DE DESARROLLO AVANCES'+ '@'+;
      '%'+ '@'+;
      'TIEMPO EMPLEADO'+ '@'+;
      'OBSERVACIONES'+;
      Salto
      cTexto = cCentraTxt(1,nLongLinea,ALLTRIM(Instituci))+;
      Salto+;
      cCentraTxt(1,nLongLinea, ALLTRIM(Escuela))+;
      Salto+;
      cCentraTxt(1,nLongLinea, 'SEGUIMIENTO DE AVANCE '+;
      DTOC(Periodo1)+' - '+DTOC(Periodo2))+;
      Salto+Salto+;
      cCentraTxt(1,nLongLinea, 'STAFF' )+;
      Salto+Salto+;
      cCentraTxt(1,nLongLinea,;
      'INSTANCIA : '+ALLTRIM(Depto)+' '+';
      'RESPONSABLE : '+Responsable );+;
      Salto+Salto+Salto+;
      'PROGRAMAS '+';
      'ETAPAS DE DESARROLLO AVANCES '+';
      '% '+';
      'TIEMPO EMPLEADO '+';
      'OBSERVACIONES'+;
      Salto+Salto
GO TOP
```

```
FOR i = 1 TO LASTREC()
  IF Periodo1 <= FechaExpo2
    IF Periodo1 < FechaExpo1
      IF FechaExpo1 <= Periodo2
        CompRepPla(Periodo2)
      ENDIF
    ELSE
      CompRepPla(Periodo2)
    ENDIF
  ENDIF
  SKIP
NEXT
HazReporte('RUP',nLongLinea)
ENDIF
ENDIF
ENDIF
SET KEY 28 TO AYUDA
RETURN NIL
```

/*

```
* CompRepPla ( <dPeriodo2> ) --> NIL
*
* Función auxiliar que permite anexas al informe
* de la unidad de planeación, aquellos registros
* que cumplan con el rango de fechas deseadas.
*
```

*/

```
FUNCTION CompRepPla( dPeriodo2 )
  LOCAL TotalDias, DiasAvance, TotMeses, TipoC, NomC

  DiasDeCur(Dias)
  TotalDias = ExtraeTotD(FechaExpo1, FechaExpo2)
  IF dPeriodo2 > FechaExpo2
    DiasAvance = ExtraeTotD(FechaExpo1, FechaExpo2)
    TotMeses = (FechaExpo2-FechaExpo1+1)/30.4
  ELSE
    DiasAvance = ExtraeTotD(FechaExpo1, dPeriodo2)
    TotMeses = (dPeriodo2-FechaExpo1+1)/30.4
  ENDIF
  TipoC = SUBSTR(CveCurso,1,1)
  DO CASE
  CASE TipoC = '0'
    NomC = 'VARIOS'
```



```
CASE TipoC = '1'
  NomC = 'IDIOMAS '
CASE TipoC = '2'
  NomC = 'COMPUTACION'
CASE TipoC = '3'
  NomC = 'CARRERA '
CASE TipoC = '4'
  NomC = 'MAESTRIA '
CASE TipoC = '5'
  NomC = 'DIPLOMADO '
ENDCASE
Texto = Texto+;
NomC+'@'+;
NombreCur+'@'+;
STR(INT(DiasAvance*100/TotalDias))+ '@'+;
STR(ROUND(TotMeses,2))+ '@'+;
Salto
cTexto = cTexto+;
NomC+' '+;
NombreCur+;
STR(INT(DiasAvance*100/TotalDias))+;
STR(ROUND(TotMeses,2))+;
Salto
RETURN NIL
```

```
/***
* DiasDeCur ( <cVarDias> ) --> NIL
*
* Agrega en el arreglo de <DiasExtraidos>
* el número de los días de la semana contenidos
* en la variable <cVarDias>. Aquí Domingo = 1,
* Lunes = 2,..., Sabado = 7.
*
*/
FUNCTION DiasDeCur(cVarDias)
  LOCAL i

  DiasExtraidos = {}
  IF SUBSTR(cVarDias,7,1) = 'S'
    ADD(DiasExtraidos,1)
  ENDIF
  FOR i = 1 TO 6
    IF SUBSTR(cVarDias,i,1) = 'S'
      ADD(DiasExtraidos,i+1)
```

```
ENDIF  
NEXT  
RETURN NIL
```

```
/*  
* ExtraeTotD ( <dFechalim1>, <dFechalim2> )  
*  
* cuenta el total de días que tiene un curso  
* dependiendo de el período de exposición dado  
* a <dFechalim1> y <dFechalim2> y los días en  
* que ese curso es impartido <DiasExtraídos>.  
*  
*/  
FUNCTION ExtraeTotD (dFechalim1, dFechalim2)  
  LOCAL Total, FechaVar, i  
  
  Total = 0  
  FechaVar = dFechalim1  
  DO WHILE FechaVar <= dFechalim2  
    FOR i = 1 TO LEN(DiasExtraídos)  
      IF Dow(FechaVar) = DiasExtraídos[i]  
        ++Total  
      ENDIF  
    NEXT  
    ++FechaVar  
  ENDDO  
  RETURN Total
```

```
/*  
* ListaProf ( ) --> NIL  
*  
* Permite seleccionar un curso por su clave  
* para elaborar su lista para profesores.  
*  
*/  
FUNCTION ListaProf  
  LOCAL Opcion, Terminar  
  
  SELECT A  
  SET FILTER TO  
  SET KEY 28 TO  
  PantasBases('<< LISTA DE ASISTENCIA DE CURSOS PARA PROFESORES >>')
```

```
IF LASTREC() = 0
  Error(5)
ELSE
  Terminar = .F.
  DO WHILE !Terminar
    PantaClave(1,"")
    IF LASTKEY() != 27
      SEEK(Clave)
      IF FOUND()
        SKIP
        IF CveCurso = Clave
          CajaCampos()
          SETCOLOR('15/2,15/0,,,')
          SET FILTER TO CveCurso = Clave
          CamposCur()
          GO TOP
          SET KEY 28 TO
            DBEDIT(11,c1,l2,c2,m1,"",m3,m2,;
              CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
          SET KEY 28 TO AYUDA
          RESTSCREEN(0,0,24,79,G)
          IF LASTKEY() != 27
            GeneraLis()
          ENDIF
          SET FILTER TO
        ELSE
          SKIP -1
          GeneraLis()
        ENDIF
      ELSE
        Error(3)
      ENDIF
    ENDIF
    IF Confirmar(8) = 2
      Terminar = .T.
    ELSE
      SELECT A
    ENDIF
  ENDDO
ENDIF
SET KEY 28 TO AYUDA
RETURN NIL
```

```
/***
* GeneraLis() --> NIL
*
* Genera la lista de profesores.
*
*/
FUNCTION GeneraLis
  LOCAL CAux
  LOCAL nLongLinea := 118

  CAux = CveCurso
  SELECT B
  LOCATE FOR CveCurso = CAux
  IF FOUND()
    Texto = ALLTRIM(Instituci)+;
      Salto+;
      ""+ALLTRIM(Escuela)+"""+;
      Salto+;
      ALLTRIM(Depto)+;
      Salto+Salto+;
      'LISTA DE ASISTENCIA'+;
      Salto+Salto+;
      'Curso o taller : '+ A->NombreCur+ ' '+;
      'Expositor : '+ C->NombreExpo+ ' '+;
      'Salón : '+ A->Salon+;
      Salto+Salto+;
      'Fecha del curso : '+ DTOC( A->FechaExpo1 )+;
      ' - '+DTOC( A->FechaExpo2 )+ ' '+;
      'Duración : '+STRTRAN(STR( A->Duracion),' ','');
      ' HRS. '+;
      Salto+Salto+;
      'Nombre del alumno'+REPLICATE('@',30)+;
      Salto
  cTexto = cCentraTxt (1,nLongLinea,ALLTRIM(Instituci))+;
    Salto+;
    cCentraTxt (1,nLongLinea,""+ALLTRIM(Escuela)+"""+;
    Salto+;
    cCentraTxt (1,nLongLinea,ALLTRIM(Depto))+;
    Salto+Salto+;
    cCentraTxt (1,nLongLinea,'LISTA DE ASISTENCIA'+;
    Salto+Salto+;
    cCentraTxt (1,nLongLinea,'Curso o taller : '+ A->NombreCur+ ' '+;
      'Expositor : '+ C->NombreExpo+ ' '+;
      'Salón : '+ A->Salon)+;
    Salto+Salto+;
    cCentraTxt (1,nLongLinea,'Fecha del curso : '+;
```

```
      DTOC( A->FechaExpo1 )+;
      '- '+DTOC( A->FechaExpo2 )+ ' '+;
      'Duración : '+;
      STRTRAN(STR( A->Duracion),' ','')+;
      ' HRS.  ')+;

      Salto+Salto+;
      ' NOMBRE DEL ALUMNO           '+;
      REPLICATE(' ',30)+;
      Salto

ContAlum = 1
DO WHILE CveCurso = CAux
  Texto = Texto+;
      RIGHT(STR(ContAlum),3)+ '- ' + RTRIM(NombreAlum)+;
      REPLICATE('@',30)+;
      Salto
  cTexto = cTexto+;
      RIGHT(STR(ContAlum),3)+ '- ' +NombreAlum+;
      REPLICATE(' ',30)+;
      Salto
  SKIP
  ++ContAlum
ENDDO
HazReporte(' .LIS',nLongLinea)
ELSE
  Error(3)
ENDIF
RETURN NIL

/****
* RegInscrip ( ) --> NIL
*
* Permite seleccionar un curso por su clave,
* para elaborar el registro de inscripción y el
* reporte de recibos.
*
*/
FUNCTION RegInscrip
  LOCAL Opcion, Terminar

  SELECT A
  SET FILTER TO
  PantaBases('<< REGISTRO DE INSCRIPCION Y REPORTE DE RECIBOS >>')
  IF LASTREC() = 0
    Error(5)
```

```
ELSE
  Terminar = .F.
  DO WHILE !Terminar
    PantaClave(1,")
    IF LASTKEY() != 27
      SEEK(Clave)
      IF FOUND()
        SKIP
        IF CveCurso = Clave
          CajaCampos()
          SETCOLOR('15/2,15/0...')
          SET FILTER TO CveCurso = Clave
          CamposCur()
          GO TOP
          SET KEY 28 TO
          DBEDIT(I1,c1,I2,c2,m1,,"m3,m2,;
            CHR(220)+CHR(220),CHR(219),CHR(223)+CHR(223))
          SET KEY 28 TO AYUDA
          RESTSCREEN(0,0,24,79,G)
          IF LASTKEY() != 27
            GeneraReg()
          ENDIF
          SET FILTER TO
        ELSE
          SKIP -1
          GeneraReg()
          SET FILTER TO
        ENDIF
      ELSE
        Error(3)
      ENDIF
    ENDIF
    IF Confirmar(10) = 2
      Terminar = .T.
    ELSE
      SELECT A
      PantaBases('<< REGISTRO DE INSCRIPCION Y REPORTE DE RECIBOS >>')
    ENDIF
  ENDDO
ENDIF
RETURN NIL
```

```
/*  
* GeneraReg ( ) --> NIL  
*  
* Permite seleccionar el registro de inscripción y el  
* reporte de recibos a exhibir y/o editar y/o imprimir.  
* Si existen recibos por reportar se procede a elaborar  
* el respectivo reporte, de lo contrario se listan los  
* reportes ya existentes y se permite seleccionar alguno  
* de ellos para generar nuevamente ese reporte.  
*  
*/  
FUNCTION GeneraReg  
  LOCAL CAux, Pos, NoOficio, Ofic, i, Opcion, ContBlanc  
  LOCAL nLong1 := 164, nLong2 := 78  
  PRIVATE ContBeca := 0 /* Cuenta el total de alumnos en un curso a ser  
                           notificados en los reportes con beca. */  
  
  CAux = CveCurso  
  SELECT B  
  LOCATE FOR CveCurso = CAux  
  IF FOUND()  
    Pos = RECNO()  
    NoOficio = {}  
    ContBlanc = 0  
    DO WHILE CveCurso = CAux  
      ReciaAux = Recibos  
      FOR i = 1 TO ASC(SUBSTR(ReciaAux,1,1))  
        Ofic = SUBSTR(ReciaAux,56*i-30,12)  
        IF Ofic = SPACE(12)  
          ++ContBlanc  
        ELSE  
          IF ASCAN(NoOficio,Ofic) = 0  
            AADD(NoOficio,Ofic)  
          ENDIF  
        ENDIF  
      NEXT  
      SKIP  
    ENDDO  
    Importe = 0  
    ContAlum = 1  
    IF LEN(NoOficio) > 0  
      IF ContBlanc != 0  
        AADD(NoOficio,'Oficio nuevo')  
        PantalLisOf()  
        SETCOLOR('15/0,0/7,...')  
        SET KEY 28 TO
```

```
Opcion = ACHOICE(10,34,14,45,NoOficio,,,1,1)
SET KEY 28 TO AYUDA
RESTSCREEN(0,0,24,79,G)
IF Opcion > 0
  Go Pos
  IF Opcion != LEN(NoOficio)
    IF AltasOfic(.T.)
      Cabecera(nLong1)
      DO WHILE CveCurso = CAux
        ReciaAux = Recibos
        FOR i = 1 TO ASC(SUBSTR(ReciaAux,1,1))
          Ofic = SUBSTR(ReciaAux,56*i-30,12)
          IF Ofic = NoOficio[Opcion]
            RepAlumnos(i)
            ++ContAlum
          ENDIF
        NEXT
        SKIP
      ENDDO
      SKIP -1
      NoOfAux = NoOficio[Opcion]
      RepRecibos(nLong2)
      HazReporte('.REG',nLong1)
      Texto = TextoAux
      cTexto = cTextoAux
      HazReporte('.REP',nLong2)
    ENDIF
  ELSE
    CapOficio()
    IF LASTKEY() != 27 .AND. NoOfAux != SPACE(12)
      IF AltasOfic(.T.)
        Cabecera(nLong1)
        DO WHILE CveCurso = CAux
          ReciaAux = Recibos
          FOR i = 1 TO ASC(SUBSTR(ReciaAux,1,1))
            Ofic = SUBSTR(ReciaAux,56*i-30,12)
            IF Ofic = SPACE(12)
              ReciaAux = STUFF(ReciaAux,56*i-30,12,NoOfAux)
              RepAlumnos(i)
              ++ContAlum
            ENDIF
          NEXT
          REPLACE Recibos WITH ReciaAux
          SKIP
        ENDDO
        SKIP -1
```



```
RepRecibos(nLong2)
HazReporte('.REG',nLong1)
Texto = TextoAux
cTexto = cTextoAux
HazReporte('.REP',nLong2)
ENDIF
ENDIF
ENDIF
ENDIF
ELSE
Caja(' No existen recibos por reportar ','15/3,...')
Ok( 12, 37,'15/7,...','0/3,...','7*16+4)
INKEY(2)
RESTSCREEN(0,0,24,79,G)
IF Confirmar(11) = 1
PantLisOf()
SETCOLOR('15/0,0/7,...')
SET KEY 28 TO
Opcion = ACHOICE(10,34,14,45,NoOficio,...,1,1)
SET KEY 28 TO AYUDA
RESTSCREEN(0,0,24,79,G)
IF Opcion > 0
Go Pos
IF AltasOfic(.T.)
Cabecera(nLong1)
DO WHILE CveCurso = CAux
ReciAux = Recibos
FOR i = 1 TO ASC(SUBSTR(ReciAux,1,1))
Ofic = SUBSTR(ReciAux,56*i-30,12)
IF Ofic = NoOficio[Opcion]
RepAlumnos(i)
++ContAlum
ENDIF
NEXT
SKIP
ENDDO
SKIP -1
NoOfAux = NoOficio[Opcion]
RepRecibos(nLong2)
HazReporte('.REG',nLong1)
Texto = TextoAux
cTexto = cTextoAux
HazReporte('.REP',nLong2)
ENDIF
ENDIF
ENDIF
```

```
ENDIF
ELSE
  CapOficio()
  IF LASTKEY() != 27 .AND. NoOfAux != SPACE(12)
    IF AltasOfic(.T.)
      Go Pos
      Cabecera(nLong1)
      DO WHILE CveCurso = CAux
        ReciAux = Recibos
        FOR i = 1 TO ASC(SUBSTR(ReciAux,1,1))
          Ofic = SUBSTR(ReciAux,56*i-30,12)
          IF Ofic = SPACE(12)
            ReciAux = STUFF(ReciAux,56*i-30,12,NoOfAux)
            RepAlumnos(i)
            ++ContAlum
          ENDIF
        NEXT
        REPLACE Recibos WITH ReciAux
        SKIP
      ENDDO
      SKIP -1
      RepRecibos(nLong2)
      HazReporte(' .REG',nLong1)
      Texto = TextoAux
      cTexto = cTextoAux
      HazReporte(' .REP',nLong2)
    ENDIF
  ENDIF
ENDIF
ELSE
  Error(3)
ENDIF
RETURN NIL
```

```
/*
* CapOficio ( ) --> NIL
*
* Permite capturar un número de oficio para
* generar un nuevo reporte.
*
*/
FUNCTION CapOficio
```

```
SET KEY 28 TO
```

```
NoOfAux = SPACE(12)
Caja(' Teclee el número de oficio ', '15/3,,,,')
SETCOLOR('15/0,,,,')
SET CURSOR ON
@ 12,34 GET NoOfAux PICT '@!'
READ
SET CURSOR OFF
SET KEY 28 TO AYUDA
RETURN NIL
```

```
/***
```

- * Cabecera (< nLongitud >) --> NIL
- *
- * Crea los encabezados del registro de inscripción
- * de cursos y los almacena en la variable <Texto>.
- * El parámetro <nLongitud> indica el total de caracteres
- * existentes por línea.
- *

```
*/
```

```
FUNCTION Cabecera ( nLongLinea )
```

```
  Texto = ALLTRIM(Escuela)+;
    Salto+;
    Depto+;
    Salto+Salto+;
    'REGISTRO DE INSCRIPCION'+;
    Salto+Salto+;
    'Curso o taller : '+ A->NombreCur+ ' '+;
    'Expositor : '+ C->NombreExpo+ ' '+;
    'Salón : '+ A->Salon+;
    Salto+Salto+;
    'Fecha del curso : '+ DTOC( A->FechaExpo1 )+;
    ' - '+DTOC( A->FechaExpo2 )+ ' '+;
    'Duración : '+ALLTRIM(STR( A-> Duracion),' ');+;
    ' HRS. '+;
    'UNAM : '+ALLTRIM(STR(A->CostoAlum))+;
    ' EGRES. : '+ALLTRIM(STR(A->CostoEgre))+;
    ' PUBLICO : '+ALLTRIM(STR(A->CostoPubl))+;
    Salto+Salto+;
    'Nombre del alumno'+ '@'+ 'Teléfono'+ '@'+ 'No.- Cuenta'+ '@'+;
    'Carrera'+ '@'+ 'Semestre'+ '@'+ 'Fecha'+ '@'+ 'Rec. Of.'+ '@'+;
    'Importe'+ '@'+ 'Notas'+;
    Salto
  cTexto = cCentraTxt (1,nLongLinea, ALLTRIM(Escuela))+;
```

```
Salto+;
cCentraTxt (1,nLongLinea, ALLTRIM(Depto))+;
Salto+Salto+;
cCentraTxt (1,nLongLinea, 'REGISTRO DE INSCRIPCION')+;
Salto+Salto+;
cCentraTxt (1,nLongLinea, 'Curso o taller : '+ A->NombreCur+ ' '+;
'Expositor : '+ C->NombreExpo+ ' '+;
'Salón : '+ A->Salon)+;
Salto+Salto+;
cCentraTxt (1,nLongLinea, 'Fecha del curso : '+ DTOC( A->FechaExpo1 )+;
' '+ DTOC( A->FechaExpo2 )+ ' '+;
'Duración : '+;
ALLTRIM(STR( A->Duracion),' '))+;
'HRS. '+;
'UNAM : '+ALLTRIM(STR(A->CostoAlum))+;
'EGRES. : '+ALLTRIM(STR(A->CostoEgre))+;
'PUBLICO : '+ALLTRIM(STR(A->CostoPubl))+;
Salto+Salto+;
' NOMBRE DEL ALUMNO '+;
'TELEFONO '+;
'No. - CUENTA '+;
'CARRERA '+;
'SEMESTRE '+;
'FECHA '+;
'REC. OF. '+;
'IMPORTE '+;
'NOTAS'+;
Salto+Salto
TextoAux = "
cTextoAux = "
RETURN NIL
```

```
/*
* RepRecibos ( <nLongitud> ) --> NIL
*
* Crea la carta de repore de ingresos.
* El parámetro <nLongitud> indica el total
* de caracteres a visualizar por renglón.
*/
FUNCTION RepRecibos(nLongLinea)
LOCAL cCompRep
```

```
cCompRep = Salto+Salto+Salto+Salto+Salto+;
```

Dirigido+;
Salto+;
Cargo1+;
Salto+;
'P r e s e n t e.'+;
Salto+Salto+Salto+Salto+Salto+;
' Por medio del presente me permito enviar a Usted '+;
STRTRAN(STR(ContAlum-ContBeca-1),' ')+;
' recibos oficiales que amparan la cantidad de : '+;
Salto+;
TRANSFORM(Importe,'\$99,999,999.99')+;
' por concepto de ingresos '+;
'recaudados en este Departamento.'+;
Salto+;
Salto+;
' Así mismo, solicito hacer el trámite para el pago de '+;
'honorarios del profesor '+;
RTRIM(C->NombreExpo)+;
' correspondientes al curso '+;
RTRIM(A->NombreCur)+;
' que se lleva a cabo los días '+;
ExtraerDias(A->Dias)+;
' del '+;
ExtraerFecha(A->FechaExpo1)+;
' al '+;
ExtraerFecha(A->FechaExpo2)+;
Salto+;
Salto+;
' Dicho honorario es por la cantidad de : '+;
TRANSFORM(Honorario,'\$99,999,999.99')+;
Salto+;
Salto+;
' Sin otro particular, aprovecho la ocasión para '+;
'enviarle un cordial saludo.'+;
Salto+;
Salto+;
Salto+;
Salto+;
'A T E N T A M E N T E'+;
Salto+;
""POR MI RAZA HABLARA EL ESPIRITU""+;
Salto+;
'San Juan de Aragón, Edo. de Méx., '+;
ExtraerFecha(Fecha)+;
Salto+;
Cargo2+;

```
Salto+;
Salto+;
Salto+;
Salto+;
Suscribe+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
' ANEXO : Recibos oficiales y la documentación '+;
'correspondiente.'+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
'c.c.p.'+;
Ccp+;
CHR(12)+; // Se obliga un salto de línea
Salto+;
Salto+;
Salto+;
'RELACION DE RECIBOS OFICIALES CORRESPONDIENTES AL CURSO '+;
RTRIM( A->NombreCur )+;
' IMPARTIDO POR EL PROFESOR '+;
RTRIM( C->NombreExpo )+;
' DEL '+;
UPPER(ExtraerFecha(A->FechaExpo1))+;
' AL '+;
UPPER(ExtraerFecha(A->FechaExpo2))+;
Salto+;
Salto+;
Salto
TextoAux = Salto+Salto+Salto+;
Escuela+;
Salto+;
Depto+;
Salto+Salto+;
'Of. Núm.'+NoOfAux+;
cCompRep+;
TextoAux
```

```
TextoAux = TextoAux+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
'TOTAL DE INGRESOS : '+';
TRANSFORM(Importe,'$99,999,999.99')+;
Salto+;
Salto+;
'HONORARIOS      : '+';
TRANSFORM(Honorario,'$99,999,999.99')+;
Salto+;
Salto+;
'N E T O ...     : '+';
TRANSFORM(Importe-Honorario,'$99,999,999.99')+;
Salto
cTextoAux = Salto+Salto+Salto+;
cCentraTxt (1,nLongLinea, Escuela)+;
Salto+;
cCentraTxt (1,nLongLinea, Depto)+;
Salto+Salto+;
cCentraTxt (1,nLongLinea, 'Of. Núm. '+NoOfAux)+;
cCompRep+;
cTextoAux
cTextoAux = cTextoAux+;
Salto+;
Salto+;
Salto+;
Salto+;
Salto+;
'TOTAL DE INGRESOS : '+';
TRANSFORM(Importe,'$99,999,999.99')+;
Salto+;
Salto+;
'HONORARIOS      : '+';
TRANSFORM(Honorario,'$99,999,999.99')+;
Salto+;
Salto+;
'N E T O ...     : '+';
TRANSFORM(Importe-Honorario,'$99,999,999.99')+;
Salto
RETURN NIL
```

```
/*
* RepAlumnos( <nNoDeRec> ) --> NIL
*
* Genera la lista de recibos del registro de
* inscripción y del reporte de recibos.
*
*/
FUNCTION RepAlumnos(nNoDeRec)
  LOCAL CadBeca

  // Acumula los datos del registro de inscripción
  IF VAL(SUBSTR(ReciAux,56*nNoDeRec-54,12)) = 0
    CadBeca = 'Beca'
  ELSE
    CadBeca = ""
  ENDIF
  Texto = Texto+;
    RIGHT(STR(ContAlum),3)+ '- ' + RTRIM(NombreAlum)+'@'+;
    ALLTRIM(STR(Telefono))+'@'+;
    Cuenta+'@'
  SELECT D
  LOCATE FOR CveProfesi = B->CveCarrera
  SELECT B
  Texto = Texto+;
    D->NombreCarr+'@'+;
    STRTRAN(RIGHT(STR(Semestre),2),' ','0')+'@'+;
    SUBSTR(ReciAux,56*nNoDeRec-8,10)+'@'+; // Fecha de entrega en DEC
    SUBSTR(ReciAux,56*nNoDeRec-54,12)+'@'+; // No. de recibo
    SUBSTR(ReciAux,56*nNoDeRec-42,12)+'@'+; // Cantidad
    CadBeca+;
    Salto
  cTexto = cTexto+;
    RIGHT(STR(ContAlum),3)+ '- ' + LEFT(NombreAlum,30) + ' ' +;
    STR(Telefono)+' ' +;
    Cuenta+' '
  SELECT D
  LOCATE FOR CveProfesi = B->CveCarrera
  SELECT B
  cTexto = cTexto+;
    SUBSTR(D->NombreCarr,1,20) + ' ' +;
    STRTRAN(RIGHT(STR(Semestre),2),' ','0')+' ' +;
    SUBSTR(ReciAux,56*nNoDeRec-8,10)+' ' +; // Fecha en DEC
    SUBSTR(ReciAux,56*nNoDeRec-54,12)+' ' +; // No. de recibo
    SUBSTR(ReciAux,56*nNoDeRec-42,12)+' ' +; // Cantidad
    CadBeca+;
    Salto
```



```
// Acumula los datos de las ordenes de pago
IF VAL(SUBSTR(ReciAux,56*nNoDeRec-54,12)) != 0
  TextoAux = TextoAux+;
  RIGHT(STR(ContAlum-ContBeca),3)+'.- '+RTRIM(NombreAlum)+'@'+;
  SUBSTR(ReciAux,56*nNoDeRec-54,12)+'@'+; // No. de recibo
  SUBSTR(ReciAux,56*nNoDeRec-8,10)+'@'+; // Fecha en DEC
  SUBSTR(ReciAux,56*nNoDeRec-42,12)+'@'+; // Cantidad
  Salto
  cTextoAux = cTextoAux+;
  RIGHT(STR(ContAlum-ContBeca),3)+'.- '+;
  LEFT(NombreAlum,25)+' '+;
  SUBSTR(ReciAux,56*nNoDeRec-54,12)+' '+; // No. de recibo
  SUBSTR(ReciAux,56*nNoDeRec-8,10)+' '+; // Fecha en DEC
  SUBSTR(ReciAux,56*nNoDeRec-42,12)+' '+; // Cantidad
  Salto
  Importe = Importe+VAL(SUBSTR(ReciAux,56*nNoDeRec-42,12))
ELSE
  ++ContBeca
ENDIF
RETURN NIL

/****
* AltasOfic( <IDat> ) --> <IConfirmar>
*
* Permite capturar los datos que lleva un
* reporte de recibos. Si <IDat> es verdadero
* se solicitara el honorario y la fecha. Si
* se confirmo la generacion del reporte
* <IConfirmar> es .T.
*
*/
FUNCTION AltasOfic( IDat )

  PantaBases('<< DATOS DEL OFICIO >>')
  PonConCan()
  PantaDatos(IDat)
RETURN ControlDat(IDat)
```

```
/*  
* ControlDat( <IDat> ) --> <IHazOficio>  
*  
* Controla la apertura de las opciones de la pantalla  
* de altas de oficinas. Si <IDat> es verdadero se solicitará  
* el honorario y la fecha.  
*/  
FUNCTION ControlDat(IDat)  
  LOCAL i, Salte, Bandera := .F.  
  
  Salte = READEXIT(.T.)  
  i = 1  
  DO WHILE i != 0  
    DO CASE  
      CASE i = 1  
        i = PedirDatos(IDat)  
      CASE i = 2  
        i = ControlCcp()  
      CASE i = 3  
        i = ConCan()  
      CASE i = 6  
        i = 0  
        Bandera = .T.  
    ENDCASE  
  ENDDO  
  READEXIT(Salte)  
  RETURN Bandera
```

```
/*  
* PedirDatos( <IDat> ) --> <nAccion>  
*  
* Solicita el ingreso de los datos de un oficina.  
* Si <IDat> es verdadero se solicitará el honorario  
* y la fecha. Devuelve en <nAccion> un código numérico  
* que corresponde a la tecla presionada.  
*/  
FUNCTION PedirDatos( IDat )  
  LOCAL T, i  
  
  SET CURSOR ON  
  SETCOLOR('0/7,,,15/0')  
  @ 2,22 GET Instituci PICT '@!'
```

```
@ 4,22 GET Escuela PICT '@!'
@ 6,22 GET Depto PICT '@!'
@ 8,22 GET Dirigido PICT '@!'
@ 10,22 GET Cargo1
@ 12,22 GET Suscribe PICT '@!'
@ 14,22 GET Cargo2 PICT '@!'
IF IDat
  @ 16,22 GET Honorario PICT '999,999,999.99'
  @ 18,22 GET Fecha
ENDIF
READ
SET CURSOR OFF
T = LASTKEY()
DO CASE
  CASE T = 27
    i = 0
  CASE T = 18 .OR. T = 5
    i = 3
  OTHERWISE
    i = 2
ENDCASE
SET CURSOR OFF
RETURN i
```

/****

* ControlCcp() --> nAccion

*

* Controla la apertura de la opción de las personas
* con copia de oficio.

*

*/

FUNCTION ControlCcp

LOCAL Tecla, i

Mod_Color(20,22,20,27,7*16+15)

Tecla = 0

DO WHILE Tecla != 13 .AND. Tecla != 27

Tecla = INKEY(0)

DO CASE

CASE Tecla = 13

Mod_Color(20,22,20,27,15)

EditCcp()

i = 3

CASE Tecla = 27

```
Mod_Color(20,22,20,27,15)
i = 0
CASE Tecla = 28
  VarAyuda = 11
  AyudaMod()
CASE Tecla = 24 .OR. Tecla = 3
  i = 3
  Mod_Color(20,22,20,27,15)
  Tecla = 27
CASE Tecla = 5 .OR. Tecla = 18
  i = 1
  Mod_Color(20,22,20,27,15)
  Tecla = 27
ENDCASE
ENDDO
RETURN i
```

/***

* EditaCcp() --> NIL

*

* Realiza la edición del contenido de la variable

* CCP, la cual almacena los datos de los funcionarios

* que tienen copia del oficio a realizar.

*

*/

FUNCTION EditaCcp

l1 = 6

c1 = 6

l2 = 18

c2 = 73

VentanaExplosiva(l1, c1, l2, c2, Marco7, '15/3,...')

SombraMarco(l1, c1, l2, c2)

RamaH(l2-2, c1, c2, RamaH6, '15/3,...')

RamaH(l1+2, c1, c2, RamaH5, '15/3,...')

SETCOLOR('0/3,...')

@ l2-1, c1+19 SAY 'ESC = Salir Ctrl-W = Salvar'

@ l1+1, c1+27 SAY 'Con copia para'

RamaV(l1+3, c2, l2-3, FlechaV, '15/7,...')

SETCOLOR('15/3,...')

SET CURSOR ON

SET KEY 28 TO

Ccp = MEMOEDIT(Ccp, l1+3, c1+1, l2-3, c2-1, .T.)

SET KEY 28 TO Ayuda

```
SET CURSOR OFF
RESTSCREEN( 0, 0, 24, 79, G)
RETURN NIL
```

```
/***
```

```
* ListaDeud() --> NIL
```

```
*
```

```
* Muestra en la pantalla la lista de deudores  
* según la fecha vigente del sistema.
```

```
*
```

```
*/
```

```
FUNCTION ListaDeud
```

```
  LOCAL i, j, Suma, Deudores
```

```
  LOCAL nLongLinea := 158
```

```
  SELECT B
```

```
  Texto = 'LISTA DE DEUDORES HASTA EL '+DTOC(DATE());
```

```
    Salto+Salto+Salto+;
```

```
    'NOMBRE DEL DEUDOR'+ '@'+;
```

```
    'TELEFONO'+ '@'+;
```

```
    'DEUDA'+ '@'+;
```

```
    'CLAVE CURSO'+ '@'+;
```

```
    'NOMBRE CURSO'+ '@'+;
```

```
    'PERIODO DE CURSO'+;
```

```
    Salto
```

```
  cTexto = cCentraTxt ( 1,nLongLinea,;
```

```
    'LISTA DE DEUDORES HASTA EL '+DTOC(DATE()));
```

```
    Salto+Salto+Salto+;
```

```
    'NOMBRE DEL DEUDOR           '+';
```

```
    'TELEFONO           '+';
```

```
    'DEUDA           '+';
```

```
    'CLAVE CURSO           '+';
```

```
    'NOMBRE CURSO           '+';
```

```
    'PERIODO DE CURSO'+;
```

```
    Salto+Salto
```

```
  IF LASTREC() = 0
```

```
    Error(7)
```

```
  ELSE
```

```
    Deudores = 0
```

```
    GO TOP
```

```
    FOR i = 1 TO LASTREC()
```

```
      Suma = 0
```

```
      FOR j = 1 TO ASC(SUBSTR(Recibos,1,1))
```

```
        Suma = Suma+VAL(SUBSTR(Recibos,56*j-42,12))
```

```
NEXT
IF (CostoCur - Suma) > 0
  Texto = Texto+;
  NombreAlum+'@'+;
  ALLTRIM(STR(Telefono))+'@'+;
  STR(CostoCur-Suma)+'@'+;
  CveCurso+'@'+;
  A->NombreCur+'@'+;
  DTOC(A->FechaExpo1)+' - '+DTOC(A->FechaExpo2)+;
  Salto
  cTexto = cTexto+;
  NombreAlum+' '+;
  STR(Telefono)+' '+;
  STR(CostoCur-Suma)+' '+;
  CveCurso+' '+;
  A->NombreCur+' '+;
  DTOC(A->FechaExpo1)+' - '+DTOC(A->FechaExpo2)+;
  Salto
  ++Deudores
ENDIF
SKIP
NEXT
IF Deudores > 0
  HazReporte('LDD',nLongLinea)
ELSE
  Caja(' No hay deudores por el momento ','15/3,,,,')
  Ok( 12, 37,'15/7,,,,','0/3,,,,',7*16+4)
  INKEY(0.7)
  RESTSCREEN(0,0,24,79,G)
ENDIF
ENDIF
RETURN NIL
```

```
/**
* CursosPorV() --> NIL
*
* Muestra en la pantalla la relación de cursos por
* vencerse según los días de anticipación requeridos.
*
*/
FUNCTION CursosPorV
  LOCAL i, Continuar, Vencidos
  LOCAL nLongLinea := 68
```

```
SELECT A
SET FILTER TO
SET KEY 28 TO
IF LASTREC() = 0
  Error(5)
ELSE
  Continuar = .T.
  DO WHILE Continuar
    Vencidos = 0
    Anticipa = SPACE(2)
    SETCOLOR('15/1,...')
    @ 4,1,19,78 BOX SPACE(9)
    GO TOP
    Caja(' Teclee los días de anticipación ','15/3,...')
    SETCOLOR('15/1,15/0,...')
    SET CURSOR ON
    @ 12,38 GET Anticipa PICT '99'
    READ
    SET CURSOR OFF
    RESTSCREEN(0,0,24,79,G)
    Texto = 'RELACION DE CURSOS POR VENCER HASTA EL '+';
      UPPER(ExtraerFecha(Hoy+VAL(Anticipa))))+;
      Salto+Salto+Salto+;
      'CLAVE CURSO'+ '@'+;
      'NOMBRE DEL CURSO'+ '@'+;
      'VENCIMIENTO'+;
      Salto
    cTexto = cCentraTxt(1,nLongLinea,;
      'RELACION DE CURSOS POR VENCER HASTA EL '+';
      UPPER(ExtraerFecha(Hoy+VAL(Anticipa))))+;
      Salto+Salto+Salto+;
      'CLAVE CURSO '+';
      'NOMBRE DEL CURSO '+';
      'VENCIMIENTO'+;
      Salto+Salto
    FOR i = 1 TO LASTREC()
      IF ( FechaExpo2 >= Hoy ) .AND. ( FechaExpo2 <= Hoy+VAL(Anticipa) )
        Texto = Texto+CveCurso+'@'+NombreCur+'@'+DIOC(FechaExpo2)+;
          Salto
        cTexto = cTexto+CveCurso+' '+'NombreCur+' '+';
          DIOC(FechaExpo2)+;
          Salto
        ++Vencidos
      ENDIF
    SKIP
  NEXT
```

```
IF Vencidos > 0
  HazReporte('CPV',nLongLinea)
ELSE
  Caja(' No hay cursos por vencerse, al menos hasta el '+
    ExtraerFecha(Hoy+VAL(Anticipa))+','15/3,...')
  Ok( 12, 37,'15/7,...','0/3,...','7*16+4)
  INKEY(10)
  Boton(12,37,42)
  RESTSCREEN(0,0,24,79,G)
ENDIF
IF Confirmar(7) = 2
  Continuar = .F.
ENDIF
ENDDO
ENDIF
SET KEY 28 TO AYUDA
RETURN NIL
```

/***

*

* HazReporte (< cExt >, < nLong >) --> NIL

*

* Genera el reporte almacenado en la variable

* <Texto> en pantalla y permite imprimirlo y/o

* grabarlo en un archivo. El parámetro < cExt >

* representa la extensión del archivo y < nLong >

* es un código que permite seleccionara el número

* de caracteres a visualizar por columna ya sea

* en pantalla o en impresora.

*

*/

```
FUNCTION HazReporte ( cExtension, nLong )
```

```
  EditaTexto(nLong)
```

```
  IF Confirmar(12) = 1
```

```
    IF CapArch(cExtension)
```

```
      GuardaArch()
```

```
    ENDIF
```

```
  ENDIF
```

```
  Imprimir(nLong)
```

```
RETURN NIL
```


/***

* Imprimir(< nLong >) -> NIL

*

* Realiza la impresión de un reporte. El parámetro
* < nLong > es indica el número de caracteres a imprimir
* por renglón.

*

*/

FUNCTION Imprimir (nLong)

LOCAL nLineas, IRoturaLinea := .T., i

IF Confirmar(15) = 1

Caja(' Imprimiendo reporte. Espere un momento... ',15/3,...')

OK(12, 37, '15/7,...', '0/3,...', 7*16+4)

nLineas := MLCOUNT(cTexto, nLong,, IRoturaLinea)

SET PRINTER ON

SET CONSOLE OFF

/* Se establecen los Reset's */

IF MisVar10 = 3

?? CHR(27)+'E'

// Reset de la impresora láser

?? CHR(27)+'&l2a0O'

// Establece tamaño carta en la página e

// impresión en posición vertical

// para impresora láser

ELSE

?? CHR(27)+'@'

// RESET de impresoras de Matriz o Chorro

ENDIF

/* Se selecciona las secuencias de escape según su longitud por línea */

IF nLong <= 85

IF MisVar10 = 3

?? CHR(27)+'(s10H' // Establece 10 CPI's en impresoras láser

ELSE

?? CHR(27)+'P'

// 10 CPI's en impresoras de Matriz o Chorro

ENDIF

ELSE

IF nLong > 85 .AND. nLong <= 102

IF MisVar10 = 3

?? CHR(27)+'(s12H' // Establece 12 CPI's en impresoras láser

ELSE

?? CHR(27)+'M'

// 12 CPI's en impresoras de Matriz o Chorro

ENDIF

ELSE

IF nLong > 102 .AND. nLong <= 141

IF MisVar10 = 3

```
    ?? CHR(27)+'(s17H' // Establece 17 CPI's en impresoras láser
ELSE
    ?? CHR(27)+'P'      // 10 CPI's en impresoras de Matriz o Chorro
    ?? CHR(27)+CHR(15) // Modo condensado impresoras de
                        // Matriz o Chorro
ENDIF
ELSE
IF MisVar10 = 3
    ?? CHR(27)+'(s20H' // Establece 20 CPI's en impresoras
                        // láser
ELSE
    ?? CHR(27)+'M'     // 12 CPI's en impresoras de Matriz o
                        // Chorro
    ?? CHR(27)+CHR(15) // Modo condensado impresoras de
                        // Matriz o Chorro
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
/* Imprime las líneas del reporte */
FOR i = 1 TO nLineas
    ?? MEMOLINE(cTexto, nLong, i,, iRoturaLinea)
    ?
NEXT
/* Realiza un avance de página */
? CHR(12) // Avance de página
/* Se establecen los Reset's */
IF MisVar10 = 3
    ? CHR(27)+'E' // Reset de la impresora láser
ELSE
    ? CHR(27)+'@' // Reset de la impresora de Chorro y de Matriz
ENDIF
RESTSCREEN(0,0,24,79,G)
SET CONSOLE ON
SET PRINTER OFF
ENDIF
RETURN NIL
```

```
/***
* ExtraerFecha ( <dFec> ) --> cFecha
*
* Devuelve la fecha <dFec> a una cadena de caracteres
* en el formato "DD de MM de YYYY" en <cFecha>.
*/
FUNCTION ExtraerFecha(dFec)

RETURN STRTRAN(STR(DAY(dFec)),',','')+ ' de '+CMONTH(dFec)+' de '+
        STRTRAN(STR(YEAR(dFec)),',','")
```

```
/***
* ExtraerDias ( <cQueDias> ) --> <cCadenaDia>
*
** Devuelve la cadena <cCadenaDia> la cual contiene
* los días de la semana indicados por <cQueDias> el
* cual tiene el formato "DDDDDDDD" donde la letra "D"
* representa un días de la semana comenzando con Lunes.
* Si "D" es igual a "S" entonces quiere decir que ese
* día de la semana debe ser tomado en cuenta e incluido
* en <cCadenaDia>.
*/
```

```
FUNCTION ExtraerDias ( cQueDias )
  LOCAL i, CadDias, TotalDias, ComDias := {}

  FOR i = 1 TO 7
  IF SUBSTR(cQueDias,i,1) = 'S'
  DO CASE
    CASE i = 1
      AADD(ComDias,'Lunes')
    CASE i = 2
      AADD(ComDias,'Martes')
    CASE i = 3
      AADD(ComDias,'Miércoles')
    CASE i = 4
      AADD(ComDias,'Jueves')
    CASE i = 5
      AADD(ComDias,'Viernes')
    CASE i = 6
      AADD(ComDias,'Sábado')
    CASE i = 7
      AADD(ComDias,'Domingo')
```

```
        ENDCASE
    ENDFI
NEXT
TotalDias = LEN(ComDias)
DO CASE
    CASE TotalDias = 1
        CadDias = ComDias[1]
    CASE TotalDias = 2
        CadDias = ComDias[1]+' y '+ComDias[2]
    OTHERWISE
        CadDias = "
        FOR i = 1 TO TotalDias-2
            CadDias = CadDias+ComDias[i]+' , '
        NEXT
        CadDias = CadDias+ComDias[TotalDias-1]+' y ' +ComDias[TotalDias]
    ENDCASE
RETURN CadDias
```

/***

* CapArch (<cExtension>) --> !Salvado

*

* Permite asignar un nombre a un archivo de texto
* con la extensión <cExtension>. En caso de confirmar
* la grabación del mismo <!Salvado> es verdadero.

*

*/

```
FUNCTION CapArch(cExtension)
```

```
    LOCAL GrabaArch := .F.
```

```
    SET KEY 28 TO
```

```
    NomArch = SPACE(8)
```

```
    Caja(' Teclee el nombre del archivo ','15/3,...')
```

```
    SETCOLOR(', 15/0,...')
```

```
    SET CURSOR ON
```

```
    @ 12,36 GET NomArch PICT 'NNNNNNNN'
```

```
    READ
```

```
    SET CURSOR OFF
```

```
    RESTSCREEN(0,0,24,79,G)
```

```
    IF LASTKEY() != 27
```

```
        NomArch = UPPER(RTRIM(NomArch))+cExtension
```

```
    IF FILE(NomArch)
```

```
        IF Confirmar(9) = 1
```

```
            GrabaArch = .T.
```

```
    ENDFI
```

```
ELSE
  GrabaArch = .T.
ENDIF
ENDIF
SET KEY 28 TO AYUDA
RETURN GrabaArch
```

```
/**
 * GuardaArch() --> NIL
 *
 * Almacena una cadena de caracteres en un
 * archivo tipo texto.
 */
FUNCTION GuardaArch

  IF MEMOWRIT(NomArch,Texto)
    Caja(' Generando Archivo '+NomArch+' ',15/3,...')
    Ok( 12, 37,'15/7....','0/3....',7*16+4)
    INKEY(0.5)
    RESTSCREEN(0,0,24,79,G)
  ELSE
    Error(31)
  ENDIF
RETURN NIL
```

```
/**
 * EscogerPer() --> NIL
 *
 * Selecciona un rango de fechas.
 */
FUNCTION EscogerPer

  Periodo1 = CTOD('')
  Periodo2 = CTOD('')
  Caja(' Teclee el periodo del cual desea reportar avance ',15/3,...')
  SETCOLOR(' ',15/0,...')
  @ 12, 40 SAY '-'
  SET CURSOR ON
  @ 12, 25 GET Periodo1
  @ 12, 46 GET Periodo2
```

```
READ  
SET CURSOR OFF  
RESTSCREEN(0,0,24,79,G)  
RETURN NIL
```

```
/***
```

```
* PonInfoAva ( <nPosReg> ) --> NIL  
*  
* Coloca la información del registro <nPosReg>  
* de la base de datos de cursos en la pantalla,  
* para el módulo de avance de cursos.  
*  
*/
```

```
FUNCTION PonInfoAva ( nPosReg )  
LOCAL i
```

```
GO nPosReg  
SETCOLOR('15/0,,,')  
@ 2,23 SAY CveCurso  
@ 3,23 SAY NombreCur  
@ 4,23 SAY FechaExpo1  
@ 4,36 SAY FechaExpo2  
@ 5,23 SAY Lugar  
@ 6,23 SAY Duracion  
@ 7,23 SAY Horario1_1  
@ 7,26 SAY Horario1_2  
@ 7,31 SAY Horario2_1  
@ 7,34 SAY Horario2_2  
@ 8,23 SAY CveExpo  
@ 9,23 SAY Cupo  
@ 10,23 SAY Salon  
@ 11,23 SAY CveCar  
FOR i = 1 TO 7  
  @ 12,8*i+15 SAY SUBSTR(Dias,i,1)  
NEXT  
SETCOLOR('14/0,,,')  
@ 08,31 SAY C->NombreExpo  
IF CveCar = '000'  
  @ 11,31 SAY 'NINGUNA'+SPACE(38)  
ELSE  
  SELECT D  
  LOCATE FOR CveProfesi = A->CveCar  
  @ 11,31 SAY NombreCarr  
  SELECT A
```

```
ENDIF  
RETURN NIL
```

```
/***
```

```
* PantaAvance() --> NIL
```

```
*
```

```
* Coloca la pantalla para ver el avance de cursos.
```

```
*
```

```
*/
```

```
FUNCTION PantaAvance
```

```
SETCOLOR('15/1,...')
```

```
@ 1,2,15,77 BOX Marco7+'
```

```
@ 16,19,21,62 BOX Marco7
```

```
@ 19,21 SAY 'Avance ='
```

```
@ 2,4 SAY 'Clave curso :'
```

```
@ 3,4 SAY 'Nombre curso :'
```

```
@ 4,4 SAY 'Periodo : / / - / /'
```

```
@ 5,4 SAY 'Lugar :'
```

```
@ 6,4 SAY 'Duración : HRS.'
```

```
@ 7,4 SAY 'Horario : - : HRS.'
```

```
@ 8,4 SAY 'Clave expositor :'
```

```
@ 9,4 SAY 'Cupo :'
```

```
@ 10,4 SAY 'Salón :'
```

```
@ 11,4 SAY 'Clave carrera :'
```

```
@ 12,4 SAY 'Días : [ ] Lu [ ] Ma [ ] Mi [ ] Ju [ ] Vi [ ]+;
```

```
' Sa [ ] Do'
```

```
@ 13,4 SAY 'Dias expuestos :'
```

```
@ 14,4 SAY 'Por exponer :'
```

```
RETURN NIL
```

```
/***
```

```
* PantaLisOf() --> NIL
```

```
*
```

```
* Coloca la pantalla en donde se pone la lista de
```

```
* oficios existenes para un curso.
```

```
*
```

```
*/
```

```
FUNCTION PantaLisOf
```

```
VentanaExplosiva(7,22,17,57,Marco7,'15/2,...')
```

```
SombraMarco(7,22,17,57)
```

```
@ 9,32,15,47 BOX Marco7 COLOR '15/2,,,'  
SETCOLOR('14/2,,,')  
@ 7,24 SAY ' Seleccione el número de oficio '  
RamaV(10,47,14,FlechaV,'15/0,,,')  
RETURN NIL
```

```
/***
```

```
* PantaDatos( <IDat> ) --> NIL
```

```
*
```

```
* Coloca la pantalla de los datos a capturar para  
* generar un oficio. Si <IDat> es verdadero entonces  
* se solician los honorarios la fecha.
```

```
*
```

```
*/
```

```
FUNCTION PantaDatos ( IDat )
```

```
    SETCOLOR('15/1,,,')
```

```
    @ 01,02,21,77 BOX Marco7
```

```
    @ 02,04 SAY 'Institución  :'
```

```
    @ 04,04 SAY 'Escuela      :'
```

```
    @ 06,04 SAY 'Departamento :'
```

```
    @ 08,04 SAY 'Dirigido a   :'
```

```
    @ 10,04 SAY 'Cargo        :'
```

```
    @ 12,04 SAY 'Suscriptor   :'
```

```
    @ 14,04 SAY 'Cargo        :'
```

```
    IF IDat
```

```
        @ 16,04 SAY 'Honorarios : $'
```

```
        @ 18,04 SAY 'Fecha      :'
```

```
    ENDIF
```

```
    @ 20,04 SAY 'Copia para   : c.c.p.'
```

```
RETURN NIL
```

```
/***
```

```
* cCentraTxt ( <nX1>, <nX2>, <cCadena> ) --> <cCadRes>
```

```
*
```

```
* Devuelve la cadena de caracteres <cCadena> en  
* <cCadRes> pero con el número total de espacios  
* en blanco requeridos a la izquierda para centrarla  
* entre los márgenes <nX1> y <nX2>.
```

```
*
```

```
*/
```

```
FUNCTION cCentraTxt ( nX1, nX2, cCadena )
```



```
LOCAL R1 := nX2-nX1-LEN(cCadena)
```

```
RETURN IF ( R1%2 > 0, REPLICATE(' ',INT(R1/2)+1)+cCadena,;  
          REPLICATE(' ',INT(R1/2))+cCadena )
```

```
-----/
```

```
-----/
```

```
* Este es el módulo de disposición o configuración del sistema *
```

```
-----/
```

```
/***
```

```
* Unidad() -> NIL
```

```
*
```

```
* Esta función permite escoger la unidad de disco
```

```
* conjuntamente con su ruta, en donde residen las
```

```
* bases de datos.
```

```
*
```

```
*/
```

```
FUNCTION Unidad()
```

```
LOCAL Continua := .T.
```

```
SET KEY 28 TO
```

```
PantaBases('<< SELECCION DE LA UNIDAD DE TRABAJO >>')
```

```
DO WHILE Continua
```

```
  Caja(' Teclee la unidad y la ruta fuente en donde residen las bases '+'  
        'de datos : ','15/5,....')
```

```
  SETCOLOR(' ',15/0,....')
```

```
  SET CURSOR ON
```

```
  @ 12,5 GET NomPath PICT '@!'
```

```
  READ
```

```
  SET CURSOR OFF
```

```
  RESTSCREEN(0,0,24,79,G)
```

```
  IF LASTKEY() <> 27
```

```
    NomPath = ALLTRIM(NomPath)
```

```
    IF LEN(NomPath) > 2 .AND. RIGHT(NomPath,1) = '\'
```

```
      NomPath = LEFT(NomPath, LEN(NomPath)-1)
```

```
    ENDIF
```

```
    IF LEN(NomPath) > 1 .AND. MEMOWRIT(NomPath+'\TMP.TMP',")
```

```
      CLOSE ALL
```

```
      DELETE FILE (NomPath+'\TMP.TMP')
```

```
      SET DEFAULT TO (NomPath)
```

```
    CargarArch()
    IF !FIN
        Relaciona()
    ENDIF
    Continua = .F.
ELSE
    Error(43)
ENDIF
NomPath = NomPath+SPACE(70-LEN(NomPath))
ELSE
    Continua = .F.
ENDIF
ENDDO
SET KEY 28 TO AYUDA
RETURN NIL
```

```
/*
* Impresora() --> NIL
*
* Permite seleccionar el tipo de impresora
* y dejarla como predeterminada.
*
*/
FUNCTION Impresora()
    LOCAL OpcionAux

    PonPanta(5)
    OpcionAux = nOpcion5
    nOpcion5 = Seleccion(5,nOpcion5)
    IF nOpcion5 = 4
        nOpcion5 = OpcionAux
    ENDIF
    MisVar10 = nOpcion5
RETURN NIL
```

```
/*
* Datos() --> NIL
*
* Permite modificar diversas cadenas empleadas en
* los reportes y dejarlos como predeterminados.
*
*/
```

FUNCTION Datos()

```
IF AltasOfic(.F.)
  MisVar01 = Instituci
  MisVar02 = Escuela
  MisVar03 = Depto
  MisVar04 = Dirigido
  MisVar05 = Cargo1
  MisVar06 = Suscribe
  MisVar07 = Cargo2
  MisVar08 = Ccp
  Responsable = Suscribe
ENDIF
RETURN NIL
```

/***

* ClaveDeAcc() --> NIL

*

* Permite modificar la clave de acceso al programa y
* cambiarla por otra, siempre y cuando el usuario tenga
* acceso o conozca la clave anterior.

*

*/

FUNCTION ClaveDeAcc()

LOCAL ConfirmCve := .F.

PantaBases('<< SELECCION DE NUEVA CLAVE DE ACCESO >>')

IF !Entrar

_Error(33)

ELSE

SET CONFIRM ON

DO WHILE !ConfirmCve

PassWord = SPACE(4)

GAux = SAVESCREEN(0,0,24,79)

Caja (' Teclee la nueva clave de acceso. Sólo se permiten '+;
'letras y/o números ',15/5,...')

SETCOLOR(',15/0,')

SET CURSOR ON

@ 12,38 GET Password PICT '@! NNNN';

VALID PassWord <> SPACE(4) .OR. PassWord <> "

READ

SET CURSOR OFF

RESTSCREEN(0,0,24,79,GAux)

IF LASTKEY() <> 27

```
IF Confirmar(14) = 1
  MisVar09 = PassWord+SPACE(4-LEN(PassWord))
  ConfirmCve = .T.
ENDIF
ELSE
  ConfirmCve = .T.
ENDIF
ENDDO
SET CONFIRM OFF
ENDIF
RETURN NIL
```

```
/*.....*/
```

```
/*.....*
*           Este módulo es el que controla errores           *
*.....*/
```

```
/***
```

```
* Error( <nNoDeError> ) --> NIL
```

```
*
```

```
* Coloca en la pantalla el mensaje de  
* error, según el parámetro <nNoDeError>.
```

```
*
```

```
*/
```

```
FUNCTION Error (nNoDeError)
```

```
  LOCAL Aviso := Array(52, X1, X2, ColoresViejos := SETCOLOR())
```

```
  Aviso[01] = ' Error 01. Clave de curso ya existente... '
```

```
  Aviso[02] = ' Error 02. Clave de expositor ya existente... '
```

```
  Aviso[03] = ' Error 03. Clave de curso inexistente o curso sin inscristos... '
```

```
  Aviso[04] = ' Error 04. Clave de expositor inexistente. '+  
  'Debe antes darlo de alta. '
```

```
  Aviso[05] = ' Error 05. No hay presente ningún curso. '+  
  'Proceda a dar de alta. '
```

- Aviso[06] = ' Error 06. No hay presente ninguna orden de pago. '+';
'Proceda a dar de alta. '
- Aviso[07] = ' Error 07. No hay presente ningún expositor. '+';
'Proceda a dar de alta. '
- Aviso[08] = ' Error 08. Clave no válida. '+';
'El TIPO de curso debe ser entre 0 y 5... '
- Aviso[09] = ' Error 09. Clave no válida. '+';
'El SUBTIPO de curso debe ser entre 0 y 2... '
- Aviso[10] = ' Error 10. Clave no válida. '+';
'Cursos internos = 1 y externos = 0... '
- Aviso[11] = ' Error 11. Clave no válida. '+';
'El curso no es un idioma... '
- Aviso[12] = ' Error 12. Clave no válida. '+';
'Curso tipo 1. Lleva comprensión o posesión '
- Aviso[13] = ' Error 13. Costo de curso negativo... '
- Aviso[14] = ' Error 14. Periodo de inicio mayor al Periodo de término... '
- Aviso[15] = ' Error 15. Horario de inicio mayor o igual '+';
'al horario de término... '
- Aviso[16] = ' Error 16. Clave de carrera inexistente '+';
'o base de datos vacía... '
- Aviso[17] = ' Error 17. La clave de la procedencia debe ser entre 0 y 2... '
- Aviso[18] = ' Error 18. Curso fuera de rango. Revise fechas o periodos '
- Aviso[19] = ' Error 19. El cupo debe ser mayor que cero... '
- Aviso[20] = ' Error 20. La duración de un curso siempre '+';
'es mayor que cero... '
- Aviso[21] = ' Error 21. El módulo de un curso siempre es mayor que cero... '
- Aviso[22] = ' Error 22. La fecha de entrega en DEC no puede '+';
'ser antes que en caja... '
- Aviso[23] = ' Error 23. No puede haber pagos negativos '

- Aviso[24] = ' Error 24. No se encontró archivo CURSOS.DBF. '+';
'Se creará uno vacío. '
- Aviso[25] = ' Error 25. No se encontró archivo ORDEN.DBF. '+';
'Se creará uno vacío. '
- Aviso[26] = ' Error 26. No se encontró archivo EXPOSITO.DBF. '+';
'Se creará uno vacío. '
- Aviso[27] = ' Error 27. No se encontró archivo CURSOS.DBT. '
- Aviso[28] = ' Error 28. No se encontró archivo ORDEN.DBT... '
- Aviso[29] = ' Error 29. No se encontró archivo EXPOSITO.DBT... '
- Aviso[30] = ' Error 30. Total de días igual a cero. '+';
'Revise fechas y/o días de la semana. '
- Aviso[31] = ' Error 31. No se pudo generar archivo : '+NomArch+' '
- Aviso[32] = ' Error 32. Un curso tiene al menos un día de la semana '+';
'para impartirse. '
- Aviso[33] = ' Error 33. Lo siento, pero su clave no tiene acceso '+';
'a este módulo... '
- Aviso[34] = ' Error 34. Debe capturar un recibo '+';
'al menos, aunque sea beca... '
- Aviso[35] = ' Error 35. Clave no válida... Intente de nuevo '
- Aviso[36] = ' Error 36. Acceso denegado a altas, bajas y modificaciones '
- Aviso[37] = ' Error 37. No hay presente ninguna carrera. '+';
'Proceda a dar de alta. '
- Aviso[38] = ' Error 38. Clave de carrera inexistente. '+';
'Debe antes darla de alta. '
- Aviso[39] = ' Error 39. Clave de carrera ya existente... '
- Aviso[40] = ' Error 40. No se encontró archivo CARRERAS.DBF. '+';
'Se creará uno vacío. '
- Aviso[41] = ' Error 41. No se encontró archivo CARRERAS.DBT... '

Aviso[42] = ' Error 42. No existen registros con esa condición... '

Aviso[43] = ' Error 43. Error en la unidad o especificación de la ruta... '

Aviso[44] = ' Error 44. La clave cero ya está asignada al mensaje '+';
"NINGUNA CARRERA". '

Aviso[45] = ' Error 45. Baja improcedente. Hay ordenes de pago asociadas '+';
'a este curso. '

Aviso[46] = ' Error 46. Cambio improcedente. Hay ordenes de pago '+';
'asociadas a este curso.'

Aviso[47] = ' Error 47. Baja improcedente. Existen cursos asociados a '+';
'este expositor. '

Aviso[48] = ' Error 48. Cambio improcedente. Existen cursos asociados '+';
'a este expositor. '

Aviso[49] = ' Error 49. Baja improcedente. Existen cursos asociados a '+';
'esta carrera. '

Aviso[50] = ' Error 50. Cambio improcedente. Existen cursos asociados '+';
'a esta carrera. '

Aviso[51] = ' Error 51. Baja improcedente. Hay ordenes de pago '+';
'asociadas a la carrera. '

Aviso[52] = ' Error 52. Cambio improcedente. Hay ordenes de pago '+';
'asociadas a la carrera.'

```
Caja(Aviso[nNoDeError],'15/4,,,')  
IF LEN(Aviso[nNoDeError]) % 2 = 0  
  X1 = 39 - INT(LEN(Aviso[nNoDeError])/2)  
ELSE  
  X1 = 38 - INT(LEN(Aviso[nNoDeError])/2)  
ENDIF  
X2 = X1 + LEN(Aviso[nNoDeError]) + 1  
Ok( 12, INT((X2-X1-5)/2)+X1,'15/7,,,,','0/4,,,,',7*16+14)  
INKEY(10)  
VentanaCML(8,x1,14,x2,Marco7,'15/4,,,,')  
SETCOLOR(ColoresViejos)  
RETURN NIL
```

...../

...../

* Este es el módulo de rutinas comunes a varias funciones *

...../

/*

* CargarArch() --> NIL

*

* Revisa la existencia de las bases de datos junto con sus
* archivos indexados y memorándum. En caso de faltar una base
* de datos o un archivo memorándum, el programa se interrumpe,
* generando un aviso de error.

*

*/

FUNCTION CargarArch

GAux = SAVESCREEN(0,0,24,79)

Caja(' Leyendo archivos de origen. Espere un momento... ','15/3,....')

OK(12, 37,'15/7,....','0/3,....','7*16+4)

IF !FILE('Cursos.dbf')

Error(24)

DBCCREATE('Cursos.dbf', Estructura[1])

SELECT A

USE Cursos ALIAS Cur

INDEX ON CveCurso TO NtxCurso

SET INDEX TO NtxCurso

ELSE

IF !FILE('Cursos.dbt')

Error(27)

Fin = .T.

nOpcion1 = 7

ELSE

SELECT A

USE Cursos ALIAS Cur

IF !FILE('NtxCurso.ntx')

INDEX ON CveCurso TO NtxCurso

ENDIF

SET INDEX TO NtxCurso

ENDIF


```
ENDIF
IF !FILE('Orden.dbf')
  Error(25)
  DBCREATE('Orden.dbf', Estructura[2])
  SELECT B
  USE Orden ALIAS Ord
  INDEX ON CveCurso+NombreAlum TO NtxOrden
  SET INDEX TO NtxOrden
ELSE
  IF !FILE('Orden.dbt')
    Error(28)
    Fin = .T.
    nOpcion1 = 7
  ELSE
    SELECT B
    USE Orden ALIAS Ord
    IF !FILE('NtxOrden.ntx')
      INDEX ON CveCurso+NombreAlum TO NtxOrden
    ENDIF
    SET INDEX TO NtxOrden
  ENDIF
ENDIF
ENDIF
IF !FILE('Exposito.dbf')
  Error(26)
  DBCREATE('Exposito.dbf', Estructura[3])
  SELECT C
  USE Exposito ALIAS Exp
  INDEX ON CveExpo TO NtxExpo
  SET INDEX TO NtxExpo
ELSE
  IF !FILE('Exposito.dbt')
    Error(29)
    Fin = .T.
    nOpcion1 = 7
  ELSE
    SELECT C
    USE Exposito ALIAS Exp
    IF !FILE('NtxExpo.ntx')
      INDEX ON CveExpo TO NtxExpo
    ENDIF
    SET INDEX TO NtxExpo
  ENDIF
ENDIF
ENDIF
IF !FILE('Carreras.dbf')
  Error(40)
  DBCREATE('Carreras.dbf', Estructura[4])
```

```
SELECT D
USE Carreras ALIAS Car
INDEX ON CveProfesi TO NtxCarr
SET INDEX TO NtxCarr
ELSE
  IF !FILE('Carreras.dbt')
    Error(41)
    Fin = .T.
    nOpcion1 = 7
  ELSE
    SELECT D
    USE Carreras ALIAS Car
    IF !FILE('NtxCarr.ntx')
      INDEX ON CveProfesi TO NtxCarr
    ENDIF
    SET INDEX TO NtxCarr
  ENDIF
ENDIF
RESTSCREEN(0,0,24,79,GAux)
RETURN NIL
```

/***

- * Seleccion (<nPanta>, <nOpcion>) --> nAlternativa
- *
- * Permite seleccionar una opción del menú <nPanta>
- * devolviendo el número de opción elegida en <nAlternativa>.
- * El primer elemento que se enfatiza para seleccionar es
- * <nOpcion>.
- *

*/

```
FUNCTION Seleccion (nPanta,nOpcion)
  LOCAL Tecla := 0, PosArreglo

  DO WHILE Tecla != 13 .AND. Tecla != 27
    Enfatiza(nPanta, nOpcion)
    Tecla = INKEY(0)
  DO CASE
    CASE Tecla = 27
      Desenfatiza(nPanta, nOpcion)
      nOpcion = LEN(Pantalla[nPanta,10])
    CASE Tecla = 28
      VarAyuda = nPanta
      AyudaMod()
    CASE Tecla = 09
```

```
Presenta()
CASE Tecla = 08
  AcercaDe()
CASE Tecla = 13
  Boton(3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8],;
    Pantalla[nPanta,8]+LEN(Pantalla[nPanta,10,nOpcion])-1)
CASE Tecla = 5 .OR. Tecla = 19
  Desenfatisa(nPanta, nOpcion)
  IF nOpcion > 1
    --nOpcion
  ELSE
    nOpcion = LEN(Pantalla[nPanta, 10])
  ENDIF
CASE Tecla = 4 .OR. Tecla = 24
  Desenfatisa(nPanta, nOpcion)
  IF nOpcion < LEN(Pantalla[nPanta,10])
    ++nOpcion
  ELSE
    nOpcion = 1
  ENDIF
OTHERWISE
  PosArreglo = ASCAN(Letras,UPPER(CHR(Tecla)))
  IF PosArreglo <> 0
    Desenfatisa(nPanta, nOpcion)
    nOpcion = PosArreglo
    Enfatisa(nPanta, nOpcion)
    Boton(3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8],;
      Pantalla[nPanta,8]+LEN(Pantalla[nPanta,10,nOpcion])-1)
    Tecla = 13
  ENDIF
ENDCASE
ENDDO
RETURN nOpcion
```

```
/*
 * Desenfatisa( nPanta, nOpcion ) --> NIL
 *
 * Desenfatisa la opción actual del menú
 * activo de la función SELECCION
 *
 */
FUNCTION Desenfatisa ( nPanta, nOpcion )

  SETCOLOR('0/7 ,,,,,')
```

```
@ 3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8] SAY ''
@ 3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8]+;
LEN(Pantalla[nPanta,10,nOpcion])-1 SAY ''
Mod_Color(3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8]+1,;
3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8]+;
LEN(Pantalla[nPanta,10,nOpcion])-2,7*16+0)

// Se enfatizan las teclas calientes del menú
Mod_Color(3*nOpcion+Pantalla[nPanta,7]-3,;
Pantalla[nPanta,8]+AT(Letras[nOpcion],;
Pantalla[nPanta,10,nOpcion])-1,;
3*nOpcion+Pantalla[nPanta,7]-3,;
Pantalla[nPanta,8]+AT(Letras[nOpcion],;
Pantalla[nPanta,10,nOpcion])-1,;
7*16+4)
RETURN NIL
```

```
/***
* Enfatiza( nPanta, nOpcion ) --> NIL
*
* Enfatiza la opción actual del menú
* activo de la función SELECCION
*/
FUNCTION Enfatiza ( nPanta, nOpcion )
```

```
SETCOLOR('15/7',,,,,')
@ 3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8] SAY CHR(16)
@ 3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8]+;
LEN(Pantalla[nPanta,10,nOpcion])-1 SAY CHR(17)
Mod_Color(3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8],;
3*nOpcion+Pantalla[nPanta,7]-3, Pantalla[nPanta,8]+;
LEN(Pantalla[nPanta,10,nOpcion])-1,7*16+15)

// Se enfatizan las teclas calientes del menú
Mod_Color(3*nOpcion+Pantalla[nPanta,7]-3,;
Pantalla[nPanta,8]+AT(Letras[nOpcion],;
Pantalla[nPanta,10,nOpcion])-1,;
3*nOpcion+Pantalla[nPanta,7]-3,;
Pantalla[nPanta,8]+AT(Letras[nOpcion],;
Pantalla[nPanta,10,nOpcion])-1,;
7*16+4)
RETURN NIL
```

/***

* Presenta() --> NIL

*

* Pone la pantalla de la versión del programa.

*

*/

FUNCTION Presenta

SET CURSOR OFF

VentanaExplosiva(4,26,19,53,Marco7,'15/7,.....')

SombraMarco(4,26,19,53)

SETCOLOR('15/7,.....')

@ 06, 36 SAY 'U N A M'

@ 07, 33 SAY 'CAMPUS ARAGON'

@ 09, 30 SAY 'Cesar Mandujano Luna'

@ 11, 36 SAY 'Presenta'

@ 15, 32 SAY 'Abr. 10, 1996'

SETCOLOR('0/7,.....')

@ 13, 34 SAY 'DEC Ver. 1.0'

OK(17,37,'15/4,.....','0/7,.....',4*16+14)

INKEY(10)

Boton(17,37,42)

VentanaCML(4,26,19,53,Marco7,'15/7,.....')

RETURN NIL

/***

* PonPanta (<nPanta>) --> NIL

*

* Coloca el número <nNoPanta> de menú.

*

*/

FUNCTION PonPanta (nPanta)

LOCAL i

Fondo()

TeclasVal()

Letras := {}

SET CURSOR OFF

VentanaExplosiva(Pantalla[nPanta,1],Pantalla[nPanta,2],Pantalla[nPanta,3];;
Pantalla[nPanta,4],Pantalla[nPanta,5],'15/1,.....')

SombraMarco(Pantalla[nPanta,1],Pantalla[nPanta,2],Pantalla[nPanta,3];;
Pantalla[nPanta,4])

```
SETCOLOR('15/0,....')
@ Pantalla[nPanta,1], Pantalla[nPanta,2] SAY ;
REPLICATE(CHR(219),INT((Pantalla[nPanta,4]-Pantalla[nPanta,2]+
1-LEN(Pantalla[nPanta,6]))/2)+1)
@ Pantalla[nPanta,1], COL() SAY Pantalla[nPanta,6]
@ Pantalla[nPanta,1], COL() SAY ;
REPLICATE(CHR(219),Pantalla[nPanta,4]-COL()+1)
SETCOLOR('0/7,.....')
FOR i = 1 TO LEN(PANTALLA[nPanta,10])
  @ 3*i+Pantalla[nPanta,7]-3, Pantalla[nPanta,8];
  SAY Pantalla[nPanta,10,i]
  SombraMensaje(3*i+Pantalla[nPanta,7]-3, Pantalla[nPanta,8];;
  Pantalla[nPanta,8]+LEN(Pantalla[nPanta,10,i])-1,'0/1,.....')

  // Se incorporan las teclas calientes
  AADD(Letras,SUBSTR(LTRIM(Pantalla[nPanta,10,i]),1,1))

  // Se enfatizan las teclas calientes del menú
  Mod_Color(3*i+Pantalla[nPanta,7]-3,;
  Pantalla[nPanta,8]+AT(Letras[i],Pantalla[nPanta,10,i])-1,;
  3*i+Pantalla[nPanta,7]-3,;
  Pantalla[nPanta,8]+AT(Letras[i],Pantalla[nPanta,10,i])-1,;
  7*16+4)
NEXT
SETCOLOR('14/1,....')
FOR i = 1 TO LEN(PANTALLA[nPanta,11])
  @ 3*i+Pantalla[nPanta,7]-3, Pantalla[nPanta,9] SAY CHR(248)+'+';
  Pantalla[nPanta,11,i]
NEXT
RETURN NIL
```

/*

```
* TeclasVal() --> NIL
*
* Coloca en el último renglón del monitor las
* teclas validas para usar un menú.
*
```

*/

```
FUNCTION TeclasVal
```

```
SETCOLOR('0/7,....')
@ 24, 00 SAY ' ESC = Salir ' + CHR(17)+CHR(196)+CHR(196)+CHR(217)+;
' = Confirmar Tab = Versión BS = Acerca F1 = Ayuda'
SETCOLOR('15/7,.....')
```

@ 24, 67 SAY CHR(179)
RETURN NIL

```
/*  
* ConCan() --> nAccion  
*  
* Permite que el usuario confirme o cancele la  
* acción a realizar, ya sea una alta, baja, consulta,  
* modificación o reporte.  
*/  
FUNCTION ConCan  
  LOCAL i, j, Tecla  
  
  j = 1  
  Tecla = 0  
  DO WHILE Tecla != 13 .AND. Tecla != 27  
    SETCOLOR('15/7,....')  
    IF j = 1  
      @ 22, 22 SAY CHR(16)+' Aceptar '+CHR(17)  
      Mod_Color(22,24,22,24,7*16+4)  
    ELSE  
      @ 22, 44 SAY CHR(16)+' Cancelar '+CHR(17)  
      Mod_Color(22,46,22,46,7*16+4)  
    ENDIF  
    Tecla = INKEY(0)  
    DO CASE  
      CASE Tecla = 13  
        Boton(22,22*j,22*j+11)  
        SETCOLOR('0/7,....')  
        IF j = 1  
          @ 22, 22 SAY ' Aceptar '  
          Mod_Color(22,24,22,24,7*16+4)  
          i = 6  
        ELSE  
          @ 22, 44 SAY ' Cancelar '  
          Mod_Color(22,46,22,46,7*16+4)  
          i = 0  
        ENDIF  
      CASE Tecla = 27  
        SETCOLOR('0/7,....')  
        IF j = 1  
          @ 22, 22 SAY ' Aceptar '  
          Mod_Color(22,24,22,24,7*16+4)
```

```
ELSE
  @ 22, 44 SAY ' Cancelar '
  Mod_Color(22,46,22,46,7*16+4)
ENDIF
i = 0
CASE Tecla = 28
  VarAyuda = 8
  AyudaMod()
CASE Tecla = 19
  SETCOLOR('0/7,.....')
  IF j = 1
    @ 22, 22 SAY ' Aceptar '
    Mod_Color(22,24,22,24,7*16+4)
    j = 2
  ELSE
    @ 22, 44 SAY ' Cancelar '
    Mod_Color(22,46,22,46,7*16+4)
    j = 1
  ENDIF
CASE Tecla = 4
  SETCOLOR('0/7,.....')
  IF j = 1
    @ 22, 22 SAY ' Aceptar '
    Mod_Color(22,24,22,24,7*16+4)
    j = 2
  ELSE
    @ 22, 44 SAY ' Cancelar '
    Mod_Color(22,46,22,46,7*16+4)
    j = 1
  ENDIF
CASE Tecla = 5 .OR. Tecla = 18
  SETCOLOR('0/7,.....')
  IF j = 1
    @ 22, 22 SAY ' Aceptar '
    Mod_Color(22,24,22,24,7*16+4)
  ELSE
    @ 22, 44 SAY ' Cancelar '
    Mod_Color(22,46,22,46,7*16+4)
  ENDIF
i = 2
Tecla = 27
CASE Tecla = 3 .OR. Tecla = 24
  SETCOLOR('0/7,.....')
  IF j = 1
    @ 22, 22 SAY ' Aceptar '
    Mod_Color(22,24,22,24,7*16+4)
```



```
ELSE
  @ 22, 44 SAY ' Cancelar '
  Mod_Color(22,46,22,46,7*16+4)
ENDIF
i = 1
Tecla = 27
OTHERWISE
IF UPPER(CHR(Tecla)) = 'A'
  SETCOLOR('0/7,....')
  @ 22, 44 SAY ' Cancelar '
  Mod_Color(22,46,22,46,7*16+4)
  j = 1
  SETCOLOR('15/7,....')
  @ 22, 22 SAY CHR(16)+' Aceptar '+CHR(17)
  Mod_Color(22,24,22,24,7*16+4)
  Boton(22,22*j,22*j+11)
  SETCOLOR('0/7,....')
  @ 22, 22 SAY ' Aceptar '
  Mod_Color(22,24,22,24,7*16+4)
  i = 6
  Tecla = 13
ELSE
  IF UPPER(CHR(Tecla)) = 'C'
    SETCOLOR('0/7,....')
    @ 22, 22 SAY ' Aceptar '
    Mod_Color(22,24,22,24,7*16+4)
    j = 2
    SETCOLOR('15/7,....')
    @ 22, 44 SAY CHR(16)+' Cancelar '+CHR(17)
    Mod_Color(22,46,22,46,7*16+4)
    Boton(22,22*j,22*j+11)
    SETCOLOR('0/7,....')
    @ 22, 44 SAY ' Cancelar '
    Mod_Color(22,46,22,46,7*16+4)
    i = 0
    Tecla = 13
  ENDIF
ENDIF
ENDCASE
ENDDO
RETURN
```

```
/*  
* PonConCan() --> NIL  
*  
* Coloca los mensajes " Aceptar " y  
* " Cancelar " junto con su sombra.  
*  
*/  
FUNCTION PonConCan
```

```
    SETCOLOR('0/7,...')  
    @ 22, 22 SAY ' Aceptar '  
    @ 22, 44 SAY ' Cancelar '  
    SombraMensaje(22,22,33,'0/1,...')  
    SombraMensaje(22,44,55,'0/1,...')  
    Mod_Color(22,24,22,24,7*16+4)  
    Mod_Color(22,46,22,46,7*16+4)  
RETURN NIL
```

```
/*  
* CajaCampos() --> NIL  
*  
* Coloca la ventana dentro de la cual se introduce  
* el contenido del campo memorándum OBSERVA del  
* registro actual, de la base de datos vigente.  
*  
*/  
FUNCTION CajaCampos
```

```
    l1 = 5  
    c1 = 5  
    l2 = 18  
    c2 = 75  
    VentanaExplosiva( l1-1, c1-1, l2+2, c2+1, Marco7, '15/2,...')  
    SombraMarco( l1-1, c1-1, l2+2, c2+1 )  
    SETCOLOR('15/2,...')  
    @ l2+1, c1+8 SAY 'ESC = Salir ' + CHR(17)+CHR(196)+CHR(196)+CHR(217)+;  
        ' = Confirmar PgDn, PgUp, '+;  
        CHR(27)+CHR(24)+CHR(25)+CHR(26)+' = Mover'  
    @ l1+1, c1-1 SAY CHR(219)  
    @ l1+1, c2+1 SAY CHR(219)  
    @ l2, c1-1 SAY CHR(219)  
    @ l2, c2+1 SAY CHR(219)  
    RamaV(l2-11,c2+1,l2-1,FlechaV,'15/0,...')  
RETURN NIL
```

```
/*  
* ControlCon( <nRen>, <nCol>, <IVerdad> ) --> NIL  
*  
* Controla la apertura de las opciones de la pantalla  
* de consultas de cursos, ordenes de pago y expositores.  
* Aquí <nRen> y <nCol> son las coordenadas en la cual  
* está ubicada la opción "Memorándum" y <IVerdad> en caso  
* de ser .F., indica que el campo memorándum no se puede  
* modificar de lo contrario este si es modificable.  
*  
*/
```

```
FUNCTION ControlCon(nRen, nCol, IVerdad)  
  LOCAL i, Salte
```

```
  Salte = READEXIT(.T.)  
  i = 2  
  DO WHILE i != 0  
    DO CASE  
      CASE i = 1  
        i = 2  
      CASE i = 2  
        i = ControlMemo(nRen,nCol,IVerdad)  
      CASE i = 3  
        i = ControlOk()  
      CASE i = 4  
        i = 3  
    ENDCASE  
  ENDDO  
  READEXIT(Salte)  
RETURN NIL
```

```
/*  
* ControlMemo( <nRen>, <nCol>, <IVerdad> ) --> nAccion  
*  
* Controla la apertura de la pantalla de memorándum's  
* para cursos, ordenes de pago y expositores. Aquí <nRen>  
* y <nCol> son las coordenadas en la cual está ubicada la  
* opción "Memorándum" y <IVerdad> en caso de ser .F.,  
* indica que el campo memorándum no se puede modificar  
* de lo contrario este si es modificable.  
*  
*/
```

```
FUNCTION ControlMemo (nRen, nCol, IVerdad)
LOCAL Tecla, i

Mod_Color(nRen,nCol,nRen,nCol+9,7*16+15)
Tecla = 0
DO WHILE Tecla != 13 .AND. Tecla != 27
  Tecla = INKEY(0)
  DO CASE
    CASE Tecla = 13
      Mod_Color(nRen,nCol,nRen,nCol+9,15)
      EditaMemo(IVerdad)
      i = 3
    CASE Tecla = 27
      Mod_Color(nRen,nCol,nRen,nCol+9,15)
      i = 0
    CASE Tecla = 28
      VarAyuda = 6
      AyudaMod()
    CASE Tecla = 24 .OR. Tecla = 3
      i = 3
      Mod_Color(nRen,nCol,nRen,nCol+9,15)
      Tecla = 27
    CASE Tecla = 5 .OR. Tecla = 18
      i = 4
      Mod_Color(nRen,nCol,nRen,nCol+9,15)
      Tecla = 27
  ENDCASE
ENDDO
RETURN i
```

/*

* ControlOk() --> nAccion

*

* Espera la pulsación de ESC, ENTER o teclas de movimiento
* para consultas de cursos, ordenes de pago y expositores.

*

*/

```
FUNCTION ControlOk
LOCAL Tecla, i

SETCOLOR('15/4,,,')
@ 22,37 SAY CHR(16)+' OK '+CHR(17)
Mod_Color(22,39,22,39,4*16+14)
Tecla = 0
```

```
DO WHILE Tecla != 13 .AND. Tecla != 27
  Tecla = INKEY(0)
  DO CASE
    CASE Tecla = 13 .OR. Tecla = 27 .OR. UPPER(CHR(Tecla)) = 'O'
      Boton(22,37,42)
      SETCOLOR('0/7,....')
      @ 22,37 SAY ' OK '
      Mod_Color(22,39,22,39,7*16+4)
      i = 0
      Tecla = 13
    CASE Tecla = 28
      VarAyuda = 7
      AyudaMod()
    CASE Tecla = 5 .OR. Tecla = 18
      SETCOLOR('0/7,....')
      @ 22,37 SAY ' OK '
      Mod_Color(22,39,22,39,7*16+4)
      i = 2
      Tecla = 27
    CASE Tecla = 3 .OR. Tecla = 24
      SETCOLOR('0/7,....')
      @ 22,37 SAY ' OK '
      Mod_Color(22,39,22,39,7*16+4)
      i = 4
      Tecla = 27
  ENDCASE
ENDDO
RETURN i
```

*/***

```
* EditaTexto( < nLong > ) -> NIL
*
* Realiza la edición en pantalla de los
* diversos reportes generados. El parámetro
* < nLong > es el total de caracteres por
* línea a visualizar.
*
```

**/*

```
FUNCTION EditaTexto(nLongLinea)
  LOCAL MemoAux

  SETCOLOR('1/3,....')
  @ 0,0,24,79 BOX REPLICATE(' ',9)
  I1 = 2
```

```
c1 = 0
l2 = 22
c2 = 79
VentanaExplosiva( l1, c1, l2, c2, Marco7, '15/1,....')
SETCOLOR('0/3,....')
@ 24, 0 SAY ' ESC y CTRL-W = Salir '+';
CHR(17)+CHR(196)+CHR(196)+CHR(217)+;
' Pg/Dn Pg/Up '+CHR(27)+CHR(24)+CHR(25)+CHR(26)+;
' HOME END TAB y BS = Desplazarse'
RamaH(22,36,77,FlechaH,'1/7,....')
RamaV(09,79,20,FlechaV,'1/7,....')
SETCOLOR('7/1,....')
SET CURSOR ON
SET KEY 28 TO
MemoAux = MEMOEDIT(cTexto,l1+1, c1+1, l2-1, c2-1,..F,..nLongLinea)
SET KEY 28 TO Ayuda
SET CURSOR OFF
RETURN NIL
```

/***

* EditaMemo (<IVerdad>) --> NIL

*

* Realiza la edición del campo memorándum **OBSERVA**

* del registro vigente de la base de datos activa. Si

* <IVerdad> es verdadero, se permite la modificación

* del contenido de dicho campo, de lo contrario sólo

* se permite su visualización.

*

*/

FUNCTION EditaMemo (IVerdad)

```
l1 = 6
```

```
c1 = 15
```

```
l2 = 18
```

```
c2 = 65
```

```
VentanaExplosiva( l1, c1, l2, c2, Marco7, '15/3,....')
```

```
SombraMarco( l1, c1, l2, c2 )
```

```
RamaH(l2-2, c1, c2, RamaH6, '15/3,....')
```

```
RamaH(l1+2, c1, c2, RamaH5, '15/3,....')
```

```
SETCOLOR('0/3,....')
```

```
@ l2-1, c1+10 SAY ' ESC = Salir Ctrl-W = Salvar'
```

```
@ l1+1, c1+18 SAY 'Observaciones'
```

```
RamaV(l1+3,c2,l2-3,FlechaV,'15/0,....')
```

```
SETCOLOR('15/3,....')
```

```
SET CURSOR ON
SET KEY 28 TO
Memorandum = MEMOEDIT(Memorandum,1+3, c1+1, l2-3, c2-1, lVerdad)
SET KEY 28 TO Ayuda
SET CURSOR OFF
RESTSCREEN( 0, 0, 24, 79, G)
RETURN NIL
```

```
/***
```

```
* PantaClave( <nExhibe>, <cRelleno> ) --> NIL
```

```
*
```

- * Coloca la pantalla que permite teclear la clave
- * de un curso, orden de pago o expositor a editar. El
- * parámetro <nExhibe> selecciona el índice de la
- * variable LISTA. En caso de que el usuario no teclee
- * toda la longitud solicitada de caracteres para la
- * clave, entonces el resto de caracteres de la clave
- * se llenan del carácter <cRelleno>.

```
*
```

```
*/
```

```
FUNCTION PantaClave (nExhibe, cRelleno)
  LOCAL Aux, x1, x2
  LOCAL Lista := { 'CVECURSO', 'CVECURSO', 'CVEEXPO', 'CVEPROFESI' }
  PRIVATE Campo, TamClave

  Campo = Lista[nExhibe]
  Aux = { 'l curso', 'l curso', 'l expositor', 'la carrera' }
  Clave = REPLICATE(' ', LEN(&Campo))
  Caja(' Teclee la clave de'+Aux[nExhibe]+' : ', '15/3, ,,,,')
  SETCOLOR('15/3, 15/0, ,,,,')
  TamClave = REPLICATE('9', LEN(&Campo))
  IF LEN(&Campo) % 2 = 0
    X1 = 39 - INT(LEN(Clave)/2)
  ELSE
    X1 = 38 - INT(LEN(Clave)/2)
  ENDIF
  X2 = X1 + LEN(Clave) + 1
  SETCOLOR('15/7, ,,,,')
  SET KEY 28 TO
  @ 12, INT((X2-X1-LEN(&Campo)-1)/2)+X1 GET Clave PICT '&TamClave'
  SETCOLOR('15/3, ,,,,')
  SET CURSOR ON
  READ
  SET KEY 28 TO Ayuda
```

```
VentanaCML(9,22,13,55,Marco7,'15/3,,,,')
SET CURSOR OFF
Clave = STRTRAN(Clave,' ',cRelleno)
IF LEN(Clave) = 0
  Clave = REPLICATE(' ',LEN(&Campo))
ENDIF
RETURN NIL
```

```
/***
* PalabraInc( <NoBD> ) --> NIL
*
* Coloca el menú correspondiente de la base de datos
* para consultas por cadena incluida. Aquí <NoBD> puede
* ser : 1 = Cursos; 2 = Ordenes; 3 = Expositores y
* 4 = Carreras.
*
*/
```

```
FUNCTION PalabraInc(NoBD)
  LOCAL i, nOpcion, Letrero, MatrizChar, Alternativ
  LOCAL X1, X2, Continuar, Activado
  PRIVATE CualCampo, Caracteres
```

```
MatrizChar = {}
Alternativ = {}
FOR i = 1 TO LEN(Estructura[NoBD])
  IF Explica[NoBD,i] != NIL
    AADD(MatrizChar,Explica[NoBD,i])
    AADD(Alternativ,i)
  ENDIF
NEXT
```

```
ACTIVATE(Alternativ)
AADD(MatrizChar,'Salir. ')
Activado = .F.
Continuar = .T.
SET WRAP ON
DO WHILE Continuar
  SETCOLOR('15/2,15/0,,,,')
  IF LEN(MatrizChar) % 2 = 0
    X1 = 12 - INT(LEN(MatrizChar)/2)
  ELSE
    X1 = 11 - INT(LEN(MatrizChar)/2)
  ENDIF
  X2 = X1 + LEN(MatrizChar) + 1
  GAux = SaveScreen(X1,20,X2+1,62)
  @ X1,20,X2,60 BOX Marco7+' '
```



```
SombraMarco(X1,20,X2,60)
FOR i = 1 TO LEN(MatrizChar)
  @ X1+i, 21 PROMPT MatrizChar[i]
NEXT
nOpcion = 1
MENU TO nOpcion
IF nOpcion > 0 .AND. nOpcion <> LEN(MatrizChar)
  RESTSCREEN(X1,20,X2+1,62,GAux)
  CualCampo = Estructura[NoBD,Alternativ[nOpcion],1]
  Caracteres = SPACE(Estructura[NoBD,Alternativ[nOpcion],3])
  Letrero = ' TECLEE LA CADENA INCLUIDA EN : '+
    UPPER(RTRIM(MatrizChar[nOpcion]))+' '
  Caja( Letrero, '15/5,....')
  SETCOLOR('15/3,15/0,....')
  IF LEN(&CualCampo) % 2 = 0
    X1 = 39 - INT(LEN(&CualCampo)/2)
  ELSE
    X1 = 38 - INT(LEN(&CualCampo)/2)
  ENDIF
  X2 = X1 + LEN(&CualCampo) + 1
  SETCOLOR('15/7,....')
  SET KEY 28 TO
  @ 12, INT((X2-X1-LEN(&CualCampo)-1)/2)+X1 GET Caracteres PICTURE '@!'
  READINSERT(.F.)
  SETCOLOR('15/3,....')
  SET CURSOR ON
  READ
  RESTSCREEN(0,0,24,79,G)
  SET CURSOR OFF
  IF LASTKEY() != 27
    Caracteres = RTRIM(Caracteres)
    LOCATE FOR '&Caracteres'&CualCampo
    IF FOUND()
      SET FILTER TO '&Caracteres'&CualCampo
      GO TOP
      Activado = .T.
      Continuar = .F.
    ELSE
      Error(42)
    ENDIF
  ENDIF
  SET KEY 28 TO AYUDA
ELSE
  Continuar = .F.
ENDIF
ENDDO
```

SET WRAP OFF
RETURN Activado

```
/***  
* Confirmar ( <nMensaje> ) --> nAccion  
*  
* Presenta en un marco el número de mensaje  
* <nMensaje> de la variable local AVISO y permite  
* al usuario elegir si desea proceder con la acción  
* especificada en el mensaje o no. En caso de que  
* elija " Si ", <nAccion> será igual a 1 de otra manera  
* <nAccion> será igual a 2.  
*  
*/
```

```
FUNCTION Confirmar (nMensaje)  
LOCAL i, Tecla, Aviso
```

```
Aviso = { ' ¿ Desea salir del programa... ? ' ;  
' ¿ Desea realizar otra alta... ? ' ;  
' ¿ Desea realizar otra baja... ? ' ;  
' ¿ Desea realizar otra consulta... ? ' ;  
' ¿ Desea realizar otra modificación... ? ' ;  
' ¿ Desea verificar otro avance... ? ' ;  
' ¿ Desea otra consulta de cursos por vencer... ? ' ;  
' ¿ Desea otra lista para profesores... ? ' ;  
' El archivo '+NomArch+' existe ¿ Desea reescribirlo ? ' ;  
' ¿ Desea otro registro de inscripción... ? ' ;  
' ¿ Desea reportar un oficio existente... ? ' ;  
' ¿ Desea grabar este reporte en un archivo... ? ' ;  
' ¿ Desea continuar con la grabación de este registro ? ' ;  
' ¿ Desea confirmar la clave '+PassWord+'... ? ' ;  
' ¿ Desea imprimir este reporte... ? ' ;  
' Se ha sobrepasado el cupo para este curso '+ ;  
' ¿ Continúa con la grabación ? ' }
```

```
Caja( Aviso[nMensaje], '15/2,...' )  
SETCOLOR( '0/7,...' )  
@ 12, 29 SAY ' Si '  
@ 12, 43 SAY ' No '  
Mod_Color( 12, 31, 12, 31, 7*16+4 )  
Mod_Color( 12, 45, 12, 45, 7*16+4 )  
SombraMensaje( 12, 29, 34, '0/2,...' )  
SombraMensaje( 12, 43, 48, '0/2,...' )
```

```
i = 1
Tecla = 0
DO WHILE Tecla != 13 .AND. Tecla != 27
  SETCOLOR('15/7,....')
  IF i = 1
    @ 12, 29 SAY Chr(16)+' Si '+Chr(17)
    Mod_Color(12,31,12,31,7*16+4)
  ELSE
    @ 12, 43 SAY Chr(16)+' No '+Chr(17)
    Mod_Color(12,45,12,45,7*16+4)
  ENDIF
  Tecla = INKEY(0)
DO CASE
  CASE Tecla = 13
    Boton(12,14*i+15,14*i+20)
  CASE Tecla = 27
    i = 2
  CASE Tecla = 19
    SETCOLOR('0/7,.....')
    IF i = 1
      @ 12, 29 SAY ' Si '
      Mod_Color(12,31,12,31,7*16+4)
      i = 2
    ELSE
      @ 12, 43 SAY ' No '
      Mod_Color(12,45,12,45,7*16+4)
      i = 1
    ENDIF
  CASE Tecla = 4
    SETCOLOR('0/7,.....')
    IF i = 1
      @ 12, 29 SAY ' Si '
      Mod_Color(12,31,12,31,7*16+4)
      i = 2
    ELSE
      @ 12, 43 SAY ' No '
      Mod_Color(12,45,12,45,7*16+4)
      i = 1
    ENDIF
  OTHERWISE
    IF UPPER(CHR(Tecla)) = 'S'
      SETCOLOR('0/7,.....')
      @ 12, 43 SAY ' No '
      Mod_Color(12,45,12,45,7*16+4)
      SETCOLOR('15/7,....')
      @ 12, 29 SAY CHR(16)+' Si '+CHR(17)
```

```
Mod_Color(12,31,12,31,7*16+4)
Boton(12,29,34)
Mod_Color(12,29,12,34,7*16)
i = 1
Tecla = 13
ELSE
  IF UPPER(CHR(Tecla)) = 'N'
    SETCOLOR('0/7,....')
    @ 12, 29 SAY ' Si '
    Mod_Color(12,31,12,31,7*16+4)
    SETCOLOR('15/7,....')
    @ 12, 43 SAY CHR(16)+' No '+CHR(17)
    Mod_Color(12,45,12,45,7*16+4)
    Boton(12,43,48)
    Mod_Color(12,43,12,48,7*16)
    i = 2
    Tecla = 13
  ENDIF
ENDIF
ENDCASE
ENDDO
VentanaCML (8,22,14,55,Marco7,'15/2,....')
RETURN i
```

```
/***
* Caja ( <cMensaje>, <cAtributos> ) --> NIL
*
* Coloca un marco de color <cAtributos> en el centro
* de la pantalla, introduciendo en el mismo el mensaje
* <cMensaje>.
*
*/
FUNCTION Caja ( cMensaje, cAtributos)
  LOCAL x1, x2

  IF LEN(cMensaje) % 2 = 0
    x1 = 39 - INT(LEN(cMensaje)/2)
  ELSE
    x1 = 38 - INT(LEN(cMensaje)/2)
  ENDIF
  x2 = x1 + LEN(cMensaje) + 1
  VentanaExplosiva(8,x1,14,x2,Marco7, cAtributos)
  SombraMarco(8,x1,14,x2)
  RamaH(10,x1,x2,RamaH6, cAtributos)
```

```
SETCOLOR(cAtributos)
@ 9, X1+1 SAY cMensaje
RETURN NIL
```

```
/***
```

```
* OK2 ( ) --> NIL
*
* Coloca el mensaje " OK " en las coordenadas 22,37.
*
*/
```

```
FUNCTION OK2
```

```
    SETCOLOR('0/7,....')
    @ 22,37 SAY ' OK '
    SombraMensaje(22,37,42,'0/1,....')
    Mod_Color(22,39,22,39,7*16+4)
RETURN NIL
```

```
/***
```

```
* PantaBases ( <cEncabeza> ) --> NIL
*
* Coloca un marco que abarca toda la pantalla teniendo
* como encabezado el parámetro <cEncabeza>.
*
*/
```

```
FUNCTION PantaBases (cEncabeza)
```

```
    @ 0,0,24,79 BOX Marco7+ ' ' COLOR '15/1,....'
    SETCOLOR('15/0,....')
    @ 0, 0 SAY REPLICATE(CHR(219),INT((80-LEN(cEncabeza))/2)+1)
    @ 0, COL() SAY cEncabeza
    @ 0, COL() SAY REPLICATE(CHR(219),79-COL()+1)
RETURN NIL
```

```
/*  
* Ok ( <nY1>, <nX1>, <cColor1>, <cColor2>, <nColor3> ) --> NIL  
*  
* Escribe en las coordenadas <nY1>, <nX1> el mensaje  
* " OK " de color <cColor1>. El parámetro <cColor2>  
* determina el color de la sombra que llevará el mensaje  
* " OK " y <nColor3> es el color de la letra "O".  
*  
*/  
FUNCTION Ok (nY1,nX1,cColor1,cColor2,nColor3)  
  LOCAL ColoresViejos := SETCOLOR()  
  
  SETCOLOR(cColor1)  
  @ nY1, nX1 SAY CHR(16)+' OK '+CHR(17)  
  SombraMensaje(nY1,nX1,nX1+5,cColor2)  
  Mod_Color(nY1,nX1+2,nY1,nX1+2,nColor3)  
  SETCOLOR(ColoresViejos)  
RETURN NIL
```

```
/*  
* Fondo() --> NIL  
*  
* Coloca el fondo de los menús.  
*  
*/  
FUNCTION Fondo  
  
  SETCOLOR('0/7,....')  
  @ 00, 00 SAY ' DEPARTAMENTO DE EDUCACION CONTINUA CAMPUS  
ARAGON '  
  @ 00, 58 SAY ' Cesar Mandujano Luna '  
  @ 24, 00 CLEAR TO 24,79  
  @ 01, 00, 23, 79 BOX REPLICATE(CHR(176),9) COLOR '3/1,....'  
  SETCOLOR('15/7,....')  
  @ 00, 57 SAY CHR(179)  
RETURN NIL
```

```
/*  
*  
* CargarRec() --> NIL  
*  
* Asigna contenido a las variable Recibo, Cantidad,  
* Oficio, FechaC y FechaD, según el registro vigente  
* del campo Recibos del archivo Ordenes.DBF.  
*  
*/  
FUNCTION CargarRec  
  LOCAL i  
  
  CadRec = Recibos  
  ArrayVacio()  
  AsignaRec()  
  FOR i = 1 TO Tope  
    AADD(Recibo, VAL(SUBSTR(CadRec,56*i-54,12)))  
    AADD(Cantidad, VAL(SUBSTR(CadRec,56*i-42,12)))  
    AADD(Oficio, SUBSTR(CadRec,56*i-30,12))  
    AADD(FechaC, CTOD(SUBSTR(CadRec,56*i-18,10)))  
    AADD(FechaD, CTOD(SUBSTR(CadRec,56*i-8,10)))  
  NEXT  
RETURN NIL
```

```
/*  
* AsignaRec() --> NIL  
*  
* Almacena el número de recibos pagados según el  
* registro vigente.  
*  
*/  
FUNCTION AsignaRec  
  
  Tope = ASC(SUBSTR(CadRec,1,1))  
  IF Tope = 0  
    Actual = 0  
  ELSE  
    Actual = 1  
  ENDIF  
RETURN NIL
```

```
/***
* ArrayVacio() --> NIL
*
* Asigna un arreglo vacío a las variables Recibo, Cantidad,
* Oficio, FechaC y FechaD, las cuales almacenan los datos de
* los recibos pagados por interesado.
*/
FUNCTION ArrayVacio

    Recibo = {}
    Cantidad = {}
    Oficio = {}
    FechaC = {}
    FechaD = {}
RETURN NIL

/***
* Relaciona()
*
* Relaciona las bases de datos de cursos con la de expositores y la
* base de datos de ordenes de pago con la de cursos.
*/
FUNCTION Relaciona()

    SELECT A
    SET RELATION TO CveExpo INTO Exp // Se relacionan las bases de datos
                                     // de CURSOS.DBF con EXPOSITO.DBF
    SELECT B
    SET RELATION TO CveCurso INTO Cur // Se relacionan las bases de datos
                                        // de ORDEN.DBF con CURSOS.DBF
RETURN NIL

/*****

/*****
*
* Este es el módulo de ayuda
*
/*****
```



```
/***
* Ayuda( <cFuncion>, <nLinea>, <cVariable> ) --> NIL
*
* Coloca en la pantalla el mensaje de ayuda
* correspondiente a la variable <cVariable>
* y módulo <cFuncion> que la solicitó.
*
*/
FUNCTION Ayuda(cFuncion,nLinea,cVariable)
  LOCAL ColoresViejos := SETCOLOR(), Aviso := ARRAY(59)

  Aviso[1] = ' Tiene que teclear la clave del curso que desea '+';
  ' dar de alta o modificar.'+';
  Salto+Salto+;
  ' Recuerde que la clave de un curso '+';
  ' se compone de diez dígitos : el primer dígito se refiere '+';
  ' al tipo de curso, cuyo rango de valores permitidos es '+';
  ' de 0 a 5. A continuación listamos los valores permitidos '+';
  ' para el primer dígito : '+';
  Salto+Salto+;
  '0 = Curso de "Varios".'+';
  Salto+;
  '1 = Curso de "Idiomas".'+';
  Salto+;
  '2 = Curso de "Computación".'+';
  Salto+;
  '3 = Curso auxilia a una "Carrera".'+';
  Salto+;
  '4 = Curso de "Maestría".'+';
  Salto+;
  '5 = curso de "Diplomado".'+';
  Salto+Salto+;
  ' El segundo dígito se refiere a el subtipo de curso, cuyo '+';
  ' rango de valores permitidos es de 0 a 2 donde : '+';
  Salto+Salto+;
  '0 = curso "Normal".'+';
  Salto+;
  '1 = curso "Juvenil".'+';
  Salto+;
  '2 = curso de "Verano".'+';
  Salto+Salto+;
  ' El tercer dígito se refiere a la modalidad del curso, '+';
  ' teniendo como valores permitidos es 0 a 1 donde : '+';
  Salto+Salto+;
  '0 = curso "Externo".'+';
```

Salto+;
'1 = curso "Interno".'+;
Salto+Salto+;
' El cuarto y quinto dígito se refieren a una clave interna '+;
'de desempate para garantizar que una clave de curso no se '+;
'repita.'+;
Salto+Salto+;
' El sexto y séptimo dígito se refieren al grupo del '+;
'curso en cuestión, ya que se puede impartir un mismo '+;
'curso en diferentes horarios.'+;
Salto+Salto+;
' El octavo y noveno dígito se refieren al módulo o grado '+;
'del curso, el cual siempre es mayor que cero.'+;
Salto+Salto+;
' El décimo dígito se refiere al nivel del curso y sólo '+;
'se captura cuando el curso es un idioma, esto es de tipo 2. '+;
'Los rangos permitidos son de 0 a 2, si es 0 entonces '+;
'el curso no es un idioma, si es 1 el curso es un '+;
'idioma y su nivel es comprensión y si es 2 el curso es '+;
'un idioma y su nivel es posesión. ' ;
+Salto+Salto+;
' Nota : cualquier valor diferente a los rangos '+;
'especificados causará un error.'

Aviso[2] = ' Debe teclear el costo del curso para un alumno. '+;
'Recuerde que cualquier costo es no negativo.'

Aviso[3] = ' Debe teclear el costo del curso para un académico. '+;
'Recuerde que cualquier costo es no negativo.'

Aviso[4] = ' Debe teclear el costo del curso para personal '+;
'administrativo. Recuerde que cualquier costo es no '+;
'negativo.'

Aviso[5] = ' Debe teclear el costo del curso para personal de '+;
'mantenimiento. Recuerde que cualquier costo es no '+;
'negativo.'

Aviso[6] = ' Debe teclear el costo del curso para un egresado. '+;
'Recuerde que cualquier costo es no negativo.'

Aviso[7] = ' Debe teclear el costo del curso para el público en '+;
'general. Recuerde que cualquier costo siempre es no '+;
'negativo.'

Aviso[8] = ' Tiene que teclear el nombre del curso.'

Aviso[9] = ' Tiene que teclear la fecha de inicio del curso. '+';
'Recuerde que la fecha de inicio de un curso siempre '+';
'es menor a la fecha de término, ya que de lo contrario '+';
'se producirá un error.'

Aviso[10]= ' Tiene que teclear la fecha de término del curso. '+';
'Recuerde que la fecha de término de un curso siempre '+';
'es mayor a la fecha de inicio, ya que de lo contrario '+';
'se producirá un error.'

Aviso[11]= ' Debe teclear el lugar de impartición de un curso, '+';
'independientemente de que este sea interno o externo.'

Aviso[12]= ' Capture la duración del curso. Recuerde que este '+';
'siempre es mayor que cero, ya que de lo contrario se '+';
'producirá un error.'

Aviso[13]= ' Capture la hora de inicio. Debe de usar en la '+';
'hora el rango de 00 a 23 Hrs. De lo contrario '+';
'no podrá pasar a otro campo.'

Aviso[14]= ' Capture la hora de inicio. Debe de usar en los '+';
'minutos el rango de 00 a 59 Min. De lo contrario '+';
'no podrá pasar a otro campo.'

Aviso[15]= ' Capture la hora de término. Debe de usar en la '+';
'hora el rango de 00 a 23 Hrs. De lo contrario '+';
'no podrá pasar a otro campo.'

Aviso[16]= ' Capture la hora de término. Debe de usar en los '+';
'minutos el rango de 00 a 59 Min. De lo contrario '+';
'no podrá pasar al otro campo.'

Aviso[17]= ' Debe teclear la clave del expositor asignado a este '+';
'curso. La clave se compone de tres dígitos y esta debe de '+';
'existir de lo contrario ocasionará un error.'

Aviso[18]= ' Debe teclear el cupo máximo del curso. Recuerde que '+';
'este siempre es mayor que cero, ya que de lo contrario se '+';
'producirá un error.'

Aviso[19]= ' Debe teclear el salón asignado al curso.'

Aviso[20]= ' Debe teclear la clave de la carrera a la cual apoya o va '+';
'enfocado el curso. La clave de la carrera es igual a la '+';

'clave empleada por las jefaturas de la institución.'+;
Salto+Salto+;
' Notas : Si el curso no apoya a ninguna carrera teclee '+;
''000". Si teclaea una clave incorrecta causará un '+;
'error ; Sugerencia ! Revise la base de datos de profesiones.'

Aviso[21]= ' Debe teclear los días de la semana en los cuales se '+;
'impartirá el curso. Aquí debe teclear exclusivamente '+;
'los caracteres "S" si el curso se impartiera ese día y '+;
''N" en caso de que no se impartiera ese día. El día en la '+;
'pantalla aparece recortado debido al espacio y denota : '+;
Salto+Salto+;
'Lu = Lunes'+Salto+;
'Ma = Martes'+Salto+;
'Mi = Miércoles'+Salto+;
'Ju = Jueves'+Salto+;
'Vi = Viernes'+Salto+;
'Sa = Sábado'+Salto+;
'Do = Domingo'+Salto+;
Salto+Salto+;
' Nota : Si presiona cualquier caracter diferente a '+;
''N" o "S" no podrá pasar a otro campo.'+;
Salto+Salto

Aviso[22]= ' Debe de teclear la clave del curso al cual desea '+;
'inscribirse el interesado. Es importante mencionar que esta '+;
'clave debe de existir, ya que de lo contrario se producirá '+;
'un error.'

Aviso[23]= ' Debe de capturar el nombre del interesado.'+;
Salto+Salto+;
' ; Sugerencia ! Como el programa ordena alfabéticamente '+;
'los nombres de los interesados, es conveniente capturar '+;
'primero los apellidos y luego el nombre.'

Aviso[24]= ' Debe de capturar el teléfono del interesado sin guiones '+;
'o separadores. En caso de teclear un número negativo no '+;
'podrá pasar a otro campo.'

Aviso[25]= ' Debe capturar la clave de la procedencia del interesado. '+;
'El rango de valores permitidos por esta clave son 3 donde :'+;
Salto+Salto+;
'0 = OTRA INSTITUCION.'+;
Salto+;
'1 = UNAM CAMPUS ARAGON.'+;

Salto+;

'2 = OTRA DEPENDENCIA DE LA UNAM.'+;

Salto+Salto+;

' Nota : Cualquier rango de valores diferentes a los '+;

'especificados causará un error.'

Aviso[26]= ' Debe capturar la procedencia del interesado sólo si la '+;
'clave de la procedencia es diferente de "1". Especifique '+;
'el nombre de la institución u organismo al cual pertenece '+;
'el interesado.'

Aviso[27]= ' En caso de ser el interesado alumno del plantel, debe '+;
'teclear la clave de la carrera a la cual pertenece. '+;
'La clave de la carrera es igual a la '+;
'clave empleada por las jefaturas de la institución.'+;
Salto+Salto+;
' Notas : Si el interesado no es alumno del plantel teclee '+;
""000". Si teclea una clave de carrera incorrecta causará un '+;
'error ; Sugerencia ! Revise la base de datos de profesiones.'

Aviso[28]= ' Si el interesado es alumno de la UNAM (clave de la '+;
'procedencia diferente de 0) capture su número de cuenta.'

Aviso[29]= ' Debe teclear la clave de la condición del interesado. '+;
'Esta clave es muy importante, ya que ella permite seleccionar '+;
'el costo del curso que le corresponde al mismo. El '+;
'rango de valores permitidos por esta clave son 6 donde :'+;
Salto+Salto+;
'0 = PUBLICO EN GENERAL.'+;
Salto+;
'1 = ALUMNO DEL PLANTEL.'+;
Salto+;
'2 = ACADEMICO.'+;
Salto+;
'3 = ADMINISTRATIVO.'+;
Salto+;
'4 = MANTENIMIENTO.'+;
Salto+;
'5 = EGRESADO.'+;
Salto+Salto+;
' Nota : No podrá pasar a otro campo si teclea cualquier '+;
'rango de valores diferentes a los especificados.'

Aviso[30]= ' Si el interesado es alumno del plantel (clave de la '+;
'procedencia igual a "1"), capture el semestre al que '+;
'actualmente esta inscrito. Si teclea un número negativo '+;

'no podrá pasar a otro campo.'

Aviso[31]= ' Este programa esta diseñado para localizar el '+';
'correspondiente costo, en función de la clave de la '+';
'condición capturada. Pero si Usted desea asignarle otro '+';
'valor, simplemente tecleelo.'+;
Salto+Salto+;
' | Recomendación ! En caso de modificar el costo de un '+';
'curso para un interesado, especifique en el campo memorándum '+';
'el motivo que originó dicho cambio de costo. Señale si es '+';
'beca o se le concedió un descuento. Esto es con el fin de '+';
'llevar un mejor control en los ingresos.'+;
Salto+Salto+;
' | Advertencia ! Si se encuentra en el módulo de '+';
'modificaciones, el costo de un curso siempre se ajusta '+';
'a la clave de la condición capturada, independientemente '+';
'que se le haya asignado otro costo al interesado al darlo '+';
'de alta. Por ello, es menester volver a teclear el mismo '+';
'costo de nuevo que cuando se dio de alta (claro, esto es '+';
'si el usuario requiere mantener el mismo precio.).'

Aviso[32]= ' Tiene que teclear la fecha de pago del recibo en caja. '+';
'Recuerde que la fecha de pago en caja siempre es menor o '+';
'igual a la fecha de entrega del mismo en las oficinas del '+';
'DEC Campus Aragón, ya que de lo contrario se producirá un '+';
'error.'

Aviso[33]= ' Tiene que teclear la fecha de entrega del pago del '+';
'recibo en las oficinas del DEC campus Aragón. Recuerde '+';
'que la fecha de entrega en las oficinas del DEC Campus '+';
'Aragón siempre es mayor o igual a la fecha de pago del '+';
'recibo en la caja, ya que de lo contrario se producirá un '+';
'error.'

Aviso[34]= ' Debe teclear el número de recibo correspondiente para '+';
'llevar un mejor control de los ingresos.'

Aviso[35]= ' Debe capturar la clave del expositor, la cual se compone '+';
'de tres dígitos.'+;
Salto+Salto+;
' Nota : Si captura una clave ya existente, por estar ésta '+';
'asignada a otro expositor, causará un error.'

Aviso[36]= ' Debe capturar el nombre completo del expositor.'

Aviso[37]= ' Debe capturar completamente el registro federal de '+';

'contribuyentes del expositor.'

Aviso[38]= ' Debe capturar el domicilio completo del expositor.'

Aviso[39]= ' Debe de capturar el teléfono del expositor sin guiones '+';
'o separadores. En caso de teclear un número negativo no '+';
'podrá pasar a otro campo.'

Aviso[40]= ' Debe capturar el nombre completo del instituto u '+';
'organismo al cual está adscrito el expositor.'

Aviso[41]= ' Deba capturar el nombre del título profesional del '+';
'expositor.'

Aviso[42]= ' Debe capturar el número de cédula profesional del '+';
'expositor.'

Aviso[43]= ' Debe teclear la fecha de expedición de la cédula '+';
'profesional del expositor.'

Aviso[44]= ' Debe teclear el importe del recibo correspondiente.'+;
Salto+Salto+;
' **NOTA** : Recuerde que todo importe es mayor que cero.'

Aviso[45]= ' Debe teclear el nombre de la escuela que está generando '+';
'el reporte. Este por defecto es : '+';
""+RTRIM(Escuela)+"""+;
' , aunque puede cambiarlo si desea.'

Aviso[46]= ' Debe teclear el nombre del departamento que está '+';
'generando el reporte. Este por defecto es : '+';
""+RTRIM(Depto)+"""+;
' , aunque puede cambiarlo si desea.'

Aviso[47]= ' Debe teclear el nombre de la persona a la cual se va '+';
'a dirigir el reporte. Este por defecto es : '+';
""+RTRIM(Dirigido)+"""+;
' , aunque puede cambiarlo si desea.'

Aviso[48]= ' Debe teclear el cargo de la persona a la cual se va '+';
'a dirigir el reporte. Este por defecto es : '+';
""+RTRIM(Cargo1)+"""+;
' , aunque puede cambiarlo si desea.'

Aviso[49]= ' Debe teclear el nombre de la persona que firma y suscribe '+';

'el reporte. Este por defecto es : '+';
''"+RTRIM(Suscribe)+"''+''+';
' , aunque puede cambiarlo si desea.'

Aviso[50]= ' Debe teclear el cargo de la persona que firma y suscribe '+';
'el reporte. Este por defecto es : '+';
''"+RTRIM(Cargo2)+"''+''+';
' , aunque puede cambiarlo si desea.'

Aviso[51]= ' Debe teclear los honorarios del exponente del curso.'

Aviso[52]= ' Debe teclear la fecha de generación del reporte.'

Aviso[53]= ' Debe teclear la clave de la carrera o profesión la cual '+';
'se compone de tres dígitos.'+;
Salto+Salto+;
' Nota : Si captura una clave ya existente por estar ésta '+';
'asignada a otra carrera, causará un error.'

Aviso[54]= ' Debe teclear el nombre de la carrera o profesión.'

Aviso[55]= ' Debe teclear la clave para tener acceso a los módulos '+';
'de altas, bajas y modificaciones. Esta clave sólo la posee '+';
'personal autorizado.'+;
Salto+Salto+;
' Si no sabe la clave y sólo desea obtener consultas y '+';
'reportes presione < ESC > , pero en caso de saberla y por '+';
'error la capturó mal, tiene dos oportunidades adicionales '+';
'para volverla a capturar. En caso de errar las tres veces '+';
'sólo tendrá acceso a consultas y reportes.'+;
Salto+Salto+;
' Presione < ESC > para salir de esta pantalla.'

Aviso[56]= ' Debe teclear el nombre de la institución que está '+';
'generando el reporte. Este por defecto es : '+';
''"+RTRIM(Instituci)+"''+''+';
' , aunque puede cambiarlo si desea.'

Aviso[57]= ' Se encuentra en el módulo de reportes. Aquí puede '+';
'seleccionar su opción con las flechas de movimiento del '+';
'cursor y posteriormente presionar < ENTER > ,o bien '+';
'tecleando el número de la opción deseada.'+;
Salto+Salto+;
' Las opciones que se disponen son las siguientes : '+';
Salto+Salto+;
' 1. PORCENTAJE DE AVANCE DE CURSOS : proporciona un '+';

'reporte visual sobre el avance en días y en porcentajes '+';
'que un curso lleva hasta la fecha que contenga el sistema.'+;
Salto+Salto+;
' 2. CURSOS POR VENCERSE : proporciona un reporte ya sea '+';
'visual, impreso y/o grabado en un archivo, sobre los '+';
'cursos por terminar, según los días de anticipación que '+';
'el usuario solicite.'+;
Salto+Salto+;
' 3. LISTA DE DEUDORES : proporciona un reporte, ya sea '+';
'visual, impreso y/o grabado en un archivo, sobre los '+';
'deudores vigentes en la base de datos de ordenes de pago.'+;
Salto+Salto+;
' 4. REGISTRO DE INSCRIPCION Y REPORTE DE RECIBOS : '+';
'proporciona un reporte, ya sea visual, impreso y/o '+';
'grabado en un archivo, sobre la relación de registros a '+';
'reportar en un oficio, así como el registro de inscripción '+';
'correspondiente. Estos reportes pueden ser nuevos o bien se '+';
'pueden generar informes ya reportados.'+;
Salto+Salto+;
' 5. LISTA DE PROFESORES : proporciona un reporte, ya sea '+';
'visual, impreso y/o grabado en un archivo, sobre las '+';
'listas de profesores, según el curso solicitado'+;
Salto+Salto+;
' 6. INFORME A LA UNIDAD DE PLANEACION : proporciona un '+';
'reporte ya sea visual, impreso y/o grabado en un '+';
'archivo, sobre el informe de cursos para la unidad de '+';
'planeación, según el periodo solicitado.'+;
Salto+Salto+;
' 7. SALIR : permite salir del módulo de reportes.'

Aviso[58]= ' Se encuentra en el menú de consultas por cadena '+';
'incluida. Seleccione el campo que servirá para '+';
'realizar la búsqueda de la cadena requerida.'

Aviso[59]= ' Debe teclear la nueva clave de acceso, la cual se '+';
'compone de cuatro caracteres alfanuméricos.'

```
PantaAyuda()  
SET CURSOR ON  
SET COLOR('0/7,....')  
SET KEY 28 TO  
DO CASE  
CASE cVariable = 'CLAVE'  
DO CASE  
CASE cFuncion= 'CAPTURACUR'  
MemoEdit(Aviso[1],6,22,18,57,..F.)
```

```
CASE cFuncion= 'CAPTURAOORD'  
  MemoEdit(Aviso[22],6,22,18,57,..F.)  
CASE cFuncion= 'CAPTURAEXP'  
  MemoEdit(Aviso[35],6,22,18,57,..F.)  
CASE cFuncion= 'CAPTURACAR'  
  MemoEdit(Aviso[53],6,22,18,57,..F.)  
ENDCASE  
CASE cVariable = 'ALUM'  
  MemoEdit(Aviso[2],6,22,18,57,..F.)  
CASE cVariable = 'ACAD'  
  MemoEdit(Aviso[3],6,22,18,57,..F.)  
CASE cVariable = 'ADMI'  
  MemoEdit(Aviso[4],6,22,18,57,..F.)  
CASE cVariable = 'MANT'  
  MemoEdit(Aviso[5],6,22,18,57,..F.)  
CASE cVariable = 'EGRE'  
  MemoEdit(Aviso[6],6,22,18,57,..F.)  
CASE cVariable = 'PUBL'  
  MemoEdit(Aviso[7],6,22,18,57,..F.)  
CASE cVariable = 'NOMBRE'  
DO CASE  
  CASE cFuncion= 'CAPTURACUR'  
    MemoEdit(Aviso[8],6,22,18,57,..F.)  
  CASE cFuncion= 'CAPTURAORD'  
    MemoEdit(Aviso[23],6,22,18,57,..F.)  
  CASE cFuncion= 'CAPTURAEXP'  
    MemoEdit(Aviso[36],6,22,18,57,..F.)  
  CASE cFuncion= 'CAPTURACAR'  
    MemoEdit(Aviso[54],6,22,18,57,..F.)  
ENDCASE  
CASE cVariable = 'FEC1'  
DO CASE  
  CASE cFuncion= 'CAPTURACUR'  
    MemoEdit(Aviso[9],6,22,18,57,..F.)  
  CASE cFuncion= 'CAPTURAEXP'  
    MemoEdit(Aviso[43],6,22,18,57,..F.)  
ENDCASE  
CASE cVariable = 'FEC2'  
  MemoEdit(Aviso[10],6,22,18,57,..F.)  
CASE cVariable = 'LUGA'  
  MemoEdit(Aviso[11],6,22,18,57,..F.)  
CASE cVariable = 'DURA'  
  MemoEdit(Aviso[12],6,22,18,57,..F.)  
CASE cVariable = 'HOR1_1'  
  MemoEdit(Aviso[13],6,22,18,57,..F.)  
CASE cVariable = 'HOR1_2'
```

```
    MemoEdit(Aviso[14],6,22,18,57,..F.)
CASE cVariable = 'HOR2_1'
    MemoEdit(Aviso[15],6,22,18,57,..F.)
CASE cVariable = 'HOR2_2'
    MemoEdit(Aviso[16],6,22,18,57,..F.)
CASE cVariable = 'CVEE'
    MemoEdit(Aviso[17],6,22,18,57,..F.)
CASE cVariable = 'CUP'
    MemoEdit(Aviso[18],6,22,18,57,..F.)
CASE cVariable = 'SALO'
    MemoEdit(Aviso[19],6,22,18,57,..F.)
CASE cVariable = 'CVECARR'
DO CASE
    CASE cFuncion= 'CAPTURACUR'
        MemoEdit(Aviso[20],6,22,18,57,..F.)
    CASE cFuncion= 'CAPTURAORD'
        MemoEdit(Aviso[27],6,22,18,57,..F.)
ENDCASE
CASE cVariable = 'LJ'
    MemoEdit(Aviso[21],6,22,18,57,..F.)
CASE cVariable = 'MA'
    MemoEdit(Aviso[21],6,22,18,57,..F.)
CASE cVariable = 'MI'
    MemoEdit(Aviso[21],6,22,18,57,..F.)
CASE cVariable = 'JU'
    MemoEdit(Aviso[21],6,22,18,57,..F.)
CASE cVariable = 'VI'
    MemoEdit(Aviso[21],6,22,18,57,..F.)
CASE cVariable = 'SA'
    MemoEdit(Aviso[21],6,22,18,57,..F.)
CASE cVariable = 'DOMINGO'
    MemoEdit(Aviso[21]+ ' Presione < ENTER > o < PgDn > para pasar '+'
        'al campo memorándum.',6,22,18,57,..F.)
CASE cVariable = 'TELE'
DO CASE
    CASE cFuncion= 'CAPTURAORD'
        MemoEdit(Aviso[24],6,22,18,57,..F.)
    CASE cFuncion= 'CAPTURAEXP'
        MemoEdit(Aviso[39],6,22,18,57,..F.)
ENDCASE
CASE cVariable = 'CVEP'
    MemoEdit(Aviso[25],6,22,18,57,..F.)
CASE cVariable = 'PR'
    MemoEdit(Aviso[26],6,22,18,57,..F.)
CASE cVariable = 'CUEN'
    MemoEdit(Aviso[28],6,22,18,57,..F.)
```

```
CASE cVariable = 'CVECOND'  
  MemoEdit(Aviso[29],6,22,18,57,..F.)  
CASE cVariable = 'SEME'  
  MemoEdit(Aviso[30],6,22,18,57,..F.)  
CASE cVariable = 'COST'  
  MemoEdit(Aviso[31],6,22,18,57,..F.)  
CASE cVariable = 'REGISTRO'  
  MemoEdit(Aviso[37],6,22,18,57,..F.)  
CASE cVariable = 'DOMI'  
  MemoEdit(Aviso[38],6,22,18,57,..F.)  
CASE cVariable = 'ADSC'  
  MemoEdit(Aviso[40],6,22,18,57,..F.)  
CASE cVariable = 'TITU'  
  MemoEdit(Aviso[41],6,22,18,57,..F.)  
CASE cVariable = 'CEDU'  
  MemoEdit(Aviso[42],6,22,18,57,..F.)  
CASE cVariable = 'DATO2'  
  MemoEdit(Aviso[44],6,22,18,57,..F.)  
CASE cVariable = 'DATO1'  
  MemoEdit(Aviso[34],6,22,18,57,..F.)  
CASE cVariable = 'FECHA1'  
  MemoEdit(Aviso[32],6,22,18,57,..F.)  
CASE cVariable = 'FECHA2'  
  MemoEdit(Aviso[33],6,22,18,57,..F.)  
CASE cVariable = 'ESCUELA'  
  MemoEdit(Aviso[45],6,22,18,57,..F.)  
CASE cVariable = 'DEPTO'  
  MemoEdit(Aviso[46],6,22,18,57,..F.)  
CASE cVariable = 'DIRIGIDO'  
  MemoEdit(Aviso[47],6,22,18,57,..F.)  
CASE cVariable = 'CARGO1'  
  MemoEdit(Aviso[48],6,22,18,57,..F.)  
CASE cVariable = 'SUSCRIBE'  
  MemoEdit(Aviso[49],6,22,18,57,..F.)  
CASE cVariable = 'CARGO2'  
  MemoEdit(Aviso[50],6,22,18,57,..F.)  
CASE cVariable = 'HONORARIO'  
  MemoEdit(Aviso[51],6,22,18,57,..F.)  
CASE cVariable = 'FECHA'  
  MemoEdit(Aviso[52],6,22,18,57,..F.)  
CASE cVariable = 'PASSWORD'  
DO CASE  
  CASE cFuncion = 'ACCESO'  
    MemoEdit(Aviso[55],6,22,18,57,..F.)  
  CASE cFuncion = 'CLAVEDEACC'  
    MemoEdit(Aviso[59],6,22,18,57,..F.)
```

```
ENDCASE
CASE cVariable = 'INSTITUCI'
  MemoEdit(Aviso[56],6,22,18,57,.F.)
CASE cVariable = 'OPCION'
  MemoEdit(Aviso[57],6,22,18,57,.F.)
CASE cVariable = 'NOPCION'
  MemoEdit(Aviso[58],6,22,18,57,.F.)
ENDCASE
SET KEY 28 TO Ayuda
RESTSCREEN (0,0,24,79,G)
SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/*
```

```
* AyudaMod() --> NIL
```

```
*
```

```
* Coloca en la pantalla el mensaje de ayuda
```

```
* correspondiente a los menús de opciones.
```

```
*
```

```
*/
```

```
FUNCTION AyudaMod
```

```
  LOCAL ColoresViejos := SETCOLOR()
```

```
  LOCAL Aviso := ARRAY(12)
```

```
Aviso[1] = ' Se encuentra en el menú principal. Aquí Usted '+';
'puede seleccionar con las flechas del cursor y '+';
'confirmar una de las siete posibles alternativas '+';
'existentes presionando < ENTER >, o bien presionando '+';
'la primera letra de la opción deseada. Cada una de estas '+';
'alternativas realiza diversas acciones ya sea para '+';
'cursos, ordenes de pago, expositores, carreras, o bien '+';
'configuración del sistema. Estas son : '+';
Salto+Salto+;
' 1) ALTAS : agrega nuevos registros a la base de datos '+';
'seleccionada.'+';
Salto+Salto+;
' 2) BAJAS : elimina un registro de la base de datos '+';
'seleccionada.'+';
Salto+Salto+;
' 3) CONSULTAS : exhibe el registro seleccionado.'+';
Salto+Salto+;
' 4) MODIFICACIONES : exhibe y edita registros de la base '+';
```

'de datos seleccionada para cambiar su contenido.'+;
Salto+Salto+;
' 5) REPORTES : genera diversos reportes en '+;
'pantalla, en impresora y/o en un archivo.'+;
Salto+Salto+;
' 6) DISPOSICION : permite configurar los principales '+;
'parámetros del sistema como son las unidades de disco, rutas, '+;
'tipos de impresora, cadenas usadas en los reportes, etc.'+;
Salto+Salto+;
' 7) SALIDA termina el programa regresando el control '+;
'al sistema operativo.'

Aviso[2] = ' Esta opción permite seleccionar la base de datos '+;
'que será empleada. Aquí se encuentran disponibles para '+;
'elegir las siguientes cuatro bases de datos : cursos, ordenes'+;
'de pago, expositores y carreras.'+;
Salto+Salto+;
' Para seleccionar utilice las flechas de movimiento del '+;
'cursor y presione < ENTER > en la opción deseada o bien '+;
'presione la primera letra de la misma.'

Aviso[3] = ' Este programa maneja tres tipos de consultas : '+;
Salto+Salto+;
' 1) GLOBALES : presenta una pantalla en la cual aparece '+;
'todos los registros de la base de datos seleccionada, '+;
'permitiendo al usuario hojear los campos y seleccionar un '+;
'registro en particular.'+;
Salto+Salto+;
' 2) POR CLAVE : presenta una pantalla en la cual puede '+;
'teclear la clave del registro deseado. ¡¡¡ Importante !!! '+;
'No es necesario que escriba toda la clave. El programa está '+;
'diseñado para que si prefiere teclear sólo una '+;
'parte de la clave, este buscará y listará aquellos '+;
'registros que comiencen con la sección de clave que '+;
'Usted tecleó.'+;
Salto+Salto+;
' 3) POR CADENA INCLUIDA : presenta una pantalla que '+;
'contiene algunos campos de la base de datos seleccionada, '+;
'y busca aquellos registros que tengan incluida la cadena '+;
'suministrada por Usted.'

Aviso[4] = ' Se encuentra en el menú de disposición o configuración '+;
'del sistema. Aquí Usted puede seleccionar con las flechas '+;
'del cursor y confirmar una de las cinco posibles '+;
'alternativas existentes presionando < ENTER > o bien '+;
'tecleando la primera letra de la opción deseada. Cada una '+;

'de estas alternativas permiten cambiar los parámetros '+';
'que rigen al sistema como son : '+';
Salto+Salto+;
' a) UNIDAD : permite seleccionar la unidad de trabajo '+';
'conjuntamente con su ruta, en la cual residen las bases '+';
'de datos.'+';
Salto+Salto+;
' b) IMPRESORA : permite seleccionar el tipo de impresora '+';
'que se dispone, para generar los diferentes informes creados '+';
'en el módulo de reportes.'+';
Salto+Salto+;
' c) DATOS : permite modificar las diferentes cadenas '+';
'utilizadas para generar los reportes, como nombre, títulos, '+';
'etc.'+';
Salto+Salto+;
' d) CLAVE : permite cambiar la clave vigente de acceso '+';
'para realizar altas, bajas y modificaciones ; Advertencia ! '+';
'Esta opción está activa, siempre y cuando el usuario haya '+';
'teclado correctamente la clave de acceso al ingresar al '+';
'programa.'+';
Salto+Salto+;
' e) SALIR : Retorna al menú anterior.'

Aviso[5] = ' Se encuentra en el menú de tipo de impresoras. '+';
'Seleccione el tipo de impresora que Usted posee.'

Aviso[6] = ' Presione la tecla < ENTER > para editar el campo '+';
'memorándum correspondiente. Aquí se deben de señalar '+';
'las observaciones más relevantes del registro que '+';
'se está editando, ya se para visualización y/o para '+';
'modificación.'+';
Salto+Salto+;
' En caso de querer grabar la información presentada '+';
'(si está editando para modificación este campo) '+';
'presione CTRL+W; si desea salir sin grabar la pantalla '+';
'presione la tecla < ESC >; si desea pasarse a otros '+';
'campos emplee las teclas mover arriba el cursor, mover '+';
'abajo el cursor, Pg/Dn y Pg/Up.'

Aviso[7] = ' Presione < ENTER > o < ESC > para pasar a la '+';
'siguiente pantalla.'

Aviso[8] = ' Si desea confirmar la operación realizada debe '+';
'presionar < ENTER > en donde dice "Aceptar" o '+';
'simplemente presione la letra "A"; si '+';
'desea arrepentirse de la operación realizada debe '+';

'presionar < ENTER > en donde dice "Cancelar", o bien '+';
'simplemente presione la tecla < ESC > o la letra "C".'

Aviso[9] = ' Presione < ENTER > para editar los recibos pagados '+';
'por el interesado.'+;
'Salto+Salto+;
' NOTA : es importante que capture siempre un recibo ya '+';
' que de lo contrario se producirá un error, incluso en '+';
' caso de haber una beca es imprescindible que le asigne '+';
' cero al número de recibo y al importe correspondiente.'

Aviso[10] = ' Se encuentra en el menú de recibos. Aquí Usted '+';
'puede seleccionar con las flechas del cursor y '+';
'confirmar una de las cuatro posibles alternativas '+';
'existentes presionando < ENTER >. Cada una de estas '+';
'alternativas permiten visualizar y/o editar el '+';
'contenido de los recibos. Según sea el caso, puede '+';
'seleccionar : '+';
'Salto+Salto+;
' a) SIGUIENTE : permite avanzar de un recibo a otro hacia '+';
'adelante. Tiene como límite máximo dar de alta hasta 100 '+';
'recibos inclusive, para un sólo interesado.'+;
'Salto+Salto+;
' b) ATRAS : permite avanzar de un recibo a otro hacia atrás.'+;
'Salto+Salto+;
' c) BORRAR : permite eliminar el recibo actual. Es '+';
'importante señalar, que sólo se activa cuando se realizan '+';
'altas o modificaciones exclusivamente.'+;
'Salto+Salto+;
' d) SALIR : Sale de la pantalla y graba la información '+';
'de los recibos. Es equivalente a presionar < ESC > en '+';
'cualquier momento.'+;
'Salto+Salto+;
' NOTA : Si modificó información de recibos ya '+';
'capturados y presionó la tecla < ESC > o seleccionó '+';
'la opción SALIR, esta información siempre se graba '+';
'automáticamente, por lo que le recomendamos en caso de '+';
'realizar cambios tener siempre la seguridad de '+';
'hacerlos. En caso de realizar cambios o altas y no quisiera '+';
'proceder con los mismos salga de esta pantalla y '+';
'seleccione CANCELAR.'

Aviso[11] = ' Presione < ENTER > para capturar las personas con copia '+';
'del registro de inscripción y/o relación de recibos.'

Aviso[12] = ' Se encuentra en el menú de recibos. Aquí Usted '+';

'puede seleccionar con las flechas del cursor y '+';
'confirmar una de las tres posibles alternativas '+';
'existentes presionando < ENTER >. Cada una de estas '+';
'alternativas permiten visualizar el contenido de los '+';
'recibos. Según sea el caso, puede seleccionar : '+';
Salto+Salto+;
' a) SIGUIENTE : permite avanzar de un recibo a otro hacia '+';
'delante, visualizando su contenido.'+;
Salto+Salto+;
' b) ATRAS : permite avanzar de un recibo a otro hacia '+';
'atrás, visualizando su contenido.'+;
Salto+Salto+;
' c) SALIR : Sale de la pantalla de consulta de recibos '+';
'Es equivalente a presionar < ESC > en cualquier momento.'

```
PantaAyuda()
SET CURSOR ON
SETCOLOR('0/7,....')
SET KEY 28 TO
MemoEdit(Aviso[VarAyuda],6,22,18,57,..F.)
SET KEY 28 TO Ayuda
SET CURSOR OFF
RESTSCREEN (0,0,24,79,G)
SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/***
* AcercaDe() -> NIL
*
* Presenta la versión y aplicación del programa.
*
*/
FUNCTION AcercaDe
LOCAL ColoresViejos := SETCOLOR()
LOCAL Aviso := ' ¡ Bienvenido a DEC Ver. 1.0 !'+ExplicaPro
```

```
PantaAyuda()
SET CURSOR ON
SETCOLOR('0/7,....')
SET KEY 28 TO
MemoEdit(Aviso,6,22,18,57,..F.)
SET KEY 28 TO Ayuda
SET CURSOR OFF
```

```
RESTSCREEN (0,0,24,79,G)
SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/*
* PantaAyuda() --> NIL
*
* Coloca el marco en cuyo interior se depositan los
* mensajes de las ayudas y de la versión del programa.
*
*/
FUNCTION PantaAyuda

  VentanaExplosiva (4,20,20,59,Marco7,'15/7,...')
  SombraMarco (4,20,20,59)
  SETCOLOR('4/7,...')
  @ 4,36 SAY ' Ayuda '
  RamaV( 7,59,18,FlechaV,'15/0,...')
RETURN NIL
```

```
...../
```

```
.....
* Este es el módulo que realiza el espectáculo de colores y ventanas *
...../
```

```
/*
* VentanaSin( <cTipoMarco>, <cColores1> ) --> NIL
*
* Coloca una ventana del tipo <cTipoMarco> de
* atributos <cColores> que crece poco a poco y
* cuya posición la determina la función matemática
* seno.
*
*/
FUNCTION VentanaSin (cTipoMarco, cColores )
```

```
LOCAL i, j, k, x, ColoresViejos := SETCOLOR()

SET CURSOR OFF
@ 0,0,24,3 BOX REPLICATE(CHR(176),9) COLOR "3/1,,,,,"
G = SAVESCREEN(0,0,24,3)
FOR i = 1 TO 19
  RESTSCREEN( 0,i*4,24,i*4+3, G )
  Delay(0.001)
NEXT
x = 0
k = 0
FOR i = 1 TO 12
  j = INT(sin(x)*20+20)
  @ i-1,j+17,i+k,j+k+24 BOX cTipoMarco+" " COLOR cColores
  SombraMarco(i-1,j+17,i+k,j+k+24)
  Delay(0.1)
  x = x + 0.2
  ++k
NEXT
FOR i = 1 TO 6
  @ 11-i,55-6*i,23-i,71-2*i BOX cTipoMarco+" " COLOR cColores
  SombraMarco(11-i,55-6*i,23-i,71-2*i)
  Delay(0.1)
NEXT
FOR i = 1 TO 4
  @ 5-i,19-i*3,17+i,3*i+59 BOX cTipoMarco+" " COLOR cColores
  SombraMarco(5-i,19-i*3,17+i,3*i+59)
  delay(0.1)
NEXT
@ 0,2,23,77 BOX cTipoMarco+" " COLOR cColores
SombraMarco(0,2,23,77)
SETCOLOR(ColoresViejos)
SET CURSOR ON
RETURN NIL
```

```
/*
* Sin( <nRadianes> ) --> <nSenoEnRad>
*
* Obtiene el seno de <nRadianes> en <nSenoEnRad>.
*
*/
FUNCTION Sin(A)
  LOCAL Sum, i
```

```
Sum = 0
FOR i = 0 TO 17
  Sum = Sum + (((-1)**i)*(A**(2*i+1)))/Factorial(2*i+1)
NEXT
RETURN Sum
```

```
/*
 * Factorial( <nValor> ) --> nFactorial
 *
 * Calcula el factorial de <nValor>.
 *
 */
```

```
FUNCTION Factorial(Tope)
  LOCAL Fac, j

  Fac = 1
  FOR j = 1 TO Tope
    Fac = Fac*j
  NEXT
RETURN Fac
```

```
/*
 * AnuncioIzq( <nY2>, <nX2>, <cCad>, <cColores> ) --> NIL
 *
 * Permite que aparezca una cadena <cCad> en
 * forma de un anuncio corredizo que se desplaza
 * de derecha a izquierda comenzado en la coordenada
 * <nY2>, <nX2> de colores <cColores>.
 *
 */
```

```
FUNCTION AnuncioIzq(nY2, nX2, cCad, cColores)
LOCAL i, j, k, ColoresViejos := SETCOLOR()
```

```
k = 0
SETCOLOR(cColores)
FOR i = 1 TO LEN(cCad)
  FOR j = k TO 0 STEP -1
    @ nY2, nX2+LEN(cCad)-j SAY SUBSTR(cCad,i,j,1)
    Temporizar(2)
  NEXT
  ++k
NEXT
```

```
SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/*
 * AnuncioDer( <nY1>, <nX1>, <cCad>, <cColor> ) --> NIL
 *
 * Permite que aparezca una cadena <cCad> en
 * forma de un anuncio corridizo que se desplaza
 * de izquierda a derecha comenzado en la coordenada
 * <nY1>, <nX1> de colores <cColor>.
 *
 */
FUNCTION AnuncioDer(nY1, nX1, cCad, cColor)
LOCAL i, j, k, ColoresViejos := SETCOLOR()
```

```
  k = 0
  SETCOLOR(cColor)
  FOR i = 1 TO LEN(cCad)
    FOR j = 0 TO k
      @ nY1, nX1+j+1 SAY SUBSTR(cCad,LEN(cCad)-k+j,1)
      Temporizar(2)
    NEXT
    ++k
  NEXT
  SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/*
 * Asteriscos( <nY1>, <nX1>, <nX2>, <cCad>, <cRelleno>, <cColor> ) --> NIL
 *
 * Permite que aparezca una cadena de caracteres <cCad>
 * en medio de las coordenadas <nY1>, <nX1>, <nY1>, <nX2>
 * de colores <cColor>. Aquí <cRelleno> es el caracter
 * con el que originalmente se rellena la línea en donde
 * la cadena va incluida.
 *
 */
FUNCTION Asteriscos ( nY1, nX1, nX2, cCad, cRelleno, cColor )
LOCAL i, j, k, l, DiferenciaX, ColoresViejos := SETCOLOR()
```

```
  IF LEN(cCad)%2 = 0
    cCad = "" + cCad
```

```
ENDIF
DiferenciaX = nX2-nX1
k = DiferenciaX - 1
SETCOLOR(cColor)
FOR i = 1 TO DiferenciaX - 1
  @ nY1, nX1+i SAY CHR(178)
NEXT
Delay(0.4)
FOR i = 1 TO INT((DiferenciaX-1)/2)+1
  @ nY1, nX1+i SAY cRelleno
  Temporizar(10)
  @ nY1, nX1+k SAY cRelleno
  Temporizar(10)
  --k
NEXT
j = 0
l = INT(LEN(cCad)/2)
FOR i = 0 to l
  FOR k = 0 to j
    @ nY1, nX1+INT((DiferenciaX-1)/2)-i+k+1 SAY SUBSTR(cCad, l-i+1+k, 1)
    Temporizar(4)
  NEXT
  j = j+2
NEXT
SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/***
* Mod_Color( <nRen1>,<nCol1>,<nRen2>,<nCol2>,<nColor> ) --> NIL
*
```

```
* Cambia los atributos del color vigente del cuadro
* especificado en la pantalla por las coordenadas
* <nRen1>, <nCol1> y <nRen2>, <nCol2> según el código
* de color numérico pasado por <nColor>.
*
```

```
*/
FUNCTION Mod_Color(nRen1,nCol1,nRen2,nCol2,nColor)
  LOCAL j, Pan, Aux

  Pan = SAVESCREEN(nRen1,nCol1,nRen2,nCol2)
  Aux = ""
  FOR j = 2 TO LEN(Pan) STEP 2
    Aux = Aux + STUFF(SUBSTR(Pan,j-1,2),2,1,CHR(nColor))
  NEXT
```

```
RESTSCREEN(nRen1,nCol1,nRen2,nCol2,Aux)  
RETURN NIL
```

```
/***  
* SombraMarco( <nRen1>,<nCol1>,<nRen2>,<nCol2> ) --> NIL  
*  
* Da la apariecia de que un marco o veritana encontrada en  
* las coordenadas <nRen1>, <nCol1> y <nRen2>, <nCol2> este  
* flotando en la pantalla  
*  
*/
```

```
FUNCTION SombraMarco(nRen1, nCol1, nRen2, nCol2)  
  
  RESTSCREEN( nRen2+1, nCol1+1, nRen2+1, nCol2+2 ;  
    TRANSFORM( SAVESCREEN(nRen2+1, nCol1+1, nRen2+1, nCol2+2);  
      REPLICATE("X"+CHR(7), nCol2 - nCol1 + 2 ) ) )  
  RESTSCREEN( nRen1+1, nCol2+1, nRen2, nCol2+2 ;  
    TRANSFORM( SAVESCREEN(nRen1+1, nCol2+1, nRen2, nCol2+2);  
      REPLICATE("X"+CHR(7), (nRen2 - nRen1 + 1)*2 ) ) )  
RETURN NIL
```

```
/***  
* SombraMensaje( <nY1>, <nX1>, <nX2>, <cAtributos> ) --> NIL  
*  
* Da la apariecia de que una opción de un menú encontrado en  
* las coordenadas <nY1>, <nX1> y <nY1>, <nX2> este flotando en  
* la pantalla.  
*  
*/
```

```
FUNCTION SombraMensaje(nY1,nX1,nX2,cAtributos)  
  LOCAL ColoresViejos := SETCOLOR()  
  
  SETCOLOR(cAtributos)  
  @ nY1+1, nX1+1 SAY REPLICATE(CHR(223),nX2-nX1+1)  
  @ nY1, nX2+1 SAY CHR(220)  
  SETCOLOR(ColoresViejos)  
RETURN NIL
```

```
/*  
* Boton ( <nY1>, <nX1>, <nX2> ) --> NIL  
*  
* Simula en la pantalla el efecto de presionar un botón  
* en las coordenadas <nY1>, <nX1> y <nY1>, <nX2>.  
*  
*/  
FUNCTION Boton ( nY1,nX1,nX2 )  
  LOCAL s1, s2, ColoresViejos := SETCOLOR(), ColorNuevo  
  
  s1 = SAVESCREEN(nY1,nX1,nY1+1,nX2+1) // Salva el mensaje y su marco  
  s2 = SAVESCREEN(nY1+1,nX1,nY1+1,nX1) // Salva el color  
  ColorNuevo = STR( ASC(RIGHT(s2,1)) % 16 )+CHR(47)+;  
    STR(INT (ASC(RIGHT(s2,1))/16))+',,,,'  
  SETCOLOR(ColorNuevo)  
  @ nY1+1, nX1+1 SAY REPLICATE(LEFT(s2,1),nX2-nX1+1)  
  RESTSCREEN(nY1,nX1,nY1,nX1,s2)  
  RESTSCREEN(nY1,nX1+1,nY1,nX2+1,s1)  
  Delay(0.18)  
  RESTSCREEN(nY1,nX1,nY1+1,nX2+1,s1)  
  Delay(0.18)  
  SETCOLOR(ColoresViejos)  
RETURN NIL
```

```
/*  
* RamaV ( <nY1>, <nX1>, <nY2>, <cTipoRama>, <cAtributos> ) --> NIL  
*  
* Dibuja una línea vertical con la punta superior en <nY1>, <nX1>  
* con el primer caracter pasado por <cTipoRama> y de color <cAtributos>  
* con la punta inferior en <nY2>, <nX1> con el tercer caracter pasado  
* por <cTipoRama> y de color <cAtributos> y con el resto de los  
* caracteres comprendidos en ambas punta por el segundo caracter pasado  
* por <cTipoRama> y de color <cAtributos>.  
*  
*/  
FUNCTION RamaV ( nY1,nX1,nY2,cTipoRama,cAtributos )  
  LOCAL j, ColoresViejos := SETCOLOR()  
  
  SETCOLOR(cAtributos)  
  @ nY1, nX1 SAY LEFT(cTipoRama,1)  
  FOR j = nY1+1 TO nY2-1  
    @ j, nX1 SAY LEFT(RIGHT(cTipoRama,2),1)  
  NEXT  
  @ nY2, nX1 SAY RIGHT(cTipoRama,1)
```



```
SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/*
* RamaH ( <nY1>, <nX1>, <nX2>, <cTipoRama>, <cAtributos> ) --> NIL
*
```

```
* Dibuja una línea horizontal con la punta izquierda en <nY1>,
* <nX1> con el primer caracter pasado por <cTipoRama> y de color
* <cAtributos> con la punta derecha en <nY1>, <nX2> con el tercer
* caracter pasado por <cTipoRama> y de color <cAtributos> con el
* resto de los caracteres comprendidos en ambas punta y por el
* segundo caracter pasado por <cTipoRama> y de color <cAtributos>.
*
```

```
*/
FUNCTION RamaH ( nY1,nX1,nX2,cTipoRama,cAtributos )
  LOCAL j, ColoresViejos := SETCOLOR()
```

```
  SETCOLOR(cAtributos)
  @ nY1, nX1 SAY LEFT(cTipoRama,1)
  FOR j = nX1+1 TO nX2-1
    @ nY1, j SAY LEFT(RIGHT(cTipoRama,2),1)
  NEXT
  @ nY1, nX2 SAY RIGHT(cTipoRama,1)
  SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/*
* VentanaExplosiva ( <nY1>,<nX1>,<nY2>,<nX2>,<cTipoMarco>,<cAtributos> )
* --> NIL
*
```

```
* Crea el efecto de una ventana explosiva o que crece con los
* caracteres pasados por <cTipoMarco> y del color <cAtributos> hasta
* llegar a las coordenadas especificadas por <nY1>, <nX1>, <nY2>, <nX2>.
*
```

```
*/
FUNCTION VentanaExplosiva(nY1,nX1,nY2,nX2,cTipoMarco,cAtributos)
  LOCAL Xp1, Yp1, ColoresViejos := SETCOLOR()
```

```
  SETCOLOR(cAtributos)
  G = SAVESCREEN(0,0,24,79)
  IF (nX2-nX1+1) % 2 = 0
    Xp1 = nX1-1
```

```
ELSE
  Xp1 = nX1
ENDIF
IF (nY2-nY1+1) % 2 = 0
  Yp1 = nY1+1
ELSE
  Yp1 = nY1
ENDIF
NumVen = 1
CentroX = INT ((nX2+Xp1) / 2)
CentroY = INT ((nY2+Yp1) / 2)
DO WHILE CentroX - INT(( 6*NumVen-3) / 2) > Xp1 .AND.;
  CentroY - INT(( 2*NumVen-1) / 2) > Yp1
  @ CentroY - INT((2*NumVen-1) / 2), CentroX - INT((6*NumVen-3) / 2),;
  CentroY + INT((2*NumVen-1) / 2), CentroX + INT((6*NumVen-3) / 2) BOX;
  cTipoMarco+CHR(32) COLOR cAtributos
  Delay(0.0005)
  ++NumVen
ENDDO
@ nY1, nX1, nY2, nX2 BOX cTipoMarco+CHR(32) COLOR cAtributos
SETCOLOR(ColoresViejos)
RETURN NIL

/****
* VentanaCML ( <nY1>,<nX1>,<nY2>,<nX2>,<cTipoMarco>,<cAtributos> ) --> NIL
*
* Crea el efecto de una ventana implosiva o que se contrae con los
* caracteres pasados por <cTipoMarco> y del color <cAtributos>
* comenzando por las coordenadas <nY1>, <nX1>, <nY2>, <nX2>.
*
*/
FUNCTION VentanaCML (nY1,nX1,nY2,nX2,cTipoMarco,cAtributos)
  LOCAL j, ColoresViejos := SETCOLOR()

  SETCOLOR(cAtributos)
  FOR j = NumVen TO 1 STEP -1
    RESTSCREEN(0,0,24,79,G)
    @ CentroY - INT((2*j-1) / 2), CentroX - INT((6*j-3) / 2),;
    CentroY + INT((2*j-1) / 2), CentroX + INT((6*j-3) / 2) BOX;
    cTipoMarco+CHR(32) COLOR cAtributos
    Delay(0.001)
  NEXT
  RESTSCREEN(0,0,24,79,G)
  SETCOLOR(ColoresViejos)
RETURN NIL
```

```
/*  
* Temporizar( <nCiclo> ) --> NIL  
*  
* Detiene momentáneamente la ejecución  
* de un proceso en <nCiclo>*100.  
*  
*/  
FUNCTION Temporizar( nCiclo )  
  LOCAL i := 0  
  
  DO WHILE i <= nCiclo*100  
    ++i  
  ENDDO  
RETURN NIL  
  
/*  
* Delay( < nSegundos > ) --> NIL  
*  
* Esta función detiene momentáneamente la ejecución de  
* un proceso en segundos según el parámetro < nSegundos >.  
*  
*/  
FUNCTION Delay(nSegundos)  
  // Se definen las variables que controlan la cantidad de tiempo de espera  
  LOCAL TiempoFinal, TiempoInicial  
  
  TiempoFinal = SECONDS()+nSegundos  
  TiempoInicial = SECONDS()  
  DO WHILE TiempoFinal >= TiempoInicial  
    TiempoInicial = SECONDS()  
  ENDDO  
RETURN NIL  
  
/*****/
```

CAPÍTULO 5

MANUAL DEL USUARIO.

En este capítulo presentamos una guía que pretende orientar tanto al usuario experto como inexperto, sobre el hardware y software mínimo que se requiere para la instalación del DEC versión 1.0 y sobre el manejo de las pantallas y opciones de los menús del mismo .

5.1. HARDWARE NECESARIO

LA MEMORIA RAM

El programa "DEC" fue desarrollado para correr en equipos PC'S sin importar el tipo de microprocesador, pero con la condición necesaria de que el equipo tenga de memoria RAM base 640 Kb, esto es debido a que el "DEC.EXE" tiene un tamaño de poco más de 427 Kb y a demás, el programa para su funcionamiento crea archivos temporales, pilas de tamaño variable y apuntadores hacia los archivos y registros empleados.

EL MICROPROCESADOR Y LA TARJETA MADRE

Idealmente sería conveniente disponer de un microprocesador de la serie PENTIUM¹¹ o compatible con su respectiva tarjeta madre, ya sea un PENTIUM a 100, 120, 133 Mhz o a mayores velocidades. Lo anterior obedece a las siguientes e importantes razones.

1) Las PC'S con microprocesador 8086 al 80486 tiene un "detalle" en cuanto a su diseño y por consiguiente en su fabricación : la fecha y hora máximas que estas

¹¹ Los microprocesadores PENTIUM, son la serie posterior a los microprocesadores 80486 fabricados por INTEL®.

pueden registrar es hasta las 23:59:59.99 del día 31/12/1999 ¡ No existe el año 2000 ni mayores a este... ! (Si el lector posee una de estas computadoras y se encuentra escéptico ante esta situación, le sugerimos intente modificar la fecha y hora de su sistema y asombrosamente descubrirá que la fecha no puede ajustarse al año 2000 o mayores) ¡ Y ese es todo el problema ! estará seguramente pensando, pero hagamos la conjetura de que el programa "DEC" siga operando hasta el año 2000 ¿ Qué pasará con aquellos procesos que realiza el programa en los cuales siempre se involucre la fecha del sistema para obtener un reporte ?

Como el programa para generar la mayor parte de los reportes emplea la fecha del sistema, las PC'S con microprocesador 8086 al 80486 son funcionales para el mismo hasta antes del año 2000. Como en los PENTIUM ya se resolvió este problema de la fecha, resultan más idóneos este tipo de microprocesadores.

2) La velocidad de procesamiento de los PENTIUM es más alta por lo cual se agilizan más los cálculos y resultados del programa.

3) Tanto el hardware como el software se diseñan para cierto número de bits, por lo que el uso de paquetes y periféricos más modernos y sofisticados que aparezcan a futuro podrá ser falible.

4) Los microprocesadores PENTIUM son multitarea, lo cual permite trabajar más efectivamente con diversas aplicaciones abiertas.

EL DISCO DURO

Debido a la gran cantidad de datos que se van almacenar, el disco duro resulta mas que necesario indispensable.

Consideramos que el disco duro posea al menos 4 Mb libres para descargar en ellos el programa y las bases de datos.

LOS MANEJADORES DE DISCO

Para realizar respaldos de la información de la base de datos es conveniente tener instalado un manejador de disco. Sugerimos uno de 3 1/2" de alta o ultra alta densidad.

LA IMPRESORA

Para imprimir los reportes es básico contar con una impresora.

El programa genera reportes muy sencillos para imprimir, pero si desea reportes con una salida más profesional con su toque personal, el programa permite generar el reporte deseado en un archivo de texto. Este último se puede importar desde el procesador de textos de su preferencia y desde ahí hacer las modificaciones e impresiones respectivas.

Consideramos que el tipo de la impresora sea apegado a las necesidades económicas, de carga de trabajo y de calidad de impresión requeridas. Aunque es necesario especificar al programa el tipo de impresora que se posee para que el mismo cargue la configuración adecuada, esto se explicará más adelante en el menú de disposición del programa.

EL NO-BREAK

Un no-break o UPS es un dispositivo electrónico, que permite suministrar energía eléctrica por breve tiempo, a todos los equipos de cómputo conectados a él cuando la energía comercial se suspende. Así, un no-break nos evita la desafortunada pérdida de datos y daño de archivos, cediendo tiempo y energía necesaria para que el usuario termine sus procesos, realice los respaldos requeridos y cierre sus archivos.

No forma parte del presente estudio indicar que tipo de no-break es el más adecuado, pero resultaría idóneo tener uno instalado.

5.2. SOFTWARE NECESARIO

Como el programa DEC fue desarrollado en CLIPPER, su ambiente natural de trabajo es en el DOS, aunque puede ejecutarse desde WINDOWS, en su modalidad de ventana, ganando con ello la ventaja de tener abiertas varias aplicaciones.

Si el programa se ejecuta desde la plataforma del DOS, es importante que la versión del mismo por cuestiones de direccionamiento del disco duro, sea superior a la 3.

¡ SUGERENCIA ! Si emplea diversas utilerías, las cuales son ejecutables desde ambiente WINDOWS, puede ejecutar mejor el programa DEC desde ahí. De esta manera puede tener habilitadas las ventanas de varias aplicaciones y agilizar sus procesos.

5.3. INSTALACIÓN Y ARRANQUE DEL PROGRAMA DEC

Debido a que es más rápido el acceso a un disco duro que a un disco blando (floppy disk), es conveniente que el programa DEC resida en el primero.

Para instalar el DEC, simplemente puede crear un subdirectorio con el nombre que Usted desee, en la unidad del disco duro. Posteriormente, cambie al subdirectorio creado y copie en ese directorio el archivo DEC.EXE. Por ejemplo suponiendo que deseamos instalar en la unidad C:\ el DEC.EXE en un subdirectorio llamado DEC1 procederíamos de la siguiente manera.

- Situarnos en la raíz de la unidad C:

```
C:\>
```

- Crear el subdirectorio DEC1

```
C:\>MD DEC1
```

- Colocarse en el subdirectorio DEC1

```
C:\>CD DEC1
```

- Colocar el disco del DEC en la unidad A:

- Copiar el archivo DEC.EXE en el directorio seleccionado.

```
C:\>DEC1>COPY A:DEC.EXE
```

- Si desea correr desde este momento el DEC simplemente teclee DEC

C:\>DEC1>DEC

No olvide presionar <ENTER> después de cada instrucción del DOS.

OBSERVACIÓN. Si se dispone de utilerías más sofisticadas como el PC TOOLS¹² o WINDOWS, puede emplearlas también para crear el subdirectorio y copiar el archivo DEC.EXE, e incluso puede ejecutar desde ahí el programa.

5.4. LOS MENÚS Y PANTALLAS DEL PROGRAMA DEC VERSIÓN 1.0

LA PANTALLA DE PRESENTACIÓN

La primer pantalla que aparece al entrar al programa DEC, es la de presentación, la cual se muestra en la figura 5.1.

Como se puede apreciar, aparece una ventana al centro, en la cual se da una explicación general sobre la aplicación y uso del programa. Si desea leer el resto del texto de esta pantalla, puede emplear las flechas del cursor, <ENTER> o las teclas de avance de página.

Para salir de esta pantalla presione <ESC>.

OBSERVACIÓN. A partir de este momento, puede presionar la tecla <ESC> para abortar o salir de cualquier situación, no importando el módulo del programa en donde se encuentre.

¹² La denominación PC Tools® es una marca registrada por Central Point Software Inc®.

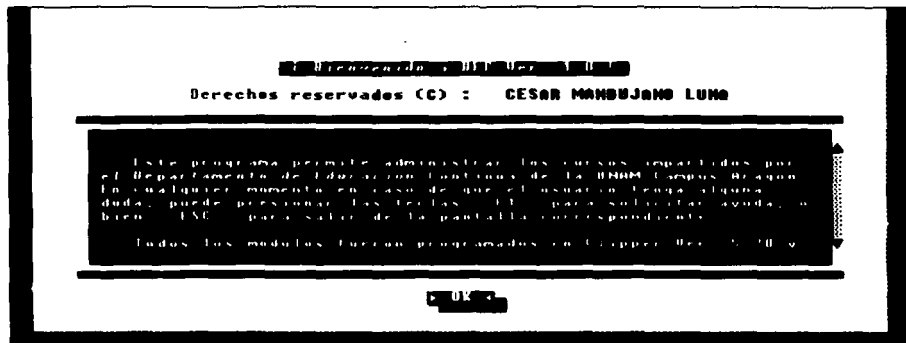


Figura 5.1. La pantalla de presentación.

OBSERVACIÓN. El programa detecta si hacen falta archivos para su funcionamiento, ya sean de bases de datos (extensión .DBF), de indexación (extensión .NTX), de memorándum (extensión .DBT), o de configuración (extensión .DEC). En caso de no encontrar algún archivo de base de datos o de indexación, el programa los creará. En caso de que sea un archivo de configuración, el programa tomará valores preestablecidos en el por defecto. Pero en caso de que el archivo faltante sea uno de memorándum, el programa se bloqueará y no podrá continuar más. Ante esta situación, es conveniente que se capture la estructura de la base de datos que falta (ver capítulo 3) ya sea desde CLIPPER, xBASE o bien desde un manejador de bases de datos compatible.

LA PANTALLA DE SOLICITUD DE CLAVE DE ACCESO

Al cerrar la pantalla de presentación, el programa DEC solicitará la clave de acceso para poder realizar altas, bajas y modificaciones a las bases de datos (ver figura 5.2).

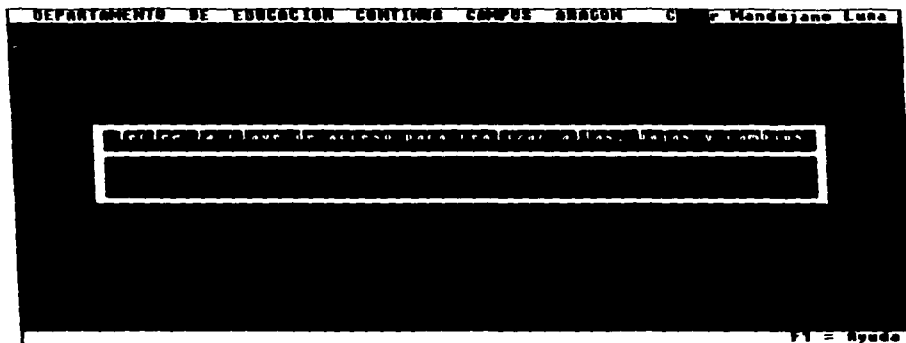


Figura 5.2. La pantalla de solicitud de clave de acceso.

Por razones de seguridad, esta clave sólo la conoce y es modificada por personal autorizado.

La clave de acceso está compuesta desde uno a cuatro caracteres y el usuario tiene hasta tres oportunidades para teclearla adecuadamente. Si fallara en los tres intentos o presiona <ESC>, el programa sólo le dará acceso a consultas, reportes y algunas opciones del módulo de disposición.

Si se tiene alguna duda sobre la acción a realizar en esta pantalla presione la tecla de ayuda F1 y aparecerá la figura 5.3.

OBSERVACIÓN. A partir de este momento, puede presionar la tecla <F1> para activar la ayuda correspondiente en caso de duda o aclaración, no importando el módulo del programa en donde se encuentre.

De antemano, todo el programa maneja ayudas de tipo modular y posicional. Estas se activan al presionar la tecla F1 y se desactivan al presionar la tecla ESC.

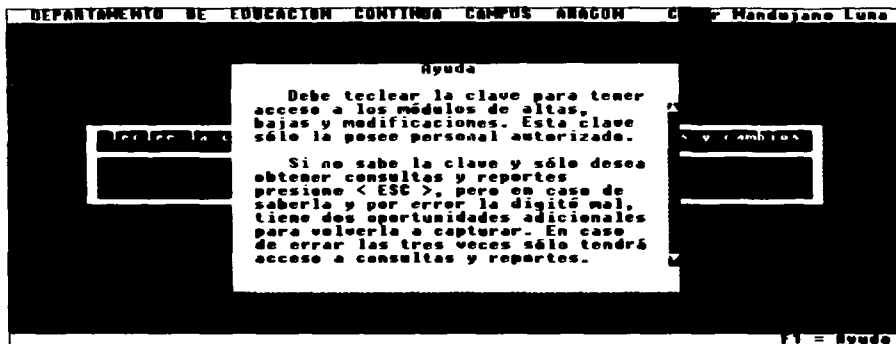


Figura 5.3. Ejemplo de la pantalla de ayuda.

LA PANTALLA DEL MENÚ PRINCIPAL.

Al salir de la pantalla de captura de clave de acceso, aparece la pantalla del menú principal la cual se ilustra en la figura 5.4. En él, se muestran siete diferentes opciones que son : altas, bajas, consultas, modificaciones, reportes, disposición y salir. También en la parte inferior del mismo aparece en conjunto de teclas válidas para este menú.

Para escoger una alternativa, puede seleccionar con las flechas del cursor, colocarse en la opción que desea y presionar <ENTER> o bien, presionar la letra inicial de la opción que desea.

Si presiona <ESC> aparecerá una pantalla que solicitará la confirmación de salida del programa. Esta pantalla se muestra en la figura 5.5.

Para conocer la versión del programa presione la tecla del tabulador (TAB). Al hacer esto aparecerá la pantalla que se muestra en la figura 5.6.

Al presionar la tecla de retroceso (back space), se presenta una pantalla con la información general sobre aplicación y uso del programa (ver figura 5.7.).

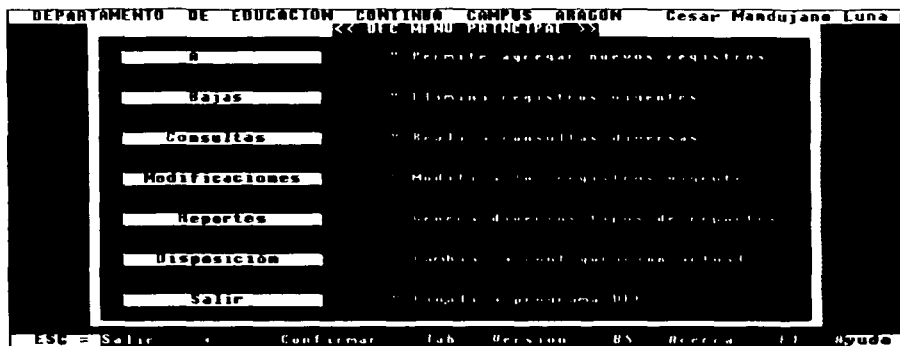


Figura 5.4. La pantalla del menú principal.

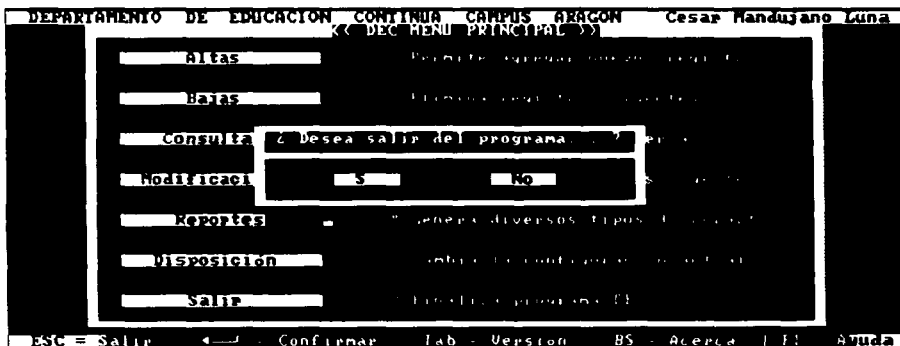


Figura 5.5. La pantalla de confirmación de terminación del programa.

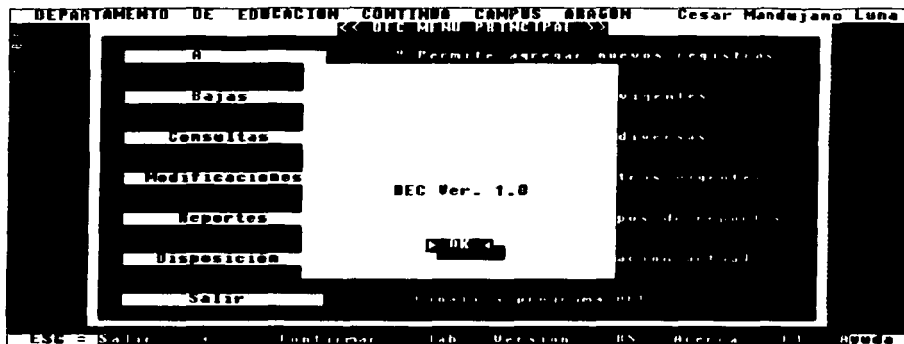


Figura 5.6. La pantalla de versión del programa.

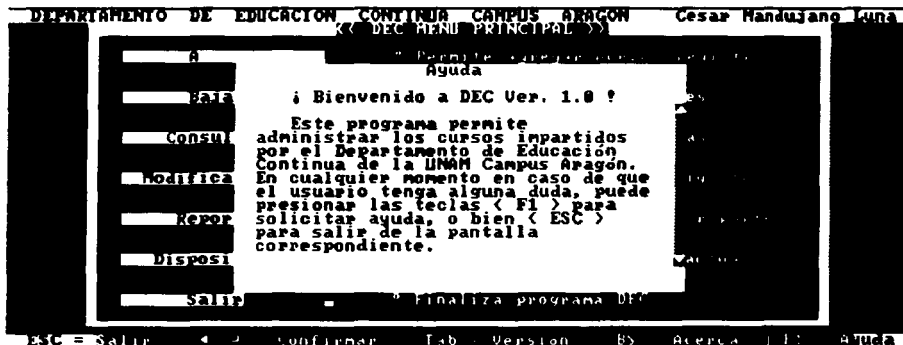


Figura 5.7. La pantalla de explicación del programa.

Puede presionar F1 para solicitar ayuda para el menú principal. Si realiza esto aparecerá la pantalla que se ilustra en la figura 5.8.

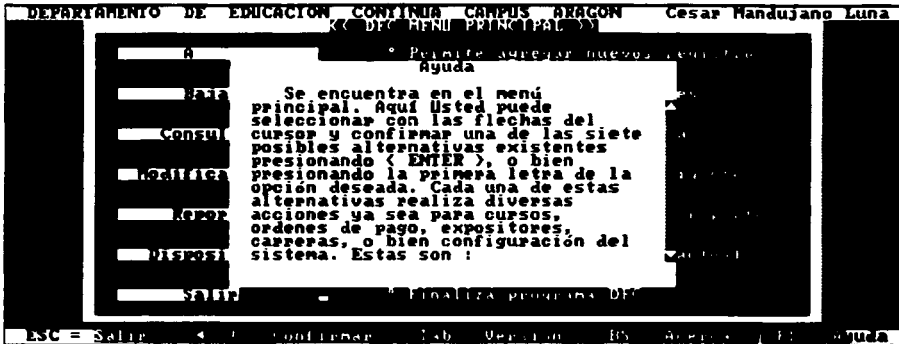


Figura 5.8. La pantalla de ayuda del menú principal.

OBSERVACIÓN. Todas las teclas válidas para el menú principal y sus acciones, aparecen en la parte inferior de la pantalla y son también válidas para todos los menús con este mismo formato. Así por ejemplo, menús de igual formato a este como lo son los de altas, bajas, consultas, cambios, disposición, impresora, etc., tienen idéntico modo de trabajo.

EL MENÚ GENERAL DE ALTAS

Al seleccionar la opción de altas del menú principal, aparecerá el menú general de altas el cual se muestra en la figura 5.9.

Como se puede observar este menú está compuesto por cinco diversas opciones que son : cursos, ordenes, expositores, carreras y salir.

Las cuatro primeras opciones realizan respectivamente las altas de cursos, ordenes de pago, expositores de cursos y carreras disponibles de la institución. La quinta opción permite regresar a la pantalla anterior que es la del menú principal.

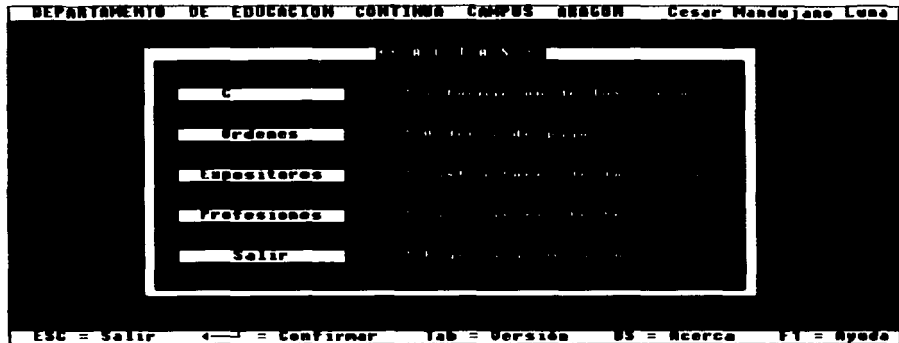


Figura 5.9. La pantalla del menú general de altas.

Las teclas válidas de este menú, son las mismas que las empleadas en el menú principal debido a que tienen el mismo formato. Por ejemplo, si deseara seleccionar un alta de expositores, puede pulsar simplemente la tecla "E" , o bien situarse en la opción "Expositores" con las flechas del cursor y pulsar < ENTER > (para mayor referencia, vuelva a consultar el manejo del menú principal.).

LA PANTALLA DE ALTAS DE CURSOS

Al seleccionar la opción "cursos" del menú de altas, aparecerá la pantalla de altas de cursos, la cual se muestra en la figura 5.10.

Esta pantalla la podemos dividir en tres áreas : a) zona de captura de datos; b) zona de memorándum, y c) zona de confirmación.

« ALTAS DE CURSOS »

Clave curso	00000000
Alumnos	5 0 00
Académicos	5 0 00
Administrativos	5 0 00
Mantenimiento	5 0 00
Expedidos	5 0 00
Público general	5 0 00
Nombre curso	
Periodo	
Lugar	
Duración	0 HRS
Horario	00 00 00 00 HRS
Clase expositiva	
Cupo	0
Salas	
Clase carrera	000
Días	[N] Lu [N] Ma [N] Mi [N] Ju [N] Vi [N] Sa [N] Do
Observaciones	Memorandum

Figura 5.10. La pantalla de altas de cursos.

a) ZONA DE CAPTURA DE DATOS

En esta zona el usuario debe de capturar los datos de un nuevo curso.

Las teclas válidas para esta zona se muestran en la figura 5.11, así como sus diferentes acciones.

Es muy importante señalar, que de capturar un valor incorrecto en cualquiera de los datos, el programa avisará que se ha incurrido en un error sólo en el momento en que se confirme la alta o modificación del registro, mientras dejará al usuario capturar libremente.

TECLA VALIDA	ACCIÓN
< ESC >	Cancelar alta.
< Enter >	Confirmar un dato capturado y pasar al siguiente campo. De ser el último campo pasar a la zona de memorándum.
< Pg/Dn >	Pasar a la zona de memorándum.
< Pg/Up >	Pasar a la zona de confirmación.
< Flecha cursor arriba >	Pasar a un campo anterior. Si es el primer campo pasar a la zona de confirmación.
< Flecha cursor abajo >	Pasar al siguiente campo. Si es el último campo pasar a la zona de memorándum.
< F1 >	Ayuda.
< Back Space >	Borrar caracter anterior.
< Del / Supr >	Borrar Caracter actual.
< Ins >	Encender/Apagar modo de inserción de caracteres.
< Home >	Pasar al primer caracter del campo actual.
< End >	Pasar al último caracter del campo actual.

Figura 5.11. El conjunto de teclas válidas para la zona de captura de datos de cursos.

b) ZONA DE MEMORÁNDUM

Como se puede ver en la pantalla de altas de cursos existe una alternativa llamada "memorándum", la cual permite capturar una serie de observaciones asociadas al curso que se desea dar de alta. Para abrir la pantalla de captura de observaciones, seleccione la opción memorándum y presione <ENTER>. Al hacer esto aparecerá la pantalla de captura la cual se muestra en la figura 5.12 y cuyo uso es muy similar a la mayoría de procesadores de textos.

A partir de este momento puede capturar todas aquellas observaciones que sean pertinentes citar de un curso. Para cerrar esta pantalla, puede presionar ya sea <CTRL>+<W > para salvar el texto capturado o bien < ESC > para no grabarlo.

OBSERVACIÓN. No es obligatorio entrar ni capturar el contenido de un campo memorándum.

Las teclas válidas para esta zona se muestran en la figura 5.13, así como sus diferentes acciones.

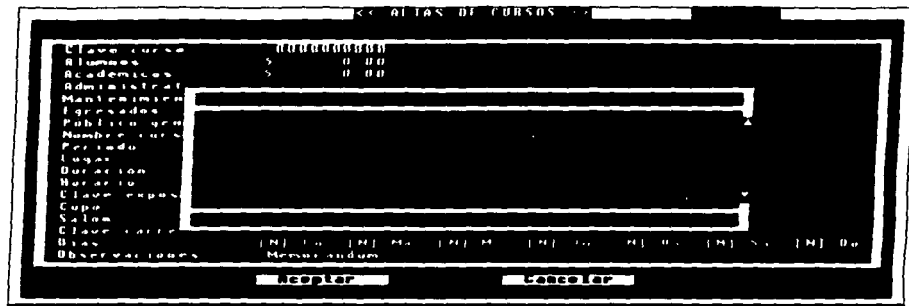


Figura 5.12. La pantalla de memorándum.

TECLA VALIDA	ACCIÓN
< ESC >	Cancelar alta.
< Enter >	Abrir pantalla de la zona de memorándum.
< Pg/Dn >	Passar a la zona de confirmación.
< Pg/Up >	Passar a la zona de captura.
< Flecha cursor arriba >	Passar a la zona de captura.
< Flecha cursor abajo >	Passar a la zona de confirmación.
< F1 >	Ayuda.

Figura 5.13. El conjunto de teclas válidas para la zona de memorándum de cursos.

c) ZONA DE CONFIRMACIÓN

Esta parte de la pantalla permite confirmar o cancelar el alta del curso dado. Si se desea cancelar seleccione la opción que dice "Cancelar" y presione <ENTER> o bien si se encuentra en esta zona presione la letra "C" o simplemente presione <ESC>. Si desea confirmar seleccione la opción "Aceptar" y presione <ENTER> o bien si se encuentra en esta zona presione la letra "A".

Las teclas válidas para esta zona se muestran en la figura 5.14, así como sus diferentes acciones.

TECLA VALIDA	ACCIÓN
< ESC >	Cancelar alta.
< Enter >	Confirmar/Cancelar alta.
< Pg/Dn >	Passar a la zona de captura.
< Pg/Up >	Passar a la zona de memorándum.
< Flecha cursor arriba >	Passar a la zona de memorándum.
< Flecha cursor abajo >	Passar a la zona de captura.
< F1 >	Ayuda.

Figura 5.14. El conjunto de teclas válidas para la zona de confirmación de cursos.

OBSERVACIÓN. Por regla general, en cualquiera de las bases de datos, no se puede dar de alta un registro con una clave ya existente.

OBSERVACIÓN. Debido a que el programa realiza consultas a las diversas bases de datos para validar la información que el usuario introduce, es muy importante tener en cuenta que el programa detecta errores de captura sólo en el momento en que se confirma el alta y no en el instante de su captura misma. Esto es con el afán de no acceder continuamente al disco y para ahorrar tiempo en la validación de datos.

OBSERVACIÓN. De tener errores de captura, el programa enviará un mensaje explicando el motivo que originó ese error y permitirá la corrección de los mismos cuantas veces sea necesario.

En caso de confirmar un alta y existan errores en la captura de los datos, el sistema lo notificará al usuario por medio de mensajes y el alta no procederá hasta que se corrijan los errores encontrados.

Por ejemplo supongamos que en el alta de un curso se captura una clave inválida, aparecerá el mensaje que se muestra en la figura 5.15.

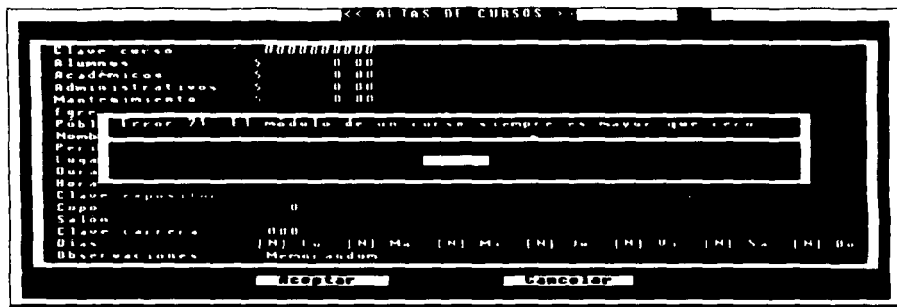


Figura 5.15. Ejemplo de la pantalla de mensaje de error.

OBSERVACIÓN. Para desaparecer cualquier pantalla de mensaje de error, puede presionar la tecla < ENTER >, o bien simplemente espere unos cuantos segundos y deje que la pantalla de error desaparezca por si sola.

OBSERVACIÓN. En caso de confirmar un alta o modificación de un curso, el programa coloca automáticamente en la pantalla el nombre del expositor y de la carrera que apoya al curso según las claves que se hayan capturado para estos y grabará las actualizaciones realizadas en el disco de trabajo.

Para terminar, independientemente de que se cancele o acepte un alta, el programa solicitará si desea realizar otra alta o no (ver figura 5.16). En caso afirmativo, se limpiará la pantalla de altas de cursos para permitir más capturas, de otro modo aparecerá el menú principal.

<< ALTAS DE CURSOS >>

Clave curso	0000000000
Alumnos	5 0 00
Académicos	5 0 00
Administrativos	5 0 00
Mantenimiento	5 0 00
Egresados	
Público general	
Nombre curso	¿ Desee realizar otra alta
Periodo	
Lugar	
Duración	
Horario	
Clase expositiva	000
Cupo	0
Salón	
Clave carrera	000
Días	[N] Lu [N] Ma [N] Mi [N] Ju [N] Vi [N] Sa [N] Do
Observaciones	Memorandum

Aceptar Cancelar

Figura 5.16. La pantalla de solicitud de nueva alta.

LA PANTALLA DE ALTAS DE ORDENES DE PAGO

Al seleccionar la opción "ordenes" del menú de altas, aparecerá la pantalla de altas de ordenes de pago, la cual se muestra en la figura 5.17.

Esta pantalla la podemos dividir en cinco áreas : a) zona de captura de datos; b) zona de costo del curso; c) zona de recibos; d) zona de memorándum, y e) zona de confirmación.

a) ZONA DE CAPTURA DE DATOS

En esta zona el usuario debe de capturar los datos de una nueva orden de pago. Esta zona de captura de datos opera de manera similar a la de cursos.

Para esta zona las teclas válidas se muestran en la figura 5.18

<< ALTAS DE ORDENES DE PAGO >>

Clave del curso
Nombre del alumno	
BIC	
Teléfono	00
Clave procedencia	00
Clave carrera	000
Número de cuenta	00000000
Condición	00
Semestre	00
Período	
Expositor	
Salón	
Costo del curso	
Número de recibo	Recibos
Observaciones	Memorandum

Figura 5.17. La pantalla de altas de ordenes de pago.

TECLA VALIDA	ACCIÓN
< ESC >	Cancelar alta.
< Enter >	Confirmar un dato capturado y pasar al siguiente campo. De ser el último campo pasar a la zona de costo del curso.
< Pg/Dn >	Pasar a la zona de costo del curso.
< Pg/Up >	Pasar a la zona de confirmación.
< Flecha cursor arriba >	Pasar a un campo anterior. Si es el primer campo pasar a la zona de confirmación.
< Flecha cursor abajo >	Pasar al siguiente campo. Si es el último campo pasar a la zona de costo del curso.
< F1 >	Ayuda.
< Back Space >	Borrar caracter anterior.
< Del / Supr >	Borrar Caracter actual.
< Ins >	Encender/Apagar modo de inserción de caracteres.
< Home >	Pasar al primer caracter del campo actual.
< End >	Pasar al último caracter del campo actual.

Figura 5.18. El conjunto de teclas válidas de la zona de captura de datos de ordenes de pago.

OBSERVACIÓN. Si tecllea la clave de un curso que no existe, se producirá un error y no podrá pasar a ninguna de las cuatro zonas restantes.

b) ZONA DE COSTO DEL CURSO

Como se puede ver en la pantalla de altas de ordenes de pago, existe una alternativa llamada "Costo del curso", la cual permite capturar el importe correspondiente del curso al cual el interesado desea inscribirse.

Note que dependiendo de la condición del alumno que se teclee en la zona de captura, el precio es colocado automáticamente, aunque de ser necesario puede ser modificado.

Las teclas válidas se muestran en la Figura 5.19.

TECLA VALIDA	ACCIÓN
< ESC >	Cancelar alta.
< Enter >	Confirmar un costo de curso y pasar a la zona de recibos.
< Pg/Dn >	Pasar a la zona de recibos.
< Pg/Up >	Pasar a la zona de captura.
< Flecha cursor arriba >	Pasar a la zona de captura.
< Flecha cursor abajo >	Pasar zona de recibos.
< F1 >	Ayuda.
< Back Space >	Borrar caracter anterior.
< Del / Supr >	Borrar Caracter actual.
< Ins >	Encender/Apagar modo de inserción de caracteres.
< Home >	Pasar al primer caracter del precio capturado.
< End >	Pasar al último caracter del precio capturado.

Figura 5.19. El conjunto de teclas válidas de la zona de costo del curso de ordenes de pago.

OBSERVACIÓN. El costo de un curso para un interesado siempre se ajusta por defecto al ya existente en la base de datos de cursos. Por ello se debe tener precaución si modificó el costo de un curso. Por ejemplo , supongamos que modificó el costo de un curso y pasa a cualquier otra zona. Después de hacer lo anterior regresa a la zona de costo del curso, notará que el valor que había modificado, cambia al que tiene grabado la base de datos de cursos. Para evitar esto procure capturar las cinco zonas en el orden en que están dadas. De esta forma se tiene la garantía de grabar en disco el valor tecleado y no el valor por defecto.

c) ZONA DE RECIBOS

Como se puede ver en la pantalla de altas de ordenes de pago, existe una alternativa llamada "Recibos", la cual permite ingresar el número, los importes y las fechas del conjunto de recibos pagados por el participante. Para poder entrar, presione la tecla <ENTER> y aparecerá el menú que se ilustra en la figura 5.20.

Como puede apreciarse en la figura 5.20, a su vez se compone de tres zonas : zona de menú de opciones; zona de captura de datos del recibo y zona de saldos.

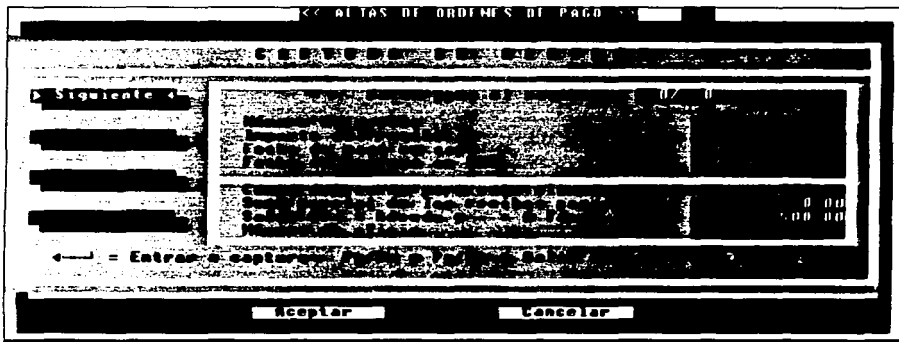


Figura 5.20. La pantalla de la zona de recibos.

La zona de menú de opciones

Las opciones que se manejan en esta zona son cuatro : 1) Siguiente, que permite entrar a la zona de captura de datos del recibo para editar el contenido del recibo posterior al actual; 2) Anterior, que permite entrar a la zona de captura de datos del recibo, para editar el contenido del recibo anterior al actual; 3) Borrar, que elimina el contenido del recibo actual, y 4) salir, que cierra esta pantalla (es equivalente a presionar la tecla < ESC > .).

Para escoger una opción, seleccione la con las flechas del cursor y presione < ENTER >.

La zona de captura de datos del recibo.

Esta zona permite ingresar el número, los importes y las fechas de pago en caja y de entrega de recibo en el DEC Aragón.

Para salir de esta zona y pasar a la zona de menú de opciones presione la tecla < Pg/Dn > o la tecla < Pg/Up >. En caso de querer salir y cerrar esta pantalla presione la tecla < ESC >.

La zona de saldos

Esta zona no es accesible para el usuario. Su única función es exhibir el costo del curso para el registro actual, la suma total de los recibos parciales pagados, el saldo que se tiene y el número de recibo oficial en donde se reportó dicho recibo.

Es obvio pensar que un recibo nuevo no reportado, todavía no tiene un número de oficio asignado.

d) ZONA DE MEMORÁNDUM

Esta zona permite capturar una serie de observaciones asociadas a la orden de pago que desea dar de alta.

Su modo de operar y conjunto de teclas válidas, es idéntico que el empleado en el de altas de cursos. (ver zona de memorándum de altas de cursos para mayor referencia.).

e) ZONA DE CONFIRMACIÓN

Esta parte de la pantalla permite confirmar o cancelar el alta de una nueva orden de pago.

Su modo de operar y conjunto de teclas válidas, es idéntico que el empleado en el de altas de cursos. (ver zona de confirmación de altas de cursos para mayor referencia.).

Como un curso tiene asociada una capacidad o cupo, el total de ordenes de pago existentes para el mismo no deben sobrepasar su capacidad, a menos de que el usuario este de acuerdo en aceptar el exceso de alumnos en un curso, o bien se amplíe la capacidad del mismo, modificando el campo cupo del curso de interés.

En caso de que al dar de altas o modificar los datos de una orden de pago y se sobrepase el cupo de un curso, el programa emitirá un aviso indicándole al usuario si esta de acuerdo en aceptar este registro. Esta pantalla se ilustra en la figura 5.21.

De aceptar, el alta queda confirmada y el nuevo registro se grabará, de lo contrario el programa permitirá cambiar los datos del registro capturado.

KX ALTAS DE ORDENES DE PAGO

Clase del curso	IIIIIIIIII
Nombre del alumno	
RFI	
Telefono	II

Se ha sobrepasado el cupo para este curso ¿ Continúa con la grabación ?

Clase del curso	IIIIIIIIII
Costo del curso	
Numero de inscrito	IIIIIIIIII
Observaciones	IIIIIIIIII

Figura 5.21. La pantalla de sobrecupo de inscritos un cursos.

En caso de confirmar en el alta de recibos uno cuyo número de recibo ya existe en la base de datos de ordenes de pago, el programa enviará un aviso al usuario indicándole que se ha capturado un recibo con número ya repetido e inmediatamente listará el conjunto de recibos en donde encontró la recurrencia. Esta pantalla se ilustra en la figura 5.22.

RECIBO ENCONTRADO	CANTIDAD	FECHA EN DIC	DE DIC	CLAVE CURSO
1	1.000.00	15/05/1996	1	10101011
1	1.000.00	17/08/1996	1	10101011

Figura 5.22. La pantalla del listado de recurrencia de recibos.

Este reporte puede ser impreso y/o almacenado en un archivo. Consulte el apartado de reportes para mayor referencia.

Para desplazarse a lo largo del reporte de la pantalla que lista la recurrencia de recibos, emplee las teclas de movimiento del cursor, el tabulador, las teclas de avance de página y retroceso de página y la tecla < ENTER >. Para salir presione la tecla < ESC >.

OBSERVACIÓN : el conjunto de teclas válidas para la lista de recurrencia de recibos, son las mismas que se deben de emplear en todos los reportes que el programa genere.

Al presionar < ESC > el programa envía un aviso indicándole al usuario si desea continuar con el alta del registro (ver figura 5.23).

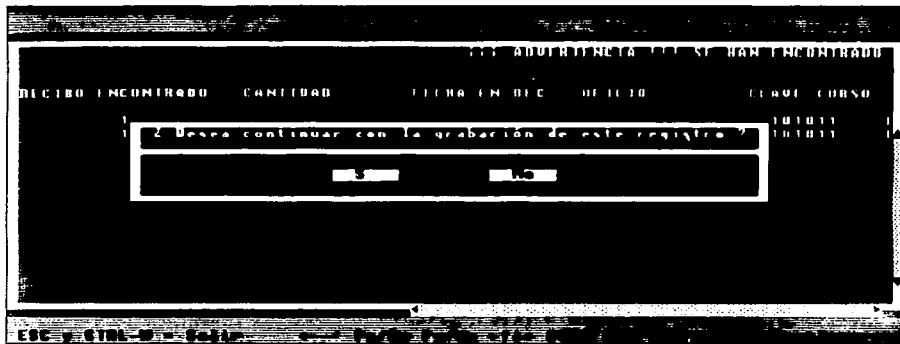


Figura 5.23. La pantalla de continuación con la grabación de un registro.

De aceptar, el alta queda confirmada y el nuevo registro se grabará, de lo contrario el programa permitirá volver a editar los datos del registro capturado.

LA PANTALLA DE ALTAS DE EXPOSITORES

Al seleccionar la opción "expositores" del menú de altas, aparecerá la pantalla de altas de expositores, la cual se muestra en la figura 5.24.

Al igual que la pantalla de altas de cursos, esta pantalla se divide en tres áreas : a) zona de captura de datos ; b) zona de memorándum y zona de confirmación. De antemano su uso de zonas y conjunto de teclas válidas son idénticas a la de cursos, lo único que cambia es que una pantalla es para cursos y la otra para expositores.

Para mayor referencia consulte la pantalla de altas de cursos.

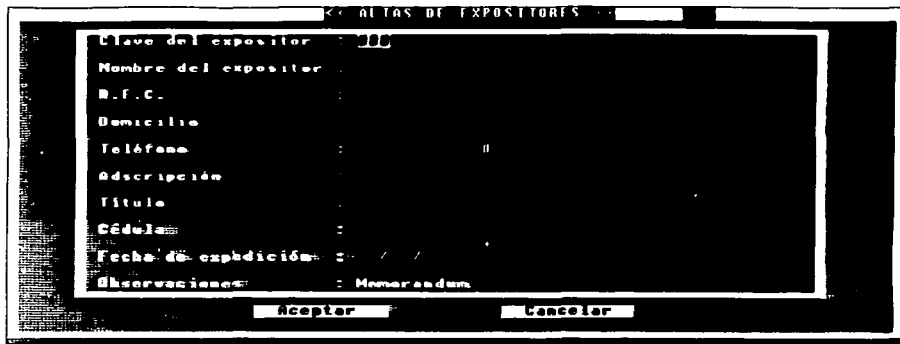


Figura 5.24. La pantalla de altas de expositores.

LA PANTALLA DE ALTAS DE CARRERAS O PROFESIONES

Al seleccionar la opción "profesiones" del menú de altas, aparecerá la pantalla de altas de carreras o profesiones, la cual se muestra en la figura 5.25.

Al igual que la pantalla de altas de cursos, esta pantalla se divide en tres áreas : a) zona de captura de datos ; b) zona de memorándum y zona de confirmación. De antemano su uso de zonas y conjunto de teclas válidas son idénticas a la de cursos, lo único que cambia es que una pantalla es para cursos y la otra es para carreras.

Para mayor referencia consulte la pantalla de altas de cursos.

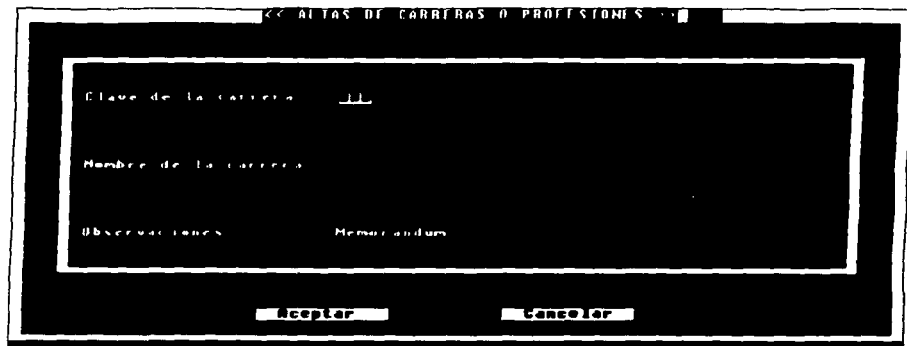


Figura 5.25. La pantalla de altas de carreras o profesiones.

EL MENÚ GENERAL DE BAJAS

Al seleccionar la opción de "bajas" del menú principal, aparecerá la pantalla del menú general de bajas la cual se muestra en la figura 5.26.

Como se puede observar este menú presenta las mismas opciones que el menú general de altas, con la diferencia de que el primero es para eliminar registros y el segundo es para agregar registros.

Las cuatro primeras opciones realizan respectivamente las bajas de cursos, ordenes de pago, expositores de cursos y carreras disponibles de la institución. La quinta opción permite regresar a la pantalla anterior que es la del menú principal.

Las teclas válidas de este menú, son las mismas que las empleadas en el menú principal debido a que tienen el mismo formato (para mayor referencia, consulte el apartado sobre el menú principal.).

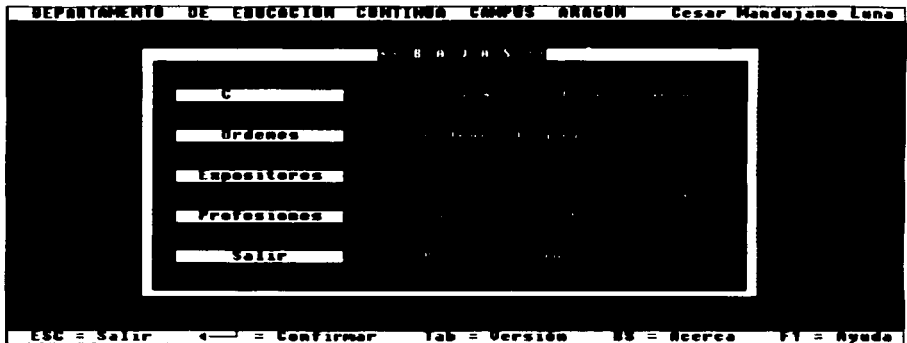


Figura 5.26. La pantalla del menú general de bajas.

LA PANTALLA DE BAJAS DE CURSOS

Al seleccionar la opción "cursos" del menú de bajas, aparecerá la pantalla de bajas de cursos, la cual se muestra en la figura 5.27.

Como se puede apreciar, esta pantalla solicita el número de clave del registro que el usuario desea dar de baja y debido a que esta clave se compone únicamente de dígitos, cualquier caracter diferente no será aceptado.

Una ventaja muy importante que presenta el programa, es que el usuario no necesita conocer ni teclear el total de dígitos que componen a una clave. Esto es, si el usuario teclea sólo una parte de la clave, el programa listará aquellos registros que cumplan con la parte de la clave tecleada. Por ejemplo, si se teclea los dígitos "10" y se presiona < ENTER >, el programa buscará aquellos registros cuya clave comience con esos dos dígitos (ver figuras 5.27 y 5.28).

<< BAJAS DE CURSOS >>

```

Clave curso
Clases          $
Académicos     $
Administrativas $
Mantenimiento  $
Egresados      $
Público general $
Nombre curso
Período
Lugar
Duración
Horario
Clave expositor
Cupo
Salón
Clave carrera
Días
Observaciones
Memorandum
  
```

Lea la clave del curso

10

[] Lu [] Ma [] Mi [] Ju [] Vi [] Sa [] Do

Aceptar Cancelar

Figura 5.27. La pantalla de solicitud de clave para bajas de cursos.

<< BAJAS DE CURSOS >>

```

Clave curso
Clave del curso  Nombre del curso
1010101011      INGLÉS INTERMEDIUM COMPRENSION MODULO 2
1010101012      INGLÉS INTERMEDIUM COMPRENSION MODULO 1
  
```

ESC Salir Confirmar PgDn, PgUp, etc Mover

Aceptar Cancelar

Figura 5.28. Ejemplo de la pantalla del listado de cursos que comienzan con sólo una parte de la clave.

En caso de que el programa no encuentre ningún registro, enviará un mensaje de error por clave inexistente; en el caso de que encuentre un sólo registro desplegará inmediatamente sus datos en la pantalla (ver figura 5.29) y en caso de encontrar más de un registro, desplegará la lista de los mismos, permitiendo al usuario visualizar el contenido de los campos de cada uno de los registros y seleccionar el requerido.

Como se puede observar en la figura 5.28, puede emplear las teclas de movimiento del cursor y de avance de página para hojear el contenido de los campos, < ENTER > para confirmar y < ESC > para cerrar esta pantalla.

<< BAJAS DE CURSOS >>

Clave curso	1010101011
Alumnos	S 500 00
Academicos	S 500 00
Administrativos	S 500 00
Maestros/asistentes	S 500 00
Ingresados	S 500 00
Público general	S 500 00
Nombre curso	ENGLISH INTENSIVO COMPRENSION MODULO I
Periodo	11-09-1996 11-11-1996
Lugar	IMP BARRION
Duracion	26 HRS
Horario	13 00 13 00 HRS
Clave expositor	001
Cupo	??
Salon	8609
Clave carrera	017
Bloq	[S] Tu [S] Mi [S] Ma [N] Ju [N] Vi [N] Sa [N] Do
Observaciones	

Aceptar Cancelar

Figura 5.29. La pantalla de bajas de cursos con los datos de un curso seleccionado.

OBSERVACIÓN : en los módulos de bajas, modificaciones y consultas por clave de cualquiera de las bases de datos puede teclear sólo una parte de la clave y el programa buscará aquellos registros que comiencen con la parte tecleada.

De la figura 5.29 se desprende que esta pantalla de bajas de cursos, también se divide en tres zonas como la de altas de cursos que son la zona de captura; la zona de memorándum, y la zona de confirmación. Solamente que la zona de captura queda restringida, o sea, no se puede ingresar a ella.

OBSERVACIÓN : La zona de captura de las cuatro bases de datos siempre queda restringida cuando se realizan bajas o consultas.

OBSERVACIÓN: Nunca se puede dar de baja un curso que tenga ya asociadas ordenes de pago, ya que esto producirá un error.

LA PANTALLA DE BAJAS DE ORDENES DE PAGO.

Al seleccionar la opción "ordenes" del menú de bajas, aparecerá la pantalla de bajas de ordenes de pago, la cual se muestra en la figura 5.30.

BAJAS DE ORDENES DE ORDENES DE PAGO

Clave del curso
Nombre del alumno
BIC
Telefono
Clave procedencia
Clave carrera
Numero de cuenta
Condición
Semestre
Periodo
Expositor
Saldo
Costo del curso
Numero de recibe
Observaciones

Recibos
Memorandum

Leer la clave del curso

Aceptar Cancelar

Figura 5.30. La pantalla de solicitud de clave para bajas de ordenes de pago.

Esta pantalla opera de manera similar que la pantalla de bajas de cursos, ya que también solicita la clave del curso del registro que se desea dar de baja.

Al igual que la pantalla de altas de ordenes de pago, se compone de cinco zonas las cuales son : zona de captura; zona de costo del curso; zona de recibos; zona de memorándum, y zona de confirmación. Solamente que las zonas de captura y de costo del curso quedan restringidas para su acceso.

Para mayor referencia consulte el apartado sobre la pantalla de altas de ordenes de pago.

LA PANTALLA DE BAJAS DE EXPOSITORES

Al seleccionar la opción "expositores" del menú de bajas, aparecerá la pantalla de bajas de expositores, la cual se muestra en la figura 5.31.

Su uso es similar a la pantalla de bajas de cursos, la única diferencia es la longitud de la clave (para mayor referencia consulte el apartado de bajas de cursos.).

<< BAJAS DE EXPOSITORES

Clave del expositor

Nombre del expositor

B I C

Banco

Telefono

Adscripcion

Titulo

Cedula

Fecha de expedicion

Memorandum

Aceptar Cancelar

Figura 5.31. La pantalla de solicitud de clave para bajas de expositores.

OBSERVACIÓN. Nunca se puede dar de baja a un expositor que ya este asociado a un curso, ya que esto ocasionará un error.

LA PANTALLA DE BAJAS DE CARRERAS O PROFESIONES.

Al seleccionar la opción "profesiones" del menú de bajas, aparecerá la pantalla de bajas de carreras o profesiones, la cual se muestra en la figura 5.32.

Su uso es similar a la pantalla de bajas de cursos, la única diferencia es la longitud de la clave (para mayor referencia consulte el apartado de bajas de cursos.).

<< BAJAS DE CARRERAS O PROFESIONES

Clave de la carrera

Nombre de la carrera

Observaciones

Memorandum

Aceptar

Cancelar

Figura 5.32. La pantalla de solicitud de clave para bajas de carreras o profesiones.

OBSERVACIÓN. Nunca se puede dar de baja una carrera que ya este asociada a un curso, ya que esto ocasionará un error.

EL MENÚ GENERAL DE CONSULTAS

Al seleccionar la opción de "consultas" del menú principal, aparecerá la pantalla del menú general de consultas la cual se muestra en la figura 5.33.

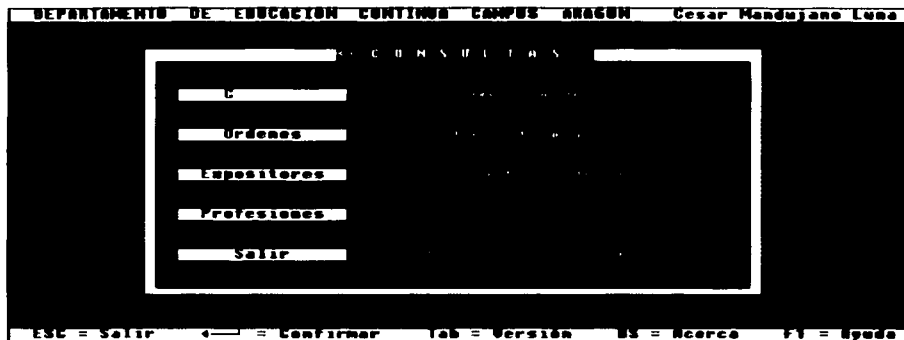


Figura 5.33. La pantalla del menú general de consultas.

Como se puede observar este menú presenta las mismas opciones que el menú general de altas, con la diferencia de que el primero es para consultar registros y el segundo es para agregar registros.

Las cuatro primeras opciones realizan respectivamente las consultas de cursos, ordenes de pago, expositores de cursos y carreras disponibles de la institución. La quinta opción permite regresar a la pantalla anterior que es la del menú principal.

Las teclas válidas de este menú, son las mismas que las empleadas en el menú principal debido a que tienen el mismo formato.

EL MENÚ DE TIPO DE CONSULTAS

Al presionar < ENTER > en cualquiera de las cuatro primeras opciones del menú general de consultas, aparecerá el menú de tipo de consultas, el cual se muestra en la figura 5.34.

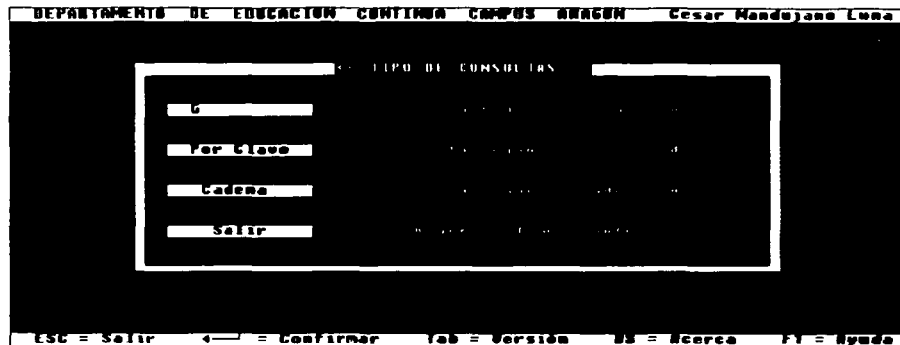


Figura 5.34. La pantalla del menú de tipo de consultas.

El menú general del tipo de consultas permite que el usuario seleccione el tipo de consulta que desea realizar. Así, la primera opción llamada "globales", exhibe el total de registros de la base de datos seleccionada, permite al usuario hojear los campos de los registros y seleccionar un registro en particular para exhibir su contenido; la segunda opción llamada "por clave", solicita el campo clave de la base de datos seleccionada, exhibe aquellos registros que tengan ya sea parcial o exactamente igual dicha clave, permite al usuario hojear los campos de los registros y seleccionar un registro en particular para exhibir su contenido; la tercera opción llamada "cadena", presenta un menú que tiene como opciones aquellos campos alfanuméricos de la base de datos seleccionada, permite al usuario escoger uno de los campos de la lista, solicita la cadena de caracteres que se desea buscar en el campo seleccionado de toda la base de datos en uso, lista aquellos registros en donde encontró la cadena incluida, permite al usuario hojear los campos de los registros y seleccionar un registro para exhibir su contenido.

Debido a que la forma de trabajar el menú general del tipo de consultas es prácticamente el mismo en las cuatro base de datos, solamente se explicará la base de datos de cursos y se mostrará los menús de las bases de datos para consultas por cadena incluida.

LA PANTALLA DE CONSULTAS DE CURSOS GLOBALES

Al seleccionar la opción "globales" del menú general de consultas de cursos globales aparecerá la pantalla que se muestra en la figura 5.35.

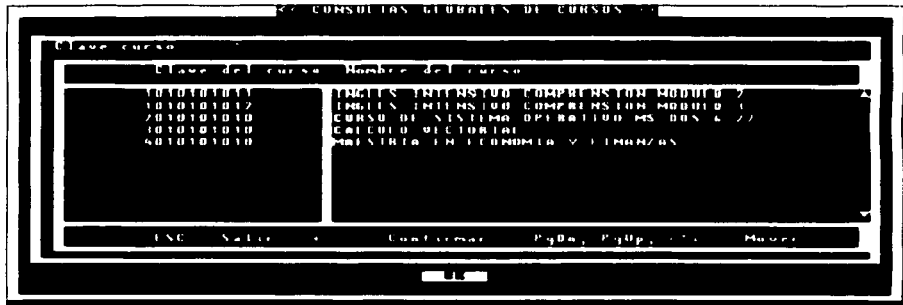


Figura 5.35. Ejemplo de la pantalla de consultas de cursos globales.

En esta pantalla se lista automáticamente el total de cursos existentes en la base de datos de cursos, permite al usuario hojear los campos de los registros, seleccionar un registro y exhibir sus datos en pantalla (ver figura 5.36.). Es importante mencionar que no existe acceso a ninguna de las zonas de la pantalla para modificación de contenidos.

Para continuar con otra consulta presiones < ENTER > en el botón de >> Ok << o bien presione < ESC >.

OBSERVACIÓN: No se puede modificar el contenido de los campos de cualquiera de las bases de datos que se en uso cuando uno se encuentra en el módulo de consultas.

CONSULTAS GLOBALES DE CURSOS	
Clave curso	00000000
Alumnos	5 500 00
Académicos	5 500 00
Administrativos	5 500 00
Mantenimiento	5 500 00
Ingresados	5 500 00
Público general	5 500 00
Nombres curso	ENGLISH INTERMEDIUM COMPRENSION MODULO 1
Periodo	11-09-1996 11-11-1996
Lugar	IMP 8060M
Duración	16 HRs
Horario	11:00 12:00 HRs
Clave expositor	001
Cupo	??
Salón	8609
Clave carrera	012
Días	[S] Tu [S] Ma [S] Mi [M] Ju [M] Vi [M] Sa [M] Do
Observaciones	

Figura 5.36. Ejemplo de la exhibición de datos de un curso seleccionado en consultas de cursos globales.

LA PANTALLA DE CONSULTAS DE CURSOS POR CLAVE

Al seleccionar la opción "por clave" del menú general de consultas de cursos por clave aparecerá la pantalla que se muestra en la figura 5.37.

Aquí el usuario debe proporcionar la clave del curso que desea consultar. Puede incluso teclear ya sea la clave completa o bien sólo una parte de ella, en cuyo caso el programa listará aquellos registros que comiencen con la clave parcial tecleada.

Al seleccionar el registro deseado, el programa exhibirá su contenido.

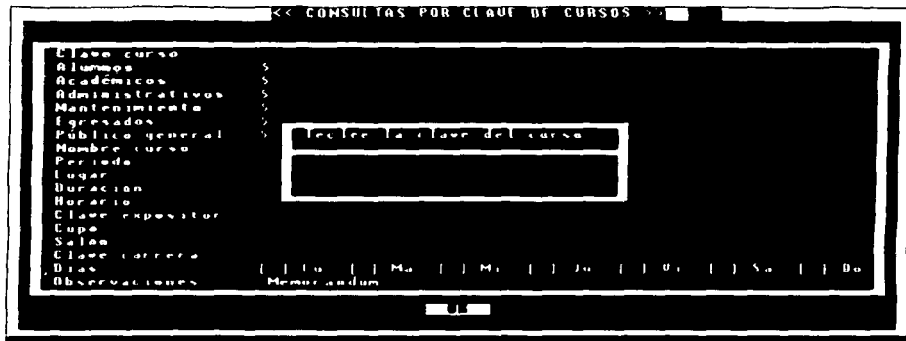


Figura 5.37. Ejemplo de la pantalla de consultas por clave de cursos.

EL MENÚ DE CONSULTAS DE CURSOS POR CADENA INCLUIDA

Al seleccionar la opción "cadena" del menú general de consultas de cursos por cadena incluida, aparecerá la pantalla que se muestra en la figura 5.38.

Esta pantalla se compone de una lista de campos alfanuméricos, que el usuario puede seleccionar, para buscar aquellos registro que tengan incluida una cadena de caracteres previamente establecida por el usuario, en la base de datos de cursos.

Por ejemplo, si se desea buscar aquellos registros de la base de datos de cursos, que tengan en su campo nombre del curso la cadena "COMPRENSION", se tiene que seleccionar la opción "nombre del curso" y aparecerá la pantalla que solicita la cadena incluida, la cual se muestra en la figura 5.39. Posteriormente se teclaea la cadena que se desea encontrar en dicho campo, en este caso se debe teclear la cadena "COMPRENSION" y presionar < ENTER >.

En caso de encontrar varios registros que tengan esta cadena incluida, el programa mostrará la lista de los mismos (ver figura 5.40), permitiendo hojear, seleccionar y exhibir el registro requerido.

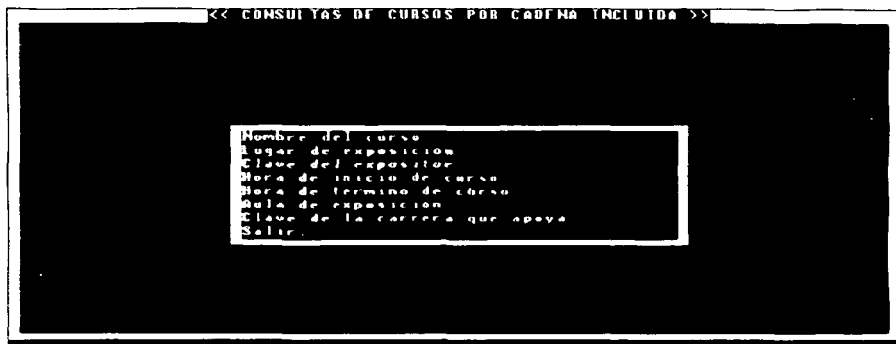


Figura 5.38. La pantalla del menú de consultas de cursos por cadena incluida.

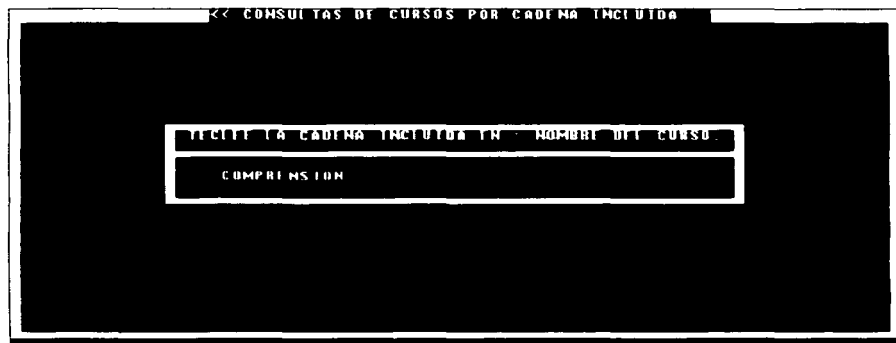


Figura 5.39. Ejemplo de la pantalla de solicitud de búsqueda de cadena.

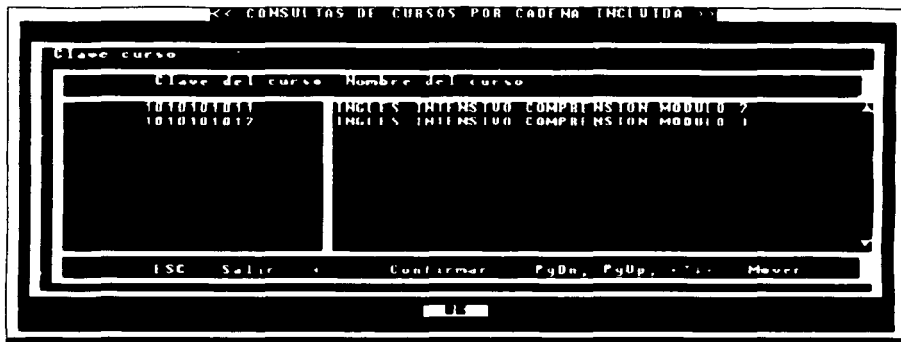


Figura 5.40. Ejemplo del listado de aquellos registros que cumplan con la palabra tecleada por el usuario.

Si encuentra sólo un registro exhibirá inmediatamente su contenido en la pantalla.

Finalmente si no encuentra ninguno emitirá un mensaje de error.

OBSERVACIÓN : Esta propiedad que maneja el programa de cadena incluida, es una herramienta muy valiosa, ya que permite realizar búsquedas y clasificaciones de la base de datos en uso según el tópico de interés del usuario.

OBSERVACIÓN. Los demás menús de cadena incluida que son los de ordenes de pago, expositores y profesiones tienen igual forma de operar que el de cursos, la única diferencia son los campos empleados para realizar las búsquedas.

LOS MENÚS DE CONSULTAS POR CADENA INCLUIDA DE ORDENES DE PAGO, EXPOSITORES Y PROFESIONES

El resto de las pantallas por cadena incluida se muestran en las figuras 5.41 a la 5.43.

Estas pantallas tienen igual forma de operar que la respectiva de cursos, por lo que para mayor referencia consulte esta sección.

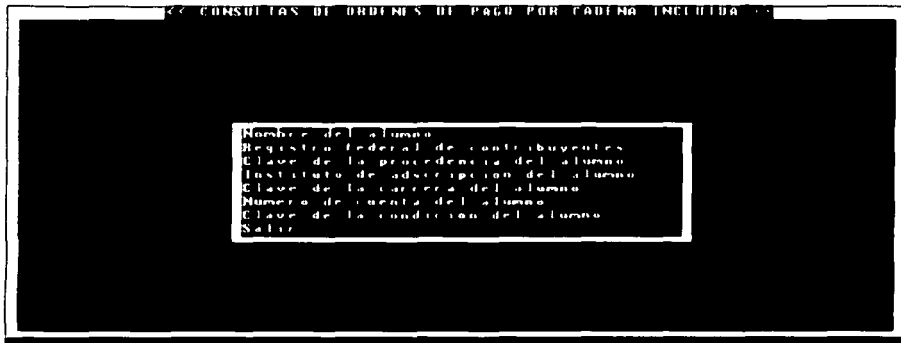


Figura 5.41. La pantalla del menú de consultas de ordenes de pago por cadena incluida.

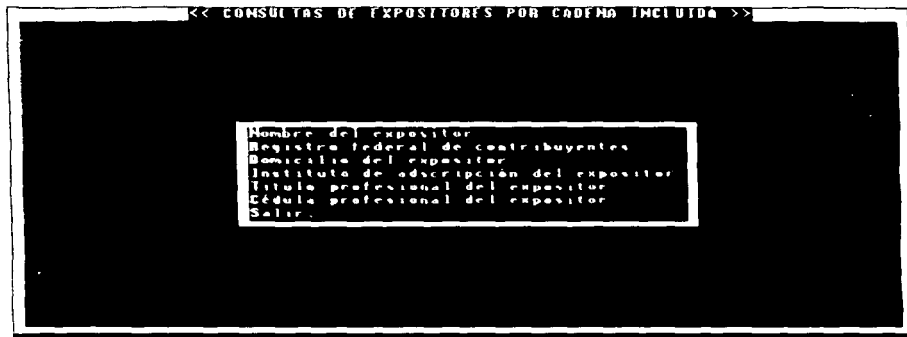


Figura 5.42. La pantalla del menú de consultas de expositores por cadena incluida.

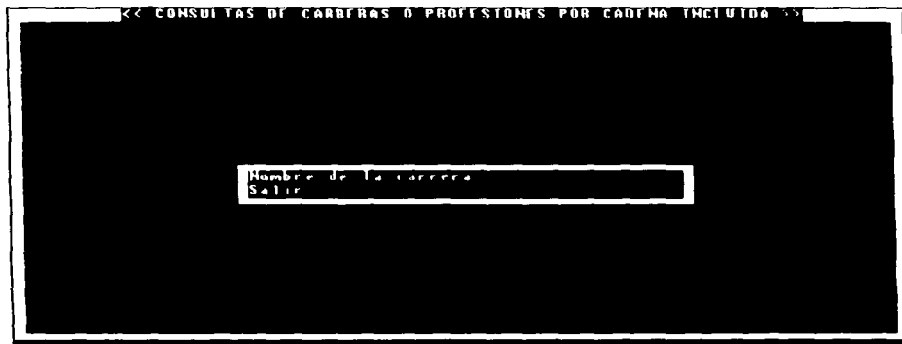


Figura 5.43. La pantalla del menú de consultas de carreras o profesiones por cadena incluida.

EL MENÚ GENERAL DE MODIFICACIONES.

Al seleccionar la opción de modificaciones del menú principal, aparecerá la pantalla del menú general de modificaciones el cual se muestra en la figura 5.44.

Como se puede observar este menú presenta las mismas opciones que el menú general de altas, con la diferencia de que este es para cambiar los datos de los registros y el otro es para agregarlos.

Las cuatro primeras opciones realizan respectivamente las modificaciones de cursos, ordenes de pago, expositores de cursos y carreras disponibles de la institución. La quinta opción permite regresar a la pantalla anterior que es la del menú principal.

Las teclas válidas de este menú, son las mismas que las empleadas en el menú principal debido a que tienen el mismo formato (para mayor referencia, consulte el manejo del menú principal.).

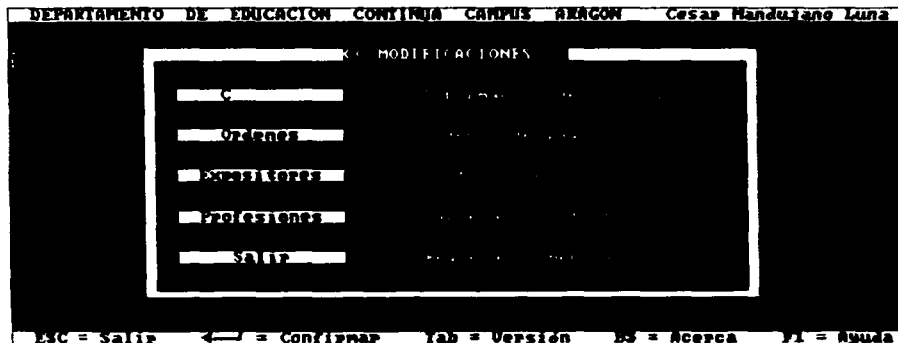


Figura 5.44. La pantalla del menú general de modificaciones.

LA PANTALLA DE MODIFICACIONES DE CURSOS.

Al seleccionar la opción "cursos" del menú de modificaciones, aparecerá la pantalla de modificaciones de cursos, la cual se muestra en la figura 5.45.

<< MODIFICACIONES DE CURSOS >>

Clave curso	
Clasificación	5
Académicas	5
Administrativas	5
Mantenimiento	5
Egresados	5
Público general	5
Número curso	
Periodo	
Lugar	
Duración	
Horario	
Clave expositor	
Cupo	
Salen	
Clave carrera	
Días	
Observaciones	

Leer la clave del curso

10

Memorandum

Aceptar Cancelar

Figura 5.45. La pantalla de modificaciones de cursos.

Esta pantalla es idéntica a la de altas de cursos, en cuanto a uso de zonas y teclas válidas, sólo que la de modificaciones solicita primero al usuario que teclee el número de clave del registro que se desea cambiar algunos de sus datos.

OBSERVACIÓN. Nunca puede modificar la clave de un curso que ya tenga asociadas ordenes de pago, ya que esto ocasionará un error.

LA PANTALLA DE MODIFICACIONES DE ORDENES DE PAGO.

Al seleccionar la opción "ordenes" del menú de modificaciones, aparecerá la pantalla de modificaciones de ordenes de pago, la cual se muestra en la figura 5.46.

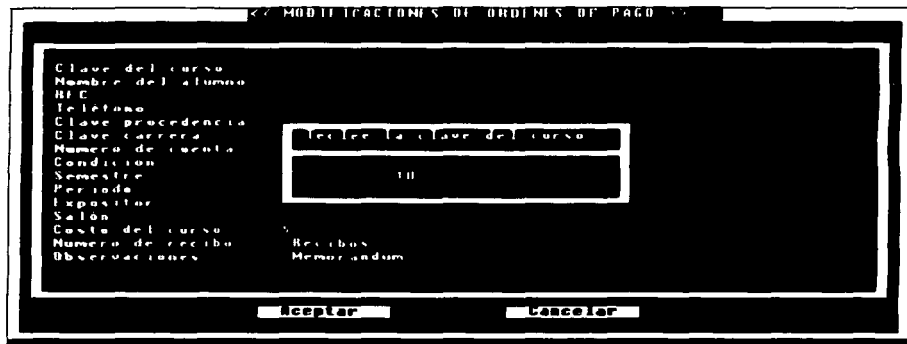


Figura 5.46. La pantalla de modificaciones de ordenes de pago.

Esta pantalla es idéntica a la de altas de ordenes de pago, en cuanto a uso de zonas y teclas válidas, sólo que la de modificaciones solicita primero al usuario que teclee el número de clave del registro que se desea cambiar algunos de sus datos.

LA PANTALLA DE MODIFICACIONES DE EXPOSITORES.

Al seleccionar la opción "expositores" del menú de modificaciones, aparecerá la pantalla de modificaciones de expositores, la cual se muestra en la figura 5.47.

<< MODIFICACIONES DE EXPOSITORES >>

Clave del expositor

Nombre del expositor

B I C

Domicilio

telefono

Descripción

Título

Cedula

Fecha de expedición

Observaciones

Memorandum

Aceptar Cancelar

Figura 5.47. La pantalla de modificaciones de expositores.

Esta pantalla es idéntica a la de altas de expositores, en cuanto a uso de zonas y teclas válidas, sólo que la de modificaciones solicita primero al usuario que teclee el número de clave del registro del cual se desea cambiar algunos de sus datos.

OBSERVACIÓN. Nunca puede modificar la clave de un expositor que ya tenga asociado un curso, ya que esto ocasionará un error.

LA PANTALLA DE MODIFICACIONES DE CARRERAS O PROFESIONES.

Al seleccionar la opción "profesiones" del menú de modificaciones, aparecerá la pantalla de modificaciones de carreras o profesiones, la cual se muestra en la figura 5.48.

Esta pantalla es idéntica a la de altas de profesiones o carreras, en cuanto a uso de zonas y teclas válidas, sólo que la de modificaciones solicita primero al usuario que teclee el número de clave del registro que se desea cambiar algunos de sus datos.

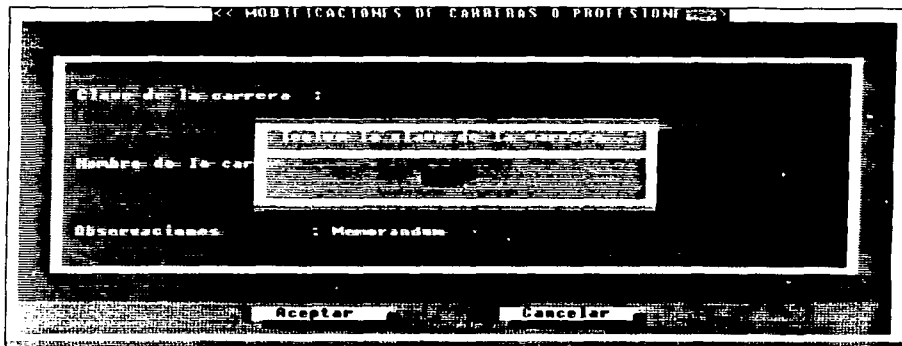


Figura 5.48. La pantalla de modificaciones de carreras profesiones.

OBSERVACIÓN. Nunca puede modificar la clave de una carrera que ya tenga asociados cursos, ya que esto ocasionará un error.

EL MENÚ GENERAL DE REPORTES.

Al seleccionar la opción de "reportes" del menú principal, aparecerá el menú de reportes el cual se muestra en la figura 5.49.

Como se puede observar este menú está compuesto por siete diversas opciones, las cuales se detallan a continuación.

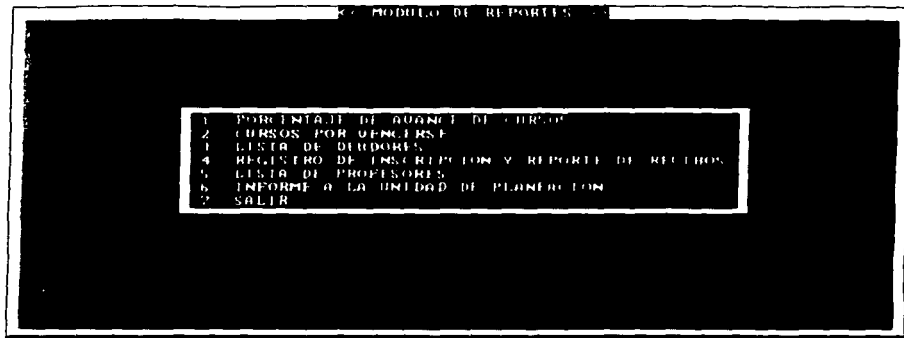


Figura 5.49. La pantalla del menú general de reportes.

LA PANTALLA DE AVANCE DE CURSOS.

Al seleccionar la opción de avance de cursos del menú de reportes, aparecerá la pantalla de avance de cursos, la cual se muestra en la figura 5.50.

Esta opción tiene como función, indicar tanto numérica como gráficamente, el porcentaje de avance que tiene un curso, previamente indicado por el usuario.

También proporciona información general del curso, el total de días que se han expuesto y el total de días que faltan por exponer.

OBSERVACIÓN. El programa realiza los cálculos del total de días de un curso, los días expuestos, los días por exponer y el porcentaje de avance de un curso, tomando en cuenta la fecha que posee actualmente el sistema. Así que, es muy importante verificar que la fecha del mismo sea la correcta, para que el programa arroje resultados reales.

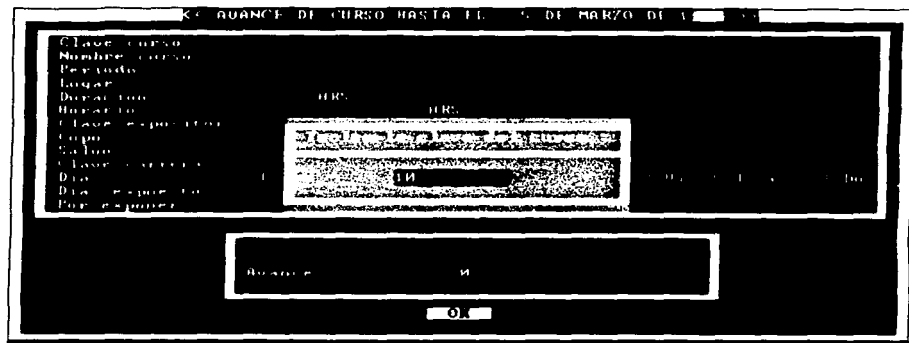


Figura 5.50. La pantalla de avance de cursos.

Como se puede observar en la figura 5.50, aparece una pantalla que solicita se digite el curso del cual se desea obtener un avance.

Al igual que los menús de bajas, modificaciones y consultas, se puede teclear sólo una parte de la clave y si el programa encuentra más de un registro que comience con la parte de la clave tecleada, proporcionará la lista de dichos registros, permitiendo al usuario hojear los campos y seleccionar el registro deseado. Por ejemplo, se requiere mostrar el avance de aquellos cursos que comiencen con "10". Para ello, tecleamos "10" en la pantalla de la figura 5.50 y en caso de encontrar más de dos registros cuya clave comience con esos dos dígitos aparecerá una figura similar a la 5.51.

Posteriormente se puede seleccionar el registro de interés, y toda la información referente al mismo aparecerá, al igual que los días expuestos, los días que faltan por exponer y su porcentaje real de avance (ver figura 5.52 .).

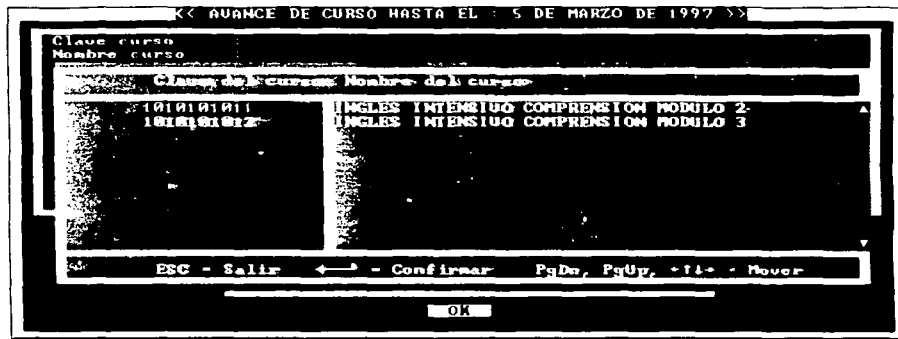


Figura 5.51. Ejemplo de la pantalla del listado de cursos que comienzan con sólo una parte de la clave.

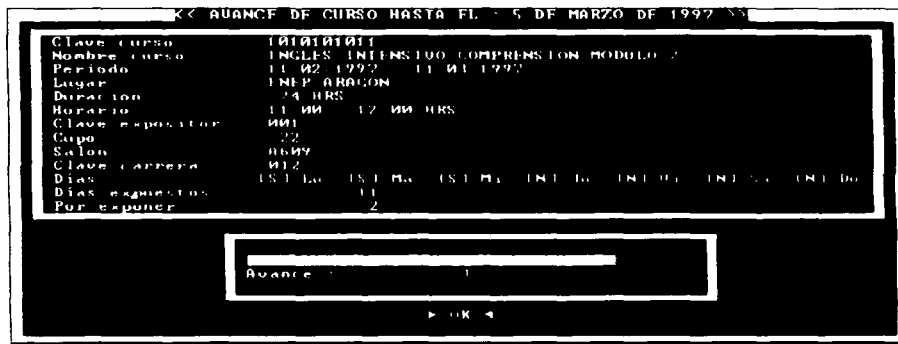


Figura 5.52. Ejemplo de la información que proporciona el reporte de avance de cursos.

Es muy importante tomar en cuenta que el programa considera que si un curso todavía no empieza, este tiene porcentaje de avance del 0% y los días que faltan por exponer son todos los considerados en el calendario de cursos, y que si un curso ya concluyó tiene un avance del 100% y los días que faltan por exponer son cero.

Para concluir con la consulta del curso por vencer presione la tecla < ESC >, < ENTER > o la letra "O". Hecho esto, el programa solicitará si desea verificar otro avance de curso (ver figura 5.53). En caso afirmativo el programa nuevamente solicitará se introduzca la clave del curso, de lo contrario el programa devolverá el control al menú de reportes.

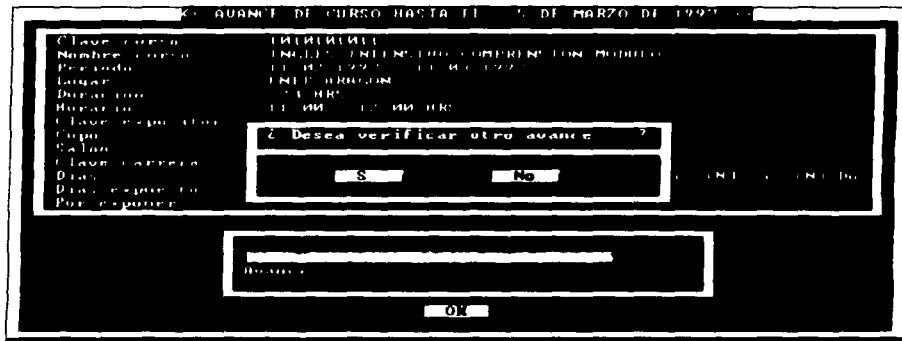


Figura 5.53. La pantalla de solicitud de verificación de avance.

En caso de que el usuario proporcione una clave de curso inexistente, o bien que el curso si exista pero este no tiene asignado ningún alumno, se emitirá un mensaje de error como el que se muestra en la figura 5.54.

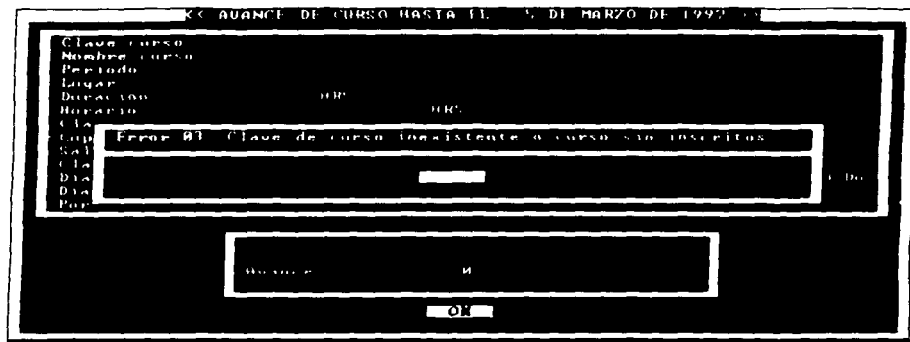


Figura 5.54. Ejemplo de la pantalla de mensaje de error.

LA PANTALLA DE CURSOS POR VENCER.

Al seleccionar la opción de "cursos por vencer" del menú de reportes, aparecerá la pantalla de cursos por vencer, la cual se muestra en la figura 5.55.

Esta opción tiene como función, proporcionar ya sea en pantalla, en impresora y/o en un archivo un reporte que muestre la lista de cursos por vencer hasta el número de días que el usuario le indique al programa.

Como se aprecia en la figura 5.55, aparece una pantalla que solicita al usuario el número de días de anticipación para listar aquellos cursos cuya vigencia estén a partir de la fecha del sistema hasta los días de anticipación.

De existir cursos por vencer en el rango indicado, el programa genera el listado completo de esos cursos en pantalla (ver figura 5.56). En caso de no existir cursos, el programa notificará esta situación.

Para cerrar este reporte se debe presionar la tecla < ESC >, a lo cual el programa responderá con un aviso indicándole al usuario si es su deseo guardar el reporte generado en un archivo. Esta situación se ilustra en la figura 5.57.

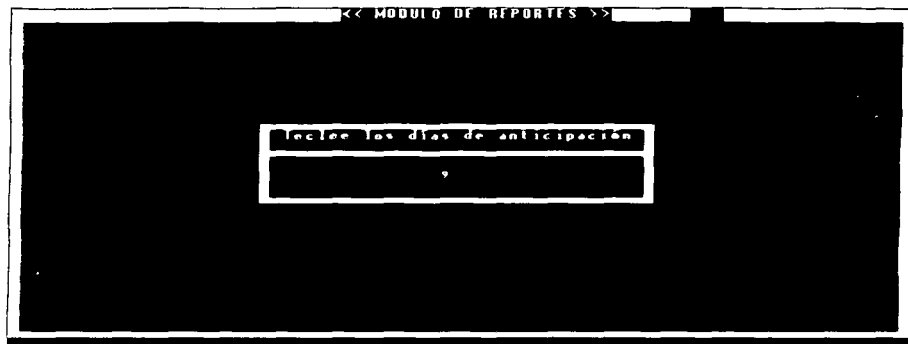


Figura 5.55. La pantalla de cursos por vencer.

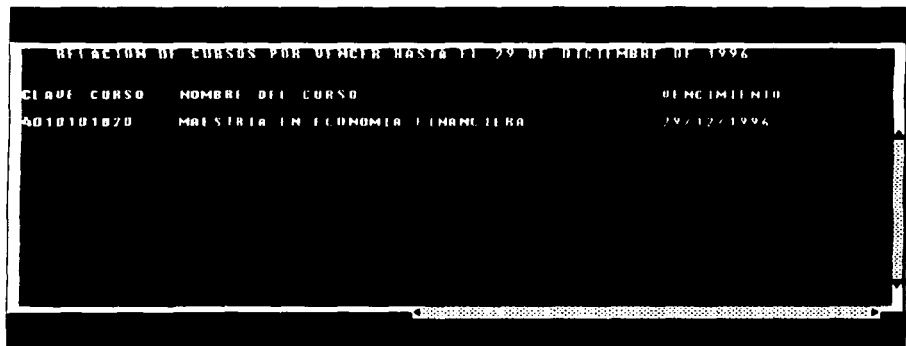


Figura 5.56. Ejemplo de la salida del reporte de cursos por vencer.

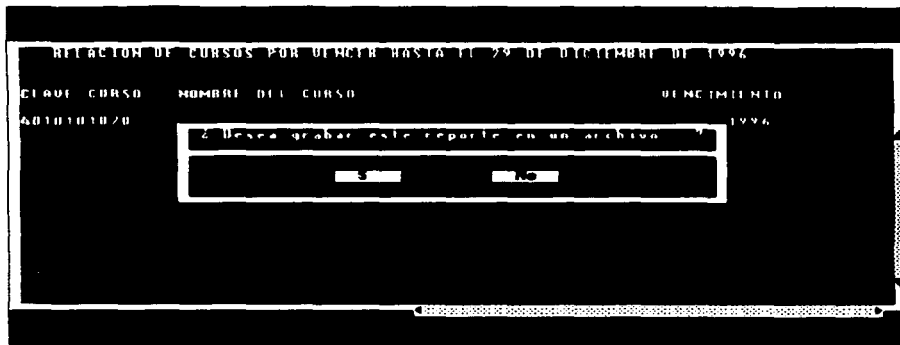


Figura 5.57. La pantalla de solicitud de grabación de un reporte en un archivo.

En caso afirmativo el programa solicitará se introduzca el nombre del archivo como se puede apreciar en la figura 5.58. Este archivo lleva siempre la extensión ".CPV".

En el supuesto caso, de que el nombre del archivo proporcionado por el usuario ya exista, el programa solicitará si desea o no reescribir el archivo (ver figura 5.59).

Este archivo generado tiene una agradable propiedad : esta escrito en código ASCII¹³, por lo que es transportable a la gran mayor parte de procesadores de textos, desde los cuales el usuario puede modificar el aspecto del reporte, dándole su sello personal. Esto es en general válido para cualquier reporte generado y almacenado por el programa en un archivo.

Al cerrar la pantalla que solicita si se desea escribir el reporte en un archivo, el programa envía un aviso indicándole al usuario si desea imprimir el reporte generado (ver figura 5.60).

¹³ ASCII son las siglas de AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE, esto es CÓDIGO ESTÁNDAR AMERICANO PARA INTERCAMBIO DE INFORMACIÓN, es el código normalizado que rige el conjunto de caracteres existentes en una computadora personal PC, asociando para ello con cada número del 0 al 255 un símbolo o caracter de control.

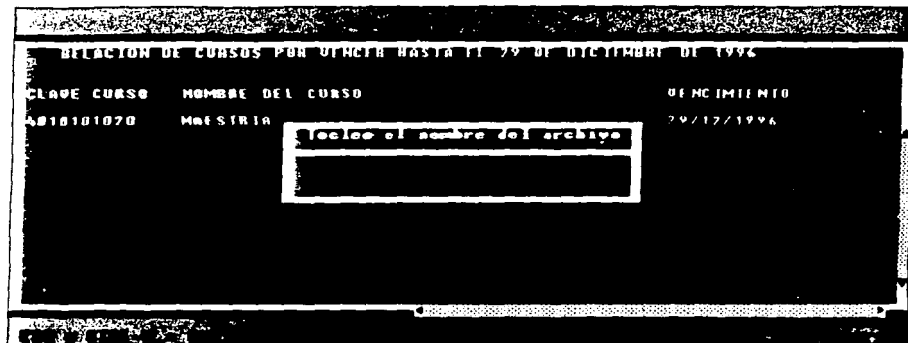


Figura 5.58. La pantalla de solicitud del nombre del archivo

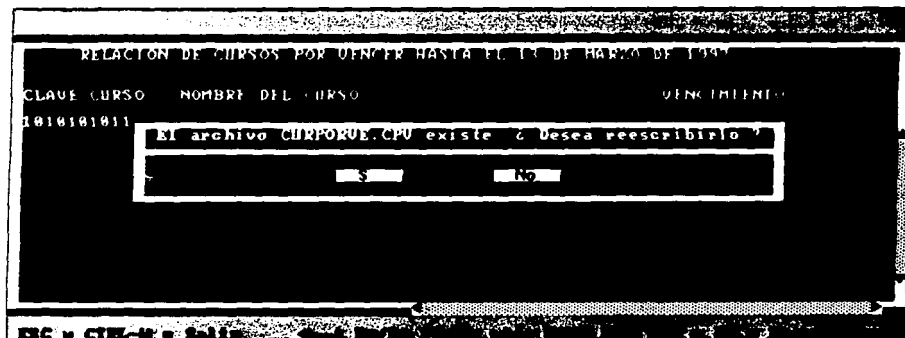


Figura 5.59. La pantalla de solicitud de rescritura de un archivo.

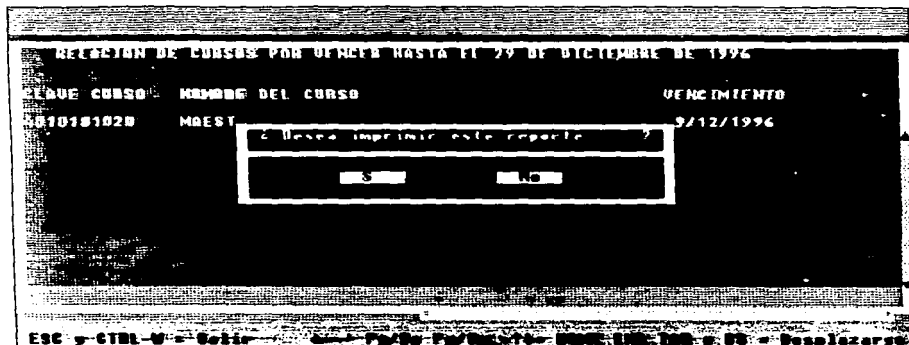


Figura 5.60. La pantalla de solicitud de impresión de un reporte.

OBSERVACIÓN. Los reportes impresos desde el programa DEC son muy sencillos y no gozan de una gran presentación, por lo que se sugiere utilizarlos sólo desde un punto de vista informativo e ilustrativo. Si se desea mayor presentación, grave el reporte en un archivo y este último llámelo desde el procesador de textos de preferencia y modifique su aspecto.

OBSERVACIÓN. Si se imprime desde el programa DEC un reporte, es importante que se haya seleccionado previamente el tipo de impresora que se tiene. Para ello, es necesario entrar al menú de "disposición", escoger la opción de "impresora" y seleccionar la adecuada.

OBSERVACIÓN. Todos los reportes generados almacenados en archivos tienen como separador de columnas o conceptos el símbolo "\".

OBSERVACIÓN. Si la mayor parte de los procesadores de texto preferidos por el usuario corren en ambiente WINDOWS, no es necesario que se salga del programa DEC para ejecutar su procesador de textos. Para ello es conveniente ejecutar el programa DEC desde WINDOWS, generar los reportes que se requieren, iconizar el programa DEC, ejecutar el procesador de textos de preferencia, cargar el archivo del reporte generado y mandar a imprimir, para posteriormente cerrar la aplicación del procesador de textos y expandir el icono del programa DEC y continuar sin necesidad de volver a cargar el programa.

OBSERVACIÓN. Para todos los reportes que se puedan generar en pantalla, en impresora y en un archivo, la forma de proceder es la misma que para este reporte.

Finalmente, el programa cuestiona al usuario si desea continuar con otra consulta de cursos por vencer como se ve en la figura 5.61. En caso afirmativo, el programa vuelve a solicitar el número de días de anticipación para otro reporte de cursos por vencer, de lo contrario, el programa devuelve el control al menú general de reportes.

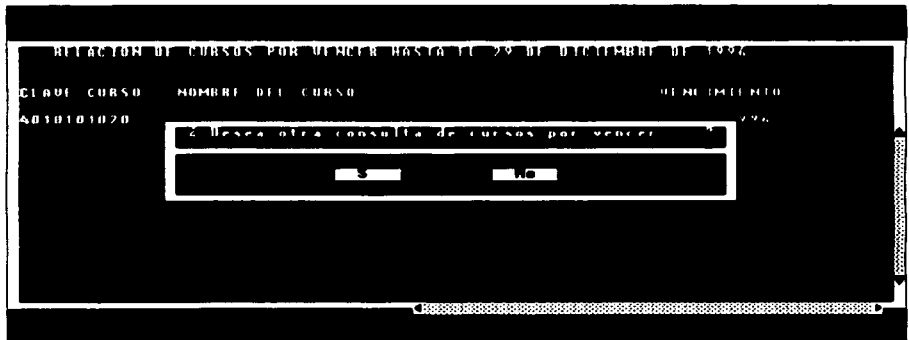


Figura 5.61. La pantalla de solicitud de otra consulta de cursos por vencer.

LA PANTALLA DE LISTA DE DEUDORES.

Al seleccionar la opción de "listas de deudores" del menú de reportes, aparecerá la pantalla de lista de deudores, la cual se muestra en la figura 5.62.

NOMBRE DEL DEUDOR	TELÉFONO	DUDA	
DAVID TORRES SANCHEZ /	7896545		499 00
GERARDO RAMIREZ / URIBE /	5367541		497 00
JAIMÉ RAMÍREZ / PÉREZ /	777619		500 00
JOHÁN BAPTISTA / JIMENEZ /		0	489 00
JESÚS MARTÍNEZ / ARANDA /	1252887		489 00
MAURICIO LUNA PÉREZ /	7936715		499 00
RENÉ ANGLIANO AGUILAR /		0	496 00
SERGIO DIMAS CAMPOS /		0	500 00

Figura 5.62. La pantalla de lista de deudores.

Esta opción tiene como función, proporcionar ya sea en pantalla, en impresora y/o en un archivo un reporte que muestre la lista de personas inscritas a los cursos que tengan saldos pendientes. Estos saldos son originados debido a que existió un acuerdo en pagar de manera diferida el importe total de un curso.

Al igual que el reporte de cursos por vencer, este reporte se puede imprimir o almacenar en un archivo (ver el apartado de cursos por vencer para mayor referencia.). La extensión de los archivos generados es siempre ".LDD".

LAS PANTALLAS DEL REGISTRO DE INSCRIPCIÓN Y REPORTE DE RECIBOS.

Al seleccionar la opción de "registro de inscripción y reporte de recibos" del menú de reportes, aparecerá la pantalla de solicitud de clave del curso, para generar el registro de inscripción y el reporte de recibos, la cual se muestra en la figura 5.63.

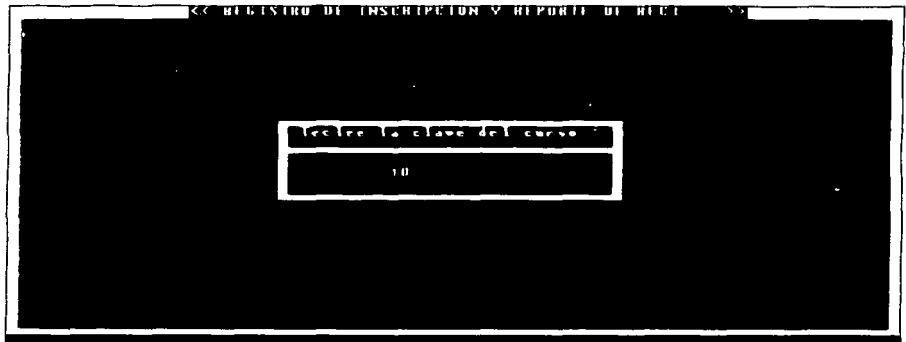


Figura 5.63. La pantalla de solicitud de clave del curso.

Esta opción tiene como función, generar ya sea en pantalla, en impresora y/o en un archivo los reportes que contengan la lista de recibos y el registro de inscripción de un número de oficio previamente determinado o seleccionado por el usuario. Si se generan los reportes en archivos, el registro de inscripción siempre llevará la extensión ".REG" y el reporte de recibos la extensión ".REP".

Como se aprecia en la figura 5.63, aparece una pantalla que solicita al usuario la clave del curso del cual se requiere obtener el registro de inscripción y el reporte de recibos. Esta clave debe existir para que no se produzca un error. Si se desea, se puede teclear una parte de la clave, y de encontrar más de un registro, el programa listará el total de registros que comiencen con la clave parcial tecleada, permitiendo hojear y seleccionar un registro (ver figura 5.64.).

Ya seleccionado un curso, el programa realiza una revisión en las ordenes de pago asociadas a dicho curso.

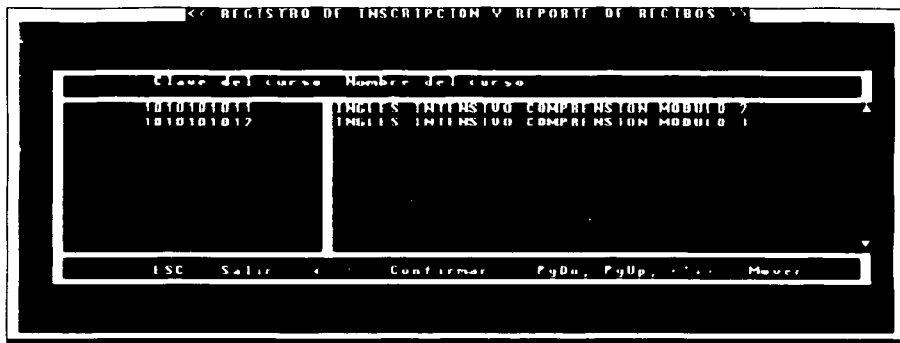


Figura 5.64. La pantalla con la lista de cursos que comienzen con una clave parcial.

Si no encontrara ningún recibo asociado, el programa notificará al usuario esta situación.

Si el programa únicamente encontrara recibos ya reportados, enviará dos mensajes al usuario, el primero le avisa que no existen recibos por reportar (ver figura 5.65) y el segundo le indica si desea reportar un oficio ya existente (ver figura 5.66) .

En caso afirmativo, el programa exhibe en una pantalla similar a la figura 5.67, en donde se listan el conjunto de oficios asociados al curso seleccionado y se le permite al usuario escoger uno de ellos.

Después de esto, el programa muestra la pantalla de datos del oficio, la cual es similar a la que se muestra en la figura 5.68. Esta pantalla solicita los datos referentes a encabezados, fechas, honorarios del profesor y los datos del personal relacionado a los oficios. Estos datos son empleados para generar el reporte de recibos. Su uso y conjunto de teclas válidas es el mismo que el empleado para las altas de cursos, ya que también se divide en tres áreas, sólo que esta tiene cambiado el nombre del área de memorándum por "Copia para", en la cual se tecldea el nombre y cargo de aquel personal que recibe copia de este reporte. Para entrar a esta zona debe presionar la tecla < ENTER > y aparecerá la zona de captura, ilustrada en la figura 5.69.

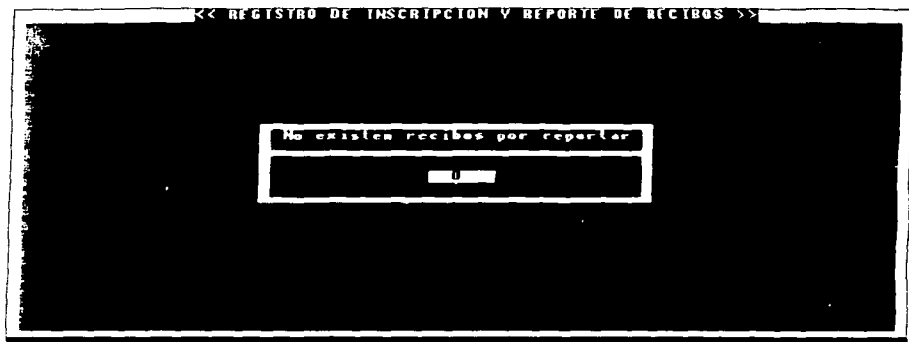


Figura 5.65. La pantalla de inexistencia de cursos por reportar.

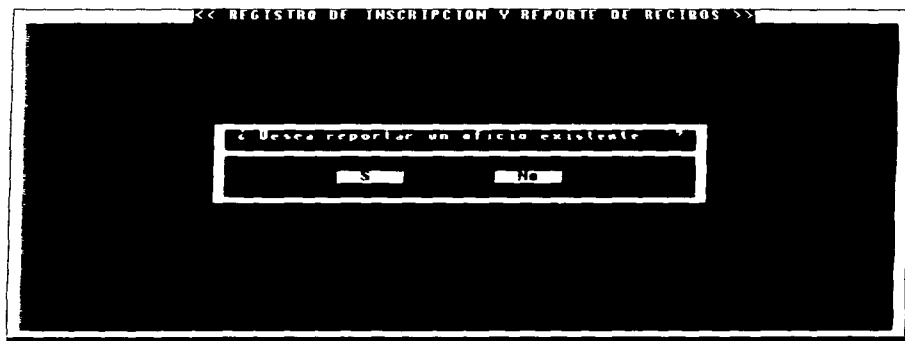


Figura 5.66. La pantalla de confirmación para reportar un oficio ya existente.

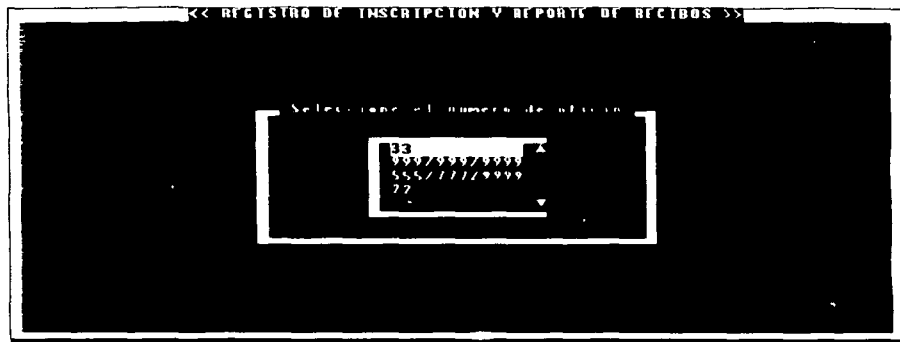


Figura 5.67. La pantalla de la lista de números de oficios asociados a un curso.

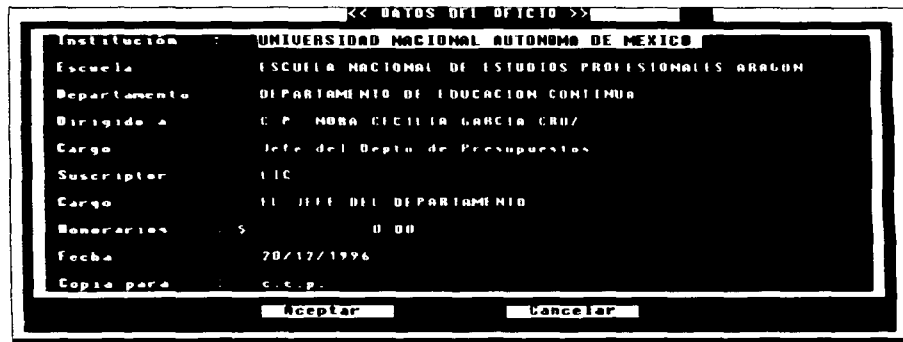


Figura 5.68. La pantalla de datos del oficio.

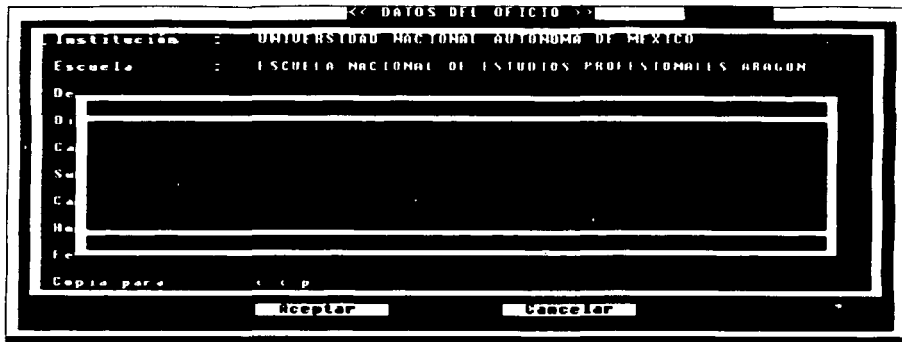


Figura 5.69. La pantalla de captura de personal con copia del reporte de recibos.

Después de capturar los datos del oficio, el programa generará el reporte del registro de inscripción (ver figura 5.70). Este reporte, al igual que el de cursos por vencer, se puede imprimir o almacenar en un archivo (ver apartado de cursos por vencer para mayor referencia.). Posteriormente generará el reporte de recibos (ver figura 5.71) que también se puede imprimir o almacenar en un archivo.

Al terminar de visualizar los reportes, el programa solicitará si desea generar otro oficio o reportar uno ya existente, como se muestra en la figura 5.72

En caso de que el programa encuentre únicamente recibos sin reportar, solicitará al usuario que teclee el número de oficio a asignar a estos recibos no reportados, lo anterior se ilustra en la figura 5.73. Posteriormente solicita los datos del oficio.

Por ultimo, si el programa detecta recibos sin reportar y al mismo tiempo recibos ya reportados, mostrará la lista de los números de oficios y al final de la lista una opción llamada "Oficio nuevo", que le indica al usuario la posibilidad de crear un nuevo oficio e incluir en el mismo los recibos sin reportar.(ver figura: 5.74.). En caso de escoger esta opción el programa solicita el número de oficio a designar y los datos del oficio.

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES DEPARTAMENTO				
REGISTRO				
Cursos o talleres	INGLES	INTENSIVO	COMPRESION	MODULO 2
Fecha del curso	11/02/1997	11/03/1997	Departación	
NOMBRE DEL ALUMNO	TELLEFONO	No. CUENTA	CARRERA	
1 DAVID TORRES SANCHEZ	3196545	IIIIIIIIII	INGENIERIA	
2 MAURA LUNA PEREZ	3136110	IIIIIIIIII		
3 SERGIO OLMOZ CAMPOS	4621285	IIIIIIIIII		

ESC o CTRL-V = Salir ← Pg/Ar Pg/Op «+» HOME END TAB y DE = Desplazarse

Figura 5.70. Ejemplo de la pantalla del reporte del registro de inscripción.

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES DEPARTAMENTO	
DEPARTAMENTO DE EDUCACION CONTINUA	
01/03/97	
C. P. NOVA CECILIA GARCIA SRIJ/ Jefe del Depto. de Preinscripción Presidencia	
Por medio del presente me permito enviar a Usted el recibo oficial que	

ESC o CTRL-V = Salir ← Pg/Ar Pg/Op «+» HOME END TAB y DE = Desplazarse

Figura 5.71. Ejemplo de la pantalla del reporte de recibos.

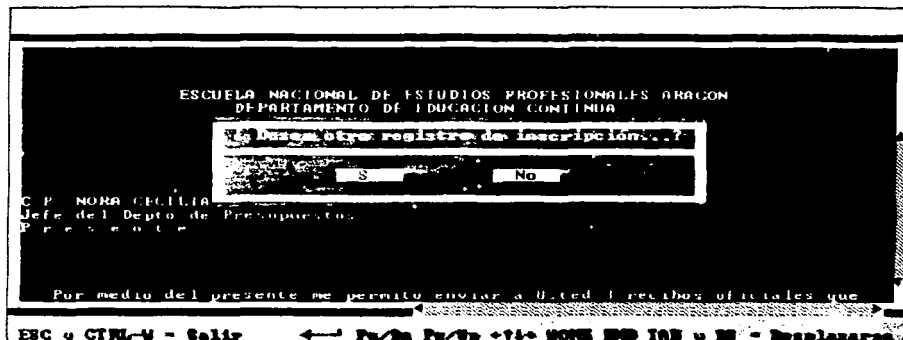


Figura 5.72. La pantalla de solicitud de otro registro de inscripción.

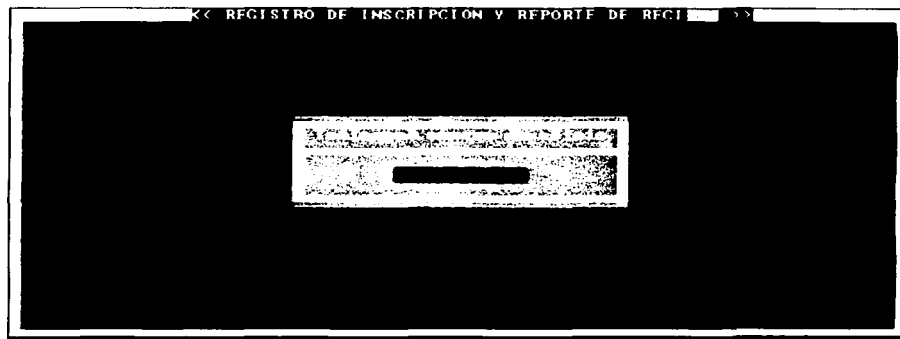


Figura 5.73. La pantalla de solicitud del número de oficio.

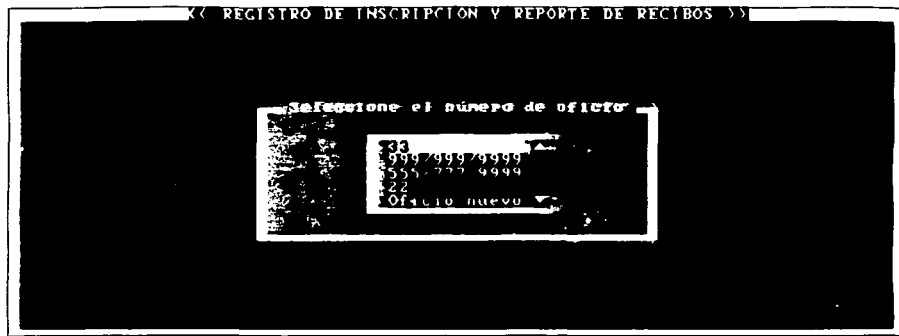


Figura 5.74. La pantalla de selección de un oficio ya existente o de creación de uno nuevo.

LA PANTALLA DE LISTA DE ASISTENCIA DE CURSOS PARA PROFESORES.

Al seleccionar la opción de "asistencia de cursos para profesores" del menú de reportes, aparecerá la pantalla de asistencia de cursos para profesores, la cual se muestra en la figura 5.75.

Esta opción tiene como función, proporcionar ya sea en pantalla, en impresora y/o en un archivo un reporte que muestre la lista de alumnos inscritos a un curso previamente determinado por el usuario. Si se genera el reporte en un archivo, este llevará la extensión ".LIS".

Como se aprecia en la figura 5.75, aparece una pantalla que solicita al usuario la clave del curso del cual se desea obtener la lista de participantes. Esta clave debe de existir para que no se produzca un error. Si se desea, se puede teclear una parte de la clave, y de encontrar más de un registro, el programa listará el total de registros que comiencen con la clave parcial tecleada, permitiendo hojear y seleccionar un registro (ver figura 5.76).

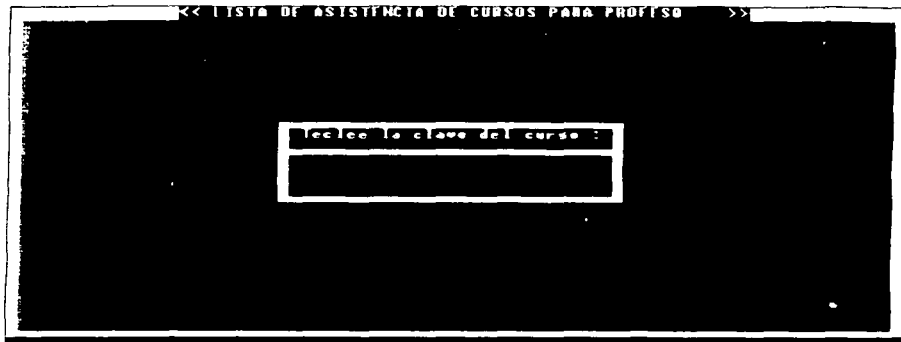


Figura 5.75. La pantalla de la lista de asistencia de cursos para profesores

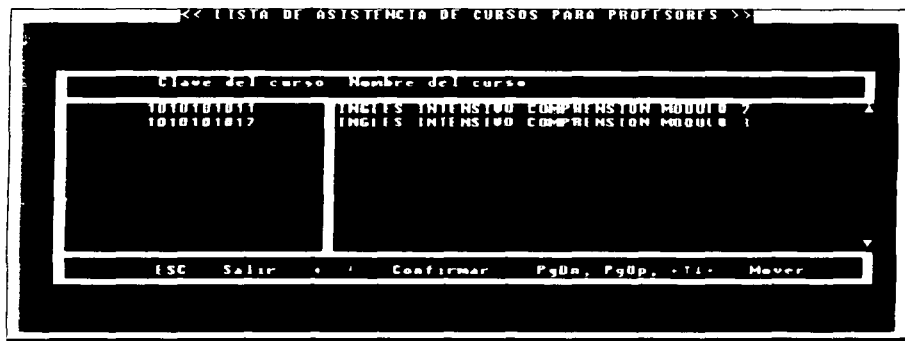


Figura 5.76. Ejemplo de la pantalla del listado de cursos que comiencen con sólo una parte de la clave.

Después de escoger la clave de un curso, el programa genera el reporte respectivo (ver figura 5.77). que al igual que el de cursos por vencer se puede imprimir o almacenar en un archivo (ver el apartado de cursos por vencer para mayor referencia.)

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO									
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES									
DEPARTAMENTO DE EDUCACION CONTINUA									
LISTA DE ASISTENCIA									
Curso a Val Per	INGLES	INTERVISTO	COMPRESION	MODULO 7	Expositor	CESR			
		Fecha del curso		11/09/1996	11/11/1996	Dura			
NOMBRE DEL ALUMNO									
1	DAVID TORRES SANCHEZ	0	0	0	0	0	0	0	0
2	GERARDO RAMIREZ URBEL	0	0	0	0	0	0	0	0
3	JAIKI GOMEZ PENA	0	0	0	0	0	0	0	0
4	JUAN BAPTISTA JIMENEZ	0	0	0	0	0	0	0	0
5	LUIS MARTINEZ RAMONA	0	0	0	0	0	0	0	0
6	MANUEL HUERTA PEREZ	0	0	0	0	0	0	0	0
7	MAURA LUNA PEREZ	0	0	0	0	0	0	0	0
8	RINI ANGUSTIANO ALULAH	0	0	0	0	0	0	0	0

Figura 5.77. Ejemplo de la salida del reporte de la lista de asistencia para profesores.

LA PANTALLA DE REPORTE A LA UNIDAD DE PLANEACION.

Al seleccionar la opción de "reporte a la unidad de planeación" del menú de reportes, aparecerá la pantalla de reporte a la unidad de planeación, la cual se muestra en la figura 5.78.

Esta opción tiene como función, proporcionar ya sea en pantalla, en impresora y/o en un archivo un reporte que muestre el porcentaje de avance de cursos y el tiempo que este lleva, en función de el rango de fechas preestablecidas por el usuario. Si se genera el reporte en un archivo, este llevará la extensión ".RUP".

Las fechas tecladas deben ser siempre válidas para que no se produzca un error. Por ejemplo la fecha inicial siempre en menor o igual a la fecha final.

Ya introducidas las fechas, aparecerá en pantalla la relación de cursos que se estén o hayan impartidos dentro del rango especificado (ver figura 5.79.). Este reporte al igual que el de cursos por vencer se puede imprimir o almacenar en un archivo (ver el apartado de cursos por vencer para mayor referencia.).

<< REPORTE A LA UNIDAD DE PLANEACION >>

leclee el periodo del cual desea reportar avance

01/01/1996 11/17/96

Figura 5.78. Ejemplo de la pantalla del solicitud del periodo para el reporte de la unidad de planeación.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO			
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES			
SEGUIMIENTO DE AVANCE 01/01/1996			
VIAJE			
INSTANCIA	DEPARTAMENTO DE EDUCACION CONTINUA	RESPONSABLE	
PROGRAMAS	ETAPAS DE DESARROLLO AVANZAS	/	TIEMPO IM
IDIOMAS	INGLES INTENSIVO COMPRENSION MODULO 7	100	7 04
IDIOMAS	INGLES INTENSIVO COMPRENSION MODULO 8	100	7 11
COMPUTACION	CURSO DE SISTEMA OPERATIVO MS DOS 6 22	100	7 40
CARRERA	CALCULO DIFERENCIAL E INTEGRAL	100	1 09
MAESTRIA	MAESTRIA EN ECONOMIA FINANCIERA	100	8 85

Figura 5.79. Ejemplo de la salida del reporte de la unidad de planeación.

EL MENÚ GENERAL DE DISPOSICIÓN O CONFIGURACIÓN.

Al seleccionar la opción de "disposición" del menú principal, aparecerá el menú general de configuración el cual se muestra en la figura 5.80. Este menú permite controlar la configuración de periféricos como las unidades de disco y la impresora y los datos empleados en el programa como la clave de acceso y los datos del oficio.

Como se puede observar este menú presenta cinco opciones que son : a) unidad; b) impresora; c) datos; d) clave, y e) salir . Estas opciones se detallan más adelante.

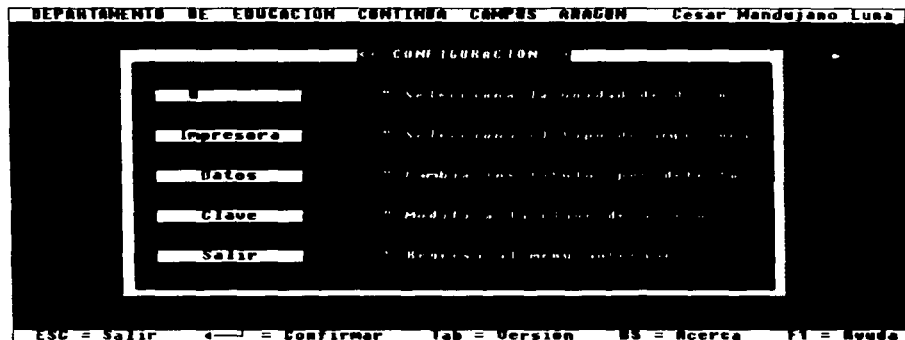


Figura 5.80. La pantalla del menú general de disposición o configuración.

LA PANTALLA DE SELECCIÓN DE UNIDAD DE TRABAJO Y/O RUTA.

Al seleccionar la opción de "unidad" del menú de disposición, aparecerá la pantalla de selección de unidad de trabajo y/o ruta la cual se muestra en la figura 5.81.

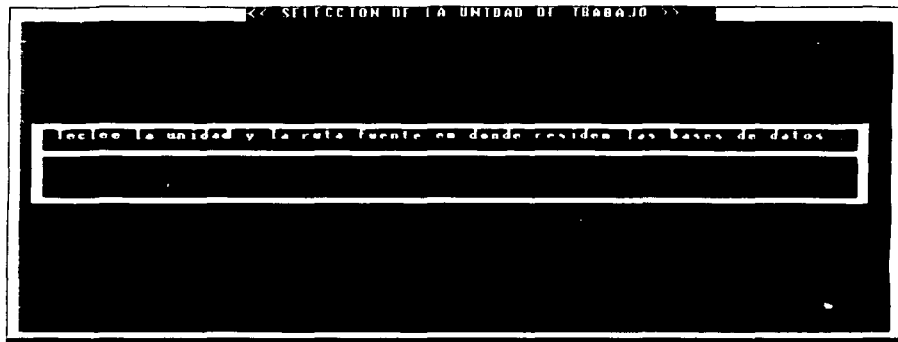


Figura 5.81. La pantalla de solicitud de la unidad de trabajo.

Esta pantalla permite especificar la unidad de discos y/o la ruta en donde residen las bases de datos. Claro está, ya deben de existir para que no ocurra ningún error.

Después de haber especificado la unidad y/o ruta, el programa cierra las bases de datos actuales y carga las que existen en dicha unidad y/o ruta.

De no existir algunas de las bases de datos en la unidad y/o ruta especificada, el programa las creará, aunque sin registros.

EL MENÚ DE TIPO DE IMPRESORA

Al seleccionar la opción de "impresora" del menú de disposición, aparecerá el menú de tipo de impresora el cual se muestra en la figura 5.82.

Como se puede observar este menú presenta tres tipos de impresoras que son : a) Matriz de puntos; b) Chorro de Tinta, y c) Láser.

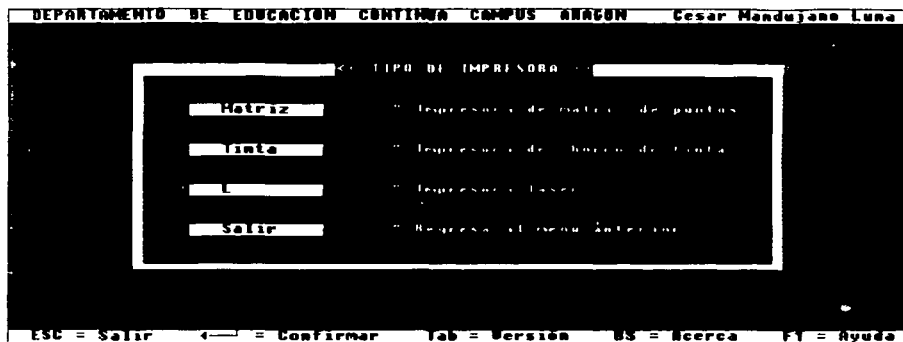


Figura 5.82. La pantalla del menú de tipo de impresora.

Es importante especificar el tipo de impresora que se dispone para que los reportes impresos desde el programa no presenten salidas no deseadas.

OBSERVACIÓN. El programa siempre recuerda el último tipo de impresora seleccionada por el usuario, y guarda dicho tipo en el archivo "CONFIG.DEC". De no existir este archivo el programa toma como defecto la impresora de tipo "Láser".

LA PANTALLA DE DATOS DEL OFICIO.

Al seleccionar la opción de "datos" del menú de disposición, aparecerá la pantalla de datos oficio la cual se muestra en la figura 5.83. Como se puede ver aparece ligeramente distinta a la empleada en el módulo de reportes, ya que le hacen falta las partes que solicitan los honorarios del profesor y la que solicita la fecha.

<< DATOS DEL OFICIO >>

Institución :	UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
Escuela :	ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ABALOM
Departamento :	DEPARTAMENTO DE EDUCACION CONTINUA
Dirigido a :	C. P. NORA CECILIA GARCIA ERIZ
Cargo :	Jefe del Depto. de Presupuestos
Subcriptor :	LIC
Cargo :	EL JEFE DEL DEPARTAMENTO
Copia para :	C. C. P.

Figura 5.83. La pantalla de datos del oficio.

Todos los datos que se muestran en esta pantalla se pueden modificar y grabar en el archivo de configuración "CONFIG.DEC", por lo que en futuros usos del programa, se recuerden los últimos datos capturados.

OBSERVACIÓN. El programa siempre recuerda los últimos datos del oficio capturados, siempre y cuando estos se modifiquen desde la opción "datos" del menú de disposición.

LA PANTALLA DE SELECCIÓN DE NUEVA CLAVE DE ACCESO.

Al seleccionar la opción de "clave" del menú de disposición, aparecerá la pantalla de selección de nueva clave de acceso la cual se muestra en la figura 5.84.

Esta pantalla permite que el usuario modifique la clave de acceso para realizar altas, bajas y modificaciones, aunque es muy importante señalar que sólo personal autorizado la puede modificar.

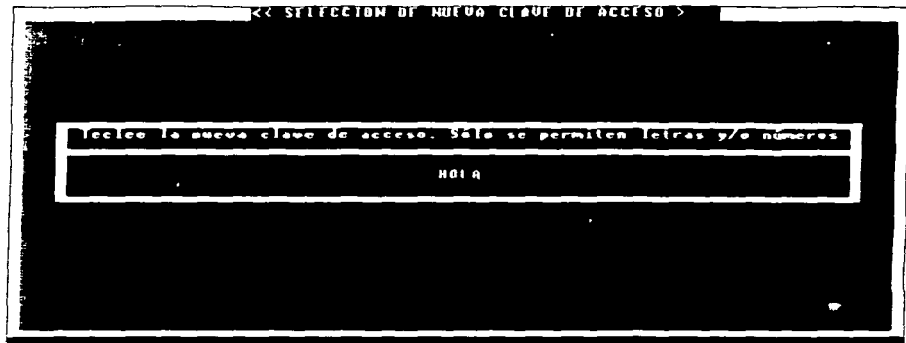


Figura 5.84. La pantalla de solicitud de nueva clave de acceso.

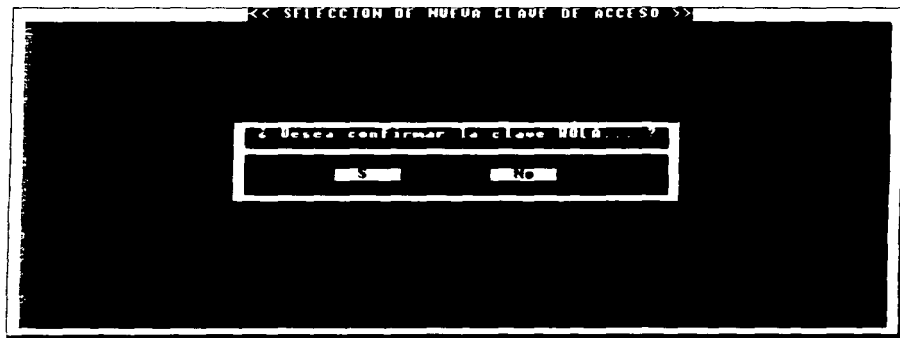


Figura 5.85. La pantalla de confirmación de una nueva clave de acceso.

La clave se compone de uno a cuatro caracteres y se confirma con la tecla < ENTER > (que en este ejemplo la nueva clave es "HOLA"). En caso de confirmar una nueva clave aparecerá una pantalla solicitando al usuario si desea confirmar la nueva clave (ver figura 5.85.). De ser afirmativo, la nueva clave se almacenará en el archivo "CONFIG.DEC" y quedará vigente para futuros accesos al programa.

CONCLUSIONES Y RECOMENDACIONES.

Como podemos apreciar, la solución mas adecuada para resolver la problemática de administración de cursos del DEC Aragón, es realizar un sistema basado en computadoras que facilite dicha administración y que sobre todo, optimice sus recursos humanos, materiales y financieros.

Esta optimización motivada por el uso del sistema, se justifica gracias a los siguientes tres puntos :

1) En lo que se refiere a los recursos materiales no se requieren gastos adicionales en instalación eléctrica, equipo de cómputo ni mantenimiento del mismo, ya que se utilizará el mismo equipo que actualmente dispone el DEC Aragón. Aunque cabe señalar, que es importante adquirir a futuro el equipo de cómputo recomendado en el manual del usuario, con el fin de obtener un mayor rendimiento y desempeño del programa propuesto.

En el caso de la papelería requerida, ya no será necesario mandar a imprimir en offset la mayor parte de los formatos y por lo mismo no preocuparse en mantener existencias mínimas, ya que estos formatos y reportes se pueden imprimir directamente en la impresora conectada a la computadora, con lo que la presentación de los mismos será de mayor calidad y presentación. Aquí el costo y uso del toner de la impresora bien se puede compensar, con el gasto realizado en la impresión de los formatos en offset y con la disminución de horas-hombre en el llenado y elaboración de formatos y reportes solicitados.

2) Con lo que respecta a los recursos humanos no es necesaria ninguna inversión, ya que sería el mismo personal de confianza el encargado de operar el sistema. La capacitación para el uso del programa es rápida debido a que el programa es muy amigable.

Por otra parte, el demás personal administrativo que también contribuye en las actividades del DEC Aragón se puede canalizar a otras labores que se requieran, de tal suerte que se entreguen los reportes sin demoras.

3) El sistema también le permitirá manejar su información de una manera más veraz y oportuna, contribuyendo a crear un mejor ambiente de trabajo y paralelamente le ayudará a aprovechar al máximo el recurso no renovable llamado tiempo .

Como podemos notar, la inversión requerida para la implementación del sistema es completamente falible y no afecta al presupuesto del DEC Aragón, ya que este cuenta con los recursos materiales y humanos necesarios para poderlo realizar, con lo cual los costos de implantación del mismo serán bajos.

De hecho, creemos pertinente mencionar, que el presente proyecto se podría extender a otros departamentos de educación continua de la UNAM y de otras instituciones. Esto es debido a formas similares de trabajar en el DEC Aragón con estos organismos. Quizás sería cuestión de utilizar el programa propuesto como proyecto piloto en el DEC Aragón y monitorear los beneficios que este obtenga para de esa manera implantarlo en otros organismos.

Cambiando de tema, el código fuente del sistema fue hecho en CLIPPER por razones de compatibilidad con los sistemas ya existentes en la UNAM Campus Aragón. No obstante, pudo haberse realizado en otros manejadores de bases de datos mas poderosos con lenguaje de programación incluido como lo son VISUAL FOX PRO® y ACCES® de MicroSoft®, o bien VISUAL DBASE® de Borland®. Debido a que estos últimos emplean conceptos de la programación orientada al objeto, se podrían aprovechar los recursos de la herencia y el poliformismo para disminuir el tamaño del código fuente del programa DEC, mejorar su entendimiento y al mismo tiempo, utilizar las clases predefinidas de estos lenguajes para emplear la plataforma y herramientas que ofrece WINDOWS, mediante las cuales se pueden agilizar los procesos y dar una mejor presentación a las pantallas del programa y a los formatos requeridos.

Ahora bien, independientemente del manejador y el lenguaje, las bases de datos generadas en CLIPPER son compatibles o exportables con los manejadores citados en el párrafo anterior y con la mayoría de los manejadores por lo que no habría necesidad de realizar de nuevo las bases de datos en caso de requerirlas para otros sistemas o programas.

Para concluir, citaremos las dos principales ventajas que encontramos al proyecto.

La primera es que por fin, el DEC Aragón cuenta con su primer programa informático para poder administrar sus cursos, que de algún modo resolverá en gran medida su problemática por hoy.

La segunda es que ya existe un estudio previo que facilitará a programadores futuros a adecuar los módulos según las nuevas necesidades que se presenten, o bien a cambiar el programa a otro lenguaje más poderoso.

Por el momento, lo más difícil ya está hecho : el comienzo.

Simplemente habrá que dar continuidad a este proyecto en función de los nuevos requerimientos que tenga el DEC Aragón.

BIBLIOGRAFÍA

- **Fundamentos de los sistemas de bases de datos**
S. M. Deen
Editorial Gustavo Gill
Barcelona, España 1987.

- **Fundamentos de bases de datos**
Henry F. Korth y Abraham Silberschatz
Editorial Mac Graw Hill
D.F., México 1993.

- **Sistemas de administración de bases de datos**
David Kruglinski
Editorial Osborne/Mac Graw Hill
Estado de México, México 1985.

- **Bases de datos**
G. Gardarin
Editorial Paraninfo
Madrid, España 1990.

- **Programación en CLIPPER 5.01 y 5.02**
José Antonio Ramalho
Editorial Mac Graw Hill
D.F., México 1994.

- CLIPPER 5.01 Avanzado**
José Antonio Ramalho
Editorial Mac Graw Hill
D.F., México 1994.

- Programación orientada al objeto en CLIPPER 5.01 y 5.02**
Francisco J. Morero Peyrona
Editorial Paraninfo
Madrid, España 1993.

- Impresoras matriciales, chorro de tinta y láser**
Juan Carlos Oros y Antonio Montes
Editorial Paraninfo
Madrid, España 1991.

- Software : The Norton Guides ver. 1.0**
Peter Norton Computing 1987.

- Manual de organización del DEC Aragón**
Editado por el DEC Aragón
Estado de México, Mexico 1998.

- Segundo informe de labores 1991 - 1992**
Claudio C. Merrifield Castro
Editado por la UNAM Campus Aragón
Estado de México, Mexico 1992.

- **Tercer informe de labores 1992 - 1993**
Claudio C. Merrifield Castro
Editado por la UNAM Campus Aragón
Estado de México, Mexico 1993.

- **Cuarto informe de labores 1993 - 1994**
Claudio C. Merrifield Castro
Editado por la UNAM Campus Aragón
Estado de México, Mexico 1994.

- **Primer informe de labores 1994 - 1995**
Claudio C. Merrifield Castro
Editado por la UNAM Campus Aragón
Estado de México, Mexico 1995.

- **Segundo informe de labores 1995 - 1996**
Claudio C. Merrifield Castro
Editado por la UNAM Campus Aragón
Estado de México, Mexico 1996.

- **Tercer informe de labores 1996 - 1997**
Claudio C. Merrifield Castro
Editado por la UNAM Campus Aragón
Estado de México, Mexico 1997.

- **Impresoras HP LaserJet 4 y 4M**
Manual del usuario.