

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

Sistemas de control escolar para instituciones educativas

Autor: Rafael Romero Díaz Barriga

**Tesis presentada para obtener el título de:
Ing. en Sistemas Computarizados [sic]**

**Nombre del asesor:
Sergio Fco. Barraza Ibarra**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar, organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





UNIVERSIDAD VASCO DE QUIROGA

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

**" SISTEMA DE CONTROL ESCOLAR
PARA INSTITUCIONES EDUCATIVAS "**

T E S I S

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN SISTEMAS COMPUTARIZADOS**

PRESENTA:

RAFAEL ROMERO DIAZ BARRIGA

**ASESOR DE TESIS
ING. Y M. A. SERGIO FCO. BARRAZA IBARRA**

No. ACUERDO 952006

CLAVE: 16PSU00 14Q



MORELIA, MICH., 20 DE SEPTIEMBRE DE 1996.

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

**" SISTEMA DE CONTROL ESCOLAR
PARA INSTITUCIONES EDUCATIVAS "**

T E S I S

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN SISTEMAS COMPUTARIZADOS**

PRESENTA:

RAFAEL ROMERO DIAZ BARRIGA

**ASESOR DE TESIS
ING. Y M. A. SERGIO FCO. BARRAZA IBARRA**



No. ACUERDO 952006

CLAVE: 16PSU00 14Q

MORELIA, MICH., 20 DE SEPTIEMBRE DE 1996.

EN PRIMER LUGAR, AGRADEZCO A DIOS POR HABERME DADO VIDA Y POR PERMITIRME LLEGAR HASTA DONDE HE LLEGADO.

HACE ALGUNOS AÑOS ME DIJERON QUE LA MÁS GRANDE HERENCIA QUE RECIBIRÍA, SERIA, "EL DARME UNA CARRERA Y EL HACERME UN HOMBRE DE PROVECHO", AHORA QUIERO DECIR, GRACIAS PAPÁ, GRACIAS MAMÁ, POR HABER HECHO HASTA LO IMPOSIBLE POR LOGRARLO; POR HABERME DADO LO MEJOR DE USTEDES: AMOR, AMISTAD, CARIÑO Y EDUCACIÓN; LES PROMETO QUE NO LOS DEFRAUDARE.

SU HIJO, RAFAEL.

A mis maestros con agradecimiento

A mi asesor de tesis, por su apoyo y ayuda :

Ing. y M.A. Sergio Fco. Barraza Ibarra

Con gratitud a mis escuelas :

Primaria "Generalísimo Morelos", Erongarícuaro, Mich.

Secundaria "María Luisa Martínez", Erongarícuaro, Mich.

"Centro de Bachillerato Industrial y de Servicios N° 94", Pátzcuaro, Mich.

"Universidad Vasco de Quiroga", Morelia, Mich.

INDICE

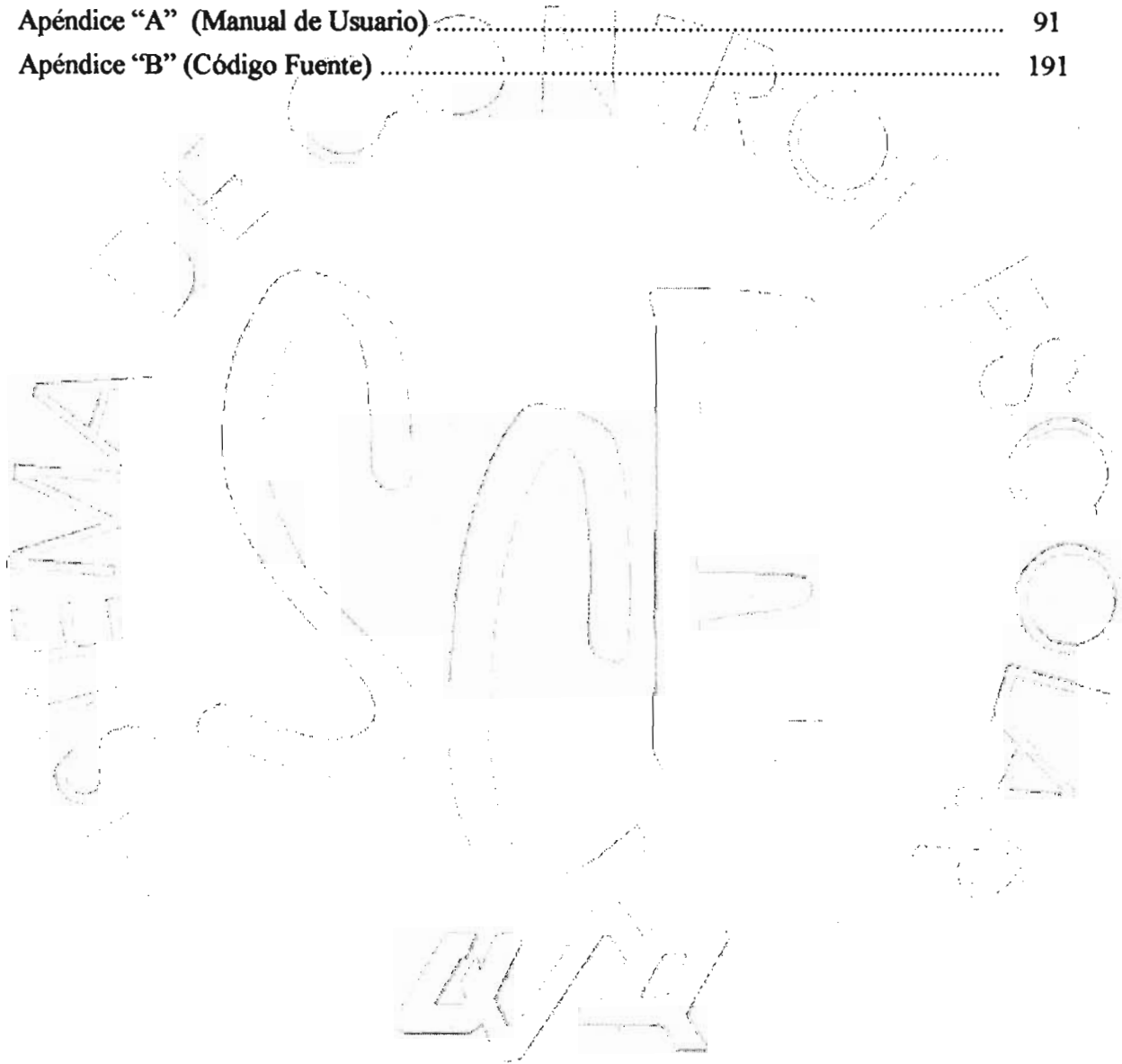
1.- Introducción	1
2.- Antecedentes	4
3.- Justificación	8
4.- Objetivo General	10
4.1.- Objetivos Específicos	11
4.2.- Metas	12
5.- Análisis del proceso actual	13
5.1.- Proceso de admisión	13
5.2.- Proceso de reingreso	15
5.3.- Registro de la información	16
5.4.- Flujo de la información	19
5.4.1.- Registro de calificaciones	19
5.4.2.- Listas de asistencia	20
5.4.3.- Documentos oficiales	20
5.5.- Sistema anterior de Control Escolar	22
6.- Teoría General de Sistemas	23
6.1.- Definición de Sistema	23
6.2.- Análisis y diseño de sistemas	25
6.2.1.- Especificación de los requerimientos de la aplicación	25
6.2.1.1.- Actividades de la determinación de requerimientos	26
6.2.1.2.- Requerimientos básicos	27
6.2.1.3.- Percepción de requerimientos	28
6.2.2.- Técnicas para la determinación de requerimientos	30
6.2.3.- Objetivos al diseñar un sistema de información	32
6.2.4.- Especificación de diseño	35
6.3.- Ciclo de vida clásico del desarrollo de sistemas	40
7.- Redes informáticas	44
7.1 - ¿ Por qué una red ?	44
7.2.- Definición de red informática	44
7.3.- Componentes de una red	45
7.4.- Características de las redes de información	47
7.5.- Ventajas de las redes de información	48
7.6.- Desventajas de las redes de información	49
7.7.- Medios de transmisión	50



7.7.1.- Distancia máxima entre terminales	51
7.7.2.- Comparación de los medios de transmisión	51
7.8.- Topología de red	52
7.8.1.- Topología en anillo o token ring	52
7.8.2.- Topología en bus	53
7.8.3.- Topología punto a punto	53
7.8.4.- Topología estrella	54
8.- Archivos y bases de datos	55
8.1.- Introducción	55
8.2.- Terminología básica de archivos	55
8.3.- Concepto de base de datos	57
8.3.1.- Concepto de Sistema de Administración de Base de Datos	58
8.3.2.- Características de Sistema de Administración de Base de Datos	59
8.4.- Modelos de datos	60
8.4.1.- Definición de Modelo de Datos	60
8.4.2.- Objetivos de un Modelo de Datos	61
8.4.3.- El Modelo de Datos Relacional	61
8.4.3.1.- Objetivos del Modelo Relacional	62
8.4.3.2.- Estructura del Modelo Relacional	63
8.4.3.3.- Dominio	64
8.4.3.4.- Restricciones del Modelo Relacional	65
8.4.3.4.1.- Restricciones Inherentes	65
8.4.3.4.2.- Restricciones de Usuario	66
8.5.- Seguridad y confidencialidad	67
9.- Descripción General del Sistema	70
9.1.- Descripción de los módulos del Sistema	70
9.1.1.- Módulo de Escuelas	70
9.1.2.- Módulo de Alumnos	71
9.1.3.- Módulo de Maestros	71
9.1.4.- Módulo de Materias	71
9.1.5.- Módulo de Grupos	71
9.1.6.- Módulo de Reportes	72
9.1.7.- Módulo de Utilerías	72
9.2.- Descripción de las bases de datos del Sistema	72
9.2.1.- Contenido de las bases de datos del Sistema	73



9.2.1.1.- Estructura de las bases de datos del Sistema	76
9.2.2.- Interrelación de las bases de datos con el Sistema	79
9.3.- Registro de la información con el sistema propuesto	83
10.- Conclusiones y recomendaciones	85
Bibliografía	88
Apéndice "A" (Manual de Usuario)	91
Apéndice "B" (Código Fuente)	191



CAPITULO 1

“INTRODUCCIÓN”

En base a las necesidades y requerimientos solicitados por el área administrativa de la Universidad Vasco de Quiroga, se observó la urgencia de contar con un sistema computarizado de Control Escolar más eficiente que el actual, con el cual se pueda ofrecer una mayor eficiencia en el manejo de la información y el servicio al alumnado de los diferentes niveles educativos de la Universidad.

Analizando lo anterior, la Rectoría de la Universidad en coordinación con la Escuela de Sistemas Computarizados, llegaron a la determinación de que la solución más adecuada sería la elaboración de un nuevo sistema, el cual será realizado por un alumno de la misma Universidad de la carrera de Sistemas Computarizados, de tal forma que el nuevo sistema esté realizado y adecuado explícitamente a las necesidades de la Universidad.

De esta forma, la Universidad busca que los alumnos pongan en práctica sus conocimientos, de tal manera que colaboren, participen activa y directamente en el mejoramiento, fortalecimiento y superación de la Universidad.

El presente trabajo está dividido en diez capítulos con sus respectivos apartados internos.

Como en todos los trabajos, es importante señalar el marco de antecedentes relacionados al mismo y por lo tanto, esto es visto en el capítulo segundo.

Por otro lado, todo proyecto debe ser realizado con base a un estudio detallado de factibilidad y tener una justificación técnica y operativa para la realización del mismo, esta es analizada en el capítulo tercero.

En el cuarto capítulo, se detallan los objetivos que se desean alcanzar con el presente trabajo, así también, se explican las metas que se pretenden alcanzar con la realización del mismo.

En el quinto capítulo se explican los procesos actuales que tienen que realizar los alumnos y el camino que sigue la información dentro de las distintas áreas de la Universidad para que pueda ser procesada; esto es para poder tener una visión general sobre el manejo de la información dentro de la institución.

Es en el sexto capítulo, donde se explica lo que son los sistemas de información, así como la forma, procedimientos, técnicas y herramientas necesarias para lograr un diseño de sistemas óptimo.

En el séptimo capítulo se aborda uno de los temas que están de moda en la actualidad, es precisamente el tema referente a “redes de información”, se comienza por explicar sus características, ventajas y desventajas, para terminar haciendo una explicación de las topologías de red adoptadas más comúnmente por las empresas.

Dado la gran cantidad de información generada y requerida cada vez con mayor rapidez y exactitud por las empresas, en el octavo capítulo, se trata el tema de las bases de datos. El tema abarca desde las definiciones, conceptos y características de los distintos sistemas de administración de bases de datos, para terminar con el estudio del modelo relacional que es la base del presente sistema.

En el capítulo noveno, es donde hago una descripción general del sistema propuesto, se explica brevemente la estructura interna, tanto del sistema, como de las bases de datos que lo integran.

En el décimo capítulo expongo las conclusiones a las que llegué al concluir el presente trabajo y las recomendaciones que consideré importantes hacer para el buen manejo del sistema.

Finalmente, se incluyen dos apéndices muy útiles para lograr un aprovechamiento y un uso óptimo del sistema : El “manual de usuario”, que es el que detalla en forma explícita, la forma de operar el sistema y el “código fuente” del sistema, que es el listado de todas las instrucciones del lenguaje de programación de que está constituido el software.

CAPITULO 2**“ANTECEDENTES”**

La Universidad Vasco de Quiroga es una institución educativa de nivel medio superior y superior, que desde hace 17 años se ha preocupado por formar profesionistas íntegros y preparados para afrontar todo tipo de situaciones en los ámbitos laborales y sociales.

La universidad nació originalmente en el mes de enero de 1979, con el nombre de Instituto de Estudios Superiores Vasco de Quiroga (IESVAQ), nombre que en el mes de abril de 1992 fue cambiado por el de Universidad Vasco de Quiroga (UVAQ), ya que el nombre de IESVAQ no se ajustaba al tipo de educación que en la institución se ofrece. Hasta hace menos de cuatro años, el hasta entonces IESVAQ funcionaba en el edificio que actualmente ocupan las preparatorias de Morelia, pero dado el número cada vez mayor de estudiantes, este edificio fue resultando poco a poco insuficiente; es por esta razón que se tomó la decisión de comprar un terreno y construir un Campus Universitario que contara con las instalaciones necesarias. Es de esta manera que hace menos de tres años, se trasladaron a su edificio en el campus Santa María, algunas de las licenciaturas que ofrece la universidad. Actualmente se encuentra en construcción el segundo edificio del Campus, el cual será más grande y en donde se instalarán las carreras que actualmente no están laborando en el campus Santa María.

Por otra parte es conveniente hacer notar que el nacimiento de la informática está relacionado con la necesidad que ha sentido siempre el hombre de disponer de un sistema que le permita manejar gran cantidad de información con relativa rapidez así como de efectuar cálculos a gran velocidad y de un modo mecánico que le libere de las penosas tareas asociadas con estas actividades. Los primeros antecedentes de sistemas, muy rudimentarios, destinados a solventar estos problemas son, por ejemplo, los ábacos, estos dispositivos rudimentarios de cálculo todavía se emplean en la actualidad.

Sin embargo, los antecedentes de la computadora son sin duda los mecanismos para la resolución de dichos problemas creados en épocas posteriores, que, en lo referente al cálculo se deben a los trabajos de Blaise Pascal (1623-1662), el cual creó una máquina capaz de sumar y restar mediante la combinación de una serie de ruedas dentadas (esta máquina es semejante a los dispositivos mecánicos que se emplean en la actualidad en los cuentakilómetros de los vehículos automóviles).

En un sentido estricto, cabe considerar que los auténticos inicios de la informática datan del siglo XIX, más concretamente de los trabajos realizados por Hermann Hollerith (1860-1929), miembro de la oficina del censo de los Estados Unidos de Norteamérica. La contribución de Hollerith consistió en emplear una cinta (que más tarde sustituyó por tarjetas), en la que se grababa la información mediante perforaciones en lugares determinados.

El siguiente paso en el camino del tratamiento automático de la información se debió a los trabajos de Howard H. Aiken (1900-1973), quien desarrolló, entre 1939 y 1944 y en el seno de compañía IBM (creada en 1924), la máquina conocida por ASCC (Automatic Sequence Controlled Calculator), mejor conocida como MARK I. Esta máquina se basaba, desde el punto de vista del sistema físico, en un dispositivo eléctrico simple, el relé, y su programación se llevaba a cabo mediante tarjetas perforadas. Esta máquina puede ser considerada como la primera computadora de la historia, disponía de una capacidad de memoria de 72 números de 23 cifras decimales. Sin embargo, era extraordinariamente lenta ya que necesitaba unos 10 segundos para llevar a cabo la multiplicación de dos números de 10 cifras. Además, su peso era de unas cinco toneladas y estaba constituida por unos 5,000 relés.

El siguiente paso se produjo gracias a la aplicación de la electrónica a la resolución de este tipo de problemas. En 1946, la escuela Moore de Ingeniería Electrónica, situada en Filadelfia, USA, construyó la primera máquina electrónica de calcular. Había sido diseñada en la universidad de Pennsylvania, entre 1943 y 1946 por J. Mauchly, J.P. Eckert y H. H. Goldstine. Se llamaba ENIAC (Electronic Numerical Integrator and Computer), estaba formada por 18,000 bulbos, pesaba unas 30 toneladas, consumía 150 Kw de corriente eléctrica y sus dimensiones eran tales que se encontraba albergada en un edificio expresamente construido para tal fin, su programación se llevaba a cabo mediante el establecimiento de conexiones entre cables eléctricos y el accionamiento de gran cantidad de interruptores o relés.

En las décadas siguientes, el progreso de este tipo de instalaciones fue cada vez más acelerado y siguió una serie de etapas que reciben el nombre de *generaciones* y que abarcan períodos determinados según se trate del sistema físico o lógico.

La *primera generación* la constituyen las computadoras que se construyeron entre los años 1950 a 1960 y su componente básico era el bulbo.

La *segunda generación* es la que comprende las computadoras construidas entre los años 1960 a 1965 y su componente básico era el transistor.

La *tercera generación* que abarca desde 1965 a 1975, su construcción y funcionamiento se basaba en el empleo de los circuitos integrados.

La *cuarta generación*, abarca desde 1975 y hasta nuestros días y se caracteriza fundamentalmente por la continuación del proceso de integración que culminó en 1975 con los circuitos integrados, asimismo está relacionada con la aparición del microprocesador.

Finalmente se habla de la llamada *quinta generación* puesta en marcha por las industrias japonesas y en la cual, a partir de 1981 se trabaja en el desarrollo de computadoras inteligentes desde el punto de vista del sistema físico.

CAPITULO 3**“JUSTIFICACIÓN”**

El área administrativa de la Universidad Vasco de Quiroga actualmente se apoya con un sistema de Control Escolar, al cual se le han detectado algunas deficiencias en el manejo de la información; esta se tomó como la principal razón por la cual es necesario llevar a cabo la elaboración de un nuevo Sistema de Control Escolar.

Algunas de las deficiencias observadas en el sistema actual se enuncian a continuación:

- ◆ Limitación para el manejo de la información a una sola escuela, siendo que la Universidad cuenta con más de 15 escuelas y está en vías de expansión.
- ◆ El sistema actual es para trabajo como monousuario.
- ◆ La limitación de manejar solamente una escuela, obliga a tener instaladas múltiples copias (una para cada escuela), dentro de un mismo disco duro, haciendo con esto que se desperdicie una gran cantidad de espacio útil.
- ◆ El sistema cuenta con llave física (centinela) de acceso, lo cual lo hace riesgoso ya que si la llave se pierde o se daña (de hecho ya se encuentra parcialmente dañada), se perdería el acceso definitivo a toda la información.

- ◆ La empresa que realizó el sistema actual ya no existe, por lo cual, no se pueden llevar a cabo actualizaciones, consultas y no se puede contar con asistencia técnica.

- ◆ Otra razón por la cual se realiza un nuevo sistema de Control Escolar se debe a que se desea la unificación en el manejo de la información de los diferentes planteles con que cuenta la Universidad Vasco de Quiroga en el Estado de Michoacán.

- ◆ Disponer de un Sistema de Control Escolar flexible a las necesidades de la Universidad y apegado a la normatividad vigente de la Secretaría de Educación en el Estado.

- ◆ Contar con un Sistema de Control Escolar que permite integrar el antecedente histórico académico del alumno, es decir, el kardex escolar, de una manera eficiente y confiable, además de lograr la emisión de listas, boletas de calificaciones, credenciales, constancias y documentos oficiales que requiere la Secretaría de Educación en el Estado.

CAPITULO 4

“OBJETIVO GENERAL”

Realizar un Sistema de Control Escolar para la Universidad Vasco de Quiroga que le permita manejar de manera integral el proceso administrativo en la gestión de la información para una mayor eficiencia y productividad del departamento de Control Escolar hacia la comunidad educativa.

4.1 .- OBJETIVOS ESPECÍFICOS

- ◆ Que dentro de un mismo sistema se maneje la información de todas las escuelas de la Universidad. Esto es para evitar el problema que se presentaba al tener múltiples copias del sistema en disco.

- ◆ Que el sistema funcione en un ambiente de red de área local.

- ◆ Evitar el riesgo de perder información por causas de llaves y/o claves de acceso.

- ◆ Agilizar el manejo tanto interno como externo de las actividades que se desarrollan dentro del departamento de Control Escolar como son: inscripciones, reinscripciones, alta de maestros, manejo de materias, etc.

- ◆ Eliminar gran parte de las formas impresas existentes, ya que con el nuevo Sistema de Control Escolar se harían innecesarias algunas de ellas.

- ◆ Facilitar el manejo de información entre las diferentes escuelas que conforman la Universidad Vasco de Quiroga.

- ◆ Se generarán los reportes básicos necesarios, como son boletas, listas de asistencia, credenciales, constancias y toda aquella información requerida por la SEE.

4.2 .- METAS

- ◆ En un plazo de 3 a 5 días a partir de la fecha en que se concluya la fase de recolección de información y análisis del problema, se instalará la red y se pondrá a funcionar en el laboratorio de cómputo del campus Sta. María.

- ◆ La fase de desarrollo del sistema esta contemplada para realizarse en un plazo de 3 meses a partir de la fecha en que se haya terminado de instalar la red.

- ◆ En un plazo de 15 días antes de terminar la fase de desarrollo se instalará el servidor y la red de cómputo en el campus de Taximaroa, que servirá de ambiente del sistema.

- ◆ Posterior a la fase de desarrollo se entrará a la fase de prueba y depuración, la cual está contemplada llevarse a cabo en un periodo de 2 semanas.

- ◆ La liberación final del sistema se pretende realizar en un período de 3 a 3 1/2 meses, a partir de la fecha de inicio del trabajo.

CAPITULO 5

“ANÁLISIS DEL PROCESO ACTUAL”

El departamento de Control Escolar como parte integrante de la Universidad Vasco de Quiroga, tiene como objetivo fundamental coordinar todos los procesos escolares necesarios para atender de una manera más ágil y eficaz el suministro de los Servicios Escolares al alumno, desde que éste solicita su admisión hasta la culminación de sus estudios, todo bajo los lineamientos y políticas establecidas en la filosofía educativa de la Institución.

Estas herramientas tienen como fin primordial el aprovechamiento de los recursos con que se cuenta y la coordinación de los esfuerzos de sus integrantes, para lograr las metas establecidas, y que permita funcionar institucionalmente de acuerdo a la estructura organizacional del departamento de Control Escolar.

5.1 .- PROCESO DE ADMISIÓN

Este proceso se describe en el diagrama 5.1, en el, detalle lo siguiente :

El aspirante solicita a la ventanilla de Servicios Escolares de la UVAQ los planes de estudio de la licenciatura o del bachillerato que haya elegido, se le proporciona también la solicitud correspondiente, la cual debe regresar debidamente requisitada. Una vez hecho esto, el aspirante debe pagar el derecho al examen de admisión, posteriormente se le proporciona la documentación necesaria y se le invita a escoger una de las fechas disponibles para presentar el examen y por último se le entrega la ficha correspondiente.

Una vez que el aspirante ha realizado los exámenes correspondientes, si el objetivo es ingresar a nivel licenciatura, se le proporcionará una fecha para la entrevista con el director de la escuela a la que haya decidido ingresar. Si en dicha entrevista, el director considera que el aspirante debe ser admitido, se debe concertar una cita para que se realice el examen médico.

Si el aspirante cumple y aprueba todos los exámenes referidos, se le solicita que entregue al departamento de Control Escolar toda la documentación necesaria para lograr su inscripción, posteriormente realiza el pago de la inscripción y por ultimo regresa a la ventanilla de servicios escolares para que se le proporcione la información final necesaria.

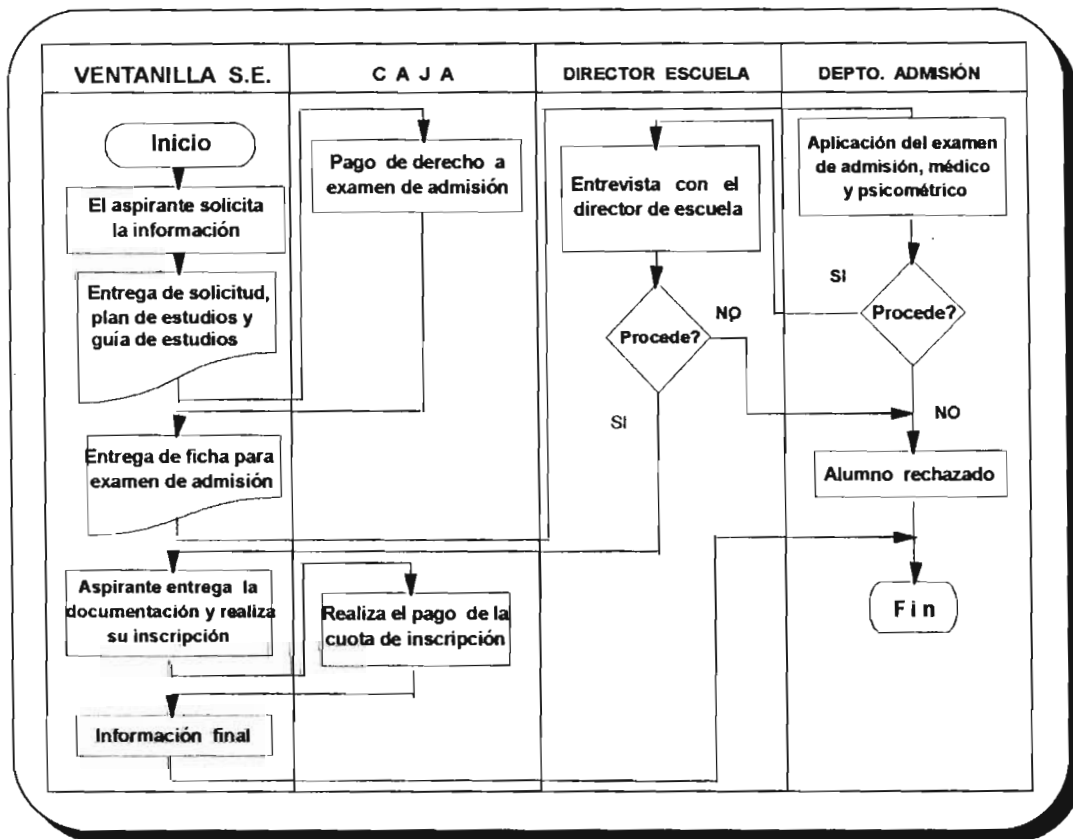


DIAGRAMA DE FLUJO 5.1

5.2 .- PROCESO DE REINGRESO

El alumno, o la persona que va a realizar la operación se ajustará al proceso gráfico del diagrama 5.2, por lo tanto en primer lugar pasa a la ventanilla de información, ahí, se revisa que no se tenga ningún adeudo, tanto económico como de material, en caso de que no se tenga ningún tipo de adeudo, se le entrega su ficha de reinscripción y de indica que pase a caja a realizar el pago correspondiente.

En el caso de que el alumno tenga algún tipo de adeudo, primero debe pasar a caja y liquidar su adeudo y después regresar a caja con su comprobante de pago para poder recibir la ficha de reingreso, para finalmente pasar nuevamente a caja a realizar el pago de reinscripción.

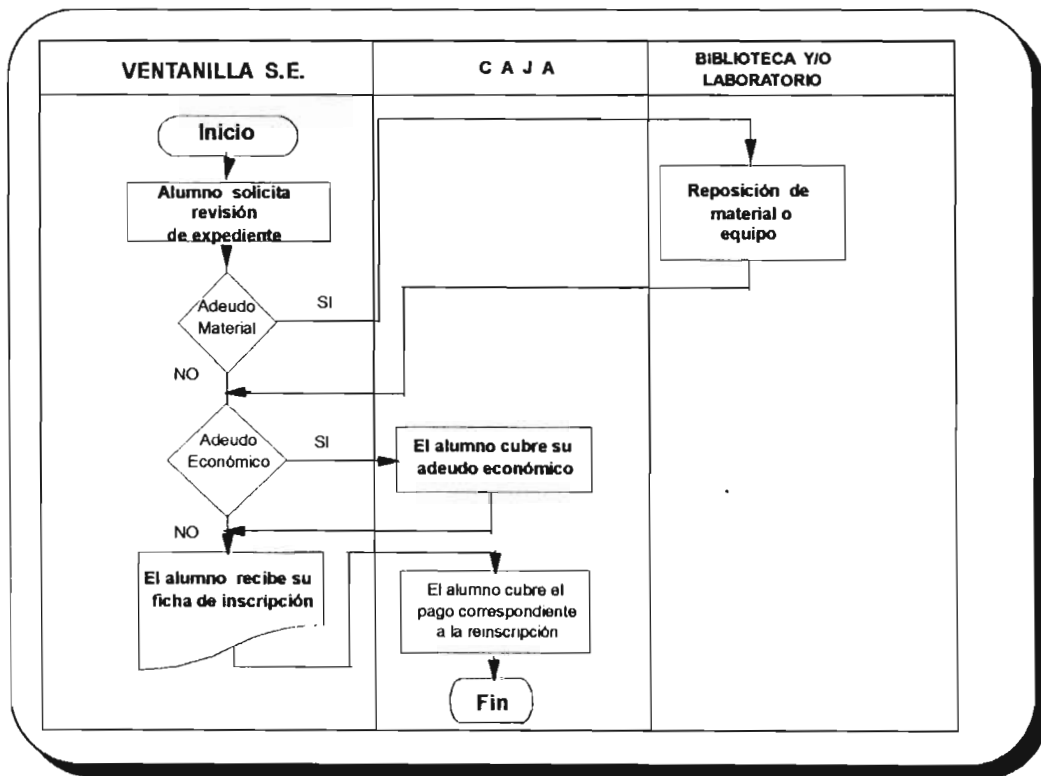


DIAGRAMA DE FLUJO 5.2

5.3 .- REGISTRO DE LA INFORMACIÓN

El proceso descrito anteriormente, es el que tiene que seguir el aspirante para poder ingresar en la UVAQ, pero ahí no termina todo, ahora toca al departamento de Control Escolar realizar el proceso “interno” necesario para poder registrar al alumno. Dicho proceso se describe en el diagrama de flujo 5.3 y se detalla a continuación :

Para dar de alta al aspirante, como alumno de la Universidad, se deben seguir los siguiente pasos : (cabe aclarar que dado que la UVAQ cuenta con licenciaturas registradas en la Secretaria de Educación Publica (SEP.), en la Cd. de México y en la Secretaria de Educación en el Estado (SEE.), ademas de contar con la preparatoria propedeutica y la bivalente, existe un proceso interno en el departamento de Control Escolar para cada caso, sin embargo, únicamente describiré, a manera de ejemplo, el proceso necesario para las licenciaturas registradas en la SEE-Michoacán.)

- ◆ Primeramente se recibe de ventanilla la documentación de los alumnos aceptados para nuevo ingreso.

- ◆ Se verifica si la documentación entregada por el alumno está completa.

- ◆ Toda la información es clasificada “manualmente” por carrera.

- ◆ Se elabora y se manda a la SEP. - Michoacán, el oficio en el cual se solicita el número de control para cada uno de los alumnos de nuevo ingreso. Esto es para cada una de las licenciaturas.

- ◆ Una vez que se cuenta con los números de control, se elaboran manualmente las CFIYAS por carrera y semestre.

- ◆ Las CFIYAS son enviadas al SEE para que sean registradas y certificadas para que de esta forma, el alumno esté dado de alta “oficialmente”.

- ◆ Los alumnos son registrados en el directorio del departamento de Control Escolar y/o capturados sus datos dentro del Sistema de Control Escolar.

- ◆ El Departamento de Control Escolar elabora manualmente el KARDEX para dar inicio al historial académico de cada uno de los alumnos.

- ◆ Se toma la foto al alumno y se manda elaborar su credencial.

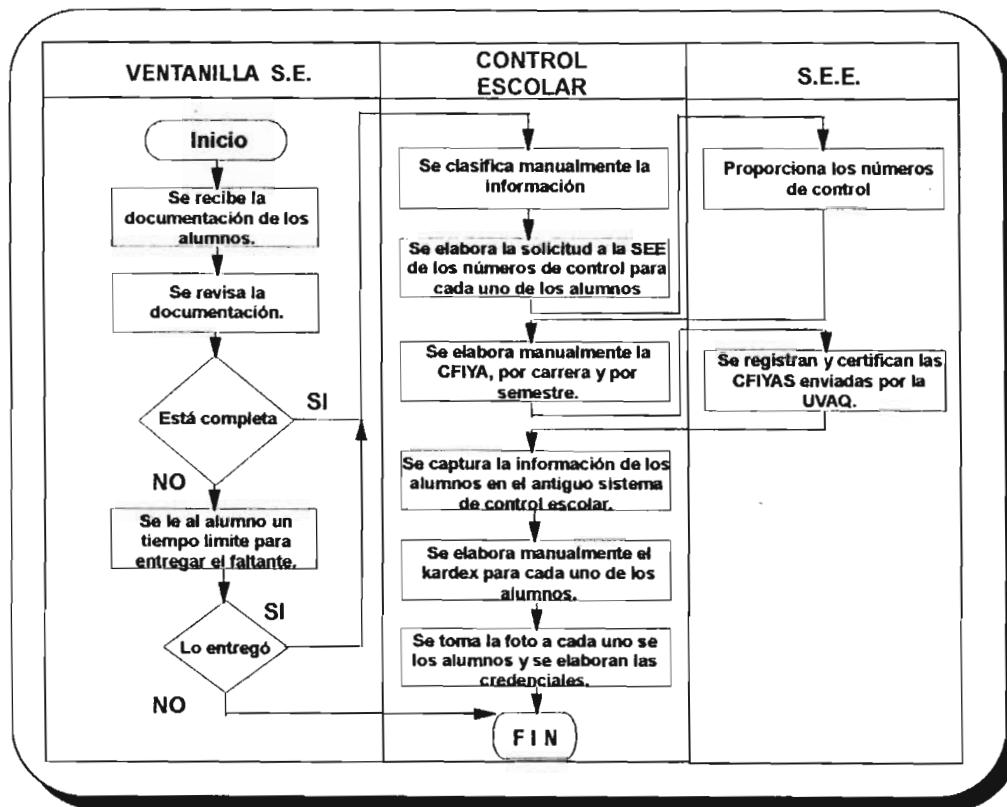


DIAGRAMA DE FLUJO 5.3

5.4 .- FLUJO DE LA INFORMACIÓN

Una vez detallados los procesos necesarios que tiene que seguir un alumno para ser dado de alta en la Universidad, se procederá a explicar en forma mas detallada la forma en que fluye la información dentro de las distintas áreas de la Universidad para que está sea finalmente procesada por el departamento de Control Escolar.

5.4.1.- REGISTRO DE CALIFICACIONES

- ◆ **Preparatoria** .- Una vez que los exámenes son aplicados y evaluados por el profesor son pasadas al coordinador de grado, este a su vez proporciona las calificaciones al departamento de Control Escolar quien las captura en el Sistema de Control Escolar (SCE), y emite las boletas parciales y finales, una vez que se tienen las boletas, estas son turnadas de nueva cuenta al coordinador de grado quien organiza la junta de padres de familia para entregarles la respectiva boleta de calificaciones.
- ◆ **Licenciatura** .- El profesor una vez que ha aplicado el examen, entrega las calificaciones a la dirección de la escuela correspondiente, que es la encargada de concentrar las calificaciones parciales de todo el semestre; al final del semestre la dirección de la escuela elabora un concentrado de calificaciones que es entregado al departamento de Control Escolar para que se capture la información en el SCE y emita las boletas *finales* que son entregadas en la ventanilla de servicios escolares.

5.4.2 .- LISTAS DE ASISTENCIA

- ◆ **Preparatoria** .- Son elaboradas por el departamento de Control Escolar quien las turna al coordinador de grado quien es el encargado de hacer llegar la lista correspondiente a cada uno de los profesores.
- ◆ **Licenciatura** .- Son elaboradas por el departamento de Control Escolar quien las entrega a la dirección de la escuela correspondiente, para que esta sea la responsable de hacerlas llegar a cada uno de los profesores.

5.4.3 .- DOCUMENTOS OFICIALES

- ◆ **Memorándum y/o constancias** .- Cuando un alumno solicita alguno de estos documentos lo hace directamente en “caja” donde se hace entrega del correspondiente recibo de pago y también se le indica el día que puede pasar a recoger el documento solicitado, posteriormente la “cajera” entrega una copia del recibo a Control Escolar quien es el encargado de elaborar el documento, posteriormente, Control Escolar pasa el documento a la ventanilla de servicios, quien es la encargada de entregar el documento al alumno, previa entrega por parte de este del original del recibo de pago.

- ◆ ***Certificados totales*** .- Una vez que el alumno ha concluido sus estudios de bachillerato o licenciatura, el Departamento de Control Escolar elabora los certificados totales para cada uno de los alumnos, los formatos utilizados para la elaboración de los certificados son obtenidos directamente en la Secretaría de Educación Pública. Una vez que las calificaciones son asentadas en el formato, se le adhiere la fotografía y se obtienen las firmas autorizadas por la UVAQ. Posteriormente son entregadas a la S.E.E. para su correspondiente legalización, una vez realizado este proceso la S.E.E. entrega nuevamente los certificados a la UVAQ, quien los registra en el libro del Departamento de Control Escolar para posteriormente poder ser entregados a los alumnos quienes previamente deben haber cubierto el costo del documento.

- ◆ ***Certificados parciales*** .- Es el mismo proceso y trámite que el anterior, con la diferencia de que es el alumno quien solicita este tipo de certificado.

- ◆ ***Títulos***.- Después del examen recepcional que presenta el alumno, el departamento de Control Escolar envía al calígrafo el pergamino con los datos necesarios para la elaboración del título. Una vez elaborado, se revisa, se adhiere la fotografía y se imprime el sello de la Universidad, además se recaban las firmas del jefe del departamento de Control Escolar, del coordinador general académico y del Rector de la Universidad. Posteriormente se lleva a Gobierno del Estado para efectuar el pago correspondiente y se entrega al departamento de Profesiones en la SEE. quienes lo devuelven a la UVAQ debidamente legalizado. Se toman los datos de legalización y se emiten copias fotostáticas que son cotejadas ante el Notario Público, y el título queda para su entrega en el departamento de Control Escolar.

5.5 .- SISTEMA ANTERIOR DE CONTROL ESCOLAR

El sistema anterior de Control Escolar, se empezó a emplear a partir del segundo semestre del ciclo 1990- 1991, capturando información y conociendo el sistema; el siguiente semestre se empleo con información real y ha seguido operando “hasta la fecha”. Este sistema funciona en modo de monousuario y es operado únicamente por una sola persona, la cual es la encargada de capturar y procesar “toda” la información que se genera en “toda” la universidad, esto como puede suponerse, crea un gran “cuello de botella” en lo que a información se refiere, de tal forma que las boletas de licenciatura, por citar un ejemplo, son entregadas a los alumnos hasta iniciado el siguiente semestre, lo cual causa molestias entre el alumnado.

En el siguiente diagrama se resume las funciones que tiene que realizar el capturista encargado del “antiguo” sistema de control escolar.

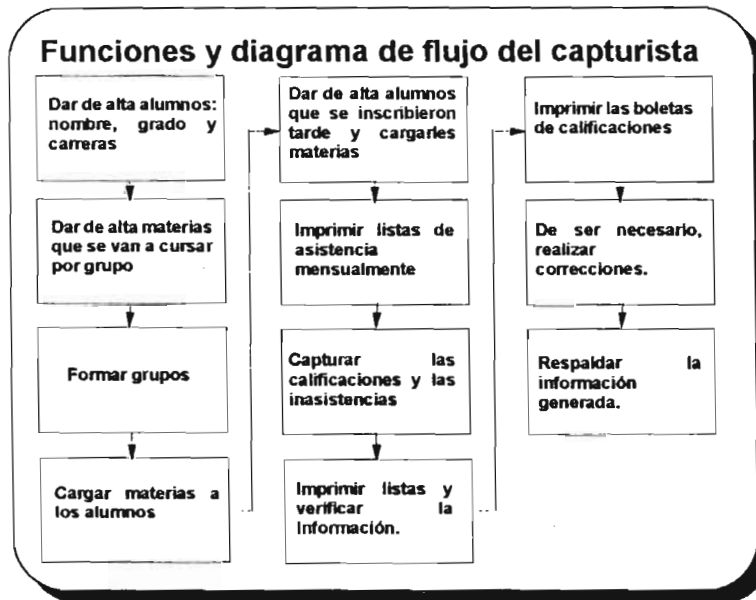


DIAGRAMA 5.3

CAPITULO 6**"TEORÍA GENERAL DE SISTEMAS"****6.1.- DEFINICIÓN DE SISTEMA.**

Sistema.- Es un conjunto de componentes que interrelacionan entre sí para lograr un objetivo común.

Sistema de Información.- Es un conjunto de personas, datos y procedimientos que se interrelacionan con el fin común de apoyar a actividades específicas dentro de una organización.

Sin importar las organizaciones a las que sirve o la forma en la que se desarrollan y diseñan, todos los sistemas de información están compuestos por seis componentes bien definidos : entradas, modelos, salidas, tecnología, base de datos y controles.

Para alcanzar sus objetivos, los sistemas interaccionan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de las fronteras de los sistemas. Los sistemas que interactúan con su medio ambiente se denominan *Sistemas Abiertos*.

En contraste, aquellos que no interactúan con su medio ambiente se conocen como *Sistemas Cerrados*. Todos los sistemas actuales son abiertos y los sistemas cerrados existen sólo como concepto.

La finalidad de los sistemas de información, como las de cualquier otro sistema dentro de una organización, son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas. Los sistemas de información están formados por subsistemas que incluyen hardware, software, medios de almacenamiento de datos y bases de datos.

Hoy en día, buena parte de nuestra sociedad se apoya en la tecnología de sistemas de información, por esto, deben ser capaces, utilizables, confiables y por encima de todo, servir como medios para alcanzar fines sin convertirse en fines por sí mismo.

6.2.- ANÁLISIS Y DISEÑO DE SISTEMAS.

El desarrollo de sistemas puede considerarse, en general, formado por dos grandes componentes: el análisis de sistemas y el diseño de sistemas.

- ◆ El análisis de sistemas, por consiguiente, es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

- ◆ El diseño de sistemas es el proceso de planificar, reemplazar o complementar un sistema organizacional existente. El diseño es el que determina el éxito del sistema.

Para determinar los requerimientos de sistemas, es necesario analizar los hechos que se tienen a la mano. Se estudian con la finalidad de evaluar el funcionamiento del sistema en uso y establecer los requerimientos que debe cumplir un nuevo diseño.

6.2.1.- ESPECIFICACIÓN DE LOS REQUERIMIENTOS DE LA APLICACIÓN.

La determinación de requerimientos es el estudio de un sistema para conocer cómo trabaja y dónde es necesario efectuar mejoras. Un *requerimiento* es una característica que debe incluirse en un nuevo sistema. Es así como la determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados con él.

6.2.1.1.- ACTIVIDADES DE LA DETERMINACIÓN DE REQUERIMIENTOS.

Es útil ver la determinación de requerimientos a través de tres grandes actividades :

- ◆ ***Anticipación de requerimientos.***- La experiencia del analista en un área en particular y el contacto con sistemas en un ambiente similar al que se encuentra bajo investigación, tienen influencia sobre el estudio que éstos realizan. La experiencia permite anticipar ciertos problemas o características y requerimientos para un nuevo sistema.

- ◆ ***Investigación de requerimientos.***- Esta actividad es la más importante del análisis de sistemas. Se estudia el sistema actual con la ayuda de varias herramientas y habilidades a fin de documentar sus características para, más adelante, emprender el análisis.

- ◆ ***Especificaciones de requerimientos.***- Los datos obtenidos durante la recopilación de hechos se analizan para determinar las características del nuevo sistema.

6.2.1.2.- REQUERIMIENTOS BÁSICOS

- ◆ ***Comprensión del proceso.***- Siempre se debe comenzar con lo básico, es decir, tener antecedentes sobre detalles fundamentales relacionados con el sistema y que sirven para describirlo. Algunos de estos detalles son: ¿Cuál es la finalidad del sistema dentro de la empresa?, ¿Qué pasos se siguen para llevarla a cabo?, ¿Quiénes lo realizan?, ¿Cuánto tiempo tardan en efectuarlos?, ¿ Con cuánta frecuencia lo hacen?, etc.

- ◆ ***Identificación de datos empleados e información generada.***- El siguiente paso es detectar qué datos se utilizan para llevar a cabo cada actividad.

- ◆ ***Frecuencia y volumen del proceso.***- Se debe de conocer con la mayor exactitud posible cual es el volumen de información que se procesa y cual es la frecuencia de dicho proceso.

- ◆ ***Identificación de controles.***- Se debe determinar si una actividad se ha llevado a cabo en forma adecuada, y si no es así, determinar la forma de evitar errores en los procesos del sistema.

6.2.1.3.- PERCEPCIÓN DE REQUERIMIENTOS

Comprende la recopilación de hechos y el análisis de los mismos. Un marco de referencia útil para examinar los hechos de un sistema y establecer los requerimientos de los usuarios, incluye las áreas de :

- ◆ **Capacidad** .- La capacidad es la habilidad que tiene el sistema actual formado por personas, equipo, espacio y procedimientos, para alcanzar las metas y objetivos básicos del sistema. Existen signos muy evidentes de la falta de capacidad de un sistema. Por ejemplo, el retraso poco deseable entre el momento en que se presenta un evento y el procesamiento de los datos asociados con él. Se tiene evidencia de un problema de capacidad cuando el sistema no puede manejar las transacciones o no cumple con los requerimientos de procesamiento y tampoco satisface los niveles de rendimiento esperados.

- ◆ **Control** .- Está relacionado con el conjunto de mecanismos que aumentan la probabilidad de que las actividades que afectan los objetivos de una organización se lleven a cabo en forma apropiada. Muchos sistemas hacen hincapié en el control de las tareas, notificando cuándo no se realizan en forma apropiada, es decir, cuando se presenta un error, ya que los datos incorrectos conducen hacia información poco confiable. Desde el punto de vista del diseño de sistemas, las siguientes estrategias se encuentran entre las que deben considerarse cuando se tiene entre manos un problema de control de sistemas :

- *Diseño para evitar fallos en el control* .- Desarrollar el sistema para prohibir la ocurrencia de fallos en los controles y con esto invalidar eventos que no pueden ocurrir.
- *Diseño para detectar y notificar problemas de control* .- Asegurar que sean incorporados en el sistema procedimientos que identifiquen dificultades o inexactitudes y las notifiquen como una excepción que debe ser corregida por la persona autorizada para hacerlo.
- *Diseño para detectar y corregir fallos en los controles*.- Proporcionar medios para emprender una acción correctiva y notificar que ésta ha sido realizada.

Cada estrategia supone, tanto en sentido favorable como desfavorable, que las *personas* que forman parte de la organización son las responsables de lo que ocurre con ella.

- ◆ *Complejidad* .- Requerir que una persona participe en tareas interrelacionadas, es algo que puede crear una complejidad innecesaria. Los procedimientos que contienen un número muy grande de tareas, pasos o actividades, a menudo dan como resultado un rendimiento inaceptable o incompleto. Para reducir la complejidad se deben considerar tres estrategias :

- *Simplificación* .- Puede obtenerse al eliminar los pasos que no son necesarios o el registro de información que no se utiliza.
- *División* .- Dividir un proceso complejo en tareas separadas disminuye con frecuencia la complejidad.
- *Cambios en la secuencia* .- Cambiar el orden en el que ocurre un proceso puede disminuir la complejidad aparente.

6.2.2 .- TÉCNICAS PARA LA DETERMINACIÓN DE REQUERIMIENTOS

- ◆ *Entrevista* .- Se utiliza por lo general para reunir información proveniente de personas o de grupos. Por lo común, los entrevistados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, los entrevistados son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él. Las entrevistas permiten al analista descubrir áreas mal comprendidas, expectativas poco realistas e incluso indicadores de resistencia hacia el sistema propuesto.

Las entrevistas pueden clasificarse como estructuradas o no estructuradas. Las entrevistas no estructuradas utilizan un formato pregunta - respuesta y son apropiadas cuando el analista desea adquirir información general acerca de un sistema. Por otro lado, las entrevistas estructuradas utilizan preguntas estándar en un formato de respuesta abierta o cerrada.

◆ **Cuestionarios** .- Permiten reunir información proveniente relacionada con varios aspectos de un sistema de un grupo grande de personas. El empleo de formatos estandarizados para las preguntas puede proporcionar datos más confiables que otras técnicas. Sin embargo, este método no permite al analista observar las expresiones o reacciones de los encuestados. Asimismo, la respuesta puede ser limitada. Al igual que con las entrevistas, se debe seleccionar a los encuestados. El analista debe asegurar que el conocimiento y experiencia de éstos los califiquen para dar respuesta a las preguntas.

◆ **Revisión de los registros** .- Varios tipos de registros y reportes pueden proporcionar información valiosa con respecto a la organización, su operación, la información que maneja y como la maneja.

◆ **Observación** .- Por medio de la observación se obtiene información de primera mano sobre la forma en que se efectúan las actividades. Este método es más útil cuando se necesita observar, por un lado, la forma en que se manejan los documentos y se llevan a cabo los procesos y por otro, para determina si se siguen todos los pasos especificados.

Aunado a todas estas técnicas está la *participación de los usuarios*, ya que está proporciona una retroalimentación que es importante a medida que avanza el diseño. Existen tres razones para que los usuarios participen en el diseño de un sistema.

- La primera de ellas es que el sistema pertenece a los usuarios no a los analistas.
- La segunda razón es la detección de omisiones y de características innecesarias o difíciles de emplear.
- La tercera razón es que cualquier error tanto en los procedimientos manuales como computarizados, almacenes de datos y otras características de diseño, deben detectarse lo más pronto posible.

Asimismo, cuando los usuarios participan en el desarrollo de sistemas, tienen la responsabilidad de conocer los requerimientos de sistemas. Ellos deben comprender el problema que están abordando al desarrollar un sistema y los datos necesarios para enfrentarse a él. Los usuarios necesitan también un conocimiento genérico de los requerimientos de datos deben ser capaces de identificar los datos en los términos de la compañía. Además se espera de ellos que sepan operar un equipo de cómputo, tal como la terminal donde se ejecuta la aplicación o la computadora de escritorio que utilizarán.

6.2.3 .- OBJETIVOS AL DISEÑAR UN SISTEMA DE INFORMACIÓN

◆ *Especificaciones de los elementos lógicos del diseño* .- El diseño de sistemas tiene dos etapas: el diseño lógico y la construcción física del sistema. Cuando el analista formula el *diseño lógico*, escribe las especificaciones detalladas del nuevo sistema, es decir aquellas que describen sus características: salidas, entradas, archivos y bases de datos y los procedimientos. El conjunto formado por todas estas características recibe el nombre de *especificaciones de diseño* del sistema.

El diseño lógico también especifica los formatos de entrada y la distribución de la salida en pantalla para todas las transacciones y archivos que son necesarios para dar mantenimiento a los datos, a los detalles de las transacciones y a los datos.

La *construcción física*, que es la siguiente actividad después del diseño lógico, produce el software y un sistema que funciona. Las especificaciones de diseño indican a los programadores lo que el sistema debe hacer. Durante la construcción física, los programadores *escriben* las instrucciones necesarias para elaborar el programa de software.

♦ ***Apoyo para las actividades de la empresa*** .- Un objetivo fundamental en el diseño de un sistema de información es asegurar que brinde apoyo a la actividad de la empresa para la que fue desarrollado.

♦ ***Asegurar que las características del sistema cumplan con los requerimientos del usuario*** .- Durante el diseño, los requerimientos del usuario se trasladan en características del sistema. Se dice que un sistema de información satisface las necesidades de los usuarios si :

- Realiza en forma apropiada los procedimientos correctos.
- Presenta información e instrucciones en forma aceptable y efectiva.
- Produce resultados exactos.
- Proporciona una interfase y métodos de interacción aceptables.
- Es percibido por los usuarios como un sistema confiable.

◆ **Proporcionar un sistema que sea fácil de utilizar** .- Es necesario diseñar el sistema con ingeniería dirigida hacia las personas y en consecuencia incluir características ergonómicas.

- **Ingeniería humana** .- Si los sistemas de información no son diseñados para las personas entonces no tendrán éxito. El analista debe procurar formular el diseño de sistemas en forma que :

- Incorpore características del sistema que sean fáciles de comprender y utilizar.
- Desaliente los errores cometidos por los usuarios.
- Se adapte a la creciente familiaridad del usuario con el sistema.
- Funcione, en general, en una forma que parezca natural al usuario.

- **Diseño ergonómico** .- En el contexto de los sistemas de información, la *ergonomía* estudia los factores físicos que afectan el rendimiento, la comodidad y la satisfacción de los usuarios directos. Se deben considerar factores ergonómicos cuando se escogen los colores para la presentación de información, la ubicación de las teclas de mandatos o los métodos de interacción con el sistema.

◆ **Ajustarse a los estándares de diseño** .- Los siguientes son ejemplos de áreas incluidas en el diseño de estándares :

-
- *Estándares para datos*.- Lineamientos para asignar nombre a los datos y especificar su longitud y tipo.
 - *Estándares de codificación* .- Abreviaturas y designaciones formales para describir actividades y entidades dentro de la organización. Por ejemplo, al escoger el nombre de las funciones o procedimientos utilizados en el programa.
 - *Estándares estructurales* .- Políticas para dividir el software en módulos, para la codificación estructurada y la relación existente entre los componentes del sistema.

6.2.4 .- ESPECIFICACIÓN DE DISEÑO

A continuación se describen las características que deben ser diseñadas y los elementos que deben tomarse en cuenta en las especificaciones formales de diseño.

- ◆ *Diseño de la salida* .- Se refiere a los resultados e información generados por el sistema. Para muchos usuarios finales la salida es la única razón para el desarrollo del sistema y la base sobre la que ellos evaluarán la utilidad de la aplicación. Cuando se diseña la salida, se debe considerar lo siguiente :

-
- Determinar qué información presentar
 - Decidir si la información será presentada en forma visual, verbal o impresa y seleccionar el medio de salida.
 - Disponer la presentación de la información en un formato aceptable.
 - Decidir cómo distribuir la salida entre los posibles destinatarios,

La disposición de la información sobre una pantalla o documento impreso de denomina distribución.

Los objetivos de la salida son :

- Expresar información relacionada con actividades pasadas.
- Señalar eventos importantes, problemas o advertencias.
- Iniciar una acción.
- Confirmar una acción.

Las salidas pueden ser :

- Un reporte.
- Un documento.
- Un mensaje.

◆ **Diseño de archivos** .- Entre las decisiones que se toman durante el diseño de archivos, se encuentran las siguientes :

- Los datos que deben incluirse en el formato de los registros contenidos en el archivo.
- La longitud de cada registro, con base en las características de los datos que contiene.
- La secuencia a disposición de los registros dentro del archivo.

Cabe aclarar que no todos los nuevos sistemas de información requieren del diseño de todos los archivos utilizados por la aplicación. Así mismo, algunos de los puntos anteriores no son aplicables para todos los modelos de archivos de datos. Este tema será tratado con mayor detalle en capítulos posteriores.

◆ **Diseño de la entrada** .- Se deben determinar los siguientes puntos para el diseño de entradas :

- ¿ Qué datos ingresan al sistema ?
- ¿ Qué medios utilizar ?
- La forma en que se deben disponer o codificar los datos.
- El diálogo que servirá de guía a los usuarios para dar entrada a los datos.
- Validación necesaria de datos y transacciones para detectar errores.
- Métodos para llevar a cabo la validación de las entradas para evitar errores.

El diseño de la entrada también incluye la especificación de los medios por los que tanto los usuarios finales como los operadores darán instrucciones al sistema sobre las acciones que deben emprender. Por ejemplo, un usuario que interactúa con el sistema por medio de una estación de trabajo, tiene que ser capaz de indicarle al sistema ya sea que acepte una entrada, genere un reporte o termine el procesamiento.

◆ **Diseño de controles** .- Se deben anticipar los errores que se cometerán al ingresar los datos en el sistema o al solicitar la ejecución de ciertas funciones. Algunos errores no tienen importancia ni consecuencias, pero otros pueden ser tan serios que ocasionarían el borrado de datos o el uso inapropiado del sistema. Aunque exista sólo la más mínima probabilidad de cometer un error serio, un buen diseño de sistema de información ofrecerá los medios para detectar y manejar el error. Los controles de entrada proporcionan medios para :

- Asegurar que sólo los usuarios autorizados tengan acceso al sistema.
- Garantizar que las transacciones sean aceptables.
- Validar los datos para comprobar su exactitud.
- Determinar si se han omitido datos que son necesarios.

◆ **Diseño de procedimientos** .- Los procedimientos especifican qué tareas deben efectuarse al utilizar el sistema y quiénes son los responsables de llevarlas a cabo. Entre los procedimientos importantes se encuentran :

-
- *Procedimientos para la entrada de datos* .- Métodos para la captura de datos de las transacciones y su ingreso al sistema.
 - *Procedimientos durante la ejecución* .- Pasos y acciones emprendidos por los operadores del sistema, como por ejemplo, el colocar en las impresoras las forma preimpresas.
 - *Procedimientos para el manejo de errores*.- Acciones a seguir cuando se presentan resultado inesperados
 - *Procedimientos de seguridad y respaldo*.- Acciones para proteger al sistema y sus recursos contra posibles daños.

6.3 .- CICLO DE VIDA CLÁSICO DEL DESARROLLO DE SISTEMAS

Es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. El método del ciclo de vida, consta de las siguientes actividades :

◆ *Investigación preliminar*

Se inicia siempre por la petición de una persona. Esta actividad tiene tres partes :

- *Aclaración de la solicitud* .- La solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea.
- *Estudio de factibilidad* .- Es sumamente importante determinar si el sistema es factible tanto técnica, económica y operacionalmente hablando.
- *Aprobación de la solicitud* .- Después de aprobar la solicitud de un proyecto se estima su costo, el tiempo necesario para terminarlo y las necesidades de personal; con esta información se determina finalmente si el sistema propuesto se desarrollará.

◆ *Determinación de los requerimientos del sistema*

Es en esta etapa donde se debe estudiar con detenimiento y contestar a las siguientes preguntas con respecto al sistema :

- ¿ Qué es lo que hace ?
- ¿ Como se hace ?
- ¿ Con que frecuencia se presenta ?
- ¿ Qué tan grande es el volumen de transacciones ?
- ¿ Cuales son los datos que se necesitan ?
- ¿ Qué salidas de información son necesarias ?
- ¿ Qué reporte impresos se generan y con que periodicidad ?
- etc.

Para ello se deben establecer pláticas con las personas involucradas directamente con el sistema propuesto, para reunir detalles relacionados con los procesos de la empresa; cuando no es posible tener un contacto directo con estas personas generalmente se utilizan cuestionarios escritos. Así mismo, las investigaciones detalladas requieren el estudio de manuales y reportes del sistema que está en operación actualmente ya sea manual o automático, la observación en condiciones reales de las actividades del trabajo y en algunas ocasiones, muestras de formas y documentos con el fin de comprender el proceso en su totalidad. Conforme se reúnen los detalles, se estudian los datos sobre requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema.

◆ *Diseño del sistema*

En esta etapa, es donde se producen los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Se comienza el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior, se determinan con toda precisión los datos específicos para cada reporte y salida, también se determinan los datos de entrada, aquellos que serán calculados y los que deben ser almacenados, se seleccionan las estructuras de archivos y los dispositivos de almacenamiento.

◆ *Diseño del software*

En esta etapa es donde se lleva a cabo la codificación del sistema, ayudándose para ello de lenguajes de programación; estos lenguajes deben ser escogidos cuidadosamente, tomando en cuenta tanto los requerimientos del sistema, como las estructuras de los archivos y la plataforma en que se va a utilizar el sistema.

◆ *Prueba del sistema*

El sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo de acuerdo con las especificaciones iniciales. Se alimenta con datos de prueba y después de analizan los resultados.

◆ *Implantación y evaluación*

En este punto es donde se instala el nuevo sistema, se capacita a los usuarios y se pone a funcionar con datos reales. Dependiendo del tamaño de la organización que empleara la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema sólo en un área de la empresa o por una o dos personas. Algunas veces es conveniente que los dos sistemas (el antiguo y el nuevo), trabajen en forma paralela con la finalidad de comparar los resultados y en caso de estos ser idénticos, abandonar definitivamente el sistema anterior. En otras circunstancias, el sistema viejo deja de utilizarse determinado día para comenzar a emplear el nuevo sistema al día siguiente.

◆ *Mantenimiento del sistema*

Es indudable que debe darse mantenimiento a las aplicaciones; realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las nuevas necesidades de los usuarios. Dado que los sistemas de las organizaciones junto con el ambiente de las empresas experimentan cambios de manera continua, los sistemas de información deben mantenerse siempre al día.

CAPITULO 7

“REDES INFORMÁTICAS”

7.1 .- ¿POR QUÉ UNA RED?

Generalmente cuando se trabaja en una computadora personal, el principal problema es el de la comunicación y compartición de la información, ya que esta tiene que ser proporcionada haciendo uso de métodos y medios externos como son los diskettes, cintas magnéticas, papel, etc.. Esto hace que los datos nunca puedan ser compartidos *en línea*, y que cualquier operación que se desee realizar sobre los mismos se tenga que hacer de forma diferida, disminuyendo la eficiencia en el manejo de información. Otro de los grandes problemas es el hecho de cada PC debe de contar por separado con todos los periféricos, como pueden ser : discos duros, impresoras, módems, etc. acarreado con esto un alto costo.

7.2 .- DEFINICIÓN DE RED INFORMÁTICA

Se puede establecer a continuación una *definición* de red de área local :

- ♦ **Es un conjunto de elementos físicos y lógicos que proporcionan interconexión a una gran variedad de dispositivos de comunicación de información en un área privada restringida.**

Las redes de área local (LAN por sus siglas en ingles), vienen en gran medida a solucionar los problemas relacionados con la compartición de información y de recursos.

Las LAN están basadas en el concepto de arquitectura cliente-servidor. Es decir, que las PC conectadas a la red pueden ser de dos tipos:

- Las que proporcionan o “sirven”, periféricos, aplicaciones, datos, etc.
- Las que hacen uso de los recursos, a las que de ahora en adelante denominaremos “Estaciones de Trabajo”.

En cualquier caso, las PC’s son del tipo “inteligente”, es decir, cuentan y trabajan con su propia memoria, sus periféricos particulares y en ocasiones, con su disco duro local.

7.3.- COMPONENTES DE UNA RED

Una red de computadoras está compuesta tanto por hardware como por software. El hardware incluye tanto las tarjetas de interfaz de red como los cables que las unen, y el software incluye los controladores (programas que se utilizan para gestionar los dispositivos periféricos) y el sistema operativo de red que gestiona la red. A continuación se listan y describen los componentes.

- ◆ **Servidor.-** El servidor ejecuta el sistema operativo de red y ofrece los servicios de red a las estaciones de trabajo. Entre estos servicios se incluyen el almacenamiento de archivos, la gestión de usuarios, la seguridad, las órdenes y opciones para usuarios de red, las órdenes del responsable de la red y otros.

- ◆ **Estaciones de trabajo.**- Cuando una computadora se conecta a una red, la primera se convierte en un nodo de la última, y se puede tratar como una *estación de trabajo* o *cliente*. Las estaciones de trabajo pueden ser computadoras personales con el DOS, sistemas Macintosh de Apple, sistemas basados en UNIX, sistemas con el OS/2 o estaciones de trabajo sin disco.

- ◆ **Tarjetas de interfaz de red.**- Toda computadora que se conecte a una red necesita de una tarjeta de interfaz de red que soporte un esquema de red específico, como Ethernet, ArcNet o Token Ring. El cable de la red se conectará a la parte trasera de la placa. También están disponibles redes sin cables por radio o infrarrojos.

- ◆ **Sistema de cableado.**- El sistema de cableado de la red está constituido por el cable utilizado para conectar entre sí el servidor y las estaciones de trabajo. En el caso de las redes sin cable que utilizan la radio o los infrarrojos no es necesario.

- ◆ **Recursos y periféricos compartidos.**- Entre los recursos compartidos se incluyen los dispositivos de almacenamiento ligados al servidor, las unidades de disco óptico, las impresoras, los trazadores y el resto de equipos que puedan ser utilizados por cualquier estación en la red.

7.4.- CARACTERÍSTICAS DE LAS REDES DE INFORMACIÓN

Las *características* más representativas de una red de área local son las siguientes :

- ◆ *Alcance.*- El área de conexión se limita a una extensión moderada, generalmente desde unos metros a unos pocos kilómetros.
- ◆ *Velocidad de transmisión.*- En estas redes, la velocidad es elevada en comparación con otros circuitos de comunicación.
- ◆ *Conectividad.*- Además de que todos los dispositivos conectados a una red de área local puedan comunicarse entre sí, también se incluye la capacidad de conexión con otras redes locales o de área extensa.
- ◆ *Propiedad privada.*- Todos los elementos de la red son propiedad de la organización o empresa.
- ◆ *Compartición de recursos.*- Permiten la integración en la misma red de una gran diversidad de dispositivos. Los recursos de almacenamiento, las impresoras y los elementos de comunicación pueden ser utilizados por todas las estaciones de la red.

7.5 .- VENTAJAS DE LAS REDES DE INFORMACIÓN

Las *ventajas* mas significativas que proporcionan las LAN son :

- ◆ *Recursos compartidos.*- Los dispositivos conectados a la red comparten datos, aplicaciones, periféricos y elementos de comunicación.

- ◆ *Conectividad a nivel local.*- Los distintos equipos que integran la red se encuentran conectados entre sí con posibilidades de comunicación.

- ◆ *Proceso distribuido.*- Las LAN permiten el trabajo distribuido, es decir, cada equipo puede trabajar independientemente o cooperativamente con el resto.

- ◆ *Flexibilidad.*- Una LAN puede adaptarse al crecimiento cuantitativo referido al número de equipos conectados, así como adaptarse a cambios cualitativos de tipo tecnológico.

- ◆ *Optimización.*- Las LAN permiten la máxima flexibilidad en la utilización de recursos, estén éstos en la computadora central o en las estaciones de trabajo.

7.6.- DESVENTAJAS DE LAS REDES DE INFORMACIÓN

Entre las *desventajas* están las siguientes :

- ◆ *Interoperatividad.*- La carencia de estándares bien definidos entre los datos que producen las aplicaciones, hace que una red local no garantice que dos dispositivos conectados a ella, funcionen correctamente entre sí al comunicar aplicaciones de distinta naturaleza.

- ◆ *Gestión de la red.*- Por la naturaleza distribuida de una red, la gestión en cuanto a control de accesos, rendimiento y fiabilidad es más compleja.

- ◆ *Integridad, seguridad y privacidad de la información.*- En todo sistema distribuido pueden surgir problemas de este tipo. Este trabajo le concierne casi en la totalidad al sistema operativo de red.

7.7 .- EL MEDIO DE TRANSMISIÓN

Uno de los primeros elementos a tomar en cuenta al instalar una red es el medio físico por donde circulara la información. Los principales medios utilizados en LAN son :

- ◆ *Cable de par trenzado.*- Está constituido por un par de hilos de cobre aislados por medio de un material plástico, trenzados o torsionados entre sí con el fin de disminuir posibles interferencias. En la actualidad tienen una gran utilización en las redes de área local, habiendo sido el principal medio en las comunicaciones telefónicas. Dentro de sus ventajas están su bajo costo, pero una gran desventaja es el hecho de que no ofrece confiabilidad de transmisión a largas distancias.

- ◆ *Cable coaxial.*- Consta de un par de conductores de cobre o aluminio, formando uno de ellos un alma central, rodeado y aislado del otro mediante pequeños hilos trenzados o una lámina metálica cilíndrica. Este tipo de cables goza de ventajas frente a los anteriores puesto que ofrecen una mayor confiabilidad de transmisión, además de una mayor rapidez.

- ◆ *Fibra óptica.*- El núcleo está formado por un pequeño hilo de vidrio o plástico transparente capaz de conducir en su interior un rayo óptico. Entre sus ventajas están la alta velocidad de transmisión, el total aislamiento a las interferencias externa, su principal desventaja es su alto costo.

7.7.1 .- DISTANCIA MÁXIMA ENTRE TERMINALES.

En la tabla 7.1 se listan los distintos tipos de topologías de red y la longitud máxima de los segmentos de cable correspondiente.

Topologías de red	Distancia máxima del segmento
Ethernet gruesa (10BASE-5)	500 Mts.
Ethernet fina (RG-58)	185 Mts.
Ethernet de par trenzado (10BASE-T)	100 Mts.
Ethernet de fibra óptica	2 Kms.
Token Ring de par trenzado	100 Mts.
ArcNet de coaxial (estrella)	609 Mts.
ArcNet de coaxial (bus)	305 Mts.
ArcNet de par trenzado (estrella)	122 Mts.
ArcNet de par trenzado (bus)	122 Mts.

TABLA 7.1

7.7.2 .- COMPARACIÓN DE LOS MEDIOS DE TRANSMISIÓN

En la tabla 7.2 se hace una comparación entre las características de los cables.

Variable	Par Trenzado	Coaxial	Fibra Óptica
Coste	Bajo	Moderado	Alto
Ancho de banda	Moderado	Alto	Muy alto
Longitud	Sobre 30 Mts.	Sobre 305 Mts.	Miles
Interferencia	Alguna	Baja	Ninguna
Fiabilidad	Alta	Alta	Muy Alta

TABLA 7.2

7.8 .- TOPOLOGÍA DE RED

Se le llama topología al modo tanto físico como lógico, que tenemos de conectar los distintos componentes de la red.

Existen varias topologías, sin embargo aquí solo mencionare algunas de ellas, ya que actualmente son las más extendidas en el medio informático corporativo a nivel pequeña, mediana y gran empresa.

7.8.1 .- TOPOLOGÍA EN ANILLO O TOKEN RING

Esta topología fue desarrollada e implementada por IBM Corp. y consiste en una serie de repetidores conectados entre sí mediante un único enlace de transmisión unidireccional que configura un camino cerrado. La información se transmite secuencialmente de un repetidor al siguiente a lo largo del anillo, de tal forma que cada repetidor generara la señal que recibe y la transmite al siguiente, salvo que la información esté dirigida a él, en cuyo caso la recibe en su memoria. A la técnica de acceso que ofrece mayor confiabilidad en este tipo de redes se le conoce como pase de testigo en bus (Token-Bus), en esta, cada estación conoce la identidad de su estación antecesora y de su sucesora dentro del anillo. La estación que recibe el testigo tiene garantizado el derecho de acceso al medio de transmisión de tal forma que al finalizar su transmisión o terminar el tiempo asignado, pasará el testigo a la siguiente estación en la secuencia lógica. Las estaciones que no precisen transmitir pueden quedar excluidas de la secuencia lógica aunque siguen estando conectadas a la red con capacidad de recepción.

7.8.2.- TOPOLOGÍA EN BUS

En esta topología todas las estaciones se conecta a un único medio bidireccional lineal o bus con puntos de terminación bien definidos. Cuando una estación transmite, su señal se propaga a ambos lados del emisor, a través del bus, hacia todas las estaciones conectadas al mismo. La mayor parte de los elementos de las redes en bus tienen la ventaja de ser elementos pasivos, es decir, todos los componentes activos se encuentran en las estaciones por lo que una avería en una estación no afecta más que a ella misma. Por otra parte, un problema originado en el bus, afecta inmediatamente a toda la red.

7.8.3.- TOPOLOGÍA PUNTO A PUNTO

Constituyen este tipo de red las conexiones exclusivas entre terminales y computadora con una línea directa. La ventaja de este tipo de conexión se encuentra en la alta velocidad de transmisión que soporta y la seguridad que presenta al no existir conexiones con otros usuarios. El inconveniente aparece en el precio, puesto que todo él recae sobre un solo usuario.

7.8.4.- TOPOLOGÍA EN ESTRELLA

En esta red, todos los nodos se encuentran conectados a un nodo o servidor central que actúa como coordinador de todas las transmisiones que se producen. Este nodo central asume el papel más importante en el conjunto ya que por él pasan todos los paquetes que se envían a la red. Tiene el inconveniente que se puede producir la paralización total de la red en caso de fallo del nodo central.

CAPITULO 8

“ARCHIVOS Y BASES DE DATOS”

8.1.- INTRODUCCIÓN.

El tratamiento de la información de un modo adecuado y cada vez en mayores proporciones ha conducido con el tiempo a la unificación de archivos, a fin de evitar el problema de las redundancias; es decir, de las repeticiones de la información contenida en archivos distintos. Esta solución ha generado, a su vez, ciertos problemas nuevos y agravando algunos de los ya existentes. Las técnicas informáticas mediante las cuales se intenta tratar este conjunto de problemas son las llamadas técnicas de bases de datos.

8.2.- TERMINOLOGÍA BÁSICA DE ARCHIVOS.

A continuación se definen los conceptos de los términos que se ven involucrados en el diseño de un registro o archivo.

◆ *Datos*

Se le llama dato a los elementos individuales de que esta formado un archivo, a estos también se les conoce como “campos” (de ahora en adelante sustituiré el termino dato por el de campo cuando me refiera a archivos). Cada campo se identifica por su nombre y tiene un valor específico asociado a él.

◆ **Registro**

Es un conjunto de campos relacionados entre si y pertenecientes a una entrada. Cuando el número y tamaño de los datos en un registro son constantes para cada registro, éste se denomina de longitud fija y a los registros donde la longitud de sus campos es relativa a la información que van a almacenar se les conoce como registros de longitud variable.

◆ **Llave de registro**

Es un campo que por sus características es capaz de identificar de manera única un registro dentro del archivo, este campo ya forma parte del registro y no es un dato adicional que existe sólo con los propósitos de identificación.

◆ **Entidad**

Es cualquier persona, lugar, cosa o evento de interés para la organización y acerca del cual se capturan, almacenan o procesan datos. En el caso específico del Sistema de Control Escolar, sus entidades básicas son los alumnos, profesores, materias y grupos con que cuenta la Universidad.

◆ **Archivo**

Es una colección de registros relacionados. Un archivo es la forma más alta de almacenamiento, ya que éste está constituido por una serie de campos, que a su vez forman registros, el tamaño del archivo se determina por el número de registros que hay en él. Algo muy importante es el incluir cada registro en un archivo que pertenezca a la misma entidad.

8.3 .- CONCEPTO DE BASE DE DATOS

Existen realmente una gran cantidad de definiciones de base de datos, sin embargo todas ellas coinciden en ciertos elementos, aquí expongo dos de éstas definiciones :

◆ *Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resultan independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos y para modificar o extraer los datos almacenados.* ¹

◆ *Colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es, también, un modelo del mundo real y como tal, debe poder servir para toda una gama de usos y aplicaciones.* ²

La principal diferencia entre una base de datos y un archivo es el hecho de que en una base de datos los registros para distintas entidades se almacenan comúnmente, mientras que los archivos almacenan registros para una única entidad; es decir, los registros referentes a distintas entidades de interés pueden almacenarse dentro de una base de datos.

1.- Martin, 1975.

2.- Conference des Statisticiens Européens, 1977.

Además las bases de datos se distinguen también de los archivos maestros comunes y de transacciones en tres formas :

- ◆ Un archivo está destinado al almacenamiento. Los registros que contienen se almacenan juntos y son recabados por medio de un método de acceso (secuencial o aleatorio).

- ◆ Agregar registros a un archivo para agrandarlo no lo convierte en una base de datos. La existencia de una base de datos no está determinada por el número de registros almacenados.

- ◆ El hecho de contar con bases de datos no elimina la necesidad de archivos en un sistema de información.

8.3.1 .- CONCEPTO DE SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS

Un Sistema de Administración de Base de Datos (SABD), es un sistema que posibilita la creación, empleo y mantenimiento de bases de datos.

8.3.2 .- CARACTERÍSTICAS DE LOS SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS (SABD).

- ◆ Una base de datos debe ser conocida por el SABD, de manera que aquélla pueda quedar bajo el control del SABD y emplearse para el procesamiento. Con el fin de elaborar una base de datos, es menester definir sus especificaciones valiéndose de un lenguaje de definición de datos que sea parte del SABD.

- ◆ La peculiaridad de estructura de datos de un SABD permite a los usuarios utilizar datos sin tener que preocuparse de cómo se encuentren almacenados estos. Los administradores de la bases de datos pueden estructurar y/o organizar los datos de acuerdo con las necesidades.

- ◆ Mediante la independencia de los datos, éstos se mantienen separados de los programas que los procesan. Así que los datos pueden variar sin que necesariamente se tengan que efectuar cambios en los programas que los emplean, es decir, los datos pueden hacerse disponibles para atender a varios usuarios con diferentes necesidades, pero sin alterar la estructura actual.

- ◆ Un administrador de base de datos puede elegir estructurar los datos lógicamente en cierta forma. De esta manera los datos pueden almacenarse con diferentes *estructuras de almacenamiento*, como pueden ser : secuencial, aleatoria, indexada.

En el caso específico del Sistema de Control Escolar, se optó por la utilización del lenguaje Clipper Ver. 5.2 como Sistema de Administración de Base de Datos, para la codificación del Sistema, ya que este cumple con estas y otras características esenciales.

8.4.- MODELOS DE DATOS

Cuando se diseña una base de datos, es importante conocer el modelo de datos subyacente al modelo del sistema de administración de bases de datos que será utilizado.

8.4.1- DEFINICIÓN DE MODELOS DE DATOS

Un *modelo de datos* es un conjunto de conceptos, reglas y convenciones que nos permiten describir los datos de las aplicaciones.

Por lo que respecta a la relación entre los modelos y los lenguajes de datos, hay que destacar que los modelos son la base para los lenguajes de datos, aunque el nivel de abstracción de estos últimos es menor.

8.4.2.- OBJETIVOS DE UN MODELO DE DATOS

Los objetivos que persigue todo modelo de datos son :

- ◆ *Formalización* .- Ya que el modelo de datos permite definir formalmente las estructuras permitidas y las restricciones a fin de representar los datos y también porque establece la definición de un lenguaje de datos.

- ◆ *Diseño*.- Es uno de los elementos básicos para el desarrollo de una metodología de diseño de base de datos, permitiendo, además, prever el impacto de los cambio en un sistema de información.

8.4.3.- MODELOS DE DATOS RELACIONAL

En la actualidad el modelo de datos mas ampliamente utilizado en las empresas es el relacional, además, es el modelo que soportan la mayoría de los Sistemas de Administración de Bases de Datos (SABD), modernos como son, Clipper, FoxPro, Visual FoxPro, Dbase, Paradox, Approach, Etc.

En el modelo relacional los datos se estructuran lógicamente en forma de tablas, siendo un objetivo fundamental del modelo el mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a otras características de tipo fisico.

8.4.3.1- OBJETIVOS DEL MODELO RELACIONAL

- ◆ *Independencia física* .- El modo en que se almacenan los datos no debe influir en su manipulación lógica

- ◆ *Independencia lógica*.- El añadir, eliminar o modificar objetos de la base de datos no debe repercutir en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismo.

- ◆ *Flexibilidad*.- En el sentido de poder representar a cada usuarios los datos de la forma en que éste prefiera.

- ◆ *Uniformidad*.- Facilita la concepción y manipulación de la base de datos por parte de los usuarios.

- ◆ *Sencillez*.- El modelo debe ser fácil de comprender y de utilizar por parte del usuario final.

Para producir los objetivos citados, se introduce el concepto de **relación** (tabla), como estructura básica del modelo. Formalmente, una relación es un conjunto de filas en la terminología relacional.

8.4.3.2.- ESTRUCTURA DEL MODELO DE DATOS RELACIONAL

Como se señaló anteriormente, la relación es el elemento básico del modelo relacional y se puede representar como una tabla (*Ver tabla 8.1*). En ella podemos distinguir un conjunto de columnas, denominadas atributos (nombre de campo), que representan propiedades de la misma y que están caracterizadas por un nombre, y un conjunto de tuplas (registros), que son las ocurrencias de la relación. El número de filas de una relación se denomina **cardinalidad**, mientras que el número de columnas es el **grado**. Existen también **dominios** de donde los atributos toman sus valores.

atributo 1	atributo 2	atributo n
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx

▶ **tupla 1**
 ▶ **tupla 2**
 ▶ **tupla n**

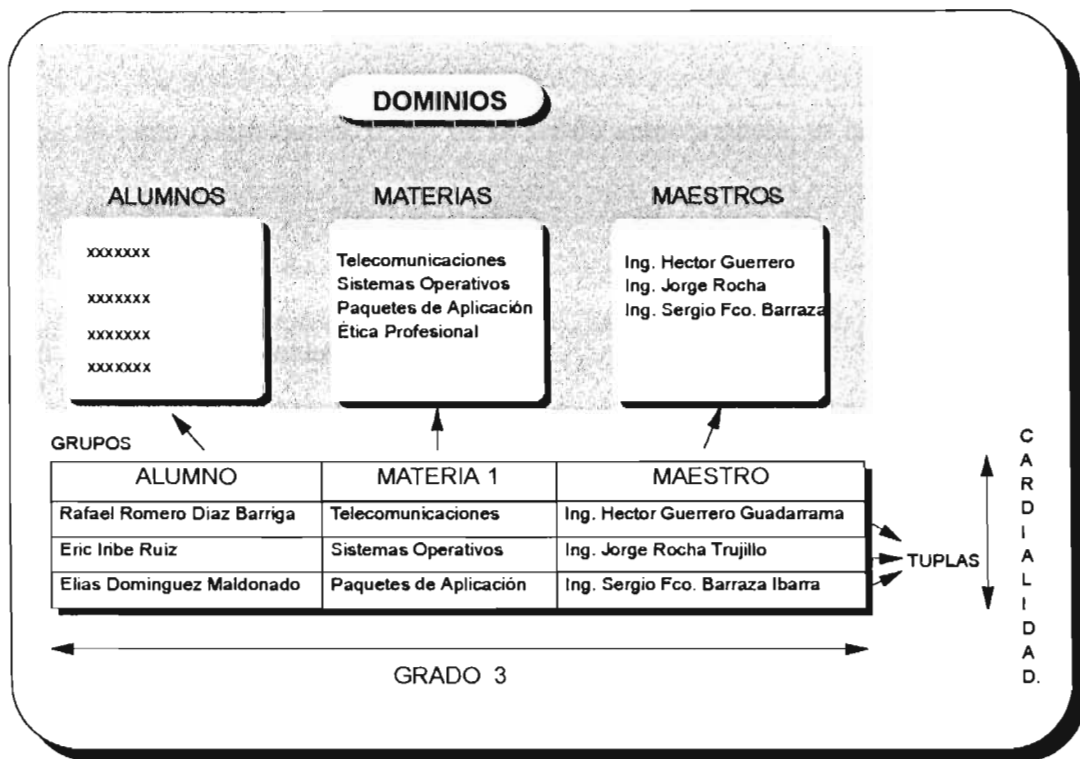
TABLA 8.1

Una relación se puede presentar en forma de tabla, aunque tiene una serie de elementos característicos que la distinguen de la tabla :

- ◆ No puede haber filas duplicadas, es decir, todas las tuplas tienen que ser distintas.
- ◆ El orden de las filas es irrelevante.
- ◆ La tabla es plana, es decir, en el cruce de una fila y una columna sólo puede haber un valor (no se admiten atributos multivaluados).

8.4.3.3.- DOMINIO

Un dominio es un conjunto finito de valores homogéneos y atómicos caracterizados por un nombre; decimos valores *homogéneos* porque son todos del mismo tipo y *atómicos* porque son indivisibles en lo que al modelo se refiere. (Ver Gráfica 8.2).



GRÁFICA 8.2

8.4.3.4.- RESTRICCIONES DEL MODELO RELACIONAL.

En el modelo relacional, al igual que en otros modelos, existen restricciones, es decir, estructuras u ocurrencias no permitidas, algunas de ellas son :

8.4.3.4.1.- RESTRICCIONES INHERENTES

Una de estas restricciones, como ya se señaló anteriormente, es la diferencia entre una relación de una tabla :

- ◆ No hay dos tuplas iguales.

- ◆ El orden de las tuplas no es significativo

- ◆ El orden de los atributos (columnas no es significativo).

- ◆ Cada atributo sólo puede tomar un único valor del dominio, no admitiéndose por tanto los grupos repetitivos.

La *regla de integridad de entidad* establece que “Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo”.

8.4.3.4.2.- RESTRICCIONES DE USUARIO

Dentro de este tipo de restricciones destaca la restricción de *Integridad referencial* que se expresa de la siguiente manera : “ Si una relación R_2 tiene un descriptor que es la clave primaria de la relación R_1 , todo valor que dicho descriptor debe concordar con el valor de la clave primaria de R_1 , o ser nulo”

Además hay que determinar las consecuencias que pueden tener ciertas operaciones realizadas por el usuario como pueden ser el “*borrado*” o la “*modificación*” realizadas sobre tuplas de la relación referenciada; pudiéndose distinguir, en principio, las siguientes opciones :

Nota .- De este momento y en lo sucesivo se cambiara el termino “Tupla” por el de “Registro”.

- ◆ ***Operación restringida.***- Esto es, el borrado o la modificación de registro de la relación que contiene la clave primaria referenciada sólo se permite si no existen registros con dicha clave en la relación que contiene la clave ajena.

- ◆ ***Operación con transmisión en cascada.***- Esto es, el borrado o la modificación de registros de la relación que contiene la clave primaria referenciada lleva consigo el borrado o modificación en cascada de los registros de la relación que contiene la clave ajena.

- ◆ **Operación con puesta a nulos.**- Esto es, el borrado o la modificación de registros de la relación que contiene la clave primaria referenciada lleva consigo poner a nulos los valores de las claves ajenas de la relación que referencia.

- ◆ **Operación con puesta a valor por defecto.**- Esto es, el borrado o la modificación de registros de la relación que contiene la clave primaria referenciada lleva consigo poner el valor por defecto a la clave ajena de la relación que referencia; valor por defecto que habría sido definido al crear la tabla correspondiente.

- ◆ **Operación que desencadena un procedimiento de usuario.**- En este caso, el borrado o la modificación de registros de la tabla referenciada pone en marcha un procedimiento definido por el usuario.

8.5.- SEGURIDAD Y CONFIDENCIALIDAD

Las *medidas de seguridad* están orientadas a preservar la información impidiendo cualquier intromisión que pudiera conducir a la destrucción de archivos maestros y de base de datos. La *protección de confidencialidad* consiste en las medidas que se toman para prevenir o evitar la difusión o distribución no autorizada de los datos. En otras palabras, la seguridad se refiere al acceso ilegal a los archivos de la computadora, ya sea físicamente o por infiltración (hacking), con el propósito de destruir, modificar o tener acceso a los datos sin permiso. La confidencialidad y privacidad por otra parte, entraña el derecho de controlar la distribución, divulgación o acceso de los datos.

La administración de las bases de datos implica la protección de la privacidad y de la seguridad, Everest (1972-), identificó tres estrategias de protección: confinamiento, reglamentación y cifrado de la información.

El *confinamiento* es el acto de alojar los datos en una ubicación física a la que no tengan fácil acceso personas no autorizadas.

La *reglamentación* se refiere a que alguien debe determinar quién debe tener acceso a los datos. Las tres fases de la reglamentación son : identificación, autorización y vigilancia.

La *identificación* de las personas que desean tener acceso a los datos se puede efectuar por medio de contraseñas y/o claves. Estas contraseñas y/o claves son palabras o nombres compuestos por números, letras o una combinación de ambos que sirven para identificar a las personas de confianza. Estos principios se pueden aplicar a la seguridad de los sistemas de cómputo.

La identificación va unida por lo general con la *autorización*, que es la aprobación para tener acceso a los archivos en particular y hacer determinados usos de los datos. Alguien debe decidir si se concede a un individuo en particular permiso para consultar todo un archivo, ciertos registros del mismo o nada más una parte de cada registro.

Una tercera fase de la reglamentación es la *vigilancia*. La conservación de registros de todas las personas que utilizan los datos y el examen de los registros de modo periódico es una forma relativamente sencilla de vigilar el acceso y tal vez detectar actividades indebidas.

Como el confinamiento completo de los datos a menudo no es factible o sensato, se aplican otras medidas de protección. Mediante el *cifrado* o *encriptación*, personas no autorizadas pueden tener dificultades para servirse de los datos que hayan extraído sin permiso o incluso para determinar cuáles son los registros a los que tuvieron acceso. El *cifrado* de la información se ejecuta desordenando los datos de acuerdo con reglas predeterminadas de información, de suerte que carezcan de sentido para quienes no puedan volverlos a ordenar. Para poder aprovechar los datos, éstos deben ser transformados de nuevo, lo cual sólo se puede efectuar a petición de una persona autorizada o por medio del sistema a través de una función de desencriptamiento.

CAPITULO 9**“DESCRIPCIÓN GENERAL DEL SISTEMA”****9.1.- DESCRIPCIÓN DE LOS MÓDULOS DEL SISTEMA**

El Sistema de Control Escolar esta compuesto por 7 módulos, a continuación se describen muy brevemente las funciones que realiza en cada uno de ellos; si se desea mayor información sobre las funciones que realiza cada modulo y la forma de operar el sistema, refiérase al “manual de usuario”.

9.1.1.- MODULO DE ESCUELAS

Este es el que se puede considerar como el modulo inicial del Sistema, ya que antes de poder realizar cualquier operación con el Sistema, es necesario dar de Alta todas las escuelas con que cuenta la universidad. Permite así mismo dar de Baja escuelas o bien, Modificar los datos de las mismas. Una vez que las escuelas han sido dadas de Alta, es aquí donde se elige a la escuela con la que se desea trabajar.

En este modulo también se cuenta con la opción de reindexar las bases principales del sistema.

9.1.2.- MODULO DE ALUMNOS

En este modulo es en donde se maneja toda la información referente a los alumnos de cada una de las escuelas, éstas opciones son : Inscripciones, Baja, Modificación, Recuperación de alumnos dados de baja y Calificaciones.

9.1.3.- MODULO DE MAESTROS

En este modulo se maneja la información de la plantilla de maestros de cada una de las escuelas, permite realizar las operaciones básicas de Alta, Baja y Modificaciones.

9.1.4.- MODULO DE MATERIAS

Es este el modulo encargado de manejar todas las materias de que constan los planes de estudio de cada una de las escuelas, permite realizar las operaciones básicas de Alta, Baja y Modificaciones.

9.1.5.- MODULO DE GRUPOS

Este es uno de los módulos mas complejos, es aquí donde se asignan horarios, materias, maestros, salones de clase y edificio a cada uno de los grupos. Posteriormente se asignan las materias a los alumnos de los grupos que hayan sido formados.

9.1.6.- MODULO DE REPORTES

Este es el modulo mas grande del sistema, ya que es el encargado de generar todos los reporte impresos que genera el Sistema, algunos de los reportes generados son : Boletas de calificaciones, listas de asistencia, credenciales, constancias de estudio, certificados, etc.

9.1.7.- MODULO DE UTILERIAS

Es este modulo el encargado de realizar el respaldo y recuperación de la información generada por el Sistema, actualizar y/o regenerar los índices de las bases de datos, además, es el encargado de preparar todas las bases de datos para iniciar un nuevo semestre.

9.2 .- DESCRIPCIÓN DE LAS BASES DE DATOS DEL SISTEMA

El Sistema de Control Escolar, para su operación, hace uso de 12 bases de datos. Éstas están divididas en dos categorías : Bases Generales y Bases Propias. Las Bases Generales son aquellas que se usan independientemente de la escuela que sea seleccionada para trabajar, éstas bases se encuentran ubicadas en el directorio principal del sistema. Las Bases Propias, son aquellas que contienen la información propia de cada escuela, éstas bases están ubicadas en directorios individuales por escuela.

9.2.1 .- CONTENIDO DE LAS BASES DE DATOS DEL SISTEMA

Las siguientes son las Bases Generales del Sistema :

- ◆ **Escuelas.dbf**.- Contiene el nombre de la escuela, su clave, el tipo (SEP, México ó SEE, Michoacán), y el nombre del directorio donde se encuentran las bases propias de la escuela.

- ◆ **Config.dbf** .- Contiene las claves de los usuarios (encriptadas), el nivel de acceso (encriptado), la impresora que le fue asignada y el puerto de impresión que usara cada uno de los usuarios dados de alta en el Sistema.

- ◆ **Impres.dbf** .- Contiene el número de identificación de cada una de las impresoras disponibles, así como el número del registro donde comienzan los códigos de cada una de las impresoras.

- ◆ **Escapes.dbf** .- Contiene los códigos de impresión de todas las impresoras disponibles.

Las Bases Propias de cada Escuela son :

- ◆ ***Alumnos.dbf*** .- Contiene toda la información necesaria para identificar a los alumnos (nombre, dirección, nombre del tutor, teléfono, etc.), las materias que le fueron asignadas, las calificaciones y las inasistencias. La estructura de esta base es distinta para licenciatura, preparatoria y postgrados.

- ◆ ***Materias.dbf*** .- Contiene el nombre, clave, número de serie y el área de todas las materias del plan de estudios.

- ◆ ***Maestros.dbf*** .- Contiene todos los datos necesarios para identificar a los maestros (nombre, dirección, R.F.C., teléfono, clave del maestro, etc.)

- ◆ ***Grupos.dbf*** .- Contiene las claves con las que se identifica a cada grupo, así como los horarios, salones, edificios, claves de las materias y de los maestros de cada uno de los grupos. Esta información se genera al formar los grupos.

- ◆ ***Asigna.dbf*** .- Contiene las claves de las materias que fueron asignadas en forma “real” a cada uno de los grupos. Esta información se genera al asignar materias a los alumnos.

-
- ◆ ***Inicio.dbf*** .- Es en esta base donde se lleva el control de ciertos procesos que no pueden ser realizados sin haber ejecutado primeramente otros. Por ejemplo, no se puede inicializar un nuevo semestre, sin antes haber generado el kardex.

 - ◆ ***Kardex.dbf*** .- Contiene todo el historial académico de cada uno de los alumnos.

 - ◆ ***Edit.dbf*** .- Contiene los documentos generados en la opción de impresión de constancias de estudio.

9.2.1.1.- ESTRUCTURA DE LAS BASES DE DATOS DEL SISTEMA

Las siguientes son las estructuras de las Bases Generales del Sistema :

ESCUELAS.DBF

Campo	Tipo	Longitud	Decimales
NomEsc	c	55	0
Clave	c	2	0
Tipo	c	1	0
Direct	c	5	0
TipoSep	l	1	0

CONFIG.DBF

Campo	Tipo	Longitud	Decimales
Clave	c	5	0
Nivel	c	1	0
Nmlmp	n	3	0
Puerto	c	8	0

IMPRES.DBF

Campo	Tipo	Longitud	Decimales
Nmlmp	n	3	0
Dslmp	c	35	0

ESCAPES.DBF

Campo	Tipo	Longitud	Decimales
Nmlmp	n	3	0
Escape	c	12	0
Valor	c	40	0

ALUMNOS.PRG

POSTGRADO

Campo	Tipo	Longitud	Decimales
Inscrito	l	1	0
MatrAlum	c	7	0
NomAlum	c	35	0
SemesAlum	c	2	0
SexAlum	c	1	0
DomAlum	c	35	0
ColAlum	c	20	0
CPAlum	c	8	0
CiudAlum	c	20	0
TelAlum1	c	15	0
TelAlum2	c	15	0
EscProAlum	c	20	0
NomTutAlum	c	35	0
FelnsAlum	d	8	0
Grupo	c	3	0
Baja	c	1	0
Mat1	c	5	0
U1	n	4	1
I1	n	2	0
MAT13	c	5	0
U13	n	4	1
I13	n	2	0

MAESTROS.DBF

Campo	Tipo	Longitud	Decimales
ClavMat	c	5	0
NomMae	c	35	0
RfcMae	c	13	0
NivMae	c	15	0
SexMae	c	1	0
DomMae	c	35	0
ColMae	c	20	0
CPMae	c	5	0
CiudMae	c	20	0
TelMae	c	15	0
FelngMae	d	8	0

MATERIAS.DBF

Campo	Tipo	Longitud	Decimales
ClavMat	c	5	0
NomMae	c	55	0
SerieMat	c	5	0
Area	c	25	0

GRUPOS.DBF

Campo	Tipo	Longitud	Decimales
Grupo	c	3	0
ClavMat	c	5	0
ClavMae	c	5	0
Hora1	c	4	0
Hora2	c	4	0
Hora3	c	4	0
Hora4	c	4	0
Hora5	c	4	0
Hora6	c	4	0
Hora7	c	4	0
Hora8	c	4	0
Salon1	c	2	0
Salon2	c	2	0
Salon3	c	2	0
Salon4	c	2	0
Edif1	c	2	0
Edif2	c	2	0
Edif3	c	2	0
Edif4	c	2	0
Dia1	c	1	0
Dia2	c	1	0
Dia3	c	1	0
Dia4	c	1	0

ASIGNA.DBF

Campo	Tipo	Longitud	Decimales
Grupo	c	3	0
ClavMat	c	5	0

INICIO.DBF

Campo	Tipo	Longitud	Decimales
Kardex	l	1	0
Cfiya	l	1	0
Cuotas	l	1	0

EDIT.DBF

Campo	Tipo	Longitud	Decimales
Edit	m	10	0

9.2.2 .- INTERRELACIÓN DE LAS BASES DE DATOS CON EL SISTEMA.

Para poder apreciar en una forma mas clara la relación que guarda cada una de las Bases de Datos con los distintos módulos de los que está compuesto el Sistema, se hace uso de diagramas de bloques. Primeramente se ve un diagrama general donde se pueden ver todos los módulos y todas las bases de datos de que esta compuesto el sistema. (Diagrama 9.1), a continuación se hace un desglose en varios diagramas (por cuestión de simplicidad), en donde se puede apreciar que bases de datos tienen relación con que modulo.



DIAGRAMA 9.1

SISTEMA DE CONTROL ESCOLAR

MODULO DE ESCUELAS

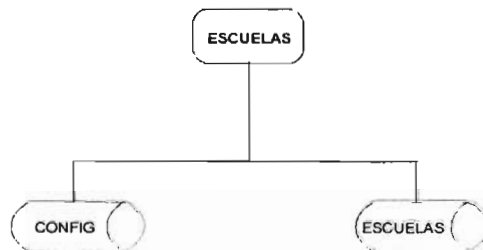


DIAGRAMA 9.2

SISTEMA DE CONTROL ESCOLAR

MODULO DE ALUMNOS

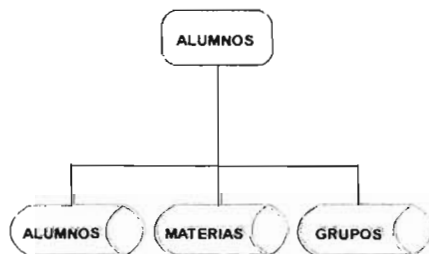


DIAGRAMA 9.3

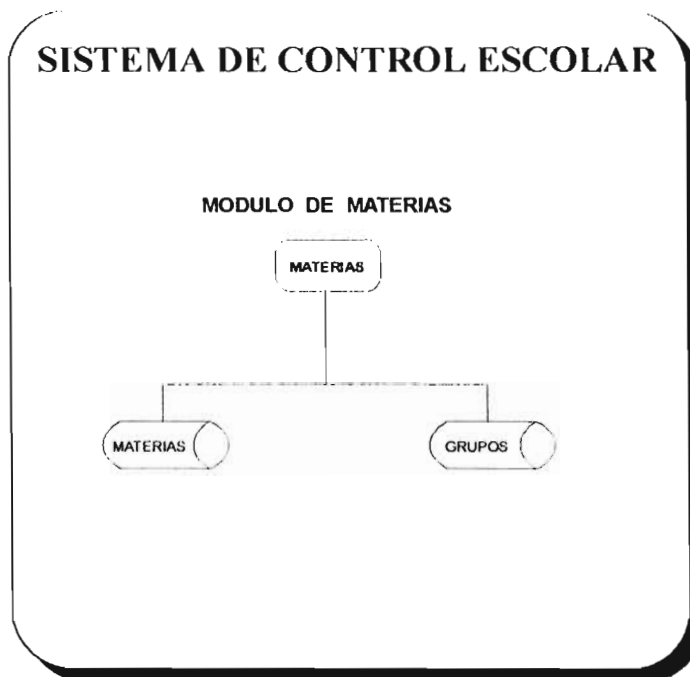


DIAGRAMA 9.4



DIAGRAMA 9.5



DIAGRAMA 9.6

9.3.- REGISTRO DE LA INFORMACIÓN CON EL SISTEMA PROPUESTO

Tal como se mencionó anteriormente, dentro del departamento de Control Escolar existen una gran variedad de procesos y uno de estos procesos es el del registro de la información de los alumnos inscritos; a continuación se explica este proceso una vez que se lleve a cabo utilizando el Sistema de Control Escolar propuesto.

- ◆ Primeramente se recibe de ventanilla la documentación de los alumnos aceptados para nuevo ingreso.

- ◆ Se verifica en la ventanilla si la documentación entregada por el alumno esta completa.

- ◆ Se captura en ventanilla, la información del alumno en el Sistema de Control Escolar propuesto

- ◆ Se elabora y se manda a la SEP - Michoacán, el oficio en el cual se solicita el número de control para cada uno de los alumnos de nuevo ingreso. Esto es para cada una de las licenciaturas.

- ◆ Una vez que se cuenta con los números de control, se imprimen las CFIYAS por carrera y semestre.

- ◆ Las CFIYAS son enviadas al SEE para que sean registradas y certificadas para que de esta forma, el alumno esté dado de alta "oficialmente".
- ◆ Se toma la foto al alumno y se manda elaborar su credencial.

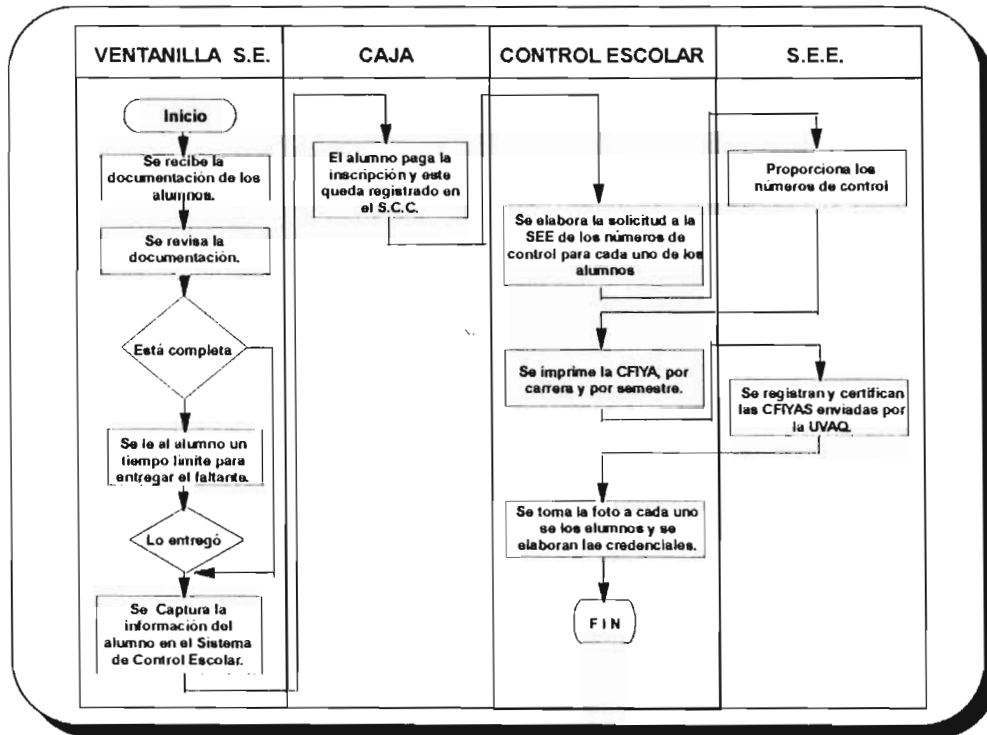


DIAGRAMA 9.7

CAPITULO 10**“CONCLUSIONES Y RECOMENDACIONES”**

Es importante considerar que con la implantación del Sistema de Control Escolar en conjunto con el Sistema de Cuentas por Cobrar en el área de Control Escolar y en el área administrativa de la Universidad Vasco de Quiroga, se lograron disminuir en forma subrayada muchos de los procedimientos manuales que anteriormente ocasionaban consumo de tiempo y descuido en otras actividades.

Con la implantación del Sistema de Control Escolar se logró que el procedimiento de inscripción, la emisión de boletas de calificaciones, listas de asistencia, credenciales, y el control del historial académico de los alumnos se pueda llevar a cabo en forma rápida y eficaz, reduciendo con ello los errores y el tiempo invertido en el manejo de la información, ya que algunos de dichos procesos se realizaban en forma manual.

El Sistema de Control Escolar fue diseñado pensando sobre todo en que fuera un programa fácil de usar y que además contará con los elementos de seguridad necesarios. En lo que respecta a la facilidad esto se consigue al diseñar menús y pantallas sencillas que en todo momento muestran mensajes informativos de las operaciones que se realizan y de las teclas rápidas de función; en el caso de la seguridad se cuenta con dos elementos sumamente útiles y efectivos para lograrla, estos son el Sistema de Protección de Acceso y las opciones de respaldo y recuperación de información disponibles en todo momento dentro del Sistema.

Finalmente a continuación presento algunas recomendaciones que sería importante seguir para obtener una mayor eficiencia del Sistema de Control Escolar.

- ◆ Aunque como ya se mencionó, el Sistema está diseñado para que pueda ser utilizado por personal con poco conocimiento en el área de la informática y la computación, es recomendable la capacitación al personal que estará a cargo del sistema, sobre todo para evitar un mal uso del mismo.

- ◆ Asignar a los usuarios claves de acceso individuales, indicándole a los usuarios que no usen como clave aquello que pueda ser fácil de deducir por personas no autorizada. Lo que no recomiendo que se use como clave de acceso es: las iniciales del usuario, de la universidad o de cualquier persona ligada con la institución o con el usuario, números de teléfono, direcciones o secuencias de caracteres como pueden ser “abcd”, “1234”, “aaaa” o “9999”. El Sistema de Protección con que cuenta el Sistema de Control Escolar, permite hasta 10 niveles de privilegios (los primeros 6 para el Sistema de Control Escolar y los restantes 4 para el Sistema de Cuentas por Cobrar), lo que permite que los usuarios puedan tener bien determinadas sus funciones y responsabilidades. (Para mayor información refiérase al manual de usuario).

- ◆ Al momento de iniciar con la inscripción de los alumnos, es recomendable que se tenga capturada toda la información de materias y maestros.

-
- ♦ Así mismo y como parte indispensable para la correcta operación del Sistema, se debe contar con una terminal en la ventanilla de servicios escolares y otra en la ventanilla de caja, esto independientemente del número de terminales destinadas al departamento de Control Escolar.

 - ♦ Realizar un **respaldo semanal** de la información para evitar la pérdida de información o el borrado accidental de las principales y más importantes bases del sistema.

"BIBLIOGRAFIA"

➤Eric Iribe Ruiz, *Sistema de cuentas por cobrar para instituciones educativas*, Tesis profesional, Universidad Vasco de Quiroga, Escuela de Sistemas Computarizados, Morelia, 1996, 276 pp.

➤Miguel/Mario Piattini. *Concepción y diseño de bases de datos*, México, 1993, 989 pp., Addison-wesley Iberoamericana.

➤Henry F. Korth. Abraham Silberschatz. *Fundamentos de bases de datos*, 1ª Ed., México, 1990, 525 pp., Mc. Graw-Hill.

➤Mark L. Gillenson. *Introducción a las bases de datos*, 1ª Ed., México, 1987, 391 pp., Mc. Graw-Hill

➤James A. Senn. *Sistemas de información para la administración*, 3ª Ed., México, 1990, 728 pp., Grupo Editorial Iberoamérica.

➤Donald H. Sanders, *Informática presente y futuro*, 3ª Ed., México, 1991, 887 pp., Mc. Graw-Hill

-
- Manoella Atziry Hernández Montiel, *Sistema de control de recursos para obras públicas del Estado de Michoacán*, Tesis profesional, Universidad Vasco de Quiroga, Escuela de Sistemas Computarizados, Morelia, 1996, 159 pp.
- Wendy Reyes Vargas, *Automatización del sistema de inscripción de las sociedades comerciales en registro público de la propiedad y del comercio del estado*, Tesis profesional, Universidad Vasco de Quiroga, Escuela de Sistemas Computarizados, Morelia, 1995, 145 pp.
- Salvador Díaz Barriga Hernández, *Implantación de sistemas de contabilidad en microcomputadoras*, Tesis profesional, Universidad del Valle de México, Escuela de Contaduría Pública, México D.F., 1983, 104 pp.
- Guillermo Levine, *Introducción a la computación y a la programación estructurada*, 2ª Ed., México, 1992, 424 pp., Mc. Graw-Hill.
- José A. Ramalho, *Clipper 5.2 avanzado*, 2ª Ed., España, 1994, 391 pp., Mc. Graw-Hill.
- Francisco Marín Quirós, Antonio Quirós Casado, Antonio Torres Lozano, *Clipper 5, referencia rápida*, 2ª Ed., México, 1991, 596 pp., Macrobit-RaMa
- Rick Spence, *Clipper 5.2, Guía avanzada para el programador*, 1ª Ed., España, 1994, 782 pp., Addison-wesley Iberoamericana.
- Stephen J. Straley, *Clipper 5.0*, 1ª Ed., México, 1993, 832 pp., Grupo noriega editores.

-
- Grupo EIDOS, *Clipper en redes de área local*, 1ª Ed., E.U.A., 1994, 658 pp., Addison-wesley iberoamericana.

 - Eduardo Alcalde, Jesús García Tomás, *Introducción a la teleinformática*, España, 1993, 296 pp., Mc. Graw-Hill.

 - Tom Sheldon, *Novell netware 4 manual de referencia*, España, 1994, 815 pp., Mc. Graw-Hill.

 - James A. Senn, *Análisis y diseño de sistemas de información*, 2ª Ed., México, 1994, 942 pp., Mc. Graw-Hill.

 - Dra. Guillermina Baena, *Manual para elaborar trabajos de investigación documental*, 1ª Ed., México, 1991, 124 pp., Editores mexicanos unidos, S.A.

 - Universidad Vasco de Quiroga, *Manual de procedimientos de control escolar*, Morelia, 50 pp.

 - Universidad Vasco de Quiroga-Departamento de integración universitaria, *Manual de criterios para admisión*, Morelia, 1994, 50 pp.

 - Sector de orientación pedagógica - Programa educativo visual, *Enciclopedia estudiantil visual / color*, México, 1996, 956 pp., THEMA

APÉNDICE “A”

MANUAL

DE

USUARIO

INDICE

Introducción	96
Bienvenida	96
Introducción al manual de usuario	96
Propósito	97
Organización	97
1.- Características y requerimientos	101
2.- Instalación	103
3.- ¿ Cómo iniciar ?	106
4.- Módulo de escuelas	109
4.1.- Introducción	109
4.2.- Opción de utilerías	110
4.2.1.- Alta de escuelas	110
4.2.2.- Baja de escuelas	112
4.2.3.- Modificación de escuelas	113
4.2.4.- Reindexar bases	113
4.3.- Opción de seleccionar	114
5.- Módulo de materias	116
5.1.- Introducción	116
5.2.- Alta de materias	117
5.3.- Baja de materias	118
5.4.- Modificación de materias	119
6.- Módulo de maestros	121
6.1.- Introducción	121
6.2.- Alta de maestros	122
6.3.- Baja de maestros	123
6.4.- Modificación de maestros	125
7.- Módulo de alumnos	126
7.1.- Alumnos	126

7.1.1.- Inscripción de alumnos	127
7.1.2.- Reinscripción de alumnos	128
7.1.3.- Baja de alumnos	129
7.1.4.- Recuperación de alumnos de baja temporal	130
7.2.- Calificaciones	132
7.2.1.- Captura de calificaciones por grupo	133
7.2.2.- Captura de calificaciones por alumno	135
8.- Módulo de grupos	138
8.1.- Introducción	138
8.2.- Formar grupos	139
8.2.1.- Modificar grupos	141
8.3.- Asignar materias a los grupos	142
8.3.1.- Asignar materias por grupos	143
8.3.1.1.- Modificar asignaciones de materias por grupo	144
8.3.2.- Asignar materias por alumno	146
8.3.2.1.- Modificar asignaciones de materias por alumno	147
9.- Módulo de reportes	149
9.1.- Introducción	149
9.2.- Listas de asistencia	151
9.2.1.- Listas de asistencia por grupo	151
9.2.2.- Listas de asistencia por materia	154
9.3.- Impresión de credenciales	155
9.3.1.- Credenciales por grupo	156
9.3.2.- Credenciales por alumno	157
9.4.- Impresión de concentrados de calificaciones (licenciaturas)	158
9.4.1.- Concentrados de calificaciones por grupo	159
9.4.2.- Concentrados de calificaciones por materia	160
9.5.- Impresión de constancias de estudio	161
9.6.- Impresión de boletas de calificaciones	163
9.6.1.- Impresión de boletas de calificaciones por grupo	164

9.6.2.- Impresión de boletas de calificaciones por alumno	165
9.7.- Consulta de calificaciones	166
9.7.1.- Consulta de calificaciones por grupo	167
9.7.2.- Consulta de calificaciones por alumno	169
9.8.- Impresión de CFIYA	171
9.9.- Actualización del KARDEX	173
10.- Módulo de utilerías	174
10.1.- Introducción	174
10.2.- Inicio de semestre	175
10.3.- Respaldo de información	176
10.4.- Recuperación de la información	178
10.5.- Reindexar las bases de datos de la escuela	179
11.- Sistema de Protección de Acceso	181
11.1.- Introducción	181
11.2.- Uso del Sistema de Protección a nivel usuario	182
11.3.- Uso del Sistema de Protección a nivel supervisor	183
11.3.1.- Alta de usuarios	184
11.3.2.- Baja de usuarios	187
11.3.3.- Modificación de usuarios	188
11.3.4.- Opción terminar	190

INTRODUCCIÓN

BIENVENIDO

Bienvenido al Sistema de Control Escolar para Instituciones Educativas.

INTRODUCCIÓN AL MANUAL DE USUARIO

El Sistema de Control Escolar fue diseñado con la finalidad de proporcionar al usuario una herramienta sumamente flexible e integrada que permita el control de todas las operaciones del departamento de Control Escolar, contemplando aspectos relacionados con :

- Alumnos
- Materias
- Horarios
- Maestros
- Grupos
- Reportes varios

Y en conjunto con el Sistema de Cuentas por Cobrar :

- Impresión de recibos
- Reporte de saldo en colegiaturas
- Cuentas por cobrar
- Colegiaturas

PROPÓSITO

Este manual tiene como propósito el de explicar de manera simple y concisa, la forma correcta de operar el Sistema de Control Escolar. El presente documento está diseñado, pensando en que pueda ser utilizado por usuarios que tengan poca experiencia con sistemas informáticos, sin embargo siempre es recomendable que se tengan las nociones básicas en DOS.

ORGANIZACIÓN

El presente manual ha sido organizado en once capítulos, a continuación se da una breve descripción de lo que podrá encontrar en cada uno de ellos.

CAPÍTULO 1

CARACTERÍSTICAS Y REQUERIMIENTOS

Se indican las principales características del sistema y los requerimientos mínimos de Hardware necesarios para su correcto funcionamiento.

CAPÍTULO 2

INSTALACIÓN

Se detalla el procedimiento necesario para la instalación del sistema en disco duro.

CAPÍTULO 3*¿ CÓMO INICIAR ?*

En este capítulo se indica la forma de iniciar el Sistema, se describe detalladamente la distribución de la pantalla, así como las opciones que están disponibles en el menú principal del Sistema.

CAPÍTULO 4*MÓDULO DE ESCUELAS*

En este capítulo se indica la forma de acceder al Sistema de Control Escolar, así como el procedimiento necesario para dar de alta, de baja, o modificar la información de las distintas escuelas de que esta compuesta la universidad. Una vez que las escuelas han sido dadas de alta., es aquí donde se elige a la escuela con la que se desea trabajar.

CAPÍTULO 5*MÓDULO DE MATERIAS*

En este capítulo se estudia la forma de integrar el plan de estudios de cada una de las escuelas de la Universidad.

CAPÍTULO 6*MÓDULO DE MAESTROS*

En este capítulo se estudia la forma de integrar la plantilla de maestros correspondiente a cada una de las escuelas con que cuenta la Institución.

CAPÍTULO 7

MÓDULO DE ALUMNOS

En este capítulo se explica como realizar la inscripción, la reinscripción, la baja y la reactivación de los alumnos. Así mismo se muestra el procedimiento para el manejo de calificaciones.

CAPÍTULO 8

MÓDULO DE GRUPOS

En esta sección se explica como llevar a cabo el proceso de formación de grupos, el cual consisten en asignar maestros y horarios a las materias que serán impartidas en cada uno de los grupos; también se detalla la forma de asignar las materias a los alumnos inscritos en cada uno de los grupos.

CAPÍTULO 9

MÓDULO DE REPORTES

En este capítulo se detalla la manera de obtener todos los reportes impresos que general el sistema. Algunos de ellos son : listas de asistencia, credenciales, boletas de calificaciones, concentrados de calificaciones, CFIYA, constancias de estudio, Etc.

CAPÍTULO 10

MÓDULO DE UTILERIAS

En este capítulo se detalla la manera de iniciar un nuevo semestre, el proceso tanto para obtener los respaldo de la información como para recuperar la información de los respaldos generados, de igual manera se explica como regenerar los índices de las bases de datos de cada una de las escuelas.

CAPÍTULO 11

SISTEMA DE PROTECCIÓN DE ACCESO

En este capítulo se detalla la manera de dar de alta, asignar los privilegios de operación, las impresoras, los puertos de impresión y las **claves de acceso** a todos los **usuarios** del sistema.

CAPÍTULO 1

“CARACTERÍSTICAS Y REQUERIMIENTOS”

El Sistema de Control Escolar fue diseñado pensando principalmente en aprovechar la facilidad que en la actualidad brindan los equipos de cómputo para el manejo de información de una manera fácil, rápida y confiable. Algunas de las principales características del programa son :

- Dado el gran auge que en los ultimo tiempos han tenido las redes informáticas, el Programa fue diseñado para operar en sistemas de red bajo ambiente NOVELL, sin embargo, de ser necesario, también existe la versión para sistemas monousuarios bajo ambiente DOS.
- El Sistema puede ser ejecutado directamente desde ambiente de Windows 3.x , esto es independiente de si se esta ejecutando el programa en red o en monousuario, de esta forma es posible tener un aprovechamiento mas optimo de los recursos de Hardware con que se cuente.
- El Sistema cuenta también con una gran variedad de “*drivers*” para impresoras.
- También se incluye un Sistema de Protección, el cual evita que personal no autorizado tenga acceso a la información.

- Por ultimo, el Sistema cuenta con un protector de pantalla, lo que evitara que en periodos prolongados de inactividad la pantalla de vídeo resulte dañada.

Los requerimientos mínimos de Hardware necesarios para que el Sistema de Control Escolar opere, son :

REQUISITOS	
Componente	Descripción
Computadora	<ul style="list-style-type: none">- IBM o compatible- Procesador 80286 o superior. (Se recomienda 80486 DX2.)- Unidad de discos de 3 1/2" de alta densidad.- Monitor CGA, VGA o de resolución superior.- Si el sistema opera en red: Tarjeta de red de 8 bits. Se recomienda tarjeta de 16 bits.
MS-DOS	Versión 5.x o posterior.
Memoria	590 Kb libres o mas. Para trabajo en ambiente Windows se recomiendan al menos 3 Mb.
Espacio en disco	El espacio requerido dependerá de la cantidad de información que se almacene, sin embargo como medida de referencia, para almacenar la información de 10,000 alumnos se requieren 20 Mb de espacio libre.

CAPÍTULO 2**"INSTALACIÓN"**

Dentro de las peculiaridades que ofrece el Sistema de Control Escolar, y dependiendo de las necesidades de los usuarios, se puede llevar a cabo la instalación del Sistema en tres ambientes de trabajo: MS-DOS, Windows y NOVELL. A continuación se establecen los pasos necesarios para realizar la instalación del Sistema según la opción de trabajo seleccionada.

- ***Para instalar el Sistema en ambiente MS-DOS***

- 1.- Inicialice la computadora
- 2.- Introduzca el disco etiquetado como *Instalar*, en la unidad de discos.
- 3.- A continuación del símbolo del sistema, escriba la letra de la unidad que esté utilizando, seguida de dos puntos (:) y presione <ENTER> como se muestra a continuación : C:\> A: <ENTER>
- 4.- Escriba **Instalar** y presione la tecla <ENTER>

- *Para instalar el Sistema en ambiente Windows*

- 1.- Inicialice la computadora

- 2.- Ejecute Windows de la siguiente manera : C:\> WIN <ENTER>

- 3.- Introduzca el disco etiquetado como *Instalar*, en la unidad de discos.

- 4.- En el menú **Archivo** del Administrador de programas, elija **Ejecutar**.

- 5.- Escriba *x:\INSTAWIN*, donde x es la letra que representa la unidad de disco y presione la tecla <ENTER>.

- 6.- Concluida la instalación, salga de Windows y vuelva a ejecutarlo como lo hizo en el paso 2 para que se configure el sistema.

- *Para instalar el Sistema en ambiente de red NOVELL*

- 1.- Asegúrese de que el servidor de la red esté encendido.
- 2.- Inicialice la terminal desde donde va a ser instalado el Sistema.
- 3.- Asegúrese de tener el grado de Supervisor, o bien de contar con los privilegios necesarios para poder instalar una aplicación en el servidor de la red.
- 4.- Introduzca el disco etiquetado como *Instalar*, en la unidad de discos.
- 5.- A continuación del símbolo del sistema, escriba la letra de la unidad que esté utilizando, seguida de dos puntos (:) y presione <ENTER> como se muestra a continuación : x:\> A: , donde x representa la letra de la unidad y presione la tecla <ENTER>.
- 6.- A continuación del símbolo del sistema escriba **INSTARED** y presiona nuevamente la tecla <ENTER>.
- 7.- Una vez finalizada la instalación, asegúrese de que el directorio que contiene al Sistema tenga los atributos de Lectura-Escritura, Modificación, Borrado, Creación, además de que todos los archivos sean tengan el atributo de Compartidos; en caso de que no se cumplan estos atributos, se presentaran errores y no podrá ser operado correctamente el programa.

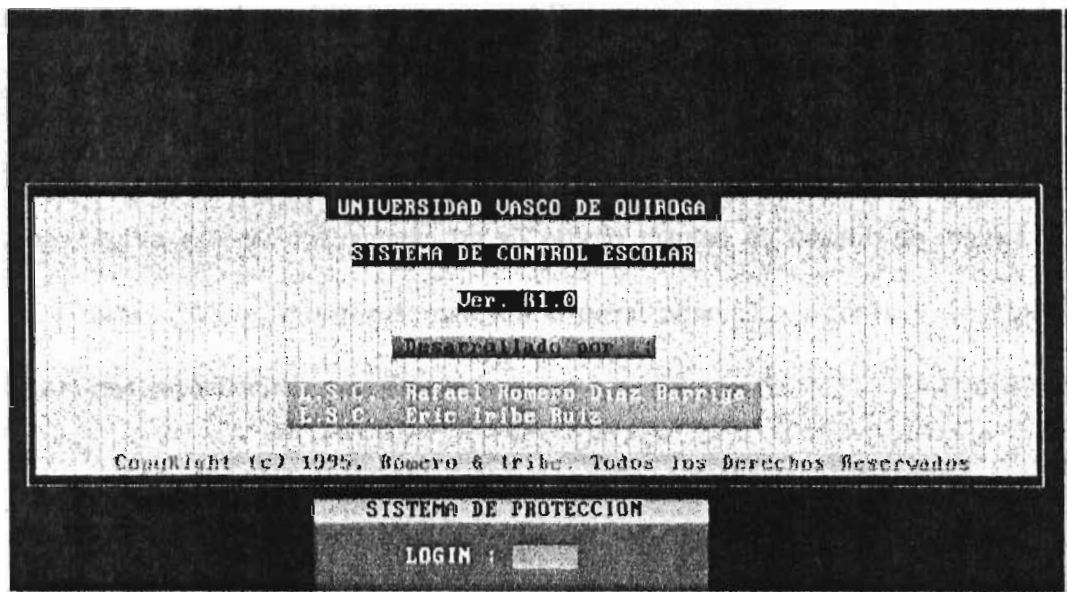
CAPÍTULO 3**“¿CÓMO INICIAR?”**

Para iniciar el Sistema de Control Escolar desde ambiente DOS o desde ambiente de red, es necesario realizar los siguientes pasos : A continuación del símbolo del sistema, escriba CD SCE y presione <ENTER> como se muestra a continuación : x:\> CD SCE: , donde x representa la letra de la unidad. Acto seguido es necesario escribir x:\>SCE>SCE

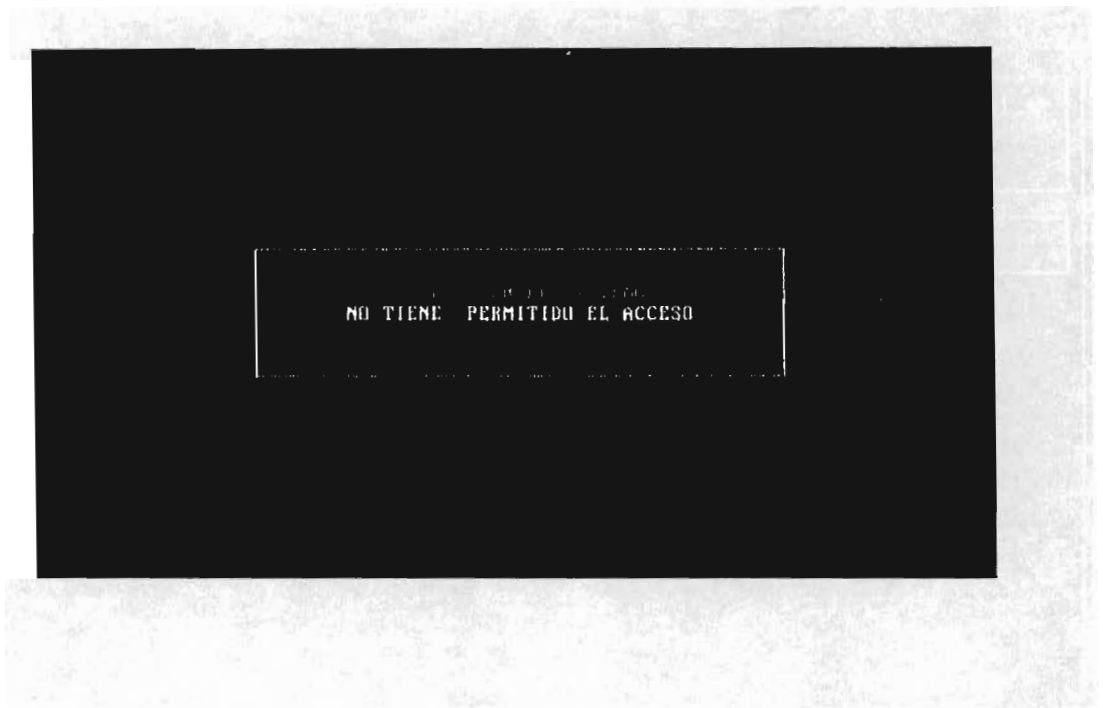
En caso de que se desee ejecutar el Sistema en ambiente Windows, será necesario que se seleccione el icono correspondiente y hacer doble clic sobre el.

Nota: Entiéndase el hacer doble clic, como el acto de presionar el botón izquierdo del mouse dos veces consecutivas con un intervalo de tiempo relativamente corto.

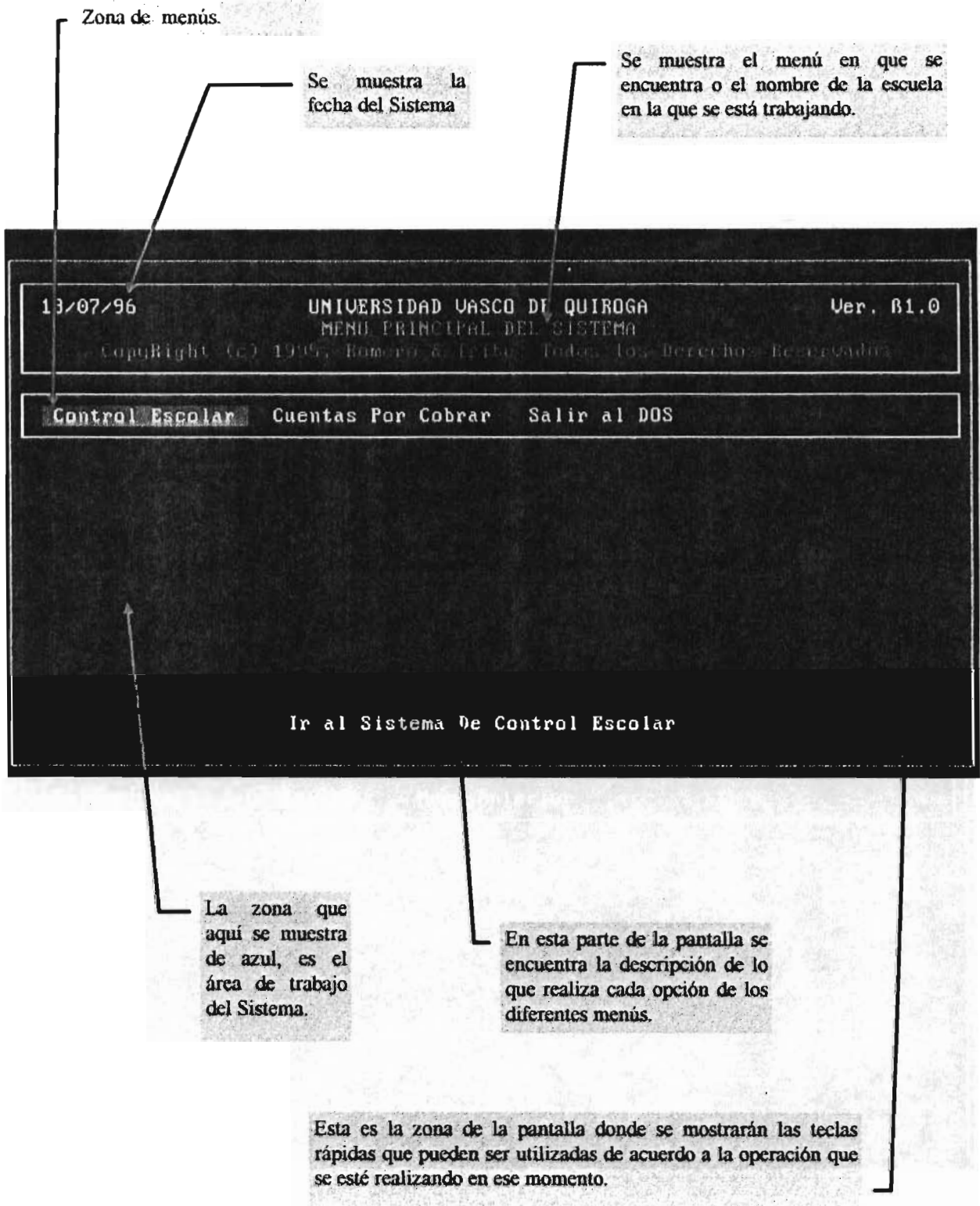
Posteriormente e independientemente del ambiente donde se ejecute el Sistema, como primera instancia se mostrará la pantalla de presentación y se le pedirá al usuario que *tecle* su clave de acceso tal como se muestra a continuación.



Como medida de protección, el Sistema permite únicamente que la clave tecleada sea errónea en tres ocasiones, en caso de que se fracase en los tres intentos por ingresar, se mostrará la pantalla que se muestra a continuación, acompañada de una alarma sonora.

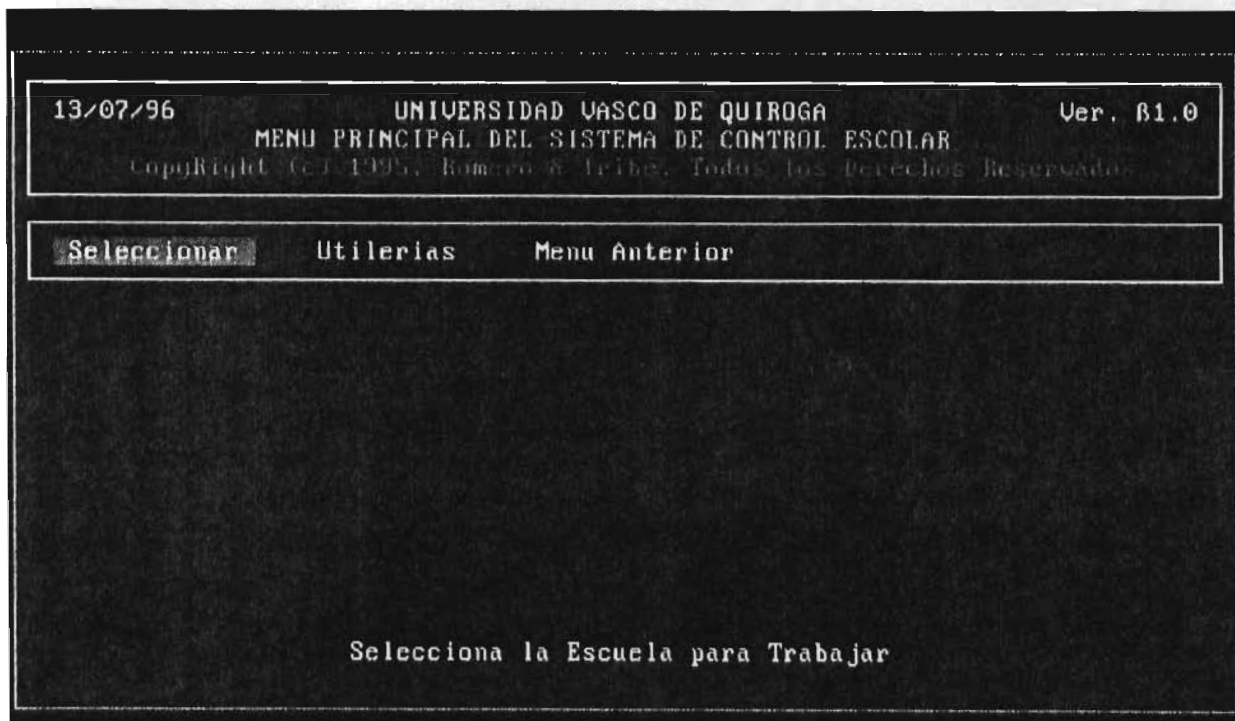


Una vez que se ha logrado tener acceso al Sistema por primera vez y/o en caso de que se tenga nivel de supervisor, la primera pantalla que se mostrará al usuario es la del menú principal, en el se tiene acceso tanto al Sistema de Control Escolar como al Sistema de Cuentas por Cobrar, sin embargo, como este último cuenta con su propio manual de usuario se omitirá cualquier referencia al mismo. A continuación se muestra el menú principal así como la descripción de la forma en que esta organizada la pantalla.



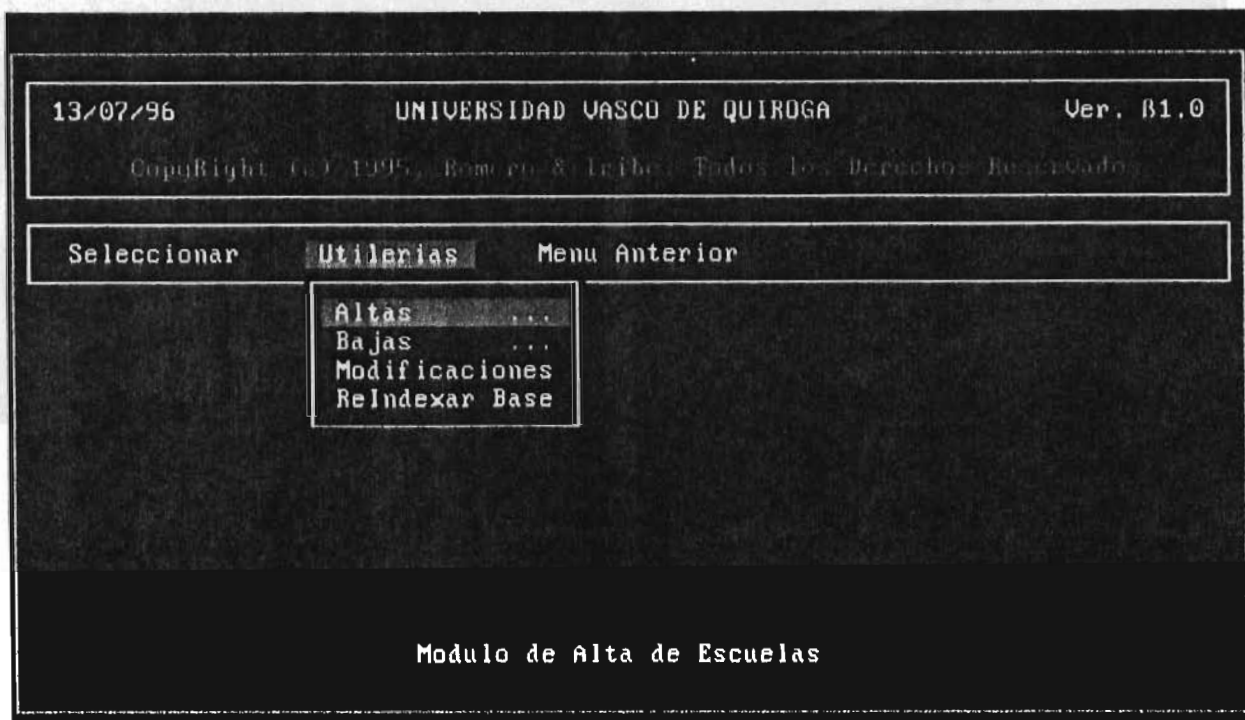
CAPÍTULO 4**"MÓDULO DE ESCUELAS"****4.1.- INTRODUCCIÓN.**

Como se puede ver, el menú principal tiene tres opciones, elija la primera opción y presione la tecla <ENTER>, una vez realizado esto, se mostrará el Menú Principal del Sistema de Control Escolar (que es a su vez el menú principal del módulo de escuelas), tal como se muestra a continuación.



4.2.- OPCIÓN DE UTILERIAS

Al seleccionar esta opción y presionar la tecla <ENTER>, se desplegara un submenú tal y como se muestra a continuación.



4.2.1.- ALTA DE ESCUELAS

Antes de poder realizar cualquier tipo de trabajo con el Sistema, es necesario dar de alta todas las escuelas de que esta integrada la Universidad. Para realizar esta operación, seleccione la opción de *altas* y presione la tecla <ENTER>, una vez realizado esto, el área de trabajo se podrá ver como se muestra a continuación :

```
13/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
  Copyright (c) 1995, Romero & Tribe. Todos los Derechos Reservados

Seleccionar      Utilerias      Salir al DOS

(P) Preparatoria, (L) Licenciatura, (G) PostGrado, (D) Diplomado : █
Nombre Esc. : ████████████████████████████████████████████████████████████
Clave Esc. : ████
(M) SEP, (E) SEE : █

Modulo de Alta de Escuelas

<ESC> Salir
```

Primeramente se pide el *nivel* de la escuela que se está dando de alta, para esto, una vez identificado el nivel, se debe presionar la tecla que corresponda a la letra que entre paréntesis precede a la opción correspondiente. A continuación se debe indicar el nombre de la escuela; es necesario aclarar que no es necesario capturar el nivel de la escuela sino solamente el nombre, por ejemplo: *es incorrecto* : “licenciatura en sistemas computarizados”, *es correcto*: “Sistemas Computarizados”. Posteriormente se pide la clave que corresponda a la escuela que se está dando de alta, para finalmente pedir que se seleccione una “M” o una “E”, dependiendo de si la escuela está registrada en la SEP. o en la SEE. Cuando se haya proporcionado toda la información, el sistema pedirá confirmación al usuario para poder realizar el proceso.

En todo momento se mostrará en la parte inferior de la pantalla información referente a las teclas especiales y las funciones que éstas realizan.

4.2.2.- BAJA DE ESCUELAS

Si en algún momento fuera necesario dar de baja alguna de las escuelas tendrá que seleccionar la opción de *bajas* y presionar la tecla <ENTER>, una vez realizado esto, el área de trabajo se podrá ver semejante a la mostrada en la opción anterior.

Para seleccionar la escuela que se dará de baja, bastara con proporcionar el nombre o parte de este y el sistema mostrará automáticamente la información correspondiente. Sin embargo, es posible que existan más de una escuela con el mismo nombre o parte de este pero de diferente nivel, en ese caso, haciendo uso de las tecla de dirección izquierda y derecha, se puede escoger (por la clave), la escuela correcta que se desea eliminar. En cualquier caso el sistema pedirá al usuario la confirmación de la baja.

Es conveniente aclarar que la baja de escuela **no elimina** físicamente del disco la información correspondiente, únicamente deja de estar disponible dentro del sistema. Si por alguna razón fuera necesario volver a dar de alta la escuela con toda la información contenida anteriormente **no use la opción de alta del sistema porque esto destruiría toda la información anterior**, preferentemente, contacte con personal técnico capacitado.

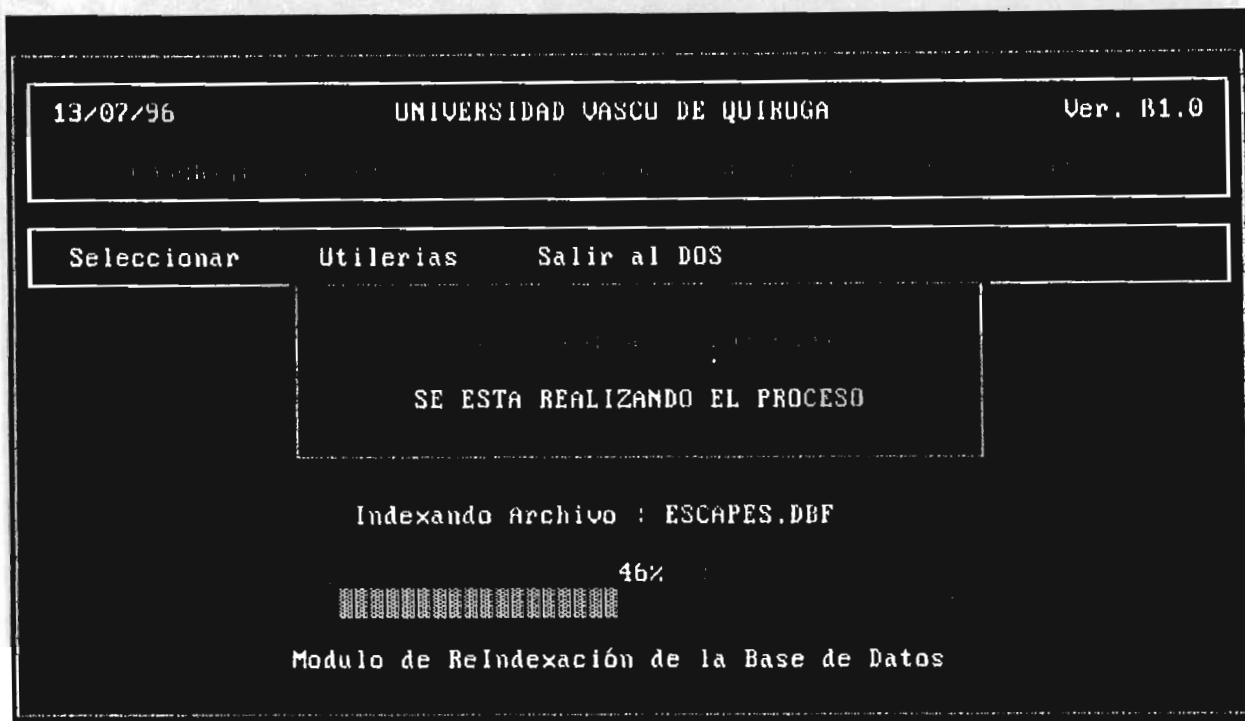
4.2.3.- MODIFICACIÓN DE ESCUELAS

Si en algún momento fuera necesario modificar el nombre, la clave o el registró de alguna de las escuelas tendrá que seleccione la opción de *modificaciones* y presionar la tecla <ENTER>, una vez realizado esto, el área de trabajo se podrá ver semejante a la mostrada en la opción de alta de escuelas mostrada anteriormente.

Para seleccionar la escuela que se desea modificar, bastara con proporcionar el nombre o parte de esté y el sistema mostrará automáticamente la información correspondiente. Sin embargo, es posible que existan más de una escuela con el mismo nombre o parte de esté pero de diferente nivel, en ese caso, haciendo uso de las tecla de dirección izquierda y derecha, se puede escoger (por la clave), la escuela correcta. En cualquier caso el sistema pedirá al usuario la confirmación de la operación.

4.2.4.- REINDEXAR BASES

Si llegase a ocurrir una falla en la corriente eléctrica y/o si el sistema muestra información incoherente o simplemente por seguridad (periódica), será necesario realizar este proceso; para ello bastara con seleccionar la opción *reindexar bases* y presionar la tecla <ENTER>, una vez hecho esto se pedirá al usuario que confirme la intención de realizar el proceso. En caso de que así sea la pantalla se podrá ver semejante a la que se muestra a continuación.



4.3.- OPCIÓN DE SELECCIONAR

Una vez que hayan dado de alta las escuelas, la opción *seleccionar* permitirá al usuario escoger la escuela con la que desea trabajar. En pantalla se mostrará la clave, el nivel y el nombre de todas las escuelas dadas de alta. Para buscar la escuela, el usuario puede hacer uso tanto de las teclas de dirección (flechas) o de las teclas PgDn (AvPág) y PgUp (RePág).

```

13/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
Copyright © 1996, Romero & Uribe. Todos los Derechos Reservados.

Seleccionar      Utilerias      Salir al DOS

Clave Nivel      Nombre de la Escuela
-----
10 Preparatoria  PROPEDEUTICA
21 Licenciatura  SISTEMAS COMPUTARIZADOS

                          Selecciona la Escuela para Trabajar
<F4>  Buscar      <ENTER>  Seleccionar      <ESC>  Salir
    
```

Para seleccionar la escuela bastara con presionar la tecla <ENTER>. Una vez que se seleccione la escuela con la que se desea trabajar, se pasa al *menú principal de trabajo por escuela*, el cual indica el nombre de la escuela seleccionada. Este menú se muestra a continuación.

```

13/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
Copyright © 1996, Romero & Uribe. Todos los Derechos Reservados.

Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior

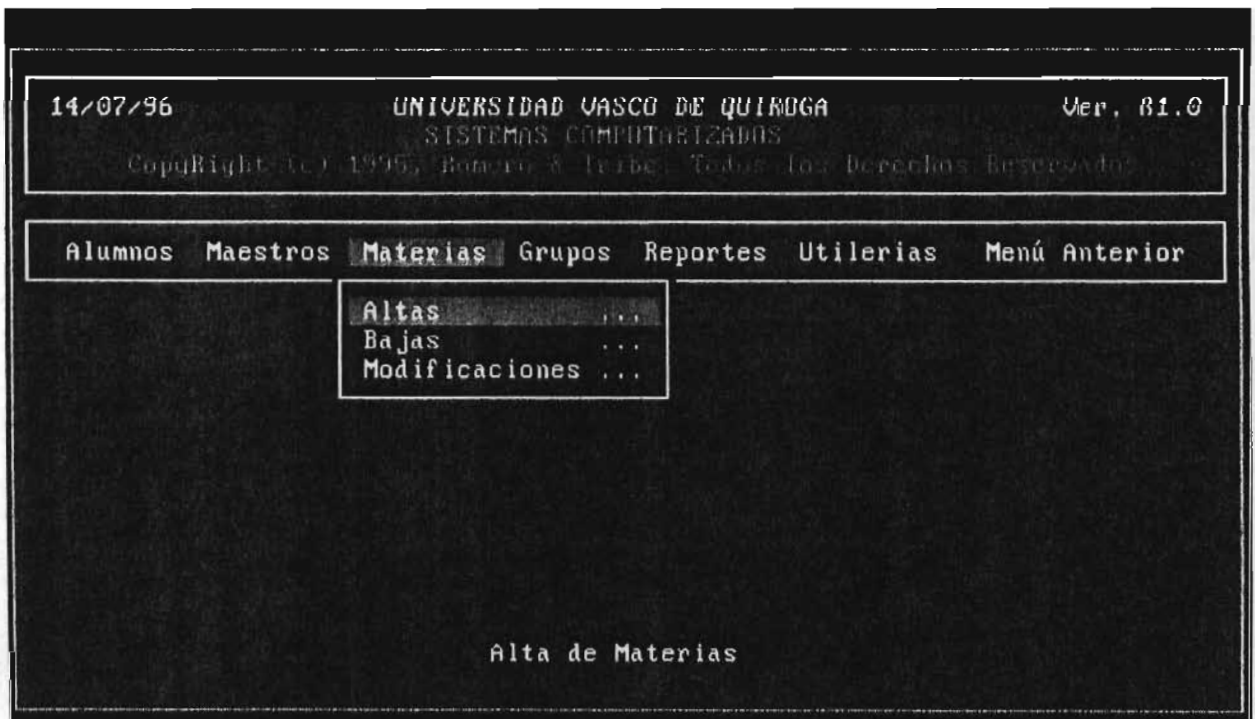
                          Manejo de la informacion referente a Alumnos
    
```

CAPÍTULO 5

"MÓDULO DE MATERIAS"

5.1.- INTRODUCCIÓN

Una vez que se han dado de alta las escuelas, el siguiente paso es el de dar de alta todas las materias de que esta compuesto el plan de estudio de cada una de las escuelas. Para realizar esta operación primeramente seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *materias* y presione la tecla <ENTER>, una vez hecho esto se desplegara un submenú tal como se puede observar a continuación.



5.2.- ALTA DE MATERIAS

Una vez realizado lo anterior, presione la tecla <ENTER> cuando la opción de *altas* esté iluminada, en ese momento el sistema mostrará el área de trabajo correspondiente.

A continuación se pide al usuario que proporcione la clave, el nombre, el número de serie y el área (tronco común, especialización, etc.), de la materia; una vez que se proporcione esta información, el sistema pedirá la confirmación por parte del usuario para poder almacenar la información; en ese momento se podrá observar la pantalla de manera semejante a la que se muestra a continuación.

```
14/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. 01.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Clave    : 00001
Nombre   : INTRODUCCION A LA CIENCIA DE LA COMPUTACION
Nº Serie : 00001      Area : COMPUTACION
Son correctos los datos (S/N) : S

Alta de Materias

<ESC> - Cancelar
```

5.3.- BAJA DE MATERIAS

Cuando sea necesario remover una materia del plan de estudio de alguna de las escuelas es necesario realizar el siguiente proceso:

Seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *materias* y presione la tecla <ENTER>, posteriormente seleccione dentro del submenú la opción de *bajas*.

Para seleccionar la materia que se desea dar de baja, bastará con proporcionar la clave de la materia y presionar la tecla <ENTER>. Sin embargo existe también la posibilidad de seleccionar una materia proporcionando únicamente parte de la clave, de ser así, el sistema mostrará automáticamente la información correspondiente a la primera materia que incluya como parte de su clave el fragmento proporcionado, para que el usuario haciendo uso de las tecla de dirección izquierda y derecha pueda seleccionar la materia adecuada y posteriormente presionar la tecla <ENTER>.

En cualquier caso el sistema pedirá la confirmación de la operación. En este momento el sistema mostrará el área de trabajo de una manera semejante a la que a continuación se muestra.

14/07/96	UNIVERSIDAD VASCO DE QUIRUGA SISTEMAS COMPUTARIZADOS Copyright (c) 1995, Romero & Iribar. Todos los Derechos Reservados	Ver. R1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Clave :	00001	
Nombre :	INTRODUCCION A LA CIENCIA DE LA COMPUTACION	
Nº Serie :	00001	Area : COMPUTACION
Baja de Materias		
<ENTER> Seleccionar	Buscar Otro	<ESC> Salir

Es conveniente mencionar que si la materia está asignada a por lo menos un alumnos, ésta no podrá ser dada de baja.

5.4.- MODIFICACIÓN DE MATERIAS

Cuando sea necesario modificar los datos de una materia del plan de estudio de alguna de las escuelas es necesario realizar el siguiente proceso.

Seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *materias* y presione la tecla <ENTER>, posteriormente seleccione dentro del submenú la opción de *modificaciones*.

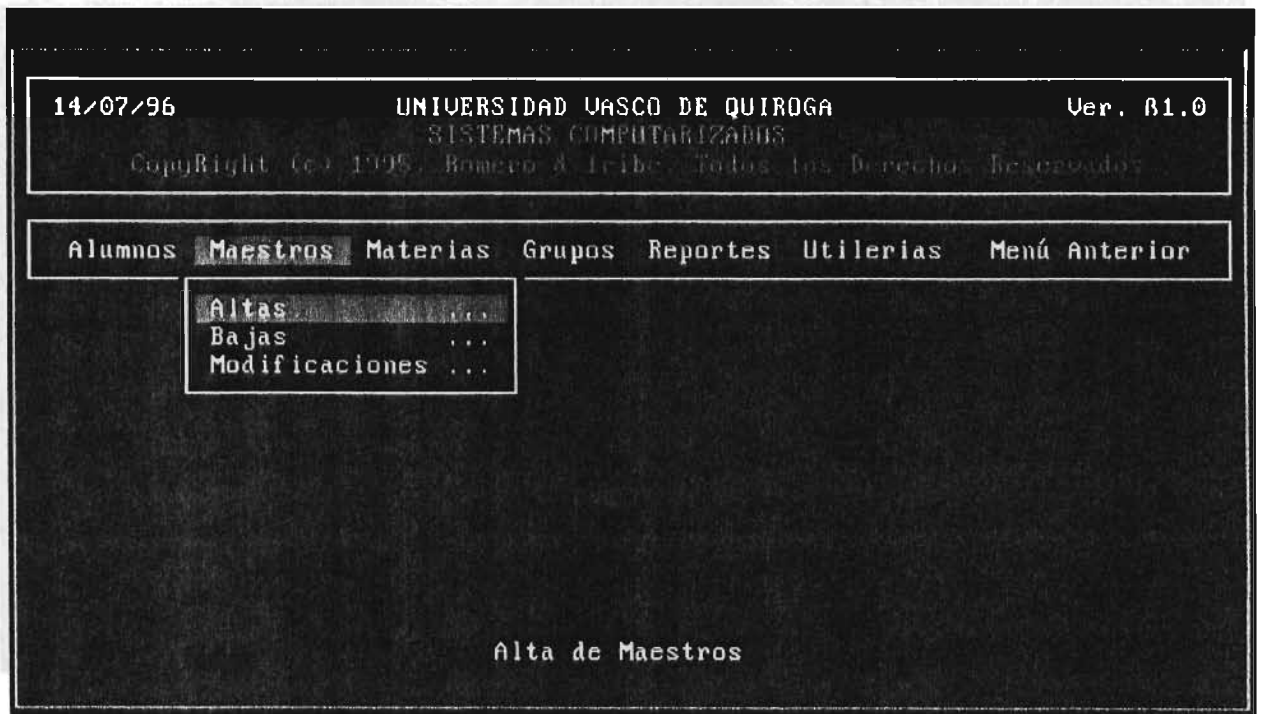
Para seleccionar la materia que se desea modificar, bastara con proporcionar la clave de la materia y presionar la tecla <ENTER>. Sin embargo existe también la posibilidad de seleccionar una materia proporcionando únicamente parte de la clave, de ser así, el sistema mostrará automáticamente la información correspondiente a la primera materia que incluya como parte de su clave el fragmento proporcionado, para que el usuario haciendo uso de las tecla de dirección izquierda y derecha pueda seleccionar la materia adecuada y posteriormente presionar la tecla <ENTER>. Una vez realizado lo anterior, bastará con hacer uso de la tecla de dirección (flechas) para posicionar el cursor sobre el texto a modificar. En cualquier caso el sistema solicitará la confirmación de la operación.

CAPÍTULO 6

"MÓDULO DE MAESTROS"

6.1.- INTRODUCCIÓN

Una vez que se han dado de alta las escuelas y que se han cargado los planes de estudio, el siguiente paso es el de dar de alta la plantilla de maestros de cada una de las escuelas. Para realizar esta operación primeramente seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *maestros* y presione la tecla <ENTER>, una vez hecho esto se desplegará un submenú tal como se puede observar a continuación.



6.2.- ALTA DE MAESTROS

Una vez realizado lo anterior, presione la tecla <ENTER> cuando la opción de *altas* esté iluminada, en ese momento el sistema mostrará el área de trabajo correspondiente.

A continuación se pide al usuario que proporcione la clave, el nombre, el R.F.C., el domicilio, el nivel de estudios, Etc. del maestro; una vez que se proporcione esta información, el sistema pedirá la confirmación por parte del usuario para poder almacenar la información; en ese momento se podrá observar la pantalla de manera semejante a la que se muestra a continuación.

14/07/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. B1.0					
Copyright © 1995, Universidad Vasco de Quiroga, México, D.F.							
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú	Anterior
Clave	: 00001	Nombre	: RAFAEL ROMERO DIAZ BARRIGA				
R.F.C.	: RDR-720915-HA4	Nivel Est	: LICENCIATURA	Sexo	: M		
Domicilio	: CIRC. UACUSECHA ED. 4 INT. 411	C.P	: 58089				
Ciudad	: MORELIA	Colonia	: FRACC. XANGARI				
Telefono	: 91(43) 27-13-00	Fecha Ingreso	: 13/09/95				
Son correctos los datos (S/N) : S							
Alta de Maestros							
<ESC> Cancelar							

6.3.- BAJA DE MAESTROS

Cuando sea necesario remover algún maestro de la plantilla de personal de alguna de las escuelas es necesario realizar el siguiente proceso.

Seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *maestros* y presione la tecla <ENTER>, posteriormente seleccione dentro del submenú la opción de *bajas*.

Para seleccionar al maestro que se desea dar de baja, bastará con proporcionar la clave y presionar la tecla <ENTER>. Sin embargo existe también la posibilidad de seleccionar un maestro proporcionando únicamente parte de la clave, de ser así, el sistema mostrará automáticamente la información correspondiente al primer maestro que incluya como parte de su clave el fragmento proporcionado, para que el usuario haciendo uso de las tecla de dirección izquierda y derecha pueda seleccionar al maestro adecuado y posteriormente presionar la tecla <ENTER>. En cualquier caso el sistema pedirá la confirmación de la operación.

En este momento el sistema mostrará el área de trabajo de una manera semejante a la que a continuación se muestra.

```

14/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. R1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero & Torres. Todos los Derechos Reservados.

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Clave      : 00001          Nombre : RAFAEL ROMERO DIAZ BARRIGA
R.F.C.     : RDDR-720915-HA4  Nivel Est : LICENCIATURA

<ENTER> Seleccionar          Baja de Maestros
                               Buscar Otro
                               <ESC> Salir
    
```

Es conveniente mencionar que si el maestro tiene asignada por lo menos una materia, el sistema permite darlo da baja, siempre y cuando se proporcione la clave del maestro que se hará cargo de la materia; esto es con el fin de no dejar materias sin maestro. En la siguiente ilustración se puede observar un ejemplo de como se mostrará el área de trabajo.

```

02/08/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. R1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero & Torres. Todos los Derechos Reservados.

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Clave      : 00001          Nombre : RAFAEL ROMERO DIAZ BARRIGA
R.F.C.     : RDDR-720915-HA4  Nivel Est : LICENCIATURA

Clave      :                Nombre :
R.F.C.     :                Nivel Est :

                               REEMPLAZANDO MAESTRO
                               <ESC> Cancelar
    
```

6.4.- MODIFICACIÓN DE MAESTROS

Cuando sea necesario modificar los datos de un maestro de alguna de las escuelas es necesario realizar el siguiente proceso.

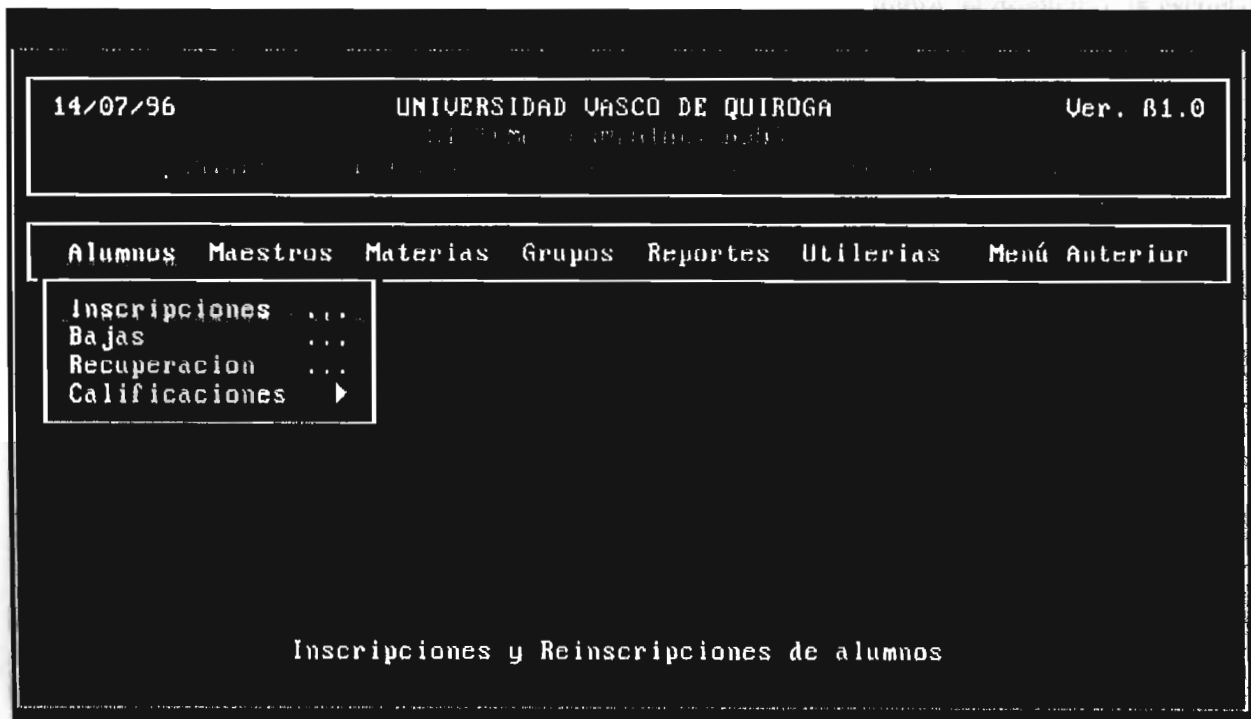
Seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *maestros* y presione la tecla <ENTER>, posteriormente seleccione dentro del submenú la opción de *modificaciones*.

Para seleccionar al maestro que se desea modificar, bastara con proporcionar la clave y presionar la tecla <ENTER>. Sin embargo existe también la posibilidad de seleccionar un maestro proporcionando únicamente parte de la clave, de ser así, el sistema mostrará automáticamente la información correspondiente al primer maestro que incluya como parte de su clave el fragmento proporcionado, para que el usuario haciendo uso de las tecla de dirección izquierda y derecha pueda seleccionar al maestro adecuado y posteriormente presionar la tecla <ENTER>. Una vez realizado lo anterior, bastara con hacer uso de la tecla de dirección (flechas) para posicionar el cursor sobre el texto a modificar. En cualquier caso el sistema pedirá la confirmación de la operación.

CAPÍTULO 7**“MÓDULO DE ALUMNOS”****7.1.- ALUMNOS**

A continuación se detallan los procedimientos necesarios para manipular la información de los alumnos de cada una de las escuelas.

Primeramente para realizar esta operación seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *alumnos* y presione la tecla <ENTER>, una vez hecho esto se desplegará un submenú tal como se puede observar a continuación.



7.1.1.- INSCRIPCIÓN DE ALUMNOS

Una vez que se encuentre en el submenú de alumnos, presione la tecla <ENTER> cuando la opción de *inscripciones* esté iluminada.

Primeramente se debe de proporcionar la matrícula del alumno; cada uno de los alumnos debe contar con una matrícula única la cual debe estar formada por el año de ingreso (los últimos dos dígitos), la clave de la escuela (dos dígitos), y un número consecutivo (tres dígitos).

Nota : Si la matrícula proporcionada ya está asignada a otro alumno, el sistema mostrará la información del alumno correspondiente.

Posteriormente se debe de capturar el semestre (en el caso de inscripción será "1"), el grupo al que será asignado (compuesto por dos números y una letra), el nombre, el domicilio, la escuela de procedencia, Etc., que correspondan al alumno.

Nota : La fecha de inscripción la proporciona el sistema, si ésta es incorrecta, será necesario salir del sistema y hacer uso del comando *DATE* del DOS para cambiar la fecha.

Una vez que sea recibida la confirmación, se almacenará toda la información del alumno y este quedará **inscrito** dentro de la escuela correspondiente.

Nota : A partir de este momento la información del alumno se encuentra disponible para el Sistema de Cuentas por Cobrar.

La siguiente figura muestra un ejemplo de la forma en que se podrá ver el área de trabajo al realizar este proceso.

```

14/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright © 1995 Romero & Leiber. Todos los derechos Reservados.

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Matricula : 9621000      Semestre : 1      Grupo : 1-U
Nombre    : MARGARITA ROMERO DIAZ BARRIGA      Sexo : F
Domicilio : CIRC. UACUSECHA ED. 4 INT. 411
Colonia   : FRACC. XANGARI                      C.P. : 58089
Ciudad    : MORELIA                             Telefono : 91(43) 27-13-00
Esc. Procedencia : C.B.T.I.S. N° 94
Nombre Padre o Tutor : ANGEL ALFONSO ROMERO ROMERO
Tel. Padre o Tutor   : 91(434) 4-00-00
Fecha de Inscripción : 14/07/96
Son correctos los datos (S/N) : S

Inscripciones y Reinscripciones de alumnos
ESC Cancelar
  
```

7.1.2.- REINSCRIPCIÓN DE ALUMNOS

Una vez que se encuentre en el submenú de alumnos, presione la tecla <ENTER> cuando la opción de *inscripciones* esté iluminada.

Una vez que se proporcione al sistema la matrícula del alumno, se mostrará en pantalla la información que le corresponda. Posteriormente se debe **modificar** la información correspondiente al **semestre** y al **grupo**; en caso de ser necesario, el sistema permite modificar toda la información correspondiente al alumno.

Una vez que sea recibida la confirmación, se actualizará la información del alumno y este quedara **reinscrito** dentro de la escuela correspondiente.

Nota. : A partir de este momento la información del alumno nuevamente queda disponible para el Sistema de Cuentas por Cobrar.

7.1.3.- BAJA DE ALUMNOS

Una vez que se encuentre en el submenú de alumnos, presione la tecla <ENTER> cuando la opción de **bajas** esté iluminada.

Para seleccionar al alumno que se desea dar de baja, bastara con proporcionar la matrícula y presionar la tecla <ENTER>. Sin embargo existe también la posibilidad de seleccionar un alumno proporcionando únicamente parte de la matrícula, de ser así, el sistema mostrará automáticamente la información correspondiente al primer alumno que incluya como parte de su matrícula el fragmento proporcionado, para que haciendo uso de las tecla de dirección izquierda y derecha pueda seleccionar al alumno adecuado y posteriormente presionar la tecla <ENTER>.

Una vez que se haya seleccionada al alumno a ser dado de baja, el sistema preguntara si la baja es **temporal** o **definitiva**. Si se indica que la baja es **definitiva**, la información del alumno será eliminada definitivamente, si se indica que la baja es **temporal**, la información del alumno podrá ser recuperada posteriormente.

Cualquiera que sea el tipo de baja seleccionado, el sistema pedirá confirmación para realizar el proceso. En ese momento se podrá observar la pantalla de manera semejante a la que se muestra a continuación.

14/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright (c) 1995. Romero & Torres. Todos los Derechos Reservados	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Matricula :	9621001	Semestre : 1
Nombre :	ROSA MARIA VARELA CAZAREZ	Grupo : 14
Domicilio :	5 DE MAYO # 39	Sexo : F
Colonia :	J. CARRILLO ARRIAGA	C.P. : 61600
Ciudad :	PATZCUARO	Telefono :
Esc. Procedencia :	C.B.T.I.S. 94	
Nombre Padre o Tutor :	ANGELINA CAZAREZ GARCIA	
Tel. Padre o Tutor :		
Fecha de Inscripción :	14/07/96	Tipo baja: (T)emporal (D)efinitiva
Baja Temporal o Total de Alumnos		
Cancelar		

7.1.4.- RECUPERACIÓN DE ALUMNOS DADOS DE BAJA TEMPORAL

Una vez que se encuentre en el submenú de alumnos, presione la tecla <ENTER> cuando la opción de *recuperación* esté iluminada.

Para seleccionar al alumno que se desea recuperar, bastará con proporcionar la matrícula y presionar la tecla <ENTER>. Sin embargo existe también la posibilidad de seleccionar al alumno proporcionando únicamente parte de la matrícula, de ser así, el sistema mostrará automáticamente la información correspondiente al primer alumno dado de baja temporalmente que incluya como parte de su matrícula el fragmento proporcionado, para que haciendo uso de las teclas de dirección izquierda y derecha pueda seleccionar al alumno adecuado y posteriormente presionar la tecla <ENTER>.

Una vez que se haya seleccionada al alumno a “recuperar”, el sistema pedirá confirmación para realizar la operación, si la respuesta es positiva, la información del alumno quedará disponible nuevamente. En ese momento se podrá observar la pantalla de manera semejante a la que se muestra a continuación.

14/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS	Ver. B1.0
Copyright (c) 1995, Romero & Tribe, Todos los Derechos Reservados		
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Matricula :	9621001	Semestre : 1
Nombre :	ROSA MARIA VARELA CAZAREZ	Grupo : 1U
Domicilio :	5 DE MAYO # 39	Sexo : F
Colonia :	J. CARRILLO ARRIAGA	C.P. : 61600
Ciudad :	PATZCUARO	Telefono :
Esc. Procedencia :	C.B.T.I.S. 94	
Nombre Padre o Tutor :	ANGELINA CAZAREZ GARCIA	
Tel. Padre o Tutor :		
Fecha de Inscripción :	14/07/96	
Esta seguro en RECUPERABLE :	N	
Recuperación de Alumnos dados de Baja		
		Cancelar

Cualquiera que sea el tipo de baja seleccionado, el sistema pedirá confirmación para realizar el proceso. En ese momento se podrá observar la pantalla de manera semejante a la que se muestra a continuación.

14/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS	Ver. B1.0							
Copyright (c) 1995. Bombero & Leibe. Todos los Derechos Reservados.									
<table border="1"> <tr> <td>Alumnos</td> <td>Maestros</td> <td>Materias</td> <td>Grupos</td> <td>Reportes</td> <td>Utilerias</td> <td>Menú Anterior</td> </tr> </table>			Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior			
Matricula :	9621001	Semestre :	1	Grupo :	111				
Nombre :	ROSA MARIA VARELA CAZAREZ	Sexo :	F						
Domicilio :	5 DE MAYO # 39								
Colonia :	J. CARRILLO ARRIAGA	C.P. :	61600						
Ciudad :	PATZCUARO	Telefono :							
Esc. Procedencia :	C.B.T.I.S. 94								
Nombre Padre o Tutor :	ANGELINA CAZAREZ GARCIA								
Tel. Padre o Tutor :									
Fecha de Inscripción :	14/07/96	Tipo baja:	(T)emporal (D)efinitiva						
Baja Temporal o Total de Alumnos						Cancelar			

7.1.4.- RECUPERACIÓN DE ALUMNOS DADOS DE BAJA TEMPORAL

Una vez que se encuentre en el submenú de alumnos, presione la tecla <ENTER> cuando la opción de *recuperación* esté iluminada.

Para seleccionar al alumno que se desea recuperar, bastará con proporcionar la matrícula y presionar la tecla <ENTER>. Sin embargo existe también la posibilidad de seleccionar al alumno proporcionando únicamente parte de la matrícula, de ser así, el sistema mostrará automáticamente la información correspondiente al primer alumno dado de baja temporalmente que incluya como parte de su matrícula el fragmento proporcionado, para que haciendo uso de las tecla de dirección izquierda y derecha pueda seleccionar al alumno adecuado y posteriormente presionar la tecla <ENTER>.

Una vez que se haya seleccionada al alumno a “recuperar”, el sistema pedirá confirmación para realizar la operación, si la respuesta es positiva, la información del alumno quedara disponible nuevamente. En ese momento se podrá observar la pantalla de manera semejante a la que se muestra a continuación.

```

14/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero & Tribe, Todos los Derechos Reservados

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

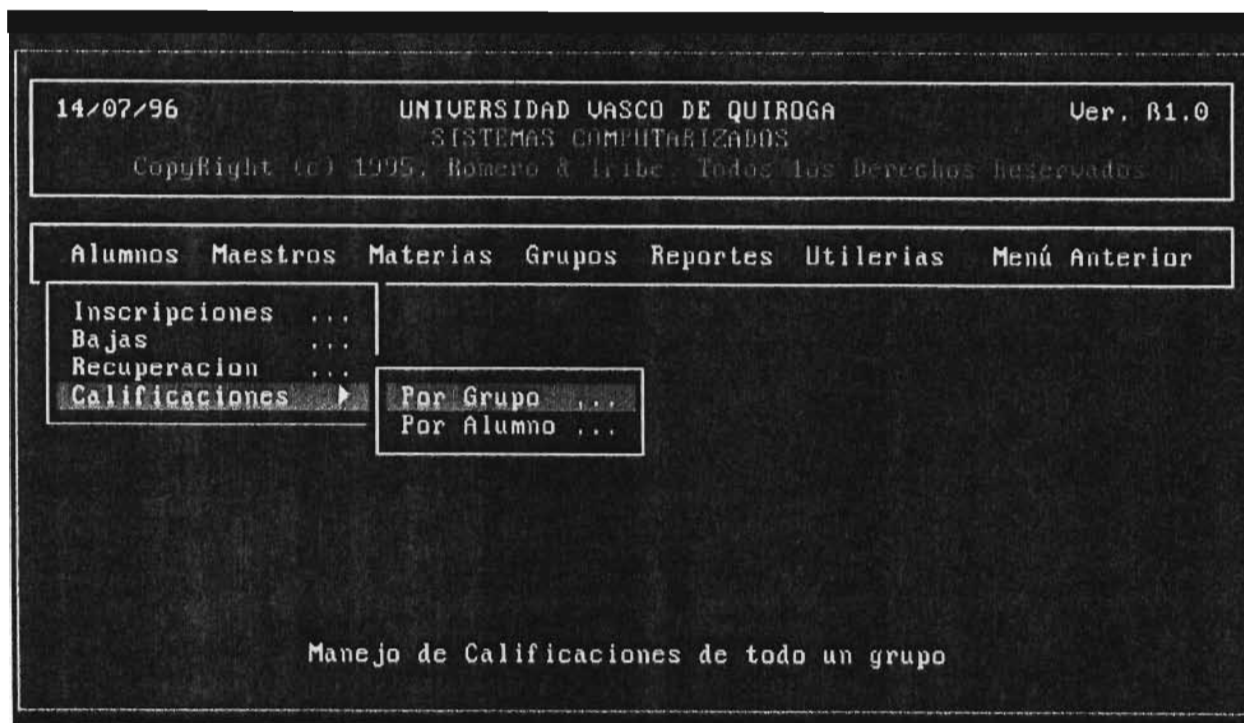
Matricula : 9521001      Semestre : 1      Grupo : 1U
Nombre     : ROSA MARIA VARELA CAZAREZ      Sexo : F
Domicilio  : 5 DE MAYO # 39
Colonia    : J. CARRILLO ARRIAGA      C.P. : 61600
Ciudad     : PATZCUARO      Telefono :
Esc. Procedencia : C.B.T.I.S. 94
Nombre Padre o Tutor : ANGELINA CAZAREZ GARCIA
Tel. Padre o Tutor   :
Fecha de Inscripción : 14/07/96
Esta seguro en RECUPERARLO      N

Recuperación de Alumnos dados de Baja      ESC Cancelar
    
```

7.2 .- CALIFICACIONES

A continuación se detallan los procedimientos necesarios para la captura de las calificaciones y de las asistencias o inasistencias.

Primeramente para realizar esta operación seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *alumnos* y presione la tecla <ENTER>, posteriormente, seleccione la opción de *calificaciones* y presione la tecla <ENTER>, una vez hecho esto se desplegará un submenú tal como se puede observar a continuación.



7.2.1 .- CAPTURA DE CALIFICACIONES POR GRUPO

Una vez que se encuentre en el submenú de calificaciones y cuando la opción de *por grupo* esté iluminada presione la tecla <ENTER>, al seleccionar esta opción, el sistema permitirá la captura de las calificaciones (por materia), de un grupo específico.

Primeramente el sistema pedirá el grupo al que se desee acceder, posteriormente solicitará la clave de la materia a la que correspondan las calificaciones a capturar, (si no se tiene a disposición la clave de la materia, bastara con presionar la tecla F1 para desplegar las materias correspondiente al plan de estudios de la escuela seleccionada; haciendo uso de las teclas de dirección (flechas), puede buscar la materia correspondiente y presionar la tecla <ENTER> para seleccionarla), a continuación es necesario indicar el número de evaluación (en el caso de las licenciaturas del 1 al 3, en las preparatorias del 1 al 8 y en postgrados solamente el 1). En ese momento se podrá observar una pantalla semejante a la que se muestra a continuación.

```
04/08/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. 01.0
                   SISTEMAS COMPUTARIZADOS
                   Copyright (c) 1995, Romeo & Iribar. Todos los derechos reservados.

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Grupo    : 1-1
Materia  : 00003
Evaluación : 1

Manejo de Calificaciones de todo un grupo
<ESC> -- Cancelar
```

Una vez que se haya proporcionado la información anterior, el sistema mostrará en forma de tabla los nombre de los alumnos, la calificación, en el caso de licenciaturas las asistencias y en el caso de las preparatorias las inasistencias; en la parte superior de la tabla podrá observarse el nombre de la materia seleccionada.

Para realizar la captura solamente tiene que hacer uso de las teclas de dirección (flechas), para buscar al alumno y presionar la tecla <ENTER> sobre la columna correspondiente, una vez hecho esto el sistema permitirá la captura de la información, para terminar y pasar a la siguiente columna o al siguiente alumno presione nuevamente la tecla <ENTER>. Este proceso se tiene que repetir tantas veces como sea necesario.

En caso de que se desee **MODIFICAR UNA CALIFICACIÓN**, se tiene que realizar este proceso de la misma manera, solamente que en este caso solamente se captura la calificación que se desee modificar. Las calificaciones del resto de los alumnos no se verán afectadas.

En el momento de la captura se podrá observar una pantalla semejante a la que se muestra a continuación.

Alumno	Calificación	Asistencias
AGUILAR RODRIGUEZ AIDA ILIANA	0.0	0
CELAYA MAGAÑA ROGER IVAN	0.0	0
CORREA GARCIA MARITZA	0.0	0
ESCUTIA HERNANDEZ JUAN GABRIEL	0.0	0
GARCIA CARRILLO GISELA	0.0	0
GUTIERREZ NUÑEZ RAUL	0.0	0
HERRERA MALDONADO MARIA GORETTI	0.0	0
HERRERA MUÑOZ SAMARA	0.0	0
HOWLAND BARRIGA JUAN JORGE	0.0	0
HUERTA AGUILAR OMAR	0.0	0

Manejo de Calificaciones de todo un grupo
 ← → - Buscar <ENTER> Seleccionar para Capturar <ESC> Salir

7.2.2 .- CAPTURA DE CALIFICACIONES POR ALUMNO

Una vez que se encuentre en el submenú de calificaciones y cuando la opción de *por alumno* esté iluminada presione la tecla <ENTER>, al seleccionar esta opción, el sistema permitirá la captura de las calificaciones de todas las por materia asignadas a un alumno.

Primeramente el sistema pedirá el grupo al que se desee acceder, posteriormente pedirá la matrícula del alumno al que correspondan las calificaciones a capturar, a continuación es necesario indicar el número de evaluación (en el caso de las licenciaturas del 1 al 3, en las preparatorias del 1 al 8 y en postgrados solamente el 1). En ese momento se podrá observar una pantalla semejante a la que se muestra a continuación.

```
04/08/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995 Romero & Tribe. Todos los Derechos Reservados.

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Grupo      : 1-U
Clave Alumno : 9621000
Evaluación :

Manejo de Calificaciones por alumno específico

Salir
```

Una vez que se haya proporcionado la información anterior, el sistema mostrará en forma de tabla las materias, la calificación, en el caso de licenciaturas las asistencias y en el caso de las preparatorias las inasistencias; en la parte superior de la tabla podrá observarse el nombre del alumno.

Para realizar la captura solamente tiene que hacer uso de las teclas de dirección (flechas), para buscar la materia y presionar la tecla <ENTER> sobre la columna correspondiente, una vez hecho esto el sistema permitirá la captura de la información, para terminar y pasar a la siguiente columna o la siguiente materia presione nuevamente la tecla <ENTER>. Este proceso se tiene que repetir tantas veces como sea necesario.

En caso de que se desee **MODIFICAR UNA CALIFICACIÓN**, se tiene que realizar este proceso de la misma manera, solamente que en este caso solamente se captura la calificación que se desee modificar. Las calificaciones del resto de las materias no se verán afectadas.

En el momento de la captura se podrá observar una pantalla semejante a la que se muestra a continuación.

Materia	Calificación
MATEMATICAS I	10.0
CONTABILIDAD	9.0
INTRODUCCION A LA ADMINISTRACION	9.0
INTRODUCCION A LA ECONOMIA	8.5
PROCESAMIENTO DE DATOS	10.0
FILOSOFIA DEL HOMBRE	8.0

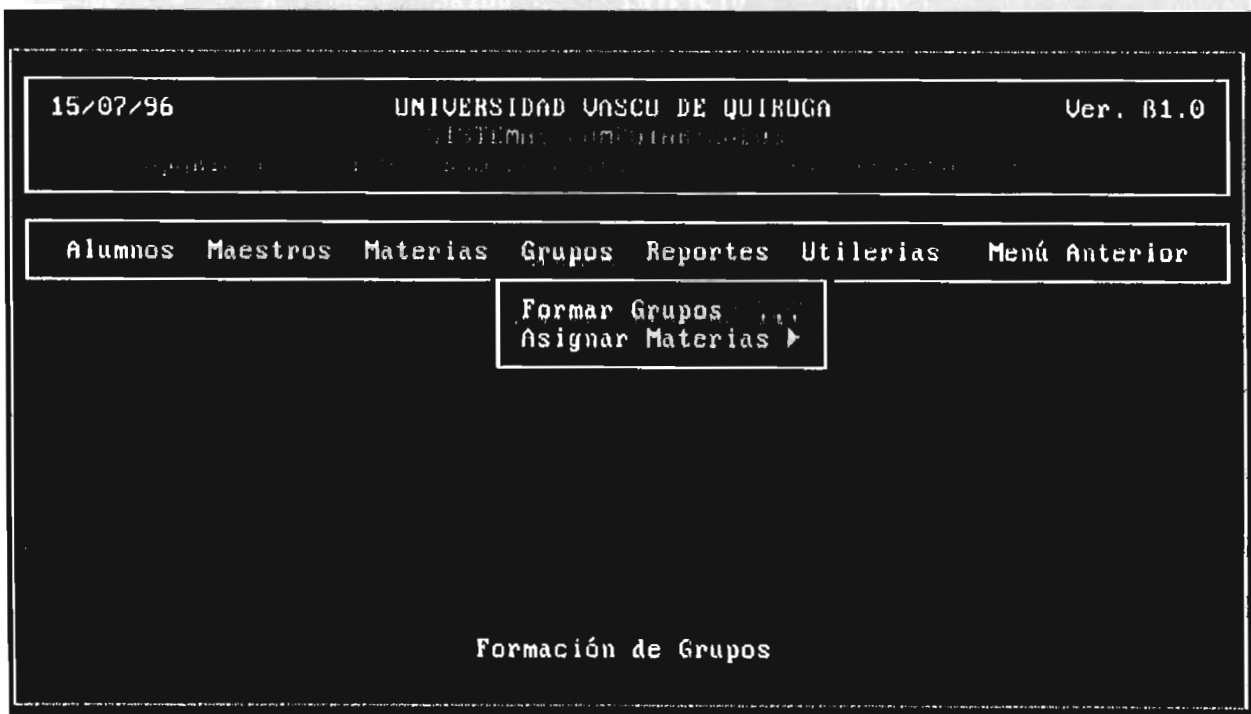
Manejo de Calificaciones por alumno especifico

F1 <←> <→> Buscar <ENTER> Selecccionar para Capturar <ESC> Salir

CAPÍTULO 8**"MÓDULO DE GRUPOS"****8.1 .- INTRODUCCIÓN**

El proceso de la formación de grupos consisten en asignar maestros a cada una de las materias, así como los horarios en que serán impartidas en los distintos grupos.

Primeramente para realizar esta operación seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *grupos* y presione la tecla <ENTER>, una vez hecho esto se desplegara un submenú tal como se puede observar a continuación.



8.2 .- FORMAR GRUPOS

Una vez que se encuentre en el submenú de grupos y cuando la opción de *formar grupos* esté iluminada presione la tecla <ENTER>. Lo primero que se tiene que indicar es el grupo que se desea formar, en este momento el sistema mostrará una pantalla semejante a la que se muestra a continuación.

```

15/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. R1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero & Leibe. Todos los Derechos Reservados

Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior

Grupo : 1-U      Clave Materia :      Clave Profesor :
Materia :
Profesor :

H O R A R I O
De :   A :   Salón :   Edificio :   Día :
De :   A :   Salón :   Edificio :   Día :
De :   A :   Salón :   Edificio :   Día :
De :   A :   Salón :   Edificio :   Día :

                                Formación de Grupos

F1  Ayuda                                Cancelar

```

Posteriormente se pide la clave de la materia que formará parte de la carga de materias del grupo, si no se tiene a la mano la clave de la materia, solamente tiene que presionar la tecla F1 y en ese momento el sistema desplegará en forma de lista, todas las materias que conforman el plan de estudios de la escuela seleccionada; haciendo uso de las teclas de dirección (flechas), busque la materia y presione la tecla <ENTER> para seleccionarla. En este momento podrá verse la pantalla semejante a la que se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright (c) 1995, Romero & Uribe. Todos los Derechos Reservados	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Grupo : 1-U	Clave Materia :	Clave Profesor :
Materia :		
Clave	Materia	
00001	INTRODUCCION A LA CIENCIA DE LA COMPUTACION	
00002	INTRODUCCION A LA VIDA UNIVERSITARIA	
00003	MATEMATICAS I	
00004	CONTABILIDAD	
00005	INTRODUCCION A LA ADMINISTRACION	
<F1> - Buscar	Formación de Grupos <ENTER> - Seleccionar	<ESC> - Salir

El siguiente paso es el de designar al profesor que quedara a cargo de la materia seleccionada, para ello se pide la clave del profesor y al igual que con el proceso anterior, si no se tiene la clave simplemente presione la tecla F1 y el sistema desplegara en forma de lista todos los maestros que estén dados de alta en la escuela seleccionada; haciendo uso de las teclas de dirección (flechas), busque al maestro y presione la tecla <ENTER> para seleccionarlo(a).

Por ultimo se tiene que especificar el horario de la materia, así como el día (para indicar el día, seleccione una de las letras en color rojo que se encuentran entre paréntesis en la esquina inferior derecha de la pantalla), el salón y el edificio en que será impartida. En ese momento se podrá observar una pantalla semejante a la que se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright © C. I.P.P., Romero A. Leide. Todos los Derechos Reservados	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Grupo : 15U	Clave Materia : 00001	Clave Profesor : 00001
Materia :	INTRODUCCION A LA CIENCIA DE LA COMPUTACION	
Profesor :	RAFAEL ROMERO DIAZ BARRIGA	
H O R A R I O		
De : 08:00	A : 09:00	Salón : 08 Edificio : 1 Día : L
De : 08:00	A : 10:00	Salón : 08 Edificio : 1 Día : U
De : :	A : :	Salón : Edificio : Día :
De : :	A : :	Salón : Edificio : Día :
Son correctos los datos (S/N) : S		
Formación de Grupos		
		<ESC> Cancelar

Este proceso se tendrá que repetir tantas veces como materias se deseen asignar al grupo especificado.

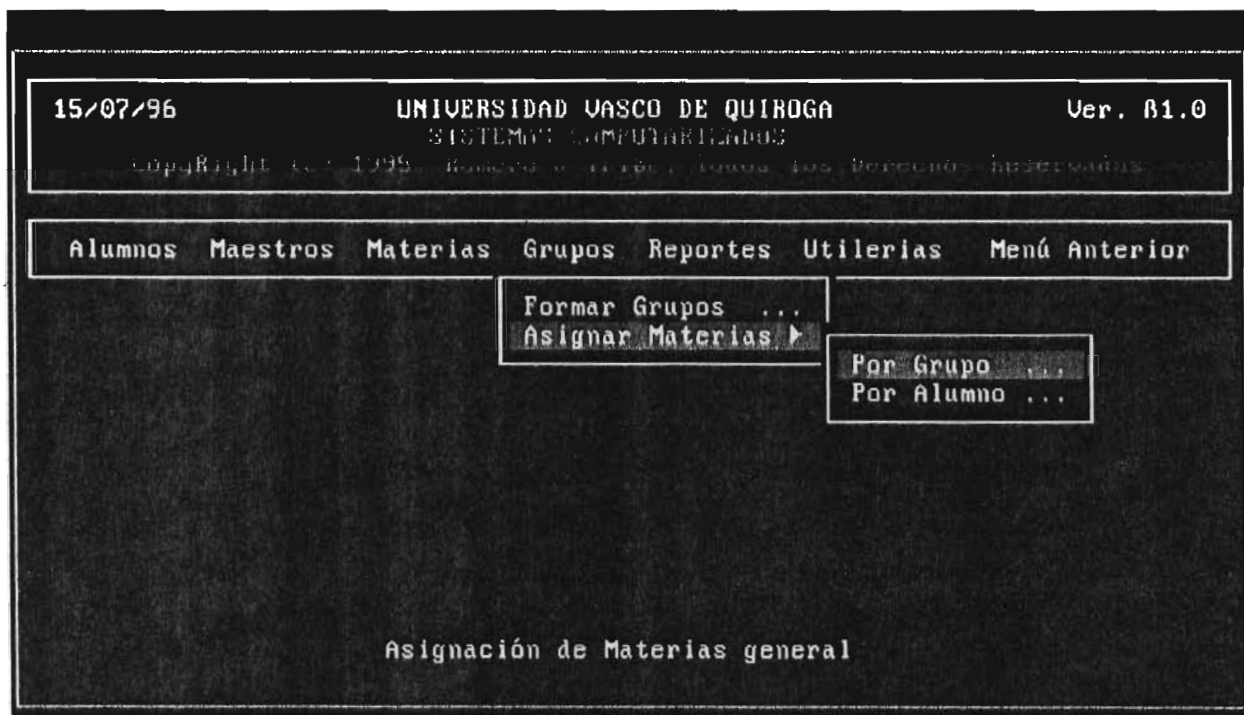
8.2.1 .- MODIFICAR GRUPOS

Si se desea cambiar al profesor asignado a una materia asignada a un grupo o si se desea cambiar el horario, siga el proceso descrito en el apartado anterior; la diferencia con respecto al proceso anterior es que al especificar la clave de la materia y presionar la tecla <ENTER>, el sistema mostrará la información correspondiente. Para modificarla tendrá que hacer uso de las teclas de dirección (flechas), para buscar y seleccionar la parte que se desee cambiar.

8.3.- ASIGNAR MATERIAS A LOS GRUPOS

El proceso de la asignación de materias a los grupos consisten en especificar cuales de las materias asignadas al grupo serán cursadas por los alumnos.

Primeramente para realizar esta operación seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *grupos* y presione la tecla <ENTER>, posteriormente dentro del submenú, seleccione la opción de *asignar materias*, una vez hecho esto se desplegara un nuevo menú tal como se puede observar a continuación



8.3.1.- ASIGNAR MATERIAS POR GRUPO

Una vez que se encuentre en el submenú de *asignar materias* y cuando la opción de *por grupos* esté iluminada presione la tecla <ENTER>, al seleccionar esta opción, el sistema solicitará que se teclee el grupo al cual se le asignaran las materias, posteriormente el sistema mostrará el área de trabajo dividida en dos ventanas. La ventana superior muestra las materias disponibles para el grupo seleccionado (ver apartado 8,2); la ventana inferior muestra las materias asignadas al grupo seleccionado; en la parte inferior de la pantalla se pueden ver las teclas necesarias para realizar la asignación y para cambiar de ventana.

Para seleccionar las materias que se desean asignar al grupo, bastara con buscarlas haciendo uso de las teclas de dirección (flechas), y seleccionarlas presionando la tecla <ENTER> (al hacer esto notara que la materia aparece en la ventana inferior). Este proceso tiene que ser repetido tantas veces como materias se deseen asignar al grupo. Durante este proceso el sistema mostrará una pantalla semejante a la que se muestra a continuación.

Grupo	Materia Registrada	Area
1U	INTRODUCCION A LA CIENCIA DE LA COMPUTACION	COMPUTACION
1U	INTRODUCCION A LA VIDA UNIVERSITARIA	TRONCO COMUN
1U	MATEMATICAS I	TRONCO COMUN
1U	CONTABILIDAD	TRONCO COMUN
Grupo	Materia Asignada	Area
1U	INTRODUCCION A LA CIENCIA DE LA COMPUTACION	COMPUTACION

Asignación de Materias general

Buscar ENTER Selecccionar ESC Cambio de Ventana Salir

8.3.1.1 .- MODIFICAR ASIGNACIONES DE MATERIAS POR**GRUPO**

Si por alguna razón fuera necesario eliminar o cambiar una materia a un grupo, siga el siguiente proceso. Una vez que se encuentre en el submenú de *asignar materias* y cuando la opción de *por grupos* esté iluminada presione la tecla <ENTER>, al seleccionar esta opción, el sistema solicitará el grupo y posteriormente mostrará la pantalla en la forma descrita en el apartado anterior.

Para pasar a la ventana inferior, presione la tecla F6. Una vez que se encuentre en la segunda ventana, haciendo uso de las tecla de dirección (flechas), busque la materia que se desee eliminar de la carga de materias del grupo, a continuación presione la tecla <SUP> (suprimir); en ese momento notara que la materia desaparece de la ventana. Repita este proceso tantas veces como materias a eliminar. Una vez terminado el proceso, presione nuevamente la tecla F6 para regresar a la ventana superior (es necesario encontrarse en la ventana superior para poder dar por terminado el proceso, mientras se encuentre en la ventana inferior la tecla <ESC> no tiene efecto). Si se desea asignar otra materia en el lugar de la que se ha dado de baja, siga el proceso descrito en el apartado anterior.

NOTA MUY IMPORTANTE .- La asignación de materias “por grupo”, solamente tiene efecto sobre los alumnos ya inscritos, si un alumno se inscribe posteriormente a la asignación de materias por grupo, consulte el apartado 8.3.2 de este manual. Igualmente es importante destacar que una vez que se hayan capturado calificaciones *no será posible la modificación de la asignación de materias* por grupo, en este caso, únicamente se puede modificar la asignación de materias de forma individual para cada alumno tal como se describe en el apartado 8.3.2.1 de este manual.

8.3.2 .- ASIGNAR MATERIAS POR ALUMNO

Este proceso es el indicado para asignar materias a los alumnos que se inscribieron en forma extemporánea o bien, a los alumnos que no cursan todas las materias que cursan el resto de los alumnos del grupo como es el caso de los alumnos que están revalidando materias o que están cursando una materia no aprobada en semestres anteriores. Para realizar este proceso, bastara con seguir los siguientes pasos.

Una vez que se encuentre en el submenú de *asignar materias* y cuando la opción de *por alumno* esté iluminada presione la tecla <ENTER>, al seleccionar esta opción, el sistema solicitará que se teclee el grupo y la matrícula del alumno al cual se le asignaran las materias, posteriormente el sistema mostrará el área de trabajo dividida en dos ventanas. La ventana superior muestra las materias disponibles para el *grupo* seleccionado (ver apartado 8,2); la ventana inferior muestra las materias asignadas al *alumno* seleccionado; en la parte inferior de la pantalla se pueden ver las teclas necesarias para realizar la asignación y para cambiar de ventana, también puede observarse en la zona superior de las ventanas, el nombre del alumno seleccionado.

Para seleccionar las materias que se desean asignar al alumno, bastara con buscarlas haciendo uso de las teclas de dirección (flechas), y seleccionarlas presionando la tecla <ENTER> (al hacer esto notara que la materia aparece en la ventana inferior). Este proceso tiene que ser repetido tantas veces como materias se deseen asignar al alumno. Durante este proceso el sistema mostrará una pantalla semejante a la que se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS	Ver. B1.0
Copyright (c) 1995, Romero & Tribe. Todos los Derechos Reservados		
Asignación de Materias por Alumno Especifico		
Grupo	Materia Registrada	Area
1U	INTRODUCCION A LA CIENCIA DE LA COMPUTACION	COMPUTACION
1U	INTRODUCCION A LA VIDA UNIVERSITARIA	TRONCO COMUN
1U	MATEMATICAS I	TRONCO COMUN
Grupo	Materia Asignada	Area
1U	INTRODUCCION A LA CIENCIA DE LA COMPUTACION	COMPUTACION
Asignación de Materias por alumno especifico		
<F1> - Buscar <ENTER> - Seleccionar <F6> - Cambio de Ventana <ESC> - Salir		

8.3.2.1 .- MODIFICAR ASIGNACIONES DE MATERIAS POR

ALUMNO

Si por alguna razón fuera necesario eliminar o cambiar una materia a un alumno, siga el siguiente proceso. Una vez que se encuentre en el submenú de *asignar materias* y cuando la opción de *por grupos* esté iluminada presione la tecla <ENTER>, al seleccionar esta opción, el sistema solicitará el grupo y la matrícula del alumno, posteriormente mostrará la pantalla en la forma descrita en el apartado anterior.

Para pasar a la ventana inferior, presione la tecla F6. Una vez que se encuentre en la segunda ventana, haciendo uso de las tecla de dirección (flechas), busque la materia que se desee eliminar de la carga de materias del alumno, a continuación presione la tecla <SUP> (suprimir); en ese momento notara que la materia desaparece de la ventana. Repita este proceso tantas veces como materias a eliminar. Una vez terminado el proceso, presione nuevamente la tecla F6 para regresar a la ventana superior (es necesario encontrarse en la ventana superior para poder dar por terminado el proceso, mientras se encuentre en la ventana inferior la tecla <ESC> no tiene efecto). Si se desea asignar otra materia en el lugar de la que se ha dado de baja, siga el proceso descrito en el apartado anterior.

NOTA MUY IMPORTANTE .- En caso de que el alumno ya tenga calificaciones capturadas de la materia eliminada, éstas se perderán definitivamente.

CAPÍTULO 9**“MÓDULO DE REPORTES”****9.1.- INTRODUCCIÓN**

Una vez que se han dado de alta las escuelas, que se han cargado los planes de estudio, se ha dado de alta la plantilla de maestros, se han inscrito los alumnos, se han formado los grupos y se han asignado las materias, cuando se puede comenzar a imprimir los reportes que genera el sistema. Para realizar esta operación primeramente seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *reportes* y presione la tecla <ENTER>.

Una vez hecho esto se desplegara uno de los dos menús que se muestran a continuación; la diferencia es que el primer menú que se muestra es el que se vera al estar trabajando en una licenciatura y el segundo que muestra la opción de *concentrado de calificaciones* atenuada, es el que se vera al trabajar con una preparatoria o con un postgrado. (En preparatoria no se genera este reporte ya que se emiten boletas de calificaciones cada evaluación, a diferencia de la licenciatura en donde se emiten las boletas hasta el final del semestre).

MENÚ DE REPORTES DE LICENCIATURAS

15/07/96 UNIVERSIDAD VASCO DE QUIROGA Ver. R1.0
 SISTEMAS COMPUTARIZADOS
 Copyright © 1995, Romero & Iribe. Todos los Derechos Reservados

Alumnos Maestros Materias Grupos **Reportes** Utilerias Menú Anterior

- Boletas de Calificaciones ▶
- Calificaciones ▶
- Credenciales ▶
- Concentrado de calificaciones ▶
- Constancia de Estudios ...
- Listas de Asistencia ▶
- Actualización del KARDEX
- Impresión de CFIYA ...

Imprime en papel las boletas de calificaciones

MENÚ DE REPORTES DE PREPARATORIA Y POSTGRADOS

15/07/96 UNIVERSIDAD VASCO DE QUIROGA Ver. R1.0
 FARMACÉUTICA
 Copyright © 1995, Romero & Iribe. Todos los Derechos Reservados

Alumnos Maestros Materias Grupos **Reportes** Utilerias Menú Anterior

- Boletas de Calificaciones ... ▶
- Calificaciones ▶
- Credenciales ▶
- Constancia de Estudios ...
- Listas de Asistencia ▶
- Actualización del KARDEX
- Impresión de CFIYA ...

Imprime en papel las boletas de calificaciones

9.2.- LISTAS DE ASISTENCIA

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *listas de asistencia* esté iluminada presione la tecla <ENTER>, al realizar esto, se desplegara un nuevo menú tal como se muestra a continuación.

15/07/96
Ver. R1.0

UNIVERSIDAD VASCO DE QUIROGA
SISTEMAS COMPUTACIONALES

Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior

Boletas de Calificaciones	▶	
Calificaciones	▶	
Credenciales	▶	
Concentrado de calificaciones	▶	
Constancia de Estudios	...	
Listas de Asistencia	▶	Por Grupo
Actualización del KARDEX	▶	Por Materia
Impresión de CFIYA	...	

Imprime todas las listas de asistencia de un Grupo

9.2.1.- LISTAS DE ASISTENCIA POR GRUPO

La impresión de listas de asistencia es una tarea que se tiene que realizar frecuentemente, por lo que la opción mas recomendable es la de imprimir todas las listas de un mismo grupo al mismo tiempo, esto con el fin de simplificar la distribución de las mismas, así como disminuir también el tiempo de elaboración las listas de las escuelas.

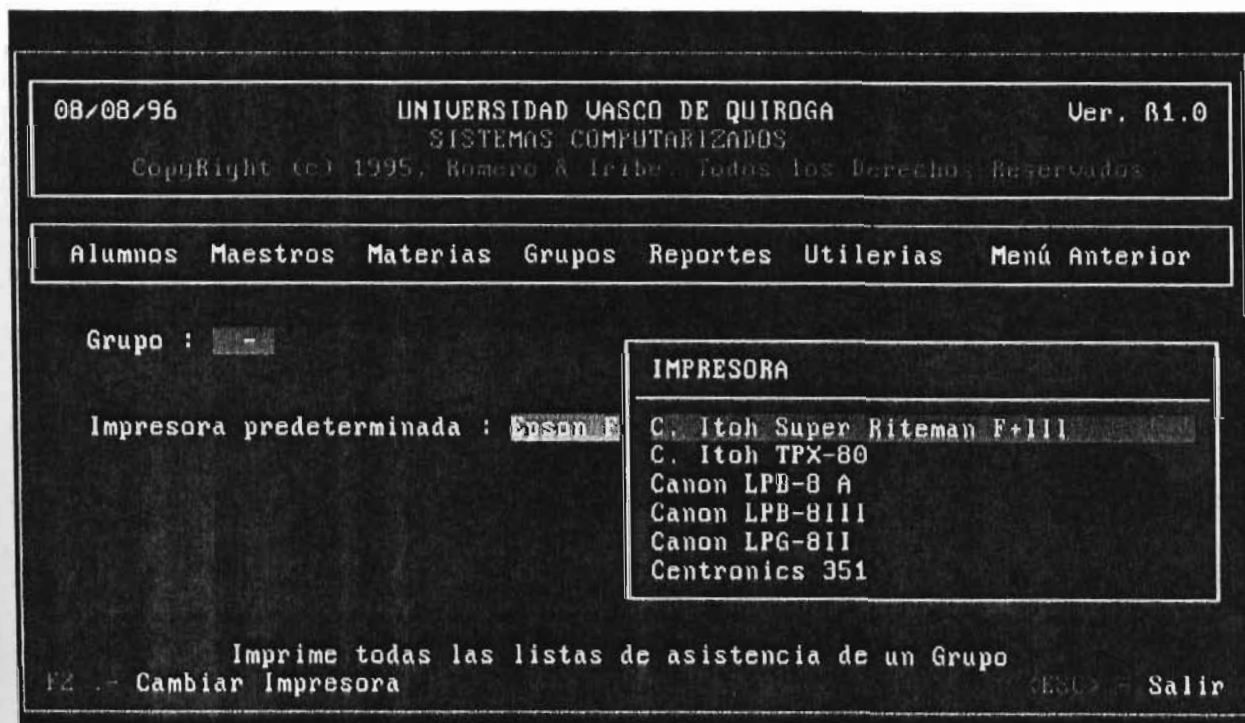
Para realizar este proceso primeramente, seleccione la opción de reportes y dentro del submenú de *listas de asistencia* la opción de *por grupo*, acto seguido lo único que se tiene que hacer es proporcionar el grupo del cual se desean imprimir las listas.

Si la impresora llegara a estar fuera de línea (on-line), apagada o desconectada, el sistema mostrará un recuadro en el cual da la opción al usuario de *reintentar* la operación o de *cancelar*. Sin embargo, si se estuviera utilizando el administrador de impresión de NOVELL y se ha activado la opción de interceptar los trabajo de impresión, el sistema nunca mostrará el recuadro aunque la impresora no este en condiciones de imprimir, en ese caso de que no se cuente con privilegios para manejar el administrador de impresión, dirigase con el administrador de la red para solucionar el problema y/o para cancelar la impresión.

08/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Grupo :	<input type="text"/>	
Impresora predeterminada :	<input type="text" value="Epson FX"/>	
Imprime todas las listas de asistencia de un Grupo		
F2 - Cambiar Impresora		<ESC> Salir

Como puede apreciarse, es posible cambiar la *impresora predeterminada*, esto es con el fin de hacer mas flexible al sistema. Si la impresora predeterminada que muestra el sistema no es con la que se cuenta, lo único que tiene que hacer es presionar la tecla F2 y al instante aparecerá una ventana con la lista de todas las impresoras disponibles en el sistema; haciendo uso de las teclas de dirección (flechas), busque la impresora correcta y presione <ENTER> para seleccionarla, una vez hecho esto, la impresora predeterminada cambiara por la que se haya seleccionado. Es **IMPORTANTE** aclarar que cualquier cambio que se realice en la impresora afectara todos los reportes que se generen en la *terminal* en la que se esté trabajando y únicamente afecta a la sección de trabajo actual. Si se requiere hacer permanente el cambio consulte el capítulo de *sistema de protección*.

Al presionar la tecla F2, el sistema mostrará una pantalla semejante a la que se muestra a continuación.



```

08/08/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. R1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero & Iribar. Todos los Derechos Reservados

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Grupo : [ ]

Impresora predeterminada : Canon B

IMPRESORA
C. Itoh Super Riteman F+111
C. Itoh TPX-80
Canon LPB-B A
Canon LPB-8111
Canon LPG-811
Centronics 351

Imprime todas las listas de asistencia de un Grupo
F2 - Cambiar Impresora          [ESC] = Salir

```

La opción de cambio de impresora está disponible en todos los reportes emitidos por el sistema y el proceso necesario para realizar el cambio es idéntico al anteriormente descrito, razón por la cual NO se volverá a mencionar en los siguientes apartados.

9.2.2.- LISTAS DE ASISTENCIA POR MATERIA

Para realizar este proceso primeramente, seleccione la opción de reportes y dentro del submenú de *listas de asistencia* la opción de *por materia*, acto seguido lo único que se tiene que hacer es proporcionar el grupo y la clave de la materia de la cual se desean imprimir las listas; si no se conoce la clave de la materia, bastara con presionar la tecla F1 para desplegar en forma de lista todas las materias incluidas en el plan de estudios, para buscar la materia haga uso de las tecla de dirección (flechas), y para seleccionar la materia presione la tecla <ENTER>.

Si la impresora llegara a estar fuera de línea (on-line), apagada o desconectada, el sistema mostrará un recuadro en el cual da la opción al usuario de *reintentar* la operación o de *cancelar*. Sin embargo, si se estuviera utilizando el administrador de impresión de NOVELL y se ha activado la opción de interceptar los trabajo de impresión, el sistema nunca mostrará el recuadro aunque la impresora no este en condiciones de imprimir, en ese caso de que no se cuente con privilegios para manejar el administrador de impresión, dirigase con el administrador de la red para solucionar el problema y/o para cancelar la impresión.

```
08/08/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero R Teiba. Todos los Derechos Reservados

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

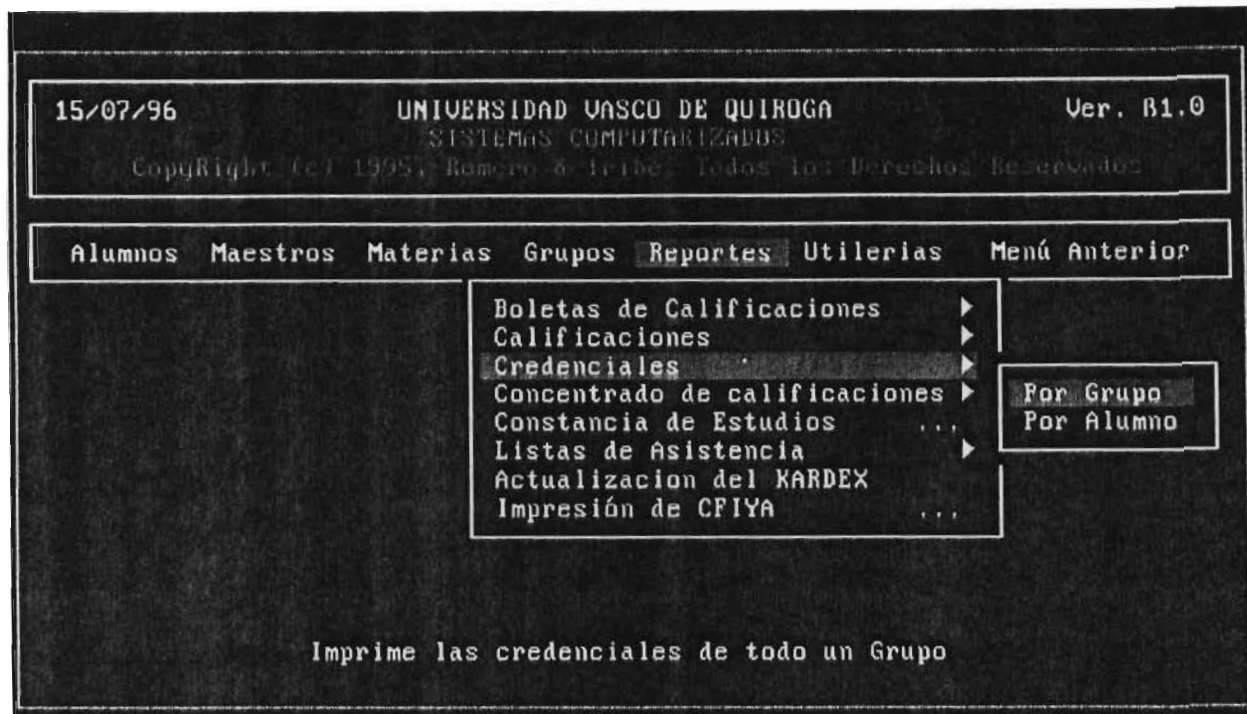
Grupo :          Clave Materia :
Materia :

Impresora predeterminada : Epson FX

Imprime la lista de asistencia por Materia dentro de un grupo en particular
F2 - Cambiar Impresora                                <ESC> - Salir
```

9.3.- IMPRESIÓN DE CREDENCIALES

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *credenciales* esté iluminada presione la tecla <ENTER>, al realizar esto, se desplegará un nuevo menú tal como se muestra a continuación.



9.3.1.- CREDENCIALES POR GRUPO

Para realizar este proceso primeramente, seleccione la opción de reportes y dentro del submenú de *credenciales* la opción de *por grupo*, acto seguido lo único que se tiene que hacer es proporcionar el grupo del cual se desean imprimir las credenciales, la vigencia de la credencial y el nombre de la persona que la firmara. Un ejemplo de este proceso se puede observar a continuación.

08/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright (C) 1995, Romero & Tribe, Todos los Derechos Reservados	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Grupo :	1-U	Vigencia : SEPTIEMBRE 1996 - FEBRERO 1997
Firma :	LIC. SALVADOR LOPEZ LIRA	
Impresora predeterminada :	Epson FX	
Esta seguro de IMPRIMIR :	N	
Imprime las credenciales de todo un Grupo		
F2 = Cambiar Impresora		<ESC> = Salir

9.3.2.- CREDENCIALES POR ALUMNO

Para realizar este proceso primeramente, seleccione la opción de reportes y dentro del submenú de *credenciales* la opción de *por alumno*, acto seguido lo único que se tiene que hacer es proporcionar el grupo, la matrícula del alumno del cual se desean imprimir la credencial, la vigencia de la credencial y el nombre de la persona que la firmara. Un ejemplo de este proceso se puede observar a continuación.

08/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright (c) 1995, Romero & Tribe. Todos los Derechos Reservados	Ver. R1.0				
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior
Grupo	:	1-U				
Clave Alumno	:	9621002				
Uigencia	:	SEPTIEMBRE 1996 - FEBRERO 1997				
Firma	:	LIC. SALVADOR LOPEZ LIRA				
Impresora Predeterminada	:	Epson FX				
Imprime las credenciales de un Alumno en particular						Salir

9.4.- IMPRESIÓN DE CONCENTRADOS DE CALIFICACIONES (LICENCIATURAS)

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *concentrado de calificaciones* esté iluminada presione la tecla <ENTER>, al realizar esto se desplegara un nuevo menú tal como se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTACIONALES	Ver. R1.0																								
Alumnos	Maestros	Materias																								
Grupos	Reportes	Utilerias																								
Menú Anterior																										
<table border="1"> <tr> <td>Boletas de Calificaciones</td> <td>▶</td> <td></td> </tr> <tr> <td>Calificaciones</td> <td>▶</td> <td></td> </tr> <tr> <td>Credenciales</td> <td>▶</td> <td></td> </tr> <tr> <td>Concentrado de calificaciones</td> <td>▶</td> <td></td> </tr> <tr> <td>Constancia de Estudios</td> <td>...</td> <td></td> </tr> <tr> <td>Listas de Asistencia</td> <td>▶</td> <td></td> </tr> <tr> <td>Actualización del KARDEX</td> <td></td> <td></td> </tr> <tr> <td>Impresión de CFIYA</td> <td>...</td> <td></td> </tr> </table>			Boletas de Calificaciones	▶		Calificaciones	▶		Credenciales	▶		Concentrado de calificaciones	▶		Constancia de Estudios	...		Listas de Asistencia	▶		Actualización del KARDEX			Impresión de CFIYA	...	
Boletas de Calificaciones	▶																									
Calificaciones	▶																									
Credenciales	▶																									
Concentrado de calificaciones	▶																									
Constancia de Estudios	...																									
Listas de Asistencia	▶																									
Actualización del KARDEX																										
Impresión de CFIYA	...																									
<table border="1"> <tr> <td>Por Grupo</td> </tr> <tr> <td>Por Materia</td> </tr> </table>			Por Grupo	Por Materia																						
Por Grupo																										
Por Materia																										
<p>Imprime el concentrado de calificaciones por grupo</p>																										

9.4.1.- CONCENTRADOS DE CALIFICACIONES POR GRUPO

Para realizar este proceso primeramente, seleccione la opción de reportes y dentro del submenú de *concentrado de calificaciones* elija la opción de *por grupo*, acto seguido lo único que se tiene que hacer es proporcionar el grupo del cual se desean imprimir las credenciales, los nombres tanto del Jefe del Departamento de Control Escolar como del Director de la Escuela seleccionada. Un ejemplo de este proceso se puede observar a continuación.

09/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS CopyRight (C) 1995, Romero & Tribe. Todos los Derechos Reservados	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Grupo :	1-U	
Jefe del D.C.E. :	LIC. SALVADOR LOPEZ LIRA	
Director de Esc. :	ING. SERGIO PEO, BARRAZA IBARRA	
Impresora Predeterminada :	Epson FX	
Esta opción de IMPRIMIR : N		
Imprime el concentrado de calificaciones por grupo		
F2 :- Cambiar Impresora	<ESC> :-	Salir

9.4.2.- CONCENTRADO DE CALIFICACIONES POR MATERIA

Para realizar este proceso primeramente, seleccione la opción de reportes y dentro del submenú de *listas de asistencia* la opción de *por materia*, acto seguido lo único que se tiene que hacer es proporcionar el grupo y la clave de la materia de la cual se desea imprimir el concentrado (si no se conoce la clave de la materia, bastara con presionar la tecla F1 para desplegar en forma de lista todas las materias incluidas en el plan de estudios, para buscar la materia haga uso de las tecla de dirección (flechas), y para seleccionar la materia presione la tecla <ENTER>), se deben proporcionar también los nombres tanto del Jefe del Departamento de Control Escolar como del Director de la Escuela seleccionada. Un ejemplo se muestra a continuación.

09/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright © 1995, Romero & Farber. Todos los derechos reservados.	Ver. R1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Grupo	: 1-U	
Materia	: 00003	
Jefe del D.C.E. :	LIC. SALVADOR LOPEZ LIRA	
Director de Esc.:	ING. SERGIO FCO. BARRAZA IBARRA	
Impresora Predeterminada :	Epson FX	
Esta seguro de IMPRIMIR :	N	
Imprime el concentrado de calificaciones por materia de un grupo particular		
F2	- Cambiar Impresora	<ESC> - Salir

9.5.- IMPRESIÓN DE CONSTANCIAS DE ESTUDIO

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *constancia de estudios* esté iluminada presione la tecla <ENTER>, al realizar esto, se desplegara una pantalla semejante a la que se muestra a continuación.

```
10/08/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
                  SISTEMAS COMPUTARIZADOS
                  Copyright (c) 1995, Romero & Iribe Todos los Derechos Reservados

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

ASUNTO:  CONSTANCIA DE ESTUDIOS

A QUIEN CORRESPONDA:

Impresora Predeterminada : Epson FX

Imprime en papel las listas de asistencia
F2 -- Cambiar Impresora  F10 -- Salvar  F11 -- Imprimir  <ESC> -- Salir
```

A continuación lo único que tiene que hacer es llenar los espacios que se encuentran en blanco. Esta opción lo que realmente proporciona es un pequeño procesador de textos y para desplazarse y escribir en él puede seguir las teclas mas comúnmente usadas en este tipo de programas.

Una vez que se complete la información y se desee **imprimir** la constancia, únicamente tiene que presionar la tecla **F11**.

Si se desea conservar los cambios efectuados en el documento únicamente tiene que presionar la tecla **F10**; si el documento ha sufrido cambios y éstos no fueron grabados, al presionar la tecla **<ESC>** el sistema pedirá confirmación para perder los cambios, si no se desea perder los cambios, el sistema regresara al documento dando oportunidad para salvar el documento. A continuación se puede ver como se podría ver la pantalla.

10/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados	Ver. R1.0									
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior					
¿ FINALIZAR SIN GUARDAR CAMBIOS ?											
NO SI											
A QUIEN CORRESPONDA:											
Impresora Predeterminada : Epson FX											
Imprime en papel las listas de asistencia											
F2	-	Cambiar Impresora	F10	-	Salvar	F11	-	Imprimir	<ESC>	-	Salir

9.6.- IMPRESIÓN DE BOLETAS DE CALIFICACIONES

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *boletas de calificaciones* esté iluminada presione la tecla <ENTER>, al realizar esto se desplegará un nuevo menú tal como se muestra a continuación.

10/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMA DE CONTROL ESCOLAR	Ver. R1.0		
Alumnos	Maestros	Materias		
Grupos	Reportes	Utilerias		
Menú Anterior				
	Boletas de Calificaciones ▶	<table border="1"> <tr> <td>Por Grupo</td> </tr> <tr> <td>Por Alumno</td> </tr> </table>	Por Grupo	Por Alumno
Por Grupo				
Por Alumno				
	Calificaciones ▶			
	Credenciales ▶			
	Concentrado de calificaciones ▶			
	Constancia de Estudios ...			
	Listas de Asistencia ▶			
	Actualizacion del KARDEX			
	Impresión de CFIYA ...			
Imprime Boletas de Calificaciones de todo un Grupo				

9.6.1.- IMPRESIÓN DE BOLETAS DE CALIFICACIONES POR GRUPO

Para realizar este proceso primeramente, seleccione la opción de *reportes* y dentro del submenú de *boletas de calificaciones* elija la opción de *por grupo*, acto seguido lo único que se tiene que hacer es proporcionar el grupo del cual se desean imprimir las boletas, el nombre y el cargo de la persona que firmara la boleta y si fuera necesario, el sistema permite incluir dos líneas de comentario y/o mensaje a los alumnos que se imprimirán en la esquina inferior izquierda de la boleta. Un ejemplo de este proceso se puede observar a continuación.

08/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright (C) 1995, Romero & Tribe. Todos los Derechos Reservados	Ver. B1.0				
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior
Grupo	:					
Firma	:					
Cargo	:					
Comentario	:					
Impresora predeterminada	:	Epson FX				
Imprime Boletas de Calificaciones de todo un Grupo						<ESC> - Salir

9.6.2.- IMPRESIÓN DE BOLETAS DE CALIFICACIONES POR ALUMNO

Para realizar este proceso primeramente, seleccione la opción de *reportes* y dentro del submenú de *boletas de calificaciones* elija la opción de *por alumno*, acto seguido lo único que se tiene que hacer es proporcionar el grupo, la matrícula del alumno del cual se desean imprimir la boleta, el nombre y el cargo de la persona que firmara la boleta y si fuera necesario, el sistema permite incluir dos líneas de comentario y/o mensaje a los alumnos que se imprimirán en la esquina inferior izquierda de la boleta. Un ejemplo de este proceso se puede observar a continuación.

```
08/08/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. R1.0
                   SISTEMAS COMPUTABILIZADOS
                   Copyright (c) 1995, Romero & Torre. Todos los Derechos Reservados.

Alumnos  Maestros  Materias  Grupos  Reportes  Utilerias  Menú Anterior

Grupo      :
Clave Alumno :
Nombre     :

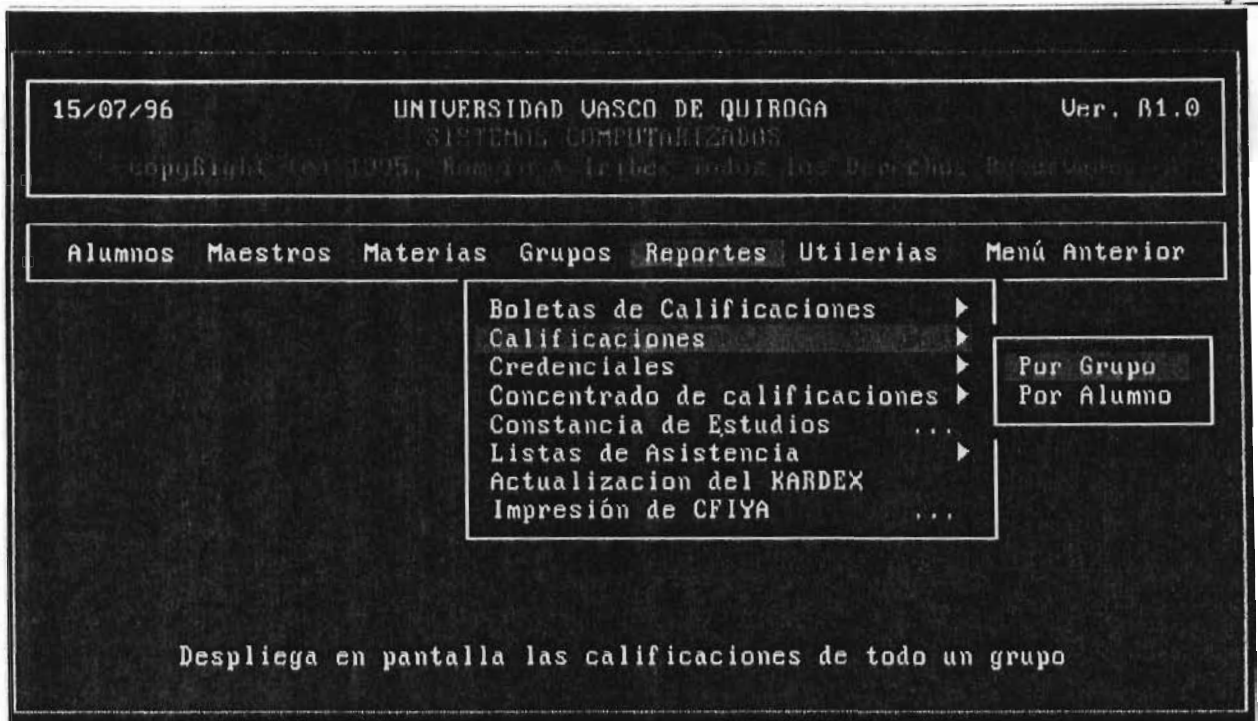
Firma      :
Cargo      :
Comentario :

Impresora predeterminada : Epson FX

Imprime Boletas de Calificaciones de un Alumno en particular
                                                                Salir
```

9.7.- CONSULTA DE CALIFICACIONES

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *calificaciones* esté iluminada presione la tecla <ENTER>, al realizar esto se desplegará un nuevo menú tal como se muestra a continuación.



9.7.1.- CONSULTA DE CALIFICACIONES POR GRUPO

Para realizar este proceso primeramente, seleccione la opción de *reportes* y dentro del submenú de *calificaciones* elija la opción de *por grupo*, acto seguido lo único que se tiene que hacer es proporcionar el grupo y la clave de la materia de la cual se desean consultar las calificaciones (si no se conoce la clave de la materia, bastara con presionar la tecla F1 para desplegar en forma de lista todas las materias incluidas en el plan de estudios, para buscar la materia haga uso de las tecla de dirección (flechas), y para seleccionar la materia presione la tecla <ENTER>), a continuación es necesario indicar el número de evaluación (en el caso de las licenciaturas del 1 al 3, en las preparatorias del 1 al 8 y en postgrados solamente el 1). En ese momento se podrá observar una pantalla semejante a la que se muestra a continuación.

04/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright © 1995, Romero A. Leizaola. todos los derechos reservados.	Ver. R1.0				
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior
Grupo	:	1-U				
Materia	:	00003				
Evaluación	:					
Manejo de Calificaciones de todo un grupo						Cancelar

Una vez que se haya proporcionado la información anterior, el sistema mostrará en forma de tabla los nombre de los alumnos, la calificación, en el caso de licenciaturas las asistencias y en el caso de las preparatorias las inasistencias; en la parte superior de la tabla podrá observarse el nombre de la materia seleccionada.

Como puede observarse a continuación, el área de trabajo es semejante a la que se muestra al momento de capturar las calificaciones por grupo, solo que en este caso, únicamente se pueden consultar, no se permite la manipulación de la información.

08/08/96			UNIVERSIDAD VASCO DE QUIROGA			Ver. B1.0																				
<table border="1"> <thead> <tr> <th>Alumno</th> <th>Calificación</th> <th>Asistencias</th> </tr> </thead> <tbody> <tr> <td>ERIC IRIBE RUIZ</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>LETICIA GUILLEN GONZALEZ</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>MARGARITA ROMERO DIAZ BARRIGA</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>ROSA MARIA VARELA CAZAREZ</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>SALVADOR DIAZ BARRIGA HERNANDEZ</td> <td>0.0</td> <td>0</td> </tr> </tbody> </table>									Alumno	Calificación	Asistencias	ERIC IRIBE RUIZ	0.0	0	LETICIA GUILLEN GONZALEZ	0.0	0	MARGARITA ROMERO DIAZ BARRIGA	0.0	0	ROSA MARIA VARELA CAZAREZ	0.0	0	SALVADOR DIAZ BARRIGA HERNANDEZ	0.0	0
Alumno	Calificación	Asistencias																								
ERIC IRIBE RUIZ	0.0	0																								
LETICIA GUILLEN GONZALEZ	0.0	0																								
MARGARITA ROMERO DIAZ BARRIGA	0.0	0																								
ROSA MARIA VARELA CAZAREZ	0.0	0																								
SALVADOR DIAZ BARRIGA HERNANDEZ	0.0	0																								
Despliega en pantalla las calificaciones de todo un grupo								Salir																		
Buscar																										

9.7.2.- CONSULTA DE CALIFICACIONES POR ALUMNO

Para realizar este proceso primeramente, seleccione la opción de *reportes* y dentro del submenú de *calificaciones* elija la opción de *por alumno*, acto seguido lo único que se tiene que hacer es proporcionar el grupo y la matrícula del alumno del la cual se desean consultar las calificaciones, a continuación es necesario indicar el número de evaluación (en el caso de las licenciaturas del 1 al 3, en las preparatorias del 1 al 8 y en postgrados solamente el 1). En ese momento se podrá observar una pantalla semejante a la que se muestra a continuación.

04/08/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright © 1995 Roberto A. Lirio - Todos los Derechos Reservados	Ver. 01.0				
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior
Grupo	:	1-U				
Clave Alumno	:	9621000				
Evaluación	:					
Manejo de Calificaciones por alumno específico						Salir

Una vez que se haya proporcionado la información anterior, el sistema mostrará en forma de tabla las materias, la calificación, en el caso de licenciaturas las asistencias y en el caso de las preparatorias las inasistencias; en la parte superior de la tabla podrá observarse el nombre del alumno seleccionado.

Como puede observarse a continuación, el área de trabajo es semejante a la que se muestra al momento de capturar las calificaciones por alumno, solo que en este caso, únicamente se pueden consultar, no se permite la manipulación de la información.

Materia	Calificación
MATEMATICAS I	0.0
CONTABILIDAD	0.0
INTRODUCCION A LA ADMINISTRACION	0.0
INTRODUCCION A LA ECONOMIA	0.0
PROCESAMIENTO DE DATOS	0.0
FILOSOFIA DEL HOMBRE	0.0

Despliega en pantalla las calificaciones de un Alumno en particular
Buscar Salir

9.8.- IMPRESIÓN DE CFIYA

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *impresión de CFIYA* esté iluminada presione la tecla <ENTER>, al realizar esto lo único que tiene que proporcionar es el grupo del cual se desea imprimir este documento, tal como se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright © 1995, Romero & Libero. Todos los derechos reservados.	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
Grupo : <input type="text" value="-"/>		
Impresora predeterminada : Epson FX <input type="text"/>		
Imprime en papel las CFIYAS		
<input type="button" value="Cambiar Impresora"/>	<input type="button" value="Salir"/>	

Este reporte se imprime en 2 partes, primeramente se imprime la relación de los alumnos con sus respectivas calificaciones y posteriormente se imprimirán las materias (en el orden adecuado), correspondientes; posteriormente será necesario escribir manualmente las materias en el espacio correspondiente del formato, en el mismo orden en que fueron impresas en la segunda hoja del reporte.

9.9.- ACTUALIZACIÓN DEL KARDEX

Una vez que se encuentre en el submenú de *reportes* y cuando la opción de *actualización del KARDEX* esté iluminada presione la tecla <ENTER>, al realizar esto el sistema pedirá la confirmación para realizar la operación. **ESTA OPERACIÓN DEBE SER REALIZADA SOLAMENTE UNA VEZ Y ÚNICAMENTE AL TERMINO DEL SEMESTRE.**

Una vez que se proporcione la confirmación para realizar esta operación, la base de datos de KARDEX quedara actualizada al igual que el historial académico de los alumnos.

CAPÍTULO 10**"MÓDULO DE UTILERIAS"****10.1.- INTRODUCCIÓN**

El modulo de utilerías incluye los procesos necesarios para el mantenimiento de las bases de datos del sistema. Estos procesos son: la *inicialización de semestres*, el *respaldo de información*, la *recuperación de la información* respaldada y la regeneración de los *índices* de las bases de datos.

Primeramente para realizar esta operación seleccione la escuela correspondiente, a continuación seleccione en el menú la opción de *utilerías* y presione la tecla <ENTER>, una vez hecho esto se desplegara un submenú tal como se puede observar a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. R1.0				
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerías	Menú Anterior
Inicio de Semestre						
Respaldo Información						
Recuperar Información						
Reindexar Bases						
Inicialización de un nuevo semestre						

10.2.- INICIO DE SEMESTRE

Una vez que haya concluido el semestre y se desee comenzar uno nuevo, será necesario ejecutar esta opción; si al termino del semestre no se ejecuta esta opción las calificaciones del semestre anterior seguirán vigentes.

Una vez que se encuentre en el submenú de *utilerias* y cuando la opción de *inicio de semestre* esté iluminada presione la tecla <ENTER>.

El proceso que se realiza, es el vaciado total y definitivo de las bases de asignaciones de materias y el borrado de todas las calificaciones de todos los alumnos de la escuela seleccionada, razón por la cual el sistema no permite realizar este proceso hasta que no se haya impreso la CFIYA y actualizado del KARDEX.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright © 1995, Romay & Torre. Todos los derechos Reservados	Ver. B1.0
Alumnos Maestros Materias Grupos Reportes Utilerias Menú Anterior		
ERROR: No se puede iniciar el semestre ya que no se han imprimido el KARDEX y/o la CFIYA o no se ha respaldado la información, son necesarios estos procesos		e c i ó n
Ok		
Inicialización de un nuevo semestre		

En caso de que los procesos de impresión de CFIYA y de actualización de KARDEX ya se hayan ejecutado y se elija esta opción, el sistema pedirá confirmación y de ser ésta confirmada, el proceso no puede ser revertido.

10.3.- RESPALDO DE INFORMACIÓN

Como parte de las medidas de seguridad que se implementaron al Sistema de Control Escolar, está la de poder obtener respaldo en discos flexibles de la información de cada una de las escuelas. Es sumamente **importante** obtener estos respaldos por lo menos una vez por semana, aunque lo ideal es obtener respaldos diarios de toda la información de las diferentes escuelas de la Universidad, ya que en caso de algún problema en el disco duro y/o en la computadora, no se perderá la información.

Para realizar este proceso es necesario realizar lo siguiente. Una vez que se encuentre en el submenú de *utilerías* y cuando la opción de *respaldo de información* esté iluminada presione la tecla <ENTER>, al realizar esto el sistema pedirá la letra de la unidad en la que se desea realizar el respaldo (unidad "A" o unidad "B"), posterior a esto, bastara con seguir las instrucciones mostradas en pantalla.

Es conveniente etiquetas inmediatamente los discos, especificando claramente, el nombre de la escuela, la fecha de respaldo y el número de la secuencia del disco.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. R1.0				
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerías	Menú Anterior
En que unidad desea realizar el respaldo A/B?						
Respaldo de las bases de datos						Salir

La información que será respaldada será únicamente la de la escuela seleccionada, si se desea respaldar la información de todas las escuelas será necesario hacerlo escuela por escuela.

Si después de indicar al sistema la letra de la unidad en que se desea realizar el respaldo se genera un mensaje de error o simplemente nunca pide que se inserten los discos, se puede deber a dos situaciones :

- Si se tiene una versión de DOS superior a la 5.0, es necesario que se cargue en memoria el comando SETVER.EXE. Para ello incluya dentro de su archivo CONFIG.SYS la siguiente línea y vuelva a iniciar su terminal :

```
DEVICE=C:\DOS\SETVER.EXE
```

- Asegúrese de tener por lo menos 590 Kb. de memoria base libre. Si se tiene un procesador 80386 o superior y DOS versión 6.x, ejecute la utileria MEMAKER para optimizar su memoria (en este caso asegúrese de incluir los controladores de red en la optimización).

- Al momento de intalarse el sistema los archivos BACKUP.EXE y RESTORE.EXE son copiados en el directorio principal del sistema. NO BORRE estos archivos ya que sin ellos no se podrá realizar el respaldo o la recuperación de la información.

10.4.- RECUPERACIÓN DE LA INFORMACIÓN

Para realizar este proceso es necesario realizar lo siguiente. Una vez que se encuentre en el submenú de *utilerias* y cuando la opción de *recuperar información* esté iluminada presione la tecla <ENTER>, al realizar esto el sistema pedirá la letra de la unidad en la que se encuentran los disco de respaldo (unidad "A" o unidad "B"), posterior a esto, bastara con seguir las instrucciones mostradas en pantalla.

Algo que se debe tomar en cuenta es lo siguiente:

- Cuando se realiza el respaldo de información, únicamente se respalda la información de la escuela seleccionada, pero en la opción de recuperación si se insertan unos discos de respaldo que no sean los de la escuela seleccionada, el sistema no mostrará error y recuperara la información “de la escuela” a la que correspondan los discos; si esto llega a suceder la información de la escuela seleccionada no sufrirá cambios.

10.5.- REINDEXAR LAS BASES DE DATOS DE LA ESCUELA

Si llegase a ocurrir una falla en la corriente eléctrica y/o si el sistema muestra información incoherente o simplemente por seguridad (periódica), será necesario realizar este proceso; para ello es necesario realizar lo siguiente. Una vez que se encuentre en el submenú de *utilerias* y cuando la opción de *reindexar bases* esté iluminada presione la tecla <ENTER>, en ese momento el sistema pedirá que se confirme la intención de realizar el proceso. En caso de que así sea la pantalla se podrá ver semejante a la que se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA SISTEMAS COMPUTARIZADOS Copyright © 1995, Romero & Iribar. Todos los Derechos Reservados.	Ver. B1.0				
Alumnos	Maestros	Materias	Grupos	Reportes	Utilerias	Menú Anterior
ESPERE UN MOMENTO POR FAVOR SE ESTA REALIZANDO EL PROCESO						Semestre Información nformación ases
Indexando Archivo : MATERIAS.DBF						
27%						
Genera los indices necesarios para las bases de datos						

La diferencia entre esta opción y la que está disponible en el modulo de escuelas es que en el modulo de escuelas se reindexan únicamente las bases principales del sistema y en esta opción se reindexan las bases propias de cada una de las escuelas.

Es recomendable realizar primero el reindexado de las bases principales (modulo de escuelas), y posteriormente el reindexado de cada una de las escuelas, por lo menos una vez al mes.

CAPÍTULO 11

“SISTEMA DE PROTECCIÓN DE ACCESO”

11.1.- INTRODUCCIÓN

En los entornos corporativos de informática distribuida, la seguridad se ha convertido en una cuestión cada vez más importante. Los recursos de datos ya no se centralizan en un solo lugar; la falta de una gestión centralizada de los datos hace que éstos sean más vulnerables a :

- La supervisión no autorizada
- La modificación de la información
- La suplantación de un usuario por otro no autorizado

Para garantizar la confidencialidad de la información se utilizan las técnicas de cifrado de claves. Es precisamente esta tarea la que realiza el Sistema de Protección de Acceso; es decir, este sistema permite mantener almacenadas un número ilimitado de claves y/o de usuarios, así como su nivel de privilegios dentro del Sistema de Control Escolar.

11.2.- USO DEL SISTEMA DE PROTECCIÓN A NIVEL USUARIO

El sistema permite 10 niveles de acceso, de los cuales 5 están dedicados al Sistema de Control Escolar, 4 al Sistema de Cuentas por Cobrar y uno de nivel supervisor en ambos sistemas.

El sistema de protección está presente en *todo* el sistema, para poder acceder a él simplemente se tiene que presionar la tecla F9.

Si se tiene un nivel de privilegio del 1 al 9, al presionar la tecla F9 el sistema pedirá que se introduzca la clave de acceso autorizada, un ejemplo de esto es el que se muestra a continuación.

The screenshot shows a terminal window with the following text:

```
15/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
MENU PRINCIPAL DEL SISTEMA DE CONTROL ESCOLAR
Copyright (c) 1995, Romero & Irujo. Todos los Derechos Reservados

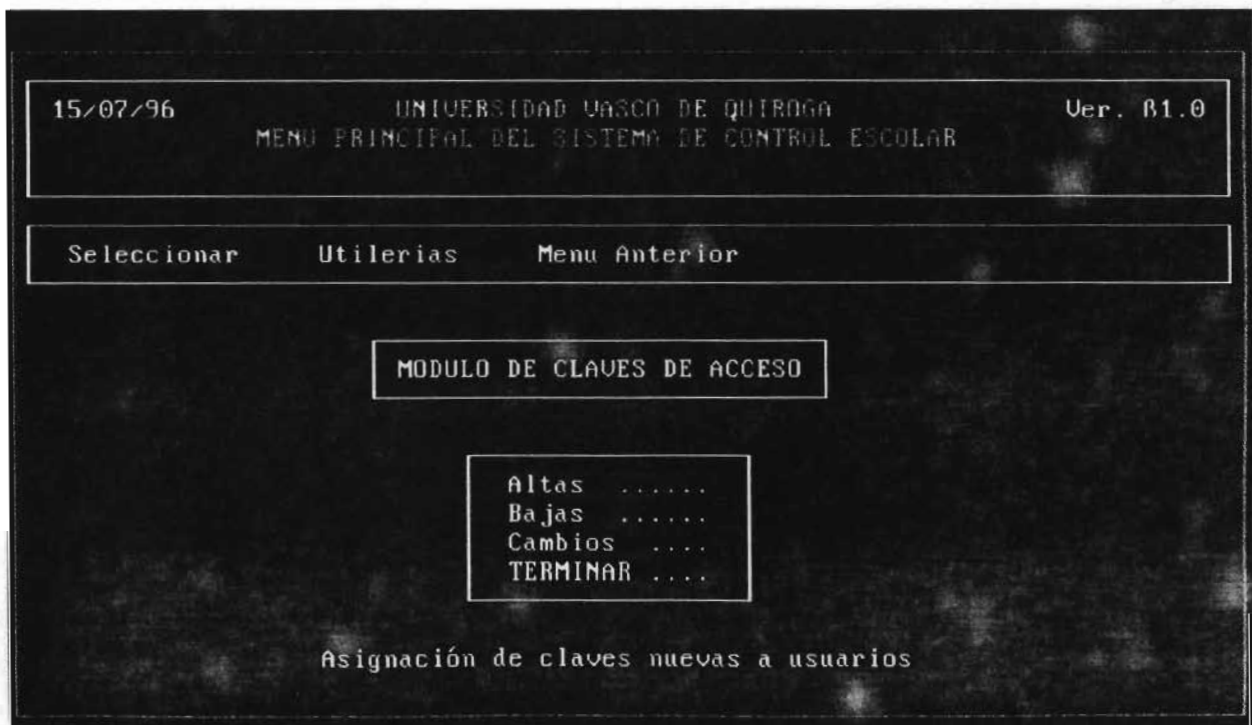
Seleccionar  Utilerias  Menu Anterior

SISTEMA DE PROTECCION
LOGIN : [ ] a jar
```

Esto es esencialmente útil cuando el usuario autorizado se tiene que separar por un período de tiempo considerable de su terminal, ya que en ese momento una persona no autorizada podría tratar de tener acceso a la información; así cuando el usuario "regrese" a su terminal, bastara con digitar nuevamente su clave de acceso para continuar trabajando en el mismo lugar en el que activó el sistema de protección.

11.3.- USO DEL SISTEMA DE PROTECCIÓN A NIVEL SUPERVISOR

Si se cuenta con nivel de supervisor al presionar la tecla F9 no se pedirá directamente la clave como sucede con los niveles de usuario, en lugar de ello se desplegara el menú del Sistema de Protección, un ejemplo de como se podría ver la pantalla se muestra a continuación.



11.3.1.- ALTA DE USUARIOS

Para realizar este proceso es necesario realizar lo siguiente. Una vez que se presione la tecla F9 y se encuentre en el menú del sistema de protección y cuando la opción de *altas* esté iluminada presione la tecla <ENTER>.

Una vez que se realice esto, el sistema pedirá que se introduzca la clave del usuario y posteriormente pedirá que se vuelva a digitar la misma clave (por cuestiones de seguridad). Es conveniente que la clave la digite el mismo usuario y no el administrador del sistema. Es conveniente también que se le indique al usuario que la clave no sean sus iniciales, ni su teléfono, o secuencias de caracteres obvias de deducir.

A continuación se pedirá al administrador del sistema proporcione el **nivel** que se desea asignar al usuario. La lista de los privilegios se muestra en un recuadro en la parte derecha de la pantalla; en primer lugar se muestra la lista de privilegios para el Sistema de Control Escolar, si se desea visualizar la lista de privilegios del Sistema de Cuentas por Cobrar bastara con presionar la tecla F4 y viceversa.

En ese momento la pantalla se vera como se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. B1.0														
MENU PRINCIPAL DEL SISTEMA DE CONTROL ESCOLAR																
Copyright (c) 1995, Bomper & Tribe. Todos los Derechos Reservados																
Seleccionar	Utilerias	Menu Anterior														
Clave Nueva :	*****															
Compara Clave :	*****															
Nivel :	5															
Impresora :																
		<table border="1"> <thead> <tr> <th>NIVEL</th> <th>PRIVILEGIOS</th> </tr> </thead> <tbody> <tr> <td>(0)</td> <td>SUPERVISOR GENERAL DEL SISTEMA</td> </tr> <tr> <td>(1)</td> <td>ADMIN. SCE SIN CAMBIO DE CLAVES</td> </tr> <tr> <td>(2)</td> <td>CALIFICS, ALTAS, BAJAS, CONSULTAS</td> </tr> <tr> <td>(3)</td> <td>ALTAS, CONSULTAS, MODIFICACIONES</td> </tr> <tr> <td>(4)</td> <td>CONSULTA DE MAESTROS, MATERIAS, CALIFICACIONES Y GRUPOS</td> </tr> <tr> <td>(5)</td> <td>CONSULTA DE CALIFICACIONES</td> </tr> </tbody> </table>	NIVEL	PRIVILEGIOS	(0)	SUPERVISOR GENERAL DEL SISTEMA	(1)	ADMIN. SCE SIN CAMBIO DE CLAVES	(2)	CALIFICS, ALTAS, BAJAS, CONSULTAS	(3)	ALTAS, CONSULTAS, MODIFICACIONES	(4)	CONSULTA DE MAESTROS, MATERIAS, CALIFICACIONES Y GRUPOS	(5)	CONSULTA DE CALIFICACIONES
NIVEL	PRIVILEGIOS															
(0)	SUPERVISOR GENERAL DEL SISTEMA															
(1)	ADMIN. SCE SIN CAMBIO DE CLAVES															
(2)	CALIFICS, ALTAS, BAJAS, CONSULTAS															
(3)	ALTAS, CONSULTAS, MODIFICACIONES															
(4)	CONSULTA DE MAESTROS, MATERIAS, CALIFICACIONES Y GRUPOS															
(5)	CONSULTA DE CALIFICACIONES															
Asignación de claves nuevas a usuarios																
Cancelar																

Una vez asignado el nivel, será necesario especificar la impresora que tendrá asignada por "default" el usuario; para ello, el sistema mostrará en un pequeño recuadro la lista de todas las impresora disponibles, para buscar la impresora utilice las teclas de dirección (flechas), y para seleccionara una de ellas presione la tecla <ENTER> una vez que la impresora adecuada se encuentre iluminada.

En ese momento la pantalla se podría ver tal como se muestra a continuación.

15/07/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. B1.0
MENU PRINCIPAL DEL SISTEMA DE CONTROL ESCOLAR		
Copyright © 1995, Homero A. Ferrer. Todos los Derechos Reservados.		
Seleccionar	Utilerias	Menu Anterior
Clave Nueva : *****		
Compara Clave : *****		
Nivel : 5		
Impresora :		
		IMPRESORA
		Epson FX-100
		Epson FX-100+
		Epson FX-1000
		Epson FX-105
		Epson FX-80
		Epson FX-80+
Asignación de claves nuevas a usuarios		
Cancelar		

A continuación se pedirá al administrador del sistema que proporcione el puerto en el cual se encuentra conectada la impresora seleccionada (generalmente el LPT1). Si el administrador del sistema no es el administrador de red y se está utilizando el administrador de impresión de la red, será conveniente que este paso sea realizado en forma conjunta.

Por ultimo, el sistema pedirá confirmación y en caso de ésta ser confirmada, el usuario quedara dado de alta y con autorización para acceder a la información a la que se le dio privilegio de uso.

11.3.2.- BAJA DE USUARIOS

Para realizar este proceso es necesario realizar lo siguiente. Una vez que se presione la tecla F9 y se encuentre en el menú del sistema de protección y cuando la opción de *bajas* esté iluminada presione la tecla <ENTER>.

Una vez que se realice esto, el sistema pedirá que se introduzca la clave del usuario y posteriormente pedirá que se vuelva a digitar la misma clave (por cuestiones de seguridad). Si las dos claves digitadas son iguales el sistema pedirá confirmación y de ser ésta confirmada, el usuario quedara dado de baja del sistema.

```
15/07/96          UNIVERSIDAD VASCO DE QUIRUGA          Ver. R1.0
MENU PRINCIPAL DEL SISTEMA DE CONTROL ESCOLAR
Copyright © 1995, Roberto Leibe. Todos los derechos reservados

Seleccionar  Utilerias  Menu Anterior

Clave a Eliminar : *****
Compara Clave   : *****

Esta seguro en DARLE DE BAJA : N

Eliminación de claves
```

11.3.3.- MODIFICACIÓN DE USUARIOS

Para realizar este proceso es necesario realizar lo siguiente. Una vez que se presione la tecla F9 y se encuentre en el menú del sistema de protección y cuando la opción de *cambios* esté iluminada presione la tecla <ENTER>.

Una vez que se realice esto, el sistema pedirá que se introduzca la clave del usuario y posteriormente pedirá que se vuelva a digitar la misma clave (por cuestiones de seguridad). Si las dos claves digitadas son iguales el sistema pedirá que se digite una nueva clave (que no puede ser la misma)

A continuación el sistema mostrará el nivel, la impresora y el puerto que ya se tenían asignados al usuario, de ser necesarios cambiar alguna de estas opciones basta con posicionarse sobre la opción a modificar.

Si fuese necesario modificar el **nivel** de usuario, la lista de los privilegios se muestra en un recuadro en la parte derecha de la pantalla; en primer lugar se muestra la lista de privilegios para el Sistema de Control Escolar, si se desea visualizar la lista de privilegios del Sistema de Cuentas por Cobrar bastara con presionar la tecla F4 y viceversa.

Una vez asignado el nivel, el sistema mostrará en un pequeño recuadro la lista de todas las impresora disponibles, para buscar la impresora utilice las teclas de dirección (flechas), y para seleccionarla presione la tecla <ENTER> una vez que la impresora adecuada se encuentre iluminada.

A continuación se pedirá al administrador del sistema que proporcione el puerto en el cual se encuentra conectada la impresora seleccionada (generalmente el LPT1). Si el administrador del sistema no es el administrador de red y se está utilizando el administrador de impresión de la red, será conveniente que este paso sea realizado en forma conjunta.

```

15/07/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. R1.0
MENU PRINCIPAL DEL SISTEMA DE CONTROL ESCOLAR
Copyright (c) 1995, Romero & Tribes. Todos los derechos reservados.

Seleccionar  Utilerias  Menu Anterior

Clave a Modificar : *****      Comparar Clave   : *****
Teclee nueva clave : *****

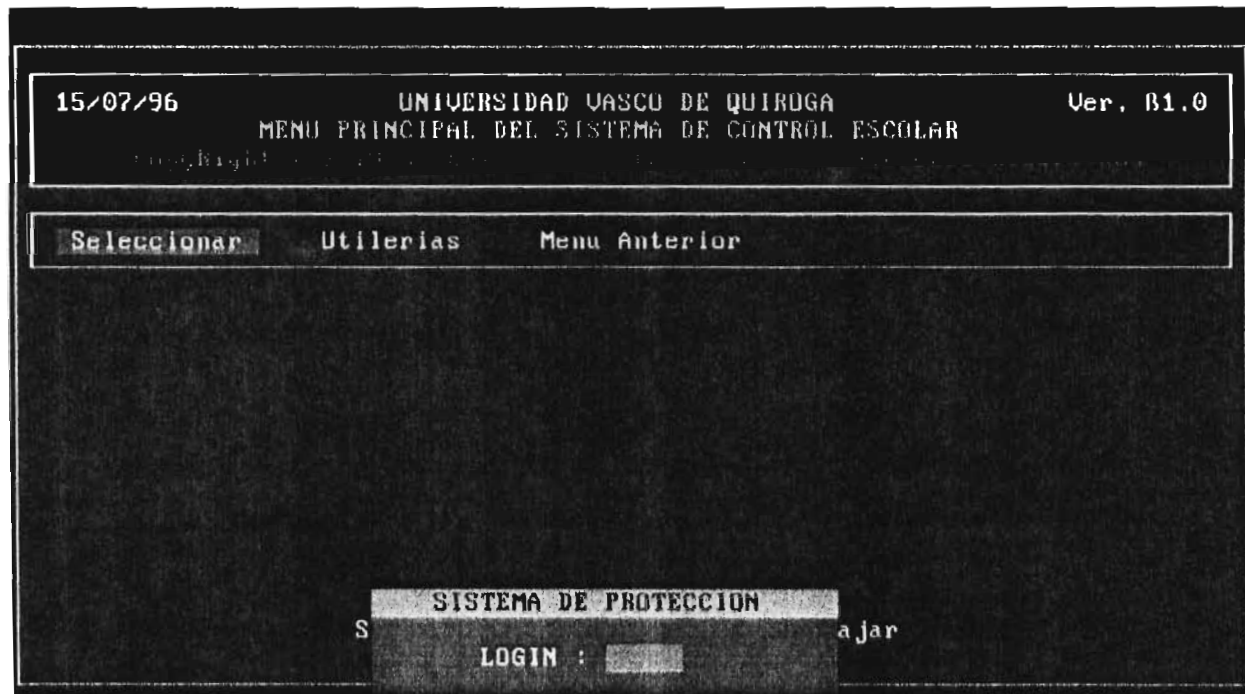
Nivel           : 5

Impresora       : AEG Laserstar 6      Puerto :

Modificación del código en claves ya asignadas
<CR> <PG> - Seleccionar  <ENTER> - Aceptar selección  <ESC> - Cancelar
    
```

11.3.4.- OPCIÓN TERMINAR

Si el administrador del sistema no desea utilizar ninguna de las opciones de Alta, baja o cambios de usuarios, sino que simplemente se tiene que ausentar por un período de tiempo considerable o desea dejar la terminar para que ésta sea utilizada por otro usuario, bastara con seleccionar la opción **terminar** para que el sistema pida la clave de acceso al sistema; el utilizar esta opción tiene el mismo efecto que tiene el que se presione la tecla F9 por parte de un usuario con nivel del 1 al 9. El efecto que se tiene en pantalla se muestra a continuación.



APENDICE “B”

CODIGO

FUENTE

INDICE

1.- SCE.PRG	194
2.- SCE2.PRG	205
3.- SCE3.PRG	227
4.- ALUMNOS.PRG	237
5.- MAESTROS.PRG	260
6.- MATERIAS.PRG	269
7.- GRUPOS.PRG	276
8.- REPORTES.PRG	298
9.- UTILERIA.PRG	347

SCE.PRГ

```

/* Programa : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje : CLIPPER 5.2
   Enlazador : RTLINK 3.14B
*/

#xcommand REPEAT => do while .t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "caos.ch"
#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla

/* Función principal del programa, se configuran la pantalla, el ambiente.
   Se dibuja la pantalla y se despliega el menú principal del sistema */
FUNCTION main ()
  LOCAL nOp := 1 // variable que contendrá la opción deseada en el menú
  LOCAL cColores // variable que contendrá los colores del entorno DOS
  LOCAL aOpcion := {"No","Si"} // opciones confirmación de salida
  LOCAL cColor, color, lineamenu,nSalir, colores
  PUBLIC nTiempo, nEspera, nPriv := NIL, cPuerto, login := SPACE(5),Nmlmp

SET EXCLUSIVE OFF
SET DECIMALS TO 1
SET MESSAGE TO MAXROW()-2 CENTER // Los mensajes de menú van en la última línea y centrados
SET DELETE ON // Evita que los registros borrados se visualicen o se consulten
SET WRAP ON // Los menús pueden rotar
SET SCOREBOARD OFF // Desactiva los mensajes de sistema en la primera línea
SET ESCAPE OFF // Evita que los procesos se aborren con <ESC>
SET TALK OFF
SET ECHO OFF // Se desactiva el eco de la pantalla
SET STATUS OFF
SET BELL OFF // Se desactiva la campana de captura
SET CURSOR OFF // Se desactiva el cursor parpadeante
SETCANCEL(.F.)

*.....//
* Establece el entorno de trabajo //
* //
run SET CLIPPER=F60;SWAPK:16 //
*.....//

colores := SETCOLOR (C_PANTALLA)
CLS // Se limpia la pantalla
color := SETCOLOR ()
CLS // Se limpia la pantalla
dispbox (7,1,20,maxcol()-1," | J - L | ")
dispbox (8,2,19,maxcol()-2," ", "w+/w+")
SETCOLOR ("W+/B+")
@ 8,24 SAY " UNIVERSIDAD VASCO DE QUIROGA "
SETCOLOR ("GR+/B+")
@ 10,26 SAY "SISTEMA DE CONTROL ESCOLAR"
@ 12,34 say "Ver. 1.0"
SETCOLOR ("B+/G")
@ 14,29 say " Desarrollado por : "
SETCOLOR ("RG+/BG")
@ 16,21 say " L.S.C. Rafael Romero Díaz Barriga "
@ 17,21 say " L.S.C. Eric Iribe Ruiz "
SETCOLOR ("R+/W+")
@ 19,8 say "CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados"
INKEY (4)
SETCOLOR (color)

IF csinet()
  SETCOLOR (colores)
  CLS
  TONE (1200,10)
  ALERT ("ESTA VERSION DEL SISTEMA ES PARA TRABAJO EN RED UNICAMENTE")

```

```
QUIT
ENDIF
```

```
*..... Pide la clave de acceso al sistema, si la clave no es.....//
*      correcta en est. misma función termina la ejecución del //
*      sistema //
claves() //
* //
* //
*.....//
```

```
TiempoDeEspera (5) // Tiempo de espera para el protector de pantalla
```

```
/* Se despliegan las distintas ventanas de la pantalla principal */
```

```
DISPBOX (1,0,MAXROW(),MAXCOL(),1)
SETCOLOR ("w+b")
DISPBOX (8,1,MAXROW()-3,MAXCOL()-1,"",C_REYENO2)
SETCOLOR (C_LIMPIA)
DISPBOX (2,1,6,MAXCOL()-1," |J-L| ,C_REYENO2")
DISPBOX (7,1,9,MAXCOL()-1," |J-L| ,C_SOMBRA")
```

```
// Se despliega el nombre del sistema, fecha, versión y registro
```

```
@ 3,3 SAY DATE() PICTURE "@e"
@ 3,68 SAY "Ver. 1.0"
cColor := SETCOLOR ("GR+N")
@ 3,24 SAY " UNIVERSIDAD VASCO DE QUIROGA "
SETCOLOR (cColor)
cColor := SETCOLOR ("R+N")
@ 5,8 SAY "CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados"
SETCOLOR (cColor)
```

```
/* Este ciclo se repite infinitamente, hasta que se pulsa una opción
valida en el men. Para salir del ciclo se presiona la tecla ESC
o la opción "Salir al Dos", tras contestar si a la pregunta de
abandonar se regresa al DOS y el sistema termina su ejecución */
```

```
IF nPriv > 0 .AND. nPriv < 6
    menu_main_sce ()
    SETCOLOR (colores)
    CLS
    QUIT
ELSEIF nPriv > 5 .AND. nPriv < 11
    main_sce ()
    SETCOLOR (colores)
    CLS
    QUIT
ENDIF
```

```
lineamenu := SAVESCREEN (4,1,4,80)
```

```
DO WHILE .T.
    SET KEY K_19 TO claves ()
    cColor2 := SETCOLOR ()
    nCentra := (78-LEN ("MENU PRINCIPAL DEL SISTEMA"))/2
    cColor := SETCOLOR ("G+N")
    @ 4, nCentra SAY "MENU PRINCIPAL DEL SISTEMA"
    SETCOLOR (cColor2)
    DISPBEGIN ()
    @ 22,1 SAY SPACE(78) // Opciones del men. principal
    @ 8,2 SAY SPACE(76) // Opciones del men. principal
    @ 8,3 PROMPT " Control Escolar " MESSAGE "Ir al Sistema De Control Escolar"
    @ 8,21 PROMPT " Cuentas Por Cobrar " MESSAGE "Ir al Sistema De Cuentas por Cobrar"
    @ 8,42 PROMPT " Salir al DOS " MESSAGE "Terminar y Salir al DOS"
    DISPEND ()
    MENU TO nOp // Se registra la selección del usuario

    IF LASTKEY() == K_PGUP .OR. LASTKEY() == K_PGDN
        LOOP
    ENDIF
```

```

IF LASTKEY() == K_ESC // Si la ultima tecla pulsada es ESC
  TONE (700)
  nSalir := ALERT ("Terminar y salir al DOS", aOpcion)
  IF nSalir == 2
    SETCOLOR (colores)
    CLS
    QUIT
  ENDIF
ENDIF

IF nOp != 0
  DO CASE
  CASE nOp == 1
    RESTSCREEN (4,1,4,80,lineamenu)
    TONE (1200)
    menu_main_sce () // Menu principal del SCE
  CASE nOp == 2
    RESTSCREEN (4,1,4,80,lineamenu)
    TONE (750)
    main_sce ()
  CASE nOp == 3 // " Salir al DOS ?
    RESTSCREEN (4,1,4,80,lineamenu)
    TONE (800)
    nSalir := ALERT ("Terminar y salir al DOS", aOpcion)
    IF nSalir == 2
      SETCOLOR (colores)
      CLS
      QUIT
    ENDIF
  ENDCASE
ENDIF
ENDDO
RETURN NIL

FUNCTION menu_main_sce ()
  LOCAL nOp := 1 // variable que contendra la opcion deseada en el menf
  LOCAL cColores // variable que contendra los colores del entorno DOS
  LOCAL aOpcion := {"No","Si"} // opciones confirmaci3n de salida
  LOCAL cColor, color, lineamenu, colores := SETCOLOR()
  PUBLIC esc_selec, nSalir

  lineamenu := SAVE:SCREEN (4,1,4,80)
  DO WHILE .T.
    cColor2 := SETCOLOR()
    nCentra := (78-LEN ("MENU PRINCIPAL DEL SISTEMA DE CONTROL ESCOLAR"))/2
    cColor := SETCOLOR ("BG+/N")
    @ 4, nCentra SAY "MENU PRINCIPAL DEL SISTEMA DE CONTROL ESCOLAR"
    SETCOLOR (cColor2)
    DISPBEGIN ()
    @ 22,1 SAY SPACE(78) // Opciones del menf principal
    @ 8 ,2 SAY SPACE(76) // Opciones del menf principal
    @ 8 ,3 PROMPT " Seleccionar " MESSAGE "Selecciona la Escuela para Trabajar"
    @ 8 ,19 PROMPT " Utilerias " MESSAGE "Modulo de utilerias"
    @ 8 ,33 PROMPT " Menu Anterior " MESSAGE "Regresar al shell del sistema"
  DISPEND ()
  MENU TO nOp // Se registra la selecci3n del usuario

  IF LASTKEY() == K_PGUP .OR. LASTKEY() == K_PGDN
    LOOP
  ENDIF

  IF LASTKEY() == K_ESC // Si la ultima tecla pulsada es ESC
    TONE (700)
    nSalir := ALERT ("Regresar al Menf Anterior", aOpcion)
    IF nSalir == 2
      RESTSCREEN (4,1,4,80,lineamenu) // Borra la linea que indica de que menu se trata
      SETCOLOR (colores)
      RETURN
    ENDIF
  ENDIF

```

```

IF nOp != 0
DO CASE
CASE nOp == 1 // se llama la rutina de seleccion de escuela
RESTSCREEN (4,1,4,80,lineamenu) // Borra la linea que indica de que menu se trata
// Para seleccionar la escuela no es necesario tener privilegios especiales
escuela := selecciona()
IF escuela > 0
esc_selec := escuela->nomesc
DBCLOSEAREA() // Se cierran las bases aqui para no perder el numero de registro seleccionado
menu_sist (esc_selec,escuela)
ELSE
DBCLOSEAREA()
ENDIF
CASE nOp == 2 // Se llama la rutina del menf de utilerias
RESTSCREEN (4,1,4,80,lineamenu) // Borra la linea que indica de que menu se trata
IF nPriv == 0
menu_utilerias ()
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

CASE nOp == 3 // ~ Salir al DOS ?
TONE (800)
nSalir := ALERT ("Regresar al Menf Anterior", aOpcion)
IF nSalir == 2
RESTSCREEN (4,1,4,80,lineamenu) // Borra la linea que indica de que menu se trata
SETCOLOR (colores)
RETURN
ENDIF
ENDCASE
ENDIF
ENDDO
RETURN

```

/* Este procedimiento es el encargado de desplegar todos los submenús del sistema, el parametro nCol indica el número del submenú que se desea desplegar. */

```

PROCEDURE menu_sec_sist (nCol)
LOCAL nOpcion := 1, pantalla, nY1,nY2,nX1,nX2, cColor3,flag := .F. // variables locales
LOCAL pantalla2

cColor3 := SETCOLOR ()
IF nCol == 1
nX1 := 9
nY1 := 3
nX2 := 15
nY2 := 24
ELSEIF nCol == 2
nX1 := 9
nY1 := 12
nX2 := 14
nY2 := 33
ELSEIF nCol == 3
nX1 := 9
nY1 := 22
nX2 := 14
nY2 := 43
ELSEIF nCol == 4
nX1 := 9
nY1 := 32
nX2 := 13
nY2 := 53
ELSEIF nCol == 5
nCol := 7
nX1 := 9
nY1 := 51
nX2 := 15
nY2 := 76
ENDIF
pantalla := SAVESCREEN (nX1,nY1-1,nX2,nY2)

```

```

DO WHILE .T.
  DISPBEGIN ()
  IF nCol == 5 .AND. flag .OR. nCol == 6 .AND. flag
    pantalla2 := SAVESCREEN (nX1,nY1-1,nX2,nY2)
  ENDIF

  DispBox(nX1+2, nY1, nX2, nY2, "██████████",C_SOMBRA)
  IF nCol == 5
    DispBox(nX1+1, nY1-1, nX2-1, nY2-1, "┌─┴─┐ ")
  ELSE
    DispBox(nX1, nY1-1, nX2-1, nY2-1, "┌─┴─┐ ")
  ENDIF
  ENDIF
  @ 22,1 SAY SPACE(78) // Borrar Mensajes del men anterior
  IF nCol == 1
    @ 10 ,nY1 PROMPT " Inscripciones ... " MESSAGE "Inscripciones y Reinscripciones de alumnos"
    @ 11 ,nY1 PROMPT " Bajas ... " MESSAGE "Baja Temporal o Total de Alumnos"
    @ 12 ,nY1 PROMPT " Recuperacion ... " MESSAGE "Recuperacin de Alumnos dados de Baja"
    @ 13 ,nY1 PROMPT " Calificaciones . " MESSAGE "SubMen de Calificaciones"
  ELSEIF nCol == 2
    @ 10 ,nY1 PROMPT " Altas ... " MESSAGE "Alta de Maestros"
    @ 11 ,nY1 PROMPT " Bajas ... " MESSAGE "Baja de Maestros"
    @ 12 ,nY1 PROMPT " Modificaciones ... " MESSAGE "Modificacin de Informacin referente a Maestros"
  ELSEIF nCol == 3
    @ 10 ,nY1 PROMPT " Altas ... " MESSAGE "Alta de Materias"
    @ 11 ,nY1 PROMPT " Bajas ... " MESSAGE "Baja de Materias"
    @ 12 ,nY1 PROMPT " Modificaciones ... " MESSAGE "Modificacin de Informacin referente a Materias"
  ELSEIF nCol == 4
    @ 10 ,nY1 PROMPT " Formar Grupos ... " MESSAGE "Formacin de Grupos"
    @ 11 ,nY1 PROMPT " Asignar Materias . " MESSAGE "Asignacin de la Carga de Materias a los alumnos"
  ELSEIF nCol == 5
    @ 12 ,nY1 PROMPT " Por Grupo ... " MESSAGE "Asignacin de Materias general"
    @ 13 ,nY1 PROMPT " Por Alumno ... " MESSAGE "Asignacin de Materias por alumno especifico"
  ELSEIF nCol == 6
    @ 13 ,nY1 PROMPT " Por Grupo ... " MESSAGE "Manejo de Calificaciones de todo un grupo"
    @ 14 ,nY1 PROMPT " Por Alumno ... " MESSAGE "Manejo de Calificaciones por alumno especifico"
  ELSEIF nCol == 7
    @ 10 ,nY1 PROMPT " Inicio de Semestre " MESSAGE "Inicializacin de un nuevo semestre"
    @ 11 ,nY1 PROMPT " Respalidar Informacin " MESSAGE "Respaldo de las bases de datos"
    @ 12 ,nY1 PROMPT " Recuperar Informacin " MESSAGE "Respaldo de las bases de datos"
    @ 13 ,nY1 PROMPT " Reindexar Bases " MESSAGE "Genera los indices necesarios para las bases de datos"
  ENDIF
  DISPEND()
  MENU TO nOpcion

  IF LASTKEY() == K_PGUP .OR. LASTKEY() == K_PGDN
    LOOP
  ENDIF

  IF LASTKEY() == K_ENTER
    IF nCol == 4 .AND. nOpcion == 1
      RESTSCREEN (nX1,nY1-1,nX2,nY2,pantalla)
    ELSEIF nCol == 1 .AND. nOpcion <= 3 .OR. nCol==2 .OR. nCol==3
      RESTSCREEN (nX1,nY1-1,nX2,nY2,pantalla)
    ENDIF
  ENDIF

  IF LASTKEY() == K_ESC // Si la ultima tecla pulsada es ESC
    IF nCol == 5
      RESTSCREEN (nX1,nY1-1,nX2,nY2,pantalla2)
      flag := .T.
      nCol := 4
      nY1 := 32
      nY2 := 53
      nX1 := 9
      nX2 := 13

      nOpcion := 2
      LOOP
    ENDIF
    IF nCol == 6
      RESTSCREEN (nX1,nY1-1,nX2,nY2,pantalla2)
      flag := .T.
    ENDIF
  ENDIF

```

```

nCol := 1
nY1 := 3
nY2 := 24
nX1 := 9
nX2 := 15
nOpcion := 4
LOOP
ENDIF

```

```

RESTSCREEN (nX1,nY1-1,nX2,nY2,pantalla)
SETCOLOR (cColor3)
RETURN
ENDIF

```

```

IF nOpcion != 0
DO CASE
CASE nopcion == 1
IF nCol == 1
IF nPriv <= 3
alta_alumnos() // Alta de alumnos
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF
ELSEIF nCol ==2
IF nPriv <= 3
alta_maes() // Alta de maestros
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF
ELSEIF nCol ==3
IF nPriv <= 3
alta_mats() // Alta de materias
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF
ELSEIF nCol ==4
IF nPriv <= 3
alta_grupos() // Formaci3n de grupos
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF
ELSEIF nCol ==5
IF nPriv <= 3
flag := .F. // Asignacion por grupo
asigna_grupo()
nOpcion := 1
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF
ELSEIF nCol == 6
IF nPriv <= 2
flag := .F.
calif_grupo (.T.) // Calificaciones por grupo
nOpcion := 1
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF
ELSEIF nCol == 7
IF nPriv == 0
inicia_sem () // Inicializaci3n de semestre
ELSE
TONE (750)
ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")

```



```

ENDIF
ENDIF
CASE nOpcion == 2
IF nCol == 1
IF nPriv <= 2
  baja_alum() // Baja de alumnos
ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ELSEIF nCol ==2
IF nPriv <= 2
  baja_mae() // Baja de maestros
ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ELSEIF nCol ==3
IF nPriv <= 2
  baja_mats() // Baja de materias
ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ELSEIF nCol ==4
IF nPriv <= 3
  nY1 := 53
  nY2 := 70
  nX1 := 10
  nX2 := 15
  nCol := 5
  nOpcion := 1
  flag := .T.
ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ELSEIF nCol == 5
IF nPriv <= 3
  flag := .F.
  alumno_asigna() // Asignación por alumno
  nOpcion := 2
ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ELSEIF nCol == 6
IF nPriv <= 2
  flag := .F.
  cal_alumno (.T.) // Calificaciones por alumno
  nOpcion := 2
ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ELSEIF nCol == 7
IF nPriv <= 1
  respaldo () // Respaldo de las bases de datos del SCE
  nOpcion := 2
ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ENDIF
CASE noopcion == 3
IF nCol == 1
IF nPriv <= 2
  recup_alum() // Recuperar alumnos dados de baja

```

```

ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF

ELSEIF nCol ==2
  IF nPriv <= 3
    modif_mae() // Modificacion de maestros
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF

ELSEIF nCol ==3
  IF nPriv <= 3
    modif_mats() // Modificacion de materias
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF

ELSEIF nCol == 7
  IF nPriv <= 1
    restaura () // Restaurar las bases de datos
    nOpcion := 3
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
ENDIF

CASE nopcion == 4
  IF nCol == 1
    IF nPriv <= 3
      nOpcion := 1
      nX1 := 12
      nY1 := 24
      nX2 := 16
      nY2 := 41
      nCol := 6
      flag := .T.
    ELSE
      TONE (750)
      ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
    ENDIF
  ENDIF
ENDIF

IF nCol == 7
  IF nPriv <= 4
    reindexa (1) // Regenera los indices de una escuela
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
ENDIF
ENDCASE
ENDIF
ENDDO
RETURN NIL

PROCEDURE menu_repor ()
  LOCAL nOpcion := 1, pantalla,cColor,cColor2,cCaso := 0 // variables locales

  SETCOLOR (C_LIMPIA)
  cColor := SETCOLOR (C_LIMPIA)
  IF cTipo # "L"
    cCaso := 1
  ENDIF

  pantalla := SAVESCREEN (9,29,19,64)
  DO WHILE .T.
    DISPBEGIN ()
    DispBox(11, 30, 19, 64, "XXXXXXXXXX",C_SOMBRA)
    DispBox(9 , 29, 18, 63, " |J-L| ")
    @ 22,1 SAY SPACE(78) // Borrar Mensajes del menL anterior

```

```

@ 10,30 PROMPT " Boletas de Calificaciones ." MESSAGE "Imprime en papel las boletas de calificaciones"
@ 11,30 PROMPT " Calificaciones ." MESSAGE "Despliega en pantalla calificaciones"
@ 12,30 PROMPT " Credenciales ." MESSAGE "Imprime credenciales de identificación"
IF ctipo == "L"
@ 13,30 PROMPT " Concentrado de calificaciones ." MESSAGE "Imprime en papel los concentrados de calificaciones"
ELSE
  cColor2 := SETCOLOR(C_SMENU)
@ 13,30 SAY " Concentrado de calificaciones ."
  SETCOLOR(cColor2)
ENDIF
@ 14,30 PROMPT " Constancia de Estudios ..." MESSAGE "Imprime en papel las listas de asistencia"
@ 15,30 PROMPT " Listas de Asistencia ." MESSAGE "Imprime en papel las listas de asistencia"
@ 16,30 PROMPT " Actualización del KARDEX " MESSAGE "Guarda la información del semestre de todos los alumnos en el
KARDEX"
@ 17,30 PROMPT " Impresión de CFIYA ..." MESSAGE "Imprime en papel las CFIYAS"
DISPEND()
MENU TO nOpcion

IF LASTKEY() == K_PGUP .OR. LASTKEY() == K_PGDN
  LOOP
ENDIF

IF LASTKEY() == K_ESC // Si la ultima tecla pulsada es ESC
  RESTSCREEN (9,29,19,64,pantalla)
  SETCOLOR (cColor)
  RETURN
ENDIF

IF nOpcion != 0 // Desistió
DO CASE
CASE nopcion == 1
  IF nPriv <= 2
    menu2_sec_sist (1) // Boletas
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nOpcion == 2
  IF nPriv <= 5
    menu2_sec_sist (2) // Calificaciones
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == 3
  IF nPriv <= 3
    menu2_sec_sist (3) // Credenciales
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == 4 .AND. cTipo == "L"
  IF nPriv <= 3
    menu2_sec_sist (5) // Concentrado de calificaciones
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == (5-cCaso)
  IF nPriv <= 3
    editor () // Constancia de estudios
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF

CASE nopcion == (6-cCaso)
  IF nPriv <= 3
    menu2_sec_sist (4) // listas de asistencia
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF

```

```

ENDIF
CASE nopcion == (7-cCaso)
  IF nPriv <= 1
    actualiza_kardex ()      // Kardex
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == (8-cCaso)
  IF nPriv <= 1
    cfiya()                // cfiya
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
OTHERWISE
  TONE (500)
ENDCASE
ENDIF
ENDDO
RETURN NIL

```

// Procedimiento que despliega todos los sub-submenús del sistema. (Excluyendo el de reportes)

```
PROCEDURE menu2_sec_sist (nCol)
```

```
  LOCAL pantalla, nY1, nY2, nX1, nX2, cColor3 // variables locales
```

```
  LOCAL flag := .T., pantalla2
```

```
  STATIC nOpcion := 1
```

```
  cColor3 := SETCOLOR ()
```

```
  IF nCol == 1
```

```
    nX1 := 10
```

```
    nY1 := 64
```

```
    nX2 := 14
```

```
    nY2 := 78
```

```
  ELSEIF nCol == 2
```

```
    nX1 := 11
```

```
    nY1 := 64
```

```
    nX2 := 15
```

```
    nY2 := 78
```

```
  ELSEIF nCol == 3
```

```
    nX1 := 12
```

```
    nY1 := 64
```

```
    nX2 := 16
```

```
    nY2 := 78
```

```
  ELSEIF nCol == 4
```

```
    /* Si se esta en licenciatura es necesario aumentar una línea ya que
    el menú de reportes contendrá la opción de concentrado de califs. */
```

```
    IF cTipo == "L"
```

```
      nX1 := 14
```

```
      nY1 := 64
```

```
      nX2 := 18
```

```
      nY2 := 78
```

```
    ELSE
```

```
      nX1 := 13
```

```
      nY1 := 64
```

```
      nX2 := 17
```

```
      nY2 := 78
```

```
    ENDIF
```

```
  ELSEIF nCol == 5
```

```
    nX1 := 13
```

```
    nY1 := 64
```

```
    nX2 := 17
```

```
    nY2 := 78
```

```
  ENDIF
```

```
  pantalla := SAVESCREEN (nX1, nY1-1, nX2, nY2)
```

```
  DO WHILE .T.
```

```
    DISPBEGIN ()
```

```
    DispBox(nX1+2, nY1, nX2, nY2, "#####", C_SOMBRA)
```

```
    DispBox(nX1, nY1-1, nX2-1, nY2-1, " |J-L| ")
```

```
    @ 22,1 SAY SPACE(78) // Borrar Mensajes del menú anterior
```

```
    IF nCol == 1
```

```

    @ nX1+1 ,nY1 PROMPT " Por Grupo " MESSAGE "Imprime Boletas de Calificaciones de todo un Grupo"
    @ nX1+2 ,nY1 PROMPT " Por Alumno " MESSAGE "Imprime Boletas de Calificaciones de un Alumno en particular"
ELSEIF nCol == 2
    @ nX1+1 ,nY1 PROMPT " Por Grupo " MESSAGE "Despliega en pantalla las calificaciones de todo un grupo"
    @ nX1+2 ,nY1 PROMPT " Por Alumno " MESSAGE "Despliega en pantalla las calificaciones de un Alumno en particular"
ELSEIF nCol == 3
    @ nX1+1 ,nY1 PROMPT " Por Grupo " MESSAGE "Imprime las credenciales de todo un Grupo"
    @ nX1+2 ,nY1 PROMPT " Por Alumno " MESSAGE "Imprime las credenciales de un Alumno en particular"
ELSEIF nCol == 4
    @ nX1+1 ,nY1 PROMPT " Por Grupo " MESSAGE "Imprime todas las listas de asistencia de un Grupo"
    @ nX1+2 ,nY1 PROMPT " Por Materia " MESSAGE "Imprime la lista de asistencia por Materia dentro de un grupo en particular"
ELSEIF nCol == 5
    @ nX1+1 ,nY1 PROMPT " Por Grupo " MESSAGE "Imprime el concentrado de calificaciones por grupo"
    @ nX1+2 ,nY1 PROMPT " Por Materia " MESSAGE "Imprime el concentrado de calificaciones por materia de un grupo particular"
ENDIF
DISPEND ()
MENU TO nOpcion

IF LASTKEY() == K_PGUP .OR. LASTKEY() == K_PGDN
    LOOP
ENDIF

IF LASTKEY() == K_ESC // Si la ultima tecla pulsada es ESC
    RESTSCREEN (nX1,nY1-1,nX2,nY2,pantalla)
    RETURN
ENDIF

IF nOpcion != 0
DO CASE
CASE nopcion == 1 .AND. nCol == 1
    gpo_boletas () // Boletas por Grupo
CASE nOpcion == 2 .AND. nCol == 1
    boletas_alumno () // Boletas por alumno
CASE nopcion == 1 .AND. nCol == 2
    calif_grupo (.F.) // Consulta de Calificaciones por grupo
CASE nOpcion == 2 .AND. nCol == 2
    cal_alumno (.F.) // Consulta de Calificaciones por alumno
CASE nopcion == 1 .AND. nCol == 3
    credenciales_grupo () // Credenciales por grupo
CASE nOpcion == 2 .AND. nCol == 3
    alumno_credenciales () // Credenciales por alumno
CASE nopcion == 1 .AND. nCol == 4
    listas_grupo () // Listas por grupo
CASE nOpcion == 2 .AND. nCol == 4
    mats_listas () // Listas por materia
CASE nOpcion == 1 .AND. nCol == 5
    concen_grupo () // Concentrado de Calif. por grupo
CASE nOpcion == 2 .AND. nCol == 5
    concen_mats () // Concentrado de Calif. por materia
END CASE
ENDIF
ENDDO
RETURN

```

SCE2.PRG

```

/* Programa   : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje     : CLIPPER 5.2
   Enlazador    : RTLINK 3.14B
*/

#xcommand REPEAT => do while .t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla
#include "caos.ch"

PROCEDURE menu_utilerias ()
LOCAL nOpcion := 1, pantalla, ok_bases // variables locales
PUBLIC cNom_esc, cClave_esc, cTipo_esc := SPACE(1), mTiposep2 := " "

aOpcion := {"Aceptar", "Cancelar"} // opciones confirmaci3n de INDEX
pantalla := SAVESCREEN (9,19,15,37)
DO WHILE .T.
  DISPBEGIN ()
  DispBox(11, 20, 15, 37, "#####", C_SOMBRA)
  DispBox(9, 19, 14, 36, " | | | | | ")
  @ 22,1 SAY SPACE(78) // Borrar Mensajes del men3 principal
  @ 10,20 PROMPT " Altas      ... " MESSAGE "Modulo de Alta de Escuelas"
  @ 11,20 PROMPT " Bajas      ... " MESSAGE "Modulo de Baja de Escuelas"
  @ 12,20 PROMPT " Modificaciones " MESSAGE "Modulo de Modificaciones de datos de escuelas"
  @ 13,20 PROMPT " ReIndexar Base " MESSAGE "Modulo de ReIndexaci3n de la Base de Datos"
  DISPEND ()
  MENU TO nOpcion

  IF LASTKEY() == K_PGUP .OR. LASTKEY() == K_PGDN
    LOOP
  ENDIF

  RESTSCREEN (9,19,15,37,pantalla)
  tecla = LASTKEY () // Se almacena la ultima tecla pulsada
  IF (tecla == K_ESC) // Si la ultima tecla pulsada es ESC
    RETURN
  ENDIF

  IF nopcion != 0
    DO CASE
      CASE nopcion == 1
        IF MEMORY (2) > 60
          alta_esc()
        ELSE
          TONE (550)
          ALERT (" NO HAY MEMORIA DISPONIBLE PARA PODER UTILIZAR ESTE MODULO DEBEN HABER 86Kb
LIBRES EN MEMORIA, PARA LIBERAR MEMORIA RETIRE ALGUNOS PROGRAMAS TSR")
        ENDIF
      CASE nOpcion == 2
        baja_esc()
      CASE nopcion == 3
        modi_esc()
      CASE nopcion == 4
        IF !ALIAS () == "ESCUELA"
          TONE (600)
          limpiamenus (1)
          reindexa (2) // Regenera los indices principales
        ELSE
          TONE(550)
          ALERT (" — NO SE PUEDE ACCEDER AL ARCHIVO !!! EN ESTE MOMENTO NO SE
PUEDE REALIZAR ESTA OPERACION ")
        ENDIF
      ENDCASE
    ENDIF
  ENDIF

```

```
ENDDO
RETURN NIL
```

```
// Función que despliega el menu de trabajo general de las escuelas
PROCEDURE menu_sist (cNom_esc, nReg_esc)
  LOCAL nOpcion := 1, pantalla, nCentra // variables locales
  LOCAL aOpcion := {"No", "Si"} // opciones confirmación de salida
  LOCAL cuadro := SAVESCREEN (9,18,16,63)
  PUBLIC cClave, cTipo, cPath, ok_bases, cTipoSep

  // Se cambia al directorio que contiene la informacion de la escuela
  // seleccionada

  ok_bases := abre_bases (1) //Base de escuelas
  IF ok_bases
    GOTO nReg_esc
    cPath := escuela->direct
    cClave := escuela->clave
    cTipo := escuela->tipo
    cTipoSep := escuela->tiposep
    SET PATH TO &cPath
    DBCLOSEAREA()
  ELSE
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
    SETCOLOR ("W/N")
    CLS
    QUIT
  ENDIF

  cColor2 := SETCOLOR()
  cNom_esc := ALLTRIM(cNom_esc)
  nCentra := (78-LEN(cNom_esc))/2
  cColor := SETCOLOR ("BG/N")
  @@ 4, nCentra SAY cNom_esc
  SETCOLOR (cColor2)
  DO WHILE .T.
    SET PATH TO &cPath
    DISPBEGIN ()
    @ 22,1 SAY SPACE(78) // Borrar Mensajes del menf principal
    @ 8,2 SAY SPACE(76) // Borra las opciones del menf anterior
    @ 8,3 PROMPT " Alumnos " MESSAGE "Manejo de la informacion referente a Alumnos"
    @ 8,12 PROMPT " Maestros " MESSAGE "Manejo de la informacion referente a Maestros"
    @ 8,22 PROMPT " Materias " MESSAGE "Manejo de la informacion referente a Materias"
    @ 8,32 PROMPT " Grupos " MESSAGE "Manejo de la informacion referente a Grupos"
    @ 8,40 PROMPT " Reportes " MESSAGE "Generador de reportes"
    @ 8,50 PROMPT " Utilerias " MESSAGE "Utilerias del sistema"
    @ 8,62 PROMPT " Menf Anterior " MESSAGE "Terminar y regresar al Menf principal"
  DISPEND ()
  MENU TO nOpcion

  IF LASTKEY() == K_PGUP .OR. LASTKEY() == K_PGDN
    LOOP
  ENDIF

  IF LASTKEY() == K_ESC // Si la ultima tecla pulsada es ESC
    TONE (700)
    nSalir := ALERT ("Regresar al Menf Anterior", aOpcion)
    IF nSalir == 2
      @ 4,2 SAY SPACE(76)
      SET PATH TO
      RETURN
    ENDIF
  ENDIF

  IF nOpcion != 0
    DO CASE
      CASE nopcion == 1
        IF nPriv <= 4
          menu_sec_sist (1) // Alumnos
```

```

ELSE
  TONE (750)
  ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
ENDIF
CASE nOpcion == 2
  IF nPriv <= 4
    menu_sec_sist (2)      // Maestros
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == 3
  IF nPriv <= 4
    menu_sec_sist (3)      // Materias
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == 4
  IF nPriv <= 4
    menu_sec_sist (4)      // Grupos
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == 5
  menu_repor ()           // Reportes
CASE nopcion == 6
  IF nPriv <= 4
    menu_sec_sist (5)      // Utilerias
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nopcion == 7
  TONE (700)
  nSalir := ALERT ("Regresar al MenL Anterior", aOpcion)
  IF nSalir == 2
    @ 4,2 SAY SPACE(76)
    SET PATH TO
    RETURN
  ENDIF
ENDCASE
ENDIF
ENDDO
RETURN NIL

```

// Apaga las opciones de los distintos menus

PROCEDURE limpiamenus (tipo_menu)

```

IF tipo_menu == 1
  @ 8,3 SAY " Seleccionar "
  @ 8,19 SAY " Utilerias "
  @ 8,33 SAY " Salir al DOS "
ELSEIF tipo_menu == 2
  @ 8,3 SAY " Alumnos "
  @ 8,12 SAY " Maestros "
  @ 8,22 SAY " Materias "
  @ 8,32 SAY " Grupos "
  @ 8,40 SAY " Reportes "
  @ 8,50 SAY " Utilerias "
  @ 8,62 SAY " MenL Anterior "
ELSEIF tipo_menu == 2
  @ 8,3 SAY " Seleccionar "
  @ 8,18 SAY " Reportes "
  @ 8,30 SAY " Utilerias "
  @ 8,43 SAY " Menu Anterior "
ENDIF
RETURN NIL

```



```

/* Se despliegan en una ventana todas las escuelas que estan dadas de alta
para que el usuario seleccione la escuela con la que va a trabajar */
FUNCTION selecciona(scc)
LOCAL nKey, ok_bases
pantalla := SAVESCREEN (10,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura
pie (4)
IIF (scc == NIL,scc := .T.,scc := scc)
IIF (scc,limpiamenus(1),limpiamenus (3)) // Se apagan las opciones del menf
DISPBOX (10,1,maxrow()-3,maxcol()-1,"┌──┐└──┘",C_REYENO) // Se dibuja la ventana de captura
@ 9,14 SAY REPLICATE ("-",15)
cColor:= SETCOLOR (C_CAP)
ok_bases := abre_bases (1) // Base de escuelas
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
SETCOLOR ("W/N")
CLS
QUIT
ENDIF

otb:= TBROWSEDB (11,2,maxrow()-4,maxcol()-2)
otb:ADDCOLUMN(TBCOLUMNNEW("Clave",{!escuela->clave}))
otb:ADDCOLUMN(TBCOLUMNNEW("Nivel",{!nivel(escuela->tipo)}))
otb:ADDCOLUMN(TBCOLUMNNEW("Nombre de la Escuela",{!escuela->nomesc}))
otb:COLSEP:=" |"
otb:HEADSEP:="-"
otb:RIGHT ()
otb:RIGHT ()
DO WHILE .T.
DO WHILE !otb:STABILIZE()
ENDDO

/* El reempinkey con estos parametros es exclusivo para los
Tbrowsers, esto es para que se realice un refresco cada
10 segundos. Esto es sobre todo para cuando se trabaja
en ambiente de red. */

nKey := reempinkey (0)

SET CURSOR OFF

DO CASE
CASE nKey == K_ENTER // Se a seleccionado una opcion
RESTSCREEN (10,1,MAXROW()-3,MAXCOL()-1,pantalla)
SETCOLOR (cColor)
@ MAXROW()-1,1 SAY SPACE (78)
IF !EMPTY (escuela->nomesc)
RETURN RECNO()
ELSE
RETURN -1
ENDIF
CASE nKey == K_ESC // Se a cancelado la operacion
escuela->(DBCLOSEAREA())
RESTSCREEN (10,1,MAXROW()-3,MAXCOL()-1,pantalla)
SETCOLOR (cColor)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN -1
CASE nKey==K_UP ;otb:UP()
CASE nKey==K_DOWN ;otb:DOWN()
CASE nKey==K_LEFT ;otb:LEFT()
CASE nKey==K_CTRL_LEFT ;otb:PANLEFT()
CASE nKey==K_RIGHT ;otb:RIGHT()
CASE nKey==K_CTRL_RIGHT ;otb:PANRIGHT()
CASE nKey==K_PGUP ;otb:PAGEUP ()
CASE nKey==K_CTRL_PGUP ;otb:GOTOP()
CASE nKey==K_PGDN ;otb:PAGEDOWN ()
CASE nKey==K_CTRL_PGDN ;otb:GOBOTTOM()
CASE nKey==K_HOME ;otb:HOME()
CASE nKey==K_CTRL_HOME ;otb:PANHOME()
CASE nKey==K_END ;otb:END()
CASE nKey==K_CTRL_END ;otb:PANEND()
OTHERWISE

```

```
TONE(125); TONE(300)
ENDCASE
ENDDO
RESTSCREEN (10,1,MAXROW()-3,MAXCOL()-1,pantalla)
SETCOLOR (cColor)
RETURN NIL
```

/* En base al registro tipo_esc, esta funcion regresa el nivel de la escuela y detecta que no haya inconsistencia en los tipo almacenados en la base */
 FUNCTION nivel (niv)

```
IF niv == "P"
    RETURN "Preparatoria"
ELSEIF niv == "L"
    RETURN "Licenciatura"
ELSEIF niv == "G"
    RETURN "PostGrado"
ELSEIF niv == "D"
    RETURN "Diplomado"
ENDIF
```

IF !EOF()
 ALERT (" ---- ERROR FATAL !!!!! Existe un error en la base de datos, NO PUEDE CONTINUAR LA EJECUCION del sistema mientras no se solucione el problema")

```
SETCOLOR (cColor)
CLS
QUIT
ENDIF
RETURN NIL
```

PROCEDURE pantallacampos (tipo_campos)

```
LOCAL cColorLocal
DISPBEGIN ()
IF tipo_campos < 7
    DISPBOX (10,1,maxrow()-3,maxcol()-1," |J-L|",C_REYENO) // Se dibuja la ventana de captura
    @ 9, 14 SAY REPLICATE ("-",45)
ENDIF
```

cColorLocal:= SETCOLOR (C_CAP)

```
IF tipo_campos == 1
    @ 12, 5 SAY "(P) Preparatoria, (L) Licenciatura, (G) PostGrado, (D) Diplomado :"  

    @ 14, 5 SAY "Nombre Esc. :"  

    @ 16, 5 SAY "Clave Esc. :"  

    @ 18, 5 SAY "(M) SEP, (E) SEE :"  

ELSEIF tipo_campos == 2
    @ 12, 5 SAY "Clave :"  

    @ 14, 5 SAY "Nombre :"  

    @ 16, 5 SAY "Nº Serie :"  

    @ 16, 30 SAY "Area :"  

ELSEIF tipo_campos == 3
    @ 11, 5 SAY "Clave :"  

    @ 11, 31 SAY "Nombre :"  

    @ 13, 5 SAY "R.F.C. :"  

    @ 13, 36 SAY "Nivel Est. :"  

    @ 13, 67 SAY "Sexo :"  

    @ 15, 5 SAY "Domicilio :"  

    @ 15, 64 SAY "C.P. :"  

    @ 17, 5 SAY "Ciudad :"  

    @ 17, 45 SAY "Colonia :"  

    @ 19, 5 SAY "Telefono :"  

    @ 19, 51 SAY "Fecha Ingreso :"  

ELSEIF tipo_campos == 4
    @ 11, 5 SAY "Matricula :"  

    @ 11, 30 SAY "Semestre :"  

    @ 11, 53 SAY "Grupo :"  

    @ 12, 5 SAY "Nombre :"  

    @ 12, 53 SAY "Sexo :"  

    @ 13, 5 SAY "Domicilio :"  

    @ 14, 5 SAY "Colonia :"  

    @ 14, 50 SAY "C.P. :"  

    @ 15, 5 SAY "Ciudad :"  

    @ 15, 50 SAY "Telefono :"  

    @ 16, 5 SAY "Esc. Procedencia :"  

    @ 17, 5 SAY "Nombre Padre o Tutor :"
```

```

@ 18, 5 SAY "Tel. Padre o Tutor :."
@ 19, 5 SAY "Fecha de Inscripción :."
ELSEIF tipo_campos == 5
@ 11, 5 SAY "Grupo :."
@ 11,20 SAY "Clave Materia :."
@ 11,48 SAY "Clave Profesor :."
@ 12, 5 SAY "Materia :."
@ 13, 5 SAY "Profesor :."
@ 15, 5 SAY "H O R A R I O"
@ 16, 5 SAY "De: A: Salón: Edificio: Día :."
@ 17, 5 SAY "De: A: Salón: Edificio: Día :."
@ 18, 5 SAY "De: A: Salón: Edificio: Día :."
@ 19, 5 SAY "De: A: Salón: Edificio: Día :."
ELSEIF tipo_campos == 6
@ 11, 5 SAY "Clave :."
@ 11,31 SAY "Nombre :."
@ 13, 5 SAY "R.F.C. :."
@ 13,36 SAY "Nivel Est :."
ELSEIF tipo_campos == 7
@ 17, 5 SAY "Clave :."
@ 17,31 SAY "Nombre :."
@ 19, 5 SAY "R.F.C. :."
@ 19,36 SAY "Nivel Est :."
ELSEIF tipo_campos == 8
@ 11, 5 SAY "Grupo :."
ELSEIF tipo_campos == 9
@ 11, 5 SAY "Grupo :."
@ 13, 5 SAY "Clave Alumno :."
ELSEIF tipo_campos == 10
@ 11, 5 SAY "Grupo :."
@ 13, 5 SAY "Materia :."
@ 15, 5 SAY "Evaluación :."
ELSEIF tipo_campos == 11
@ 11, 5 SAY "Grupo :."
@ 13, 5 SAY "Clave Alumno :."
@ 15, 5 SAY "Evaluación :."
ELSEIF tipo_campos == 12
@ 11, 5 SAY "Grupo :."
@ 14, 5 SAY "Impresora predeterminada :."
ELSEIF tipo_campos == 13
@ 11, 5 SAY "Grupo :."
@ 11,20 SAY "Clave Materia :."
@ 12, 5 SAY "Materia :."
@ 14, 5 SAY "Impresora predeterminada :."
ELSEIF tipo_campos == 14
@ 11, 5 SAY "Grupo :."
@ 12, 5 SAY "Clave Alumno :."
@ 13, 5 SAY "Nombre :."
@ 15, 5 SAY "Firma :."
@ 16, 5 SAY "Cargo :."
@ 17, 5 SAY "Comentario :."
ELSEIF tipo_campos == 14.5
@ 11, 5 SAY "Grupo :."
@ 13, 5 SAY "Firma :."
@ 14, 5 SAY "Cargo :."
@ 15, 5 SAY "Comentario :."
ELSEIF tipo_campos == 15
@ 12, 5 SAY "Grupo :."
@ 12, 26 SAY "Vigencia :."
@ 14, 5 SAY "Impresora predeterminada :."
@ 16, 5 SAY "Firma :."
ELSEIF tipo_campos == 16
@ 11, 5 SAY "Grupo :."
@ 13, 5 SAY "Clave Alumno :."
@ 15, 5 SAY "Vigencia :."
@ 17, 5 SAY "Firma :."
ELSEIF tipo_campos == 17
@ 20, 5 SAY "Impresora Predeterminada :."
ELSEIF tipo_campos == 17.5
@ 19, 5 SAY "Impresora Predeterminada :."
ELSEIF tipo_campos == 18
@ 11, 5 SAY "Grupo :."

```

```

@ 13, 5 SAY "Materia   :."
@ 15, 5 SAY "Jefe del D.C.E. :."
@ 16, 5 SAY "Director de Esc.:"
@ 18, 5 SAY "Impresora Predeterminada :."
ELSEIF tipo_campos == 19
@ 11, 5 SAY "Grupo   :."
@ 12, 5 SAY "Jefe del D.C.E. :."
@ 13, 5 SAY "Director de Esc.:"
@ 15, 5 SAY "Impresora Predeterminada :."
*.....Campos del SCC
ELSEIF tipo_campos == 20
@ 11, 5 SAY "Grupo   :."
@ 12, 5 SAY "Matricula  :."
@ 13, 5 SAY "Alumno    :."
@ 15, 5 SAY "Inscripción :."
@ 16, 5 SAY "Colegiatura :."
ELSEIF tipo_campos == 21
@ 14, 5 SAY "Inscripción :."
@ 16, 5 SAY "Colegiatura :."
ELSEIF tipo_campos == 21.5
@ 11, 5 SAY "Grupo   :."
@ 12, 5 SAY "Matricula  :."
@ 13, 5 SAY "Alumno    :."
@ 15, 5 SAY "Porcentaje de Beca :."
ELSEIF tipo_campos == 22
@ 11, 5 SAY "Grupo   :."
@ 11, 45 SAY "Matricula  :."
@ 12, 5 SAY "Alumno    :."
@ 14, 5 SAY "Mes a pagar :."
@ 15, 5 SAY "Importe    :."
@ 16, 5 SAY "Cantidad   :."
@ 17, 5 SAY "Recargos   :."
@ 18, 5 SAY "Total     :."
@ 19, 5 SAY "Forma de pago :."
@ 20, 5 SAY "Impresora predeterminada :."
ELSEIF tipo_campos == 23
@ 11, 5 SAY "Grupo   :."
@ 11, 45 SAY "Matricula  :."
@ 12, 5 SAY "Alumno    :."
@ 14, 5 SAY "Concepto   :."
@ 16, 5 SAY "Importe    :."
@ 17, 5 SAY "Recargos   :."
@ 18, 5 SAY "Total     :."
@ 19, 5 SAY "Forma de pago :."
@ 20, 5 SAY "Impresora predeterminada :."
ELSEIF tipo_campos == 24
@ 14, 5 SAY "Impresora predeterminada :."
ELSEIF tipo_campos == 25
@ 12, 5 SAY "Grupo   :."
@ 14, 5 SAY "Impresora predeterminada :."
ELSEIF tipo_campos == 26
@ 12, 5 SAY "Matricula  :."
@ 12, 31 SAY "Alumno    :."
@ 14, 5 SAY "Impresora predeterminada :."
ENDIF
DISPEND ()
RETURN NIL

```

```

/* Crea las bases de datos dependiendo del tipo de escuela de que se trate,
esto se conoce por medio del parametro tipo_base */
FUNCTION crea_bases (tipo_base)

```

```

aAlumnos := {}
AADD(aAlumnos, {"MatriAlum", "C", 7, 0})
AADD(aAlumnos, {"Nomalum", "C", 35, 0})
AADD(aAlumnos, {"SemesAlum", "C", 2, 0})
AADD(aAlumnos, {"SexAlum", "C", 1, 0})
AADD(aAlumnos, {"DomAlum", "C", 35, 0})
AADD(aAlumnos, {"ColAlum", "C", 20, 0})
AADD(aAlumnos, {"CpAlum", "C", 5, 0})

```

```

AADD(aAlumnos,{"CiudAlum","C",20,0})
AADD(aAlumnos,{"TelAlum1","C",15,0})
AADD(aAlumnos,{"EscProAlum","C",20,0})
AADD(aAlumnos,{"NomTutAlum","C",35,0})
AADD(aAlumnos,{"TelAlum2","C",15,0})
AADD(aAlumnos,{"FelsAlum","D",8,0})
AADD(aAlumnos,{"Grupo","C",3,0})
AADD(aAlumnos,{"Baja","C",1,0})
IF tipo_base == 1 //Licenciatura
  FOR i = 1 TO 13
    AADD(aAlumnos,{"Mat"+ALLTRIM(STR(i)),"C",5,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+1,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+1,"N",2,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+2,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+2,"N",2,0})
    AADD(aAlumnos,{"F"+ALLTRIM(STR(i))+3,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+3,"N",2,0})
  NEXT
ELSEIF tipo_base == 2 //Preparatoria
  FOR i = 1 TO 13
    AADD(aAlumnos,{"Mat"+ALLTRIM(STR(i)),"C",5,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+1,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+1,"N",2,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+2,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+2,"N",2,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+3,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+3,"N",2,0})
    AADD(aAlumnos,{"S"+ALLTRIM(STR(i)),"N",4,1})
    AADD(aAlumnos,{"IS"+ALLTRIM(STR(i)),"N",2,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+4,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+4,"N",2,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+5,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+5,"N",2,0})
    AADD(aAlumnos,{"P"+ALLTRIM(STR(i))+6,"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i))+6,"N",2,0})
    AADD(aAlumnos,{"F"+ALLTRIM(STR(i)),"N",4,1})
    AADD(aAlumnos,{"IF"+ALLTRIM(STR(i)),"N",2,0})
  NEXT
ELSEIF tipo_base == 3 .OR. tipo_base == 4 //Postgrados
  FOR i = 1 to 13
    AADD(aAlumnos,{"Mat"+ALLTRIM(STR(i)),"C",5,0})
    AADD(aAlumnos,{"U"+ALLTRIM(STR(i)),"N",4,1})
    AADD(aAlumnos,{"I"+ALLTRIM(STR(i)),"N",2,0})
  NEXT
ENDIF
DBCREATE("Alumnos",aAlumnos)

aMaestros := {}
AADD(aMaestros,{"Clavmae","C",5,0})
AADD(aMaestros,{"Nommae","C",35,0})
AADD(aMaestros,{"ricmae","C",13,0})
AADD(aMaestros,{"Nivmae","C",15,0})
AADD(aMaestros,{"Sexmae","C",1,0})
AADD(aMaestros,{"Dommae","C",35,0})
AADD(aMaestros,{"Colmae","C",20,0})
AADD(aMaestros,{"CPmae","C",5,0})
AADD(aMaestros,{"Ciudmae","C",20,0})
AADD(aMaestros,{"Telmae","C",15,0})
AADD(aMaestros,{"Teingmae","D",8,0})
DBCREATE("Maestros",aMaestros)

aMaterias := {}
AADD(aMaterias,{"Clavmat","C",5,0})
AADD(aMaterias,{"Nommat","C",55,0})
AADD(aMaterias,{"Seriemat","C",5,0})
AADD(aMaterias,{"area","C",25,0})
DBCREATE("Materias",aMaterias)

aGrupos := {}
AADD(aGrupos,{"Grupo","C",3,0})
AADD(aGrupos,{"Clavmat","C",5,0})
AADD(aGrupos,{"Clavmae","C",5,0})

```

```

AADD(aGrupos,{"Hora1","C",4,0})
AADD(aGrupos,{"Hora2","C",4,0})
AADD(aGrupos,{"Hora3","C",4,0})
AADD(aGrupos,{"Hora4","C",4,0})
AADD(aGrupos,{"Hora5","C",4,0})
AADD(aGrupos,{"Hora6","C",4,0})
AADD(aGrupos,{"Hora7","C",4,0})
AADD(aGrupos,{"Hora8","C",4,0})
AADD(aGrupos,{"Salon1","C",2,0})
AADD(aGrupos,{"Salon2","C",2,0})
AADD(aGrupos,{"Salon3","C",2,0})
AADD(aGrupos,{"Salon4","C",2,0})
AADD(aGrupos,{"Edifi1","C",2,0})
AADD(aGrupos,{"Edifi2","C",2,0})
AADD(aGrupos,{"Edifi3","C",2,0})
AADD(aGrupos,{"Edifi4","C",2,0})
AADD(aGrupos,{"Dia1","C",1,0})
AADD(aGrupos,{"Dia2","C",1,0})
AADD(aGrupos,{"Dia3","C",1,0})
AADD(aGrupos,{"Dia4","C",1,0})
DBCREATE("Grupos",aGrupos)

```

```

aasigna := {}
AADD(aasigna,{"grupo","C",3,0})
AADD(aasigna,{"clavmat","C",5,0})
AADD(aasigna,{"area","C",25,0})
DBCREATE("asigna",aasigna)

```

```

ainicio := {}
AADD(ainicio,{"kardex","L",1,0})
AADD(ainicio,{"cfiya","L",1,0})
AADD(ainicio,{"cuotas","L",1,0})
DBCREATE("inicio",ainicio)

```

```

aedit := {}
AADD(aedit,{"edit","M",10,0})
DBCREATE("edit",aedit)

```

```

aKardex := {}
AADD(aKardex,{"matrerialum","C",7,0})
AADD(aKardex,{"grupo","C",3,0})
AADD(aKardex,{"clavmat","C",5,0})
IF tipo_base == 1 //Licenciatura
  AADD(aKardex,{"CAL","N",2,0})
  AADD(aKardex,{"EXTRA","N",2,0})
  AADD(aKardex,{"REGULA","N",2,0})
  AADD(aKardex,{"FEXAM","D",8,0})

```

```

ELSEIF tipo_base == 2 //Preparatoria
  AADD(aKardex,{"CALM1","N",2,0})
  AADD(aKardex,{"CALM2","N",2,0})
  AADD(aKardex,{"CALM3","N",2,0})
  AADD(aKardex,{"PROMM","N",2,0})
  AADD(aKardex,{"CAL5","N",2,0})
  AADD(aKardex,{"PROMS","N",2,0})
  AADD(aKardex,{"CALM4","N",2,0})
  AADD(aKardex,{"CALM5","N",2,0})
  AADD(aKardex,{"CALM6","N",2,0})
  AADD(aKardex,{"PROMM2","N",2,0})
  AADD(aKardex,{"CALF","N",2,0})
  AADD(aKardex,{"PROMS2","N",2,0})
ENDIF
AADD(aKardex,{"actualiza","D",8,0})
AADD(aKardex,{"semestre","C",2,0})
DBCREATE("kardex",aKardex)

```

```

aBase := {}
AADD(aBase,{"matrerialum","C",7,0})
AADD(aBase,{"beca","n",3,0})
AADD(aBase,{"inscripcion","N",7,2})
AADD(aBase,{"Cargo1","N",7,2})
AADD(aBase,{"Pagado1","L",1,0})

```

```

AADD(aBase,{"saldo1","N", 7,2})
AADD(aBase,{"Cargo2","N", 7,2})
AADD(aBase,{"Pagado2","L", 1,0})
AADD(aBase,{"saldo2","N", 7,2})
AADD(aBase,{"Cargo3","N", 7,2})
AADD(aBase,{"Pagado3","L", 1,0})
AADD(aBase,{"saldo3","N", 7,2})
AADD(aBase,{"Cargo4","N", 7,2})
AADD(aBase,{"Pagado4","L", 1,0})
AADD(aBase,{"saldo4","N", 7,2})
AADD(aBase,{"Cargo5","N", 7,2})
AADD(aBase,{"Pagado5","L", 1,0})
AADD(aBase,{"saldo5","N", 7,2})
AADD(aBase,{"Cargo6","N", 7,2})
AADD(aBase,{"Pagado6","L", 1,0})
AADD(aBase,{"saldo6","N", 7,2})
AADD(aBase,{"saldoant","N", 7,2})
DBCREATE("edocucnt",aBase)

IF NETERR ()
  ALERT ("NO SE PUEDE REALIZAR LA OPERACION, ESPERE UN MOMENTO E INTENTE NUEVAMENTE")
  RETURN 0
ELSE
  RETURN 1
ENDIF
RETURN NIL

```

```

FUNCTION pie (tipo_pie)
  LOCAL cLocalColor, cColor := SETCOLOR (), x1, x2

  DISPBEGIN ()
  cLocalColor := SETCOLOR ("N/N") // Mensajes al pie de pantalla
  @ MAXROW()-1,1 SAY SPACE (78)
  SETCOLOR (cLocalColor)

  IF tipo_pie == 1
    cLocalColor := SETCOLOR (C_PIE1)
    @ MAXROW()-1,62 SAY "<ESC>.-"
    SETCOLOR (cLocalColor)
    cLocalColor := SETCOLOR (C_PIE2)
    @ MAXROW()-1,70 SAY "Cancelar"
    SETCOLOR (cColor)
  ELSEIF tipo_pie == 2
    cLocalColor := SETCOLOR (C_PIE1) // Mensajes al pie de pantalla
    @ MAXROW()-1,65 SAY "<ESC>.-"
    SETCOLOR (cLocalColor)
    cLocalColor := SETCOLOR (C_PIE2)
    @ MAXROW()-1,73 SAY "Salir"
    SETCOLOR (cColor)
  ELSEIF tipo_pie == 3
    cLocalColor := SETCOLOR (C_PIE1)
    @ MAXROW()-1,2 SAY "<ENTER>.-"
    @ MAXROW()-1,30 SAY "<- ->.-"
    @ MAXROW()-1,65 SAY "<ESC>.-"
    SETCOLOR (cLocalColor)
    cLocalColor := SETCOLOR (C_PIE2)
    @ MAXROW()-1,12 SAY "Seleccionar"
    @ MAXROW()-1,40 SAY "Buscar Otro"
    @ MAXROW()-1,73 SAY "Salir"
    SETCOLOR (cColor)
  ELSEIF tipo_pie == 4
    cLocalColor := SETCOLOR (C_PIE1)
    @ MAXROW()-1,2 SAY "<.>.-"
    @ MAXROW()-1,30 SAY "<ENTER>.-"
    @ MAXROW()-1,65 SAY "<ESC>.-"
    SETCOLOR (cLocalColor)
    cLocalColor := SETCOLOR (C_PIE2)
    @ MAXROW()-1,9 SAY "Buscar"
    @ MAXROW()-1,40 SAY "Seleccionar"
    @ MAXROW()-1,73 SAY "Salir"

```

```

SETCOLOR (cColor)
ELSEIF tipo_pie == 5
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY "F1 .-"
  @ MAXROW()-1,65 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,8 SAY "Busqueda por Nombre"
  @ MAXROW()-1,73 SAY "Salir"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 6
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY "F1 .-"
  @ MAXROW()-1,33 SAY "F2 .-"
  @ MAXROW()-1,62 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,8 SAY "Ayuda"
  @ MAXROW()-1,39 SAY "Baja"
  @ MAXROW()-1,70 SAY "Cancelar"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 7.OR. tipo_pie == 8
  IF tipo_pie == 7
    x1 := 40
    x2 := 47
  ELSEIF tipo_pie == 8
    x1 := 54
    x2 := 61
  ENDIF
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY "< >.-"
  IF tipo_pie == 7
    @ MAXROW()-1,17 SAY "<ENTER>.-"
  ENDIF
  @ MAXROW()-1,x1 SAY "<F6>.- "
  IF tipo_pie == 8
    @ MAXROW()-1,27 SAY "<SUP>.- "
  ENDIF
  IF tipo_pie # 8
    @ MAXROW()-1,65 SAY "<ESC>.-"
  ENDIF
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,9 SAY "Buscar"
  IF tipo_pie == 7
    @ MAXROW()-1,27 SAY "Seleccionar"
  ENDIF
  @ MAXROW()-1,x2 SAY "Cambio de Ventana"
  IF tipo_pie == 8
    @ MAXROW()-1,35 SAY "Eliminar "
  ENDIF
  IF tipo_pie # 8
    @ MAXROW()-1,73 SAY "Salir"
  ENDIF
  SETCOLOR (cColor)
ELSEIF tipo_pie == 9
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY "F1 .-"
  @ MAXROW()-1,62 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,8 SAY "Ayuda"
  @ MAXROW()-1,70 SAY "Cancelar"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 10
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY ".. < -> .-"
  @ MAXROW()-1,25 SAY "<ENTER>.-"
  @ MAXROW()-1,65 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,15 SAY "Buscar"

```



```

@ MAXROW()-1,35 SAY "Seleccionar para Capturar"
@ MAXROW()-1,73 SAY "Salir"
SETCOLOR (cColor)
ELSEIF tipo_pie == 11
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY "<RE_PAG>.-"
  @ MAXROW()-1,28 SAY "<ENTER>.-"
  @ MAXROW()-1,62 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,13 SAY "Seleccionar"
  @ MAXROW()-1,38 SAY "Aceptar selecci3n"
  @ MAXROW()-1,70 SAY "Cancelar"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 12
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY ".. <- ->.-"
  @ MAXROW()-1,65 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,15 SAY "Buscar"
  @ MAXROW()-1,73 SAY "Salir"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 13
  cLocalColor := SETCOLOR (C_PIE1) // Mensajes al pie de pantalla
  @ MAXROW()-1,2 SAY "F2 .-"
  @ MAXROW()-1,65 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,8 SAY "Cambiar Impresora"
  @ MAXROW()-1,73 SAY "Salir"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 14
  cLocalColor := SETCOLOR (C_PIE1) // Mensajes al pie de pantalla
  @ MAXROW()-1,2 SAY "F2 .-"
  @ MAXROW()-1,24 SAY "F10 .-"
  @ MAXROW()-1,43 SAY "F11 .-"
  @ MAXROW()-1,65 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,8 SAY "Cambiar Impresora"
  @ MAXROW()-1,31 SAY "Salvar"
  @ MAXROW()-1,49 SAY "Imprimir"
  @ MAXROW()-1,73 SAY "Salir"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 15
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY "F1 .-"
  @ MAXROW()-1,26 SAY "F2 .-"
  @ MAXROW()-1,62 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,8 SAY "Ayuda"
  @ MAXROW()-1,32 SAY "Cambiar Impresora"
  @ MAXROW()-1,70 SAY "Cancelar"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 16
  cLocalColor := SETCOLOR (C_PIE1)
  @ MAXROW()-1,2 SAY "<ALT+A>.-"
  @ MAXROW()-1,21 SAY "<ALT+B>.-"
  @ MAXROW()-1,40 SAY "<ALT+E>.-"
  @ MAXROW()-1,62 SAY "<ESC>.-"
  SETCOLOR (cLocalColor)
  cLocalColor := SETCOLOR (C_PIE2)
  @ MAXROW()-1,12 SAY "Alta"
  @ MAXROW()-1,31 SAY "Baja"
  @ MAXROW()-1,50 SAY "Editar"
  @ MAXROW()-1,70 SAY "Salir"
  SETCOLOR (cColor)
ELSEIF tipo_pie == 17
  cLocalColor := SETCOLOR (C_PIE1) // Mensajes al pie de pantalla
  @ MAXROW()-1,2 SAY "<F2>.-"

```

```

@ MAXROW()-1,33 SAY "<F3>.-"
@ MAXROW()-1,65 SAY "<ESC>.-"
SETCOLOR (cLocalColor)
cLocalColor := SETCOLOR (C_PIE2)
@ MAXROW()-1,9 SAY "Cambiar Impresora"
@ MAXROW()-1,40 SAY "Alinear Recibos"
@ MAXROW()-1,73 SAY "Salir"
SETCOLOR (cColor)
ELSEIF tipo_pie == 18
cLocalColor := SETCOLOR (C_PIE1)
@ MAXROW()-1,2 SAY "<F2>.-"
@ MAXROW()-1,33 SAY "<F3>.-"
@ MAXROW()-1,62 SAY "<ESC>.-"
SETCOLOR (cLocalColor)
cLocalColor := SETCOLOR (C_PIE2)
@ MAXROW()-1,9 SAY "Cambiar Impresora"
@ MAXROW()-1,40 SAY "Alinear Recibos"
@ MAXROW()-1,70 SAY "Cancelar"
SETCOLOR (cColor)
ELSEIF tipo_pie == 19
cLocalColor := SETCOLOR (C_PIE1)
@ MAXROW()-1,2 SAY "<F1>.-"
@ MAXROW()-1,18 SAY "<F2>.-"
@ MAXROW()-1,37 SAY "<F3>.-"
@ MAXROW()-1,62 SAY "<ESC>.-"
SETCOLOR (cLocalColor)
cLocalColor := SETCOLOR (C_PIE2)
@ MAXROW()-1,9 SAY "Nombre"
@ MAXROW()-1,25 SAY "Impresora"
@ MAXROW()-1,44 SAY "Alinear Recibos"
@ MAXROW()-1,70 SAY "Cancelar"
SETCOLOR (cColor)
ENDIF
DISPEND ()
RETURN NIL

```

```

// Realiza la búsqueda y recorrido del archivo, en el sentido de la tecla de
// flecha presionada

```

```

PROCEDURE flechas (tipo_flecha)
IF tipo_flecha == 1 // flecha izq.
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",58)
SETCOLOR (cColor)
IF BOF () // Se verifica que no sea el principio del archivo
TONE(450)
cColor2 := SETCOLOR ("W+/R")
@ 20,5 SAY "Es el primero"
SETCOLOR (cColor2)
INKEY (1)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",13)
SETCOLOR (cColor)
ELSE
SKIP -1 // Se retrocede un registro
ENDIF
ELSEIF tipo_flecha == 2
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",58)
SETCOLOR (cColor)
IF !EOF () // Se verifica que no sea el fin de archivo
SKIP // Se avanza en registro
ENDIF
ENDIF
RETURN

```

```

// Abre la(s) base(s) de datos y comprueba que no haya un error al hacerlo
FUNCTION abre_bases (tipo_bases)
LOCAL cuadro, color
IF tipo_bases == 1
OPEN FILE escuelas TIME 30 SHARED INDEX escuela ALIAS escuela

```

```

ELSEIF tipo_bases == 2
  OPEN FILE escuelas TIME 30 EXCLUSIVE INDEX escuela ALIAS escuela
ELSEIF tipo_bases == 3
  OPEN FILE materias TIME 30 SHARED INDEX materia1, materia2 ALIAS materias
ELSEIF tipo_bases == 4
  OPEN FILE materias TIME 30 EXCLUSIVE ALIAS materias
ELSEIF tipo_bases == 5
  OPEN FILE maestros TIME 30 SHARED INDEX maestro1, maestro2 ALIAS maestros
ELSEIF tipo_bases == 6
  OPEN FILE maestros TIME 30 EXCLUSIVE ALIAS maestros
ELSEIF tipo_bases == 7
  OPEN FILE alumnos TIME 30 SHARED INDEX alumno1, alumno2, alumno3, alumno4 ALIAS alumnos
ELSEIF tipo_bases == 8
  OPEN FILE alumnos TIME 30 EXCLUSIVE ALIAS alumnos
ELSEIF tipo_bases == 9
  OPEN FILE alumnos TIME 30 SHARED ALIAS alumnos
ELSEIF tipo_bases == 10
  OPEN FILE grupos TIME 30 SHARED INDEX grupo1, grupo2, grupo3, grupo4 ALIAS grupos
ELSEIF tipo_bases == 11
  OPEN FILE grupos TIME 30 EXCLUSIVE ALIAS grupos
ELSEIF tipo_bases == 12
  OPEN FILE grupos TIME 30 SHARED ALIAS grupos
ELSEIF tipo_bases == 13
  OPEN FILE alumnos TIME 30 EXCLUSIVE INDEX alumno1, alumno2, alumno3, alumno4 ALIAS alumnos
ELSEIF tipo_bases == 14
  OPEN FILE asigna TIME 30 EXCLUSIVE INDEX asigna1, asigna2 ALIAS asigna
ELSEIF tipo_bases == 15
  OPEN FILE config TIME 30 EXCLUSIVE INDEX config ALIAS config
ELSEIF tipo_bases == 15.5
  OPEN FILE config TIME 30 EXCLUSIVE ALIAS config
ELSEIF tipo_bases == 18
  OPEN FILE config TIME 30 SHARED INDEX config ALIAS config
ELSEIF tipo_bases == 16
  OPEN FILE asigna TIME 30 EXCLUSIVE ALIAS asigna
ELSEIF tipo_bases == 17
  OPEN FILE asigna TIME 30 SHARED INDEX asigna1, asigna2 ALIAS asigna
ELSEIF tipo_bases == 19
  OPEN FILE impres TIME 30 SHARED INDEX impres1, impres2 ALIAS impres
ELSEIF tipo_bases == 19.5
  OPEN FILE impres TIME 30 EXCLUSIVE ALIAS impres
ELSEIF tipo_bases == 20
  OPEN FILE escapes TIME 30 SHARED INDEX escape1 ALIAS escapes
ELSEIF tipo_bases == 20.5
  OPEN FILE escapes TIME 30 EXCLUSIVE ALIAS escapes
ELSEIF tipo_bases == 21
  OPEN FILE inicio TIME 30 SHARED ALIAS inicio
ELSEIF tipo_bases == 22
  OPEN FILE inicio TIME 30 EXCLUSIVE ALIAS inicio
ELSEIF tipo_bases == 23
  OPEN FILE edit TIME 30 SHARED ALIAS edit
ELSEIF tipo_bases == 24
  OPEN FILE kardex TIME 30 EXCLUSIVE ALIAS kardex
ELSEIF tipo_bases == 25
  OPEN FILE kardex TIME 30 SHARED INDEX kardex1, kardex2, kardex3, kardex4 ALIAS kardex
ELSEIF tipo_bases == 26
  OPEN FILE concepto TIME 30 SHARED INDEX concepto ALIAS concepto
ELSEIF tipo_bases == 27
  OPEN FILE mes TIME 30 SHARED ALIAS mes
ELSEIF tipo_bases == 28
  OPEN FILE pago TIME 30 SHARED INDEX pago ALIAS pago
ELSEIF tipo_bases == 28.5
  OPEN FILE pago TIME 30 EXCLUSIVE ALIAS pago
ELSEIF tipo_bases == 29
  OPEN FILE edocuent TIME 30 SHARED INDEX edocuent ALIAS edocuent
ELSEIF tipo_bases == 30
  OPEN FILE edocuent TIME 30 EXCLUSIVE ALIAS edocuent
ELSE
  TONE (900)
  TONE (500)
  TONE (750)
  ALERT ("!!!! ERROR !!!!! NO SE PUDO HABRIR UNA BASE DE DATOS, SUSPENDA TODO
  TRABAJO CON EL SISTEMA Y CONTACTE CON ALGUIEN ESPECIALIZADO")

```

```

TONE (900)
TONE (500)
TONE (750)
cuadro := SAVESCREEN (9,18,16,63)
DISPBOX (9,18,15,62," | | - | ",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")
@ 11,23 SAY " EL CODIGO DE LA BASE ES"
SETCOLOR (color)
color := SETCOLOR ("GR+*/N")
@ 12,23 SAY tipo_bases
SETCOLOR (color)
INKEY (0)
RETURN .F.
ENDIF

IF NETERR ()
TONE (800)
ALERT ("ARCHIVO OCUPADO, ESPERE UN MOMENTO Y VUELVA A INTENTAR")
RETURN .F.
ELSE
RETURN .T.
ENDIF
RETURN NIL

/* Se capturan los datos de las distintas escuelas que van a existir en el
sistema, se crean sus bases de datos especificas en su directorio especifico */
PROCEDURE alta_esc()
LOCAL pantalla, cColor
LOCAL ok_bases, ok_esc := .T.

pie(2)
SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
limpiamenus(1) // Apaga el menf
pantallacampos (1) // Despliega los campos a capturar
IF !(ALIAS () == "ESCUELA")
ok_bases := abre_bases (1) // Base de escuelas
IF lok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS")
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR(cColor)
RETURN
ENDIF
ENDIF
DO WHILE ok_esc
var_ini()
mTiposep2 := " "
ok_esc := escu_alt(.T.)
ENDDO
DBCLOSEAREA()
SET CURSOR OFF
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL

/* Permite dar de baja en el sistema a una escuela. El directorio y las bases
de datos no se perderan. La baja es logica */
FUNCTION baja_esc ()
LOCAL tipo := "N", pantalla, ok:= "N", flag := 0, cColor, cColor2, cColor3
LOCAL ok_bases
pantalla := SAVESCREEN (10,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

cColor3 := SETCOLOR()
limpiamenus (1) // Se apagan las opciones del menf principal
pantallacampos (1) // Se muestran todos los campos
ok_bases := abre_bases (1) // Base de escuelas
IF lok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS")

```

```

SETCOLOR (cColor3)
RETURN
ENDIF
SET CURSOR ON
/* Este ciclo termina unicamente cuando se pulsa la tecla ESC o la tecla
  ENTER (sin ningun valor), esto hace que termine el proceso de captura */
DO WHILE .T.
  cColor := SETCOLOR ("N/N")
  @ MAXROW()-1,1 SAY SPACE (78)
  SETCOLOR (cColor)
  GOTO TOP
DO WHILE !FOUND()
  var_ini()
  @ 12,72 SAY mTipo // y se limpian los valores anteriores
  @ 14,19 SAY mNomesc
  @ 16,19 SAY mClave
  @ 18,24 SAY SPACE(1)

  @ 14,19 GET mNomesc PICTURE "@!"
  READ
  tecla := LASTKEY ()

  // ***** Si la ultima tecla pulsada fue ESC o un valor vacio, termina el proceso de captura
  IF tecla == K_ESC .OR. EMPTY(mNomesc)
    DBCLOSEAREA()
    RESTSCREEN (10,1,MAXROW()-3,MAXCOL()-1,pantalla)
    SETCOLOR (cColor3)
    @ MAXROW()-1,1 SAY SPACE (78)
    RETURN
  ENDIF

  /* ***** Se busca el nombre (o parte de el) en la base de datos.
  Si existe se posiciona el apuntador en el registro donde se encuentre */
  SET FILTER TO ALLTRIM(mNomesc) $UPPER(escuela->nomesc)
  GOTO TOP
  LOCATE FOR ALLTRIM(mNomesc) $UPPER(escuela->nomesc)

  /* Si no se encuentra el nombre en la base de datos se piden los
  datos necesarios para darlo de alta */
  IF .NOT. FOUND ()
    TONE (350)
    cColor := SETCOLOR (C_BORRADO)
    @ 14,19 SAY REPLICATE ("█",55)
    SETCOLOR (cColor)
    @ 14,19 SAY "NO EXISTE"
    INKEY (1)
    @ 14,19 SAY " "
  ENDIF
ENDDO

// Si se encuentra el nombre (o parte de el) en la base ...
// se leen los campos de la base de datos

DO WHILE (FOUND () .OR. LASTKEY() <> K_ESC)
  pic(3)
  IF EOF () // Se verifica que no sea el fin de archivo
    TONE(450)
    cColor2 := SETCOLOR ("W+/R")
    @ 20,5 SAY "Es el ultimo"
    SETCOLOR (cColor2)
    INKEY (1)
    cColor := SETCOLOR (C_BORRADO)
    @ 20, 5 SAY REPLICATE ("█",13)
    SETCOLOR (cColor)
    DBSKIP(-1)
  ELSE
    mTiposep2 := IIF ( escuela->Tiposep, "M","E")
    cColor := SETCOLOR ("GR+/G")
    @ 18,24 SAY mTiposep2 PICTURE "!A"
    SETCOLOR (cColor)
    @ 12,72 GET escuela->tipo PICTURE "!A"
    @ 14,19 GET escuela->nomesc PICTURE "@!"
  
```

```

@ 16,19 GET escuela->clave PICTURE "@!"
CLEAR GETS
ENDIF
INKEY (0)
DO CASE
CASE LASTKEY () == K_LEFT // Si la tecla pulsada fue <-
flechas (1)
CASE LASTKEY () == K_RIGHT // Si la tecla pulsada fue ->
flechas (2)
CASE LASTKEY () == K_ENTER // Se a seleccionado una escuela
pie(1)
@@ 20,5 SAY SPACE(57)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro en DARLO DE BAJA : "
SETCOLOR (cColor2)
@ 20,36 GET ok PICTURE "!A" VALID (ok$"SN")
READ
IF ok == "N".OR. LASTKEY () == K_ESC
TONE (550)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "El Registro NO a sido dado de baja"
SETCOLOR (cColor2)
inkey (2)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
EXIT
ELSEIF ok == "S"
DBSEEK (escuela->nomesc)
IF FOUND()
ok:="N"
IF lockr (30)
DELETE
DBUNLOCK()
TONE (550)
cColor2 := SETCOLOR ("W+/R")
@ 20,5 SAY "La escuela a sido dada de BAJA"
SETCOLOR (cColor2)
inkey (1)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
GOTO TOP
EXIT
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE
MOMENTO")
EXIT
ENDIF
ELSE
ALERT (" LA ESCUELA ACABA DE SER DADA DE BAJA EN ALGUNA OTRA ESTACION DE LA RED ")
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
ENDIF
ENDIF

CASE LASTKEY () == K_ESC // Se a cancelado la operacion
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
EXIT
ENDCASE
ENDDO
ENDDO
DBCLOSEAREA()
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)

```

```
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL
```

// Permite modificar los datos previamente capturados de las escuelas

```
FUNCTION modi_esc()
  LOCAL tipo := " ", pantalla, ok:="N", flag := 0, cColor2, cColor3,ok_bases
  pantalla := SAVESCREEN (10,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

  cColor3 := SETCOLOR()
  limpiamenus (1) // Se apagan las opciones del menL principal
  pantallacampos (1) // Se muestran todos los campos
  ok_bases := abre_bases (1) // Base de escuelas
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
    SETCOLOR ("W/N")
    CLS
    QUIT
  ENDIF
  SET CURSOR ON
  /* Este ciclo termina unicamente cuando se pulsa la tecla ESC o la tecla
  . ENTER (sin ningun valor), esto hace que termine el proceso de captura */
  DO WHILE .T.
    flag := 0
    cColor := SETCOLOR ("N/N")
    @ MAXROW()-1,1 SAY SPACE (78)
    SETCOLOR (cColor)
    GOTO TOP
    DO WHILE !FOUND()
      var_ini()
      @ 12,72 SAY mTipo // y se limpian los valores anteriores
      @ 14,19 SAY mNomesc
      @ 16,19 SAY mClave
      @ 18,24 SAY SPACE(1)

      @ 14,19 GET mNomesc PICTURE "@!"
      READ
      tecla := LASTKEY ()

      // ***** Si la ultima tecla pulsada fue ESC o un valor vacio, termina el proceso de captura
      IF tecla == K_ESC .OR. EMPTY(mNomesc)
        DBCLOSEAREA()
        RESTSCREEN (10,1,MAXROW()-3,MAXCOL()-1,pantalla)
        SETCOLOR (cColor3)
        @ MAXROW()-1,1 SAY SPACE (78)
        RETURN
      ENDIF

      /* ***** Se busca el nombre (o parte de el) en la base de datos.
      Si existe se posiciona el apuntador en el registro donde se encuentre */
      SET FILTER TO ALLTRIM(mNomesc) $UPPER(escuela->nomesc)
      GOTO TOP
      LOCATE FOR ALLTRIM(mNomesc) $UPPER(escuela->nomesc)

      /* Si no se encuentra el nombre en la base de datos se piden los
      datos necesarios para darlo de alta */
      IF .NOT. FOUND ()
        TONE (350)
        cColor := SETCOLOR (C_BORRADO)
        @ 14,19 SAY REPLICATE ("█",55)
        SETCOLOR (cColor)
        @ 14,19 SAY "NO EXISTE"
        INKEY (1)
        @ 14,19 SAY " " "
          ENDIF
        ENDDO

        // Si se encuentra el nombre (o parte de el) en la base ...
        // se leen los campos de la base de datos
```

```

DO WHILE (FOUND () .OR. LASTKEY() <> K_ESC)
  IF flag == 1
    exit
  ENDIF
  pie(3)
  IF EOF () // Se verifica que no sea el fin de archivo
    TONE(450)
    cColor2 := SETCOLOR ("W+R")
    @ 20,5 SAY "Es el ultimo"
    SETCOLOR (cColor2)
    INKEY (1)
    cColor := SETCOLOR (C_BORRADO)
    @ 20, 5 SAY REPLICATE ("■",13)
    SETCOLOR (cColor)
    DBSKIP(-1)
  ELSE
    mTiposep2 := IIF ( escuela->Tiposep, "M","E")
    cColor := SETCOLOR ("GR+/G")
    @ 18,24 SAY mTiposep2 PICTURE "!A"
    SETCOLOR (cColor)
    @ 12,72 GET escuela->tipo PICTURE "!A"
    @ 14,19 GET escuela->nomesc PICTURE "@!"
    @ 16,19 GET escuela->clave PICTURE "@!"
    CLEAR GETS
  ENDIF
  INKEY (0)
  DO CASE
    CASE LASTKEY () == K_LEFT // Si la tecla pulsada fue <-
      flechas (1)
    CASE LASTKEY () == K_RIGHT // Si la tecla pulsada fue ->
      flechas (2)
    CASE LASTKEY () == K_ENTER // Se a seleccionado una escuela
      flag := 1
      pie(1)
      igual_var_cam()
      ok_esc := escu_alt(.F.)
      IF ok_esc
        EXIT
      ELSE
        DBCLOSEAREA()
        SET CURSOR OFF
        RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
        SETCOLOR (cColor3)
        @ MAXROW()-1,1 SAY SPACE (78)
        RETURN
      ENDIF
    CASE LASTKEY () == K_ESC // Se a cancelado la operacion
      cColor := SETCOLOR (C_BORRADO)
      @ 20, 5 SAY REPLICATE ("■",55)
      SETCOLOR (cColor)
      DBCLEARFILTER ()
      EXIT
    ENDCASE
  ENDDO
  ENDDO
  DBCLOSEAREA()
  RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
  SETCOLOR (cColor3)
  @ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL

/* Verdadero == altas */
FUNCTION escu_alt(tipo)
  LOCAL ok_base:= 2, ok_almacen := 2, correcto := "N"
  LOCAL direct := SPACE(5), ok_dir := 1, arch, nTipo_esc,i,verifica
  LOCAL bBloque, cDirectorio
  correcto := "N"
  IF tipo
    @ 12,72 GET mTipo PICTURE "!A" VALID (mTipo$"LPGD")
  ENDIF

```



```

@ 14,19 GET mNomesc PICTURE "@!" VALID (!EMPTY(mNomesc))
@ 16,19 GET mClave PICTURE "@!" VALID (!EMPTY(mClave))
@ 18,24 GET mTiposep2 PICTURE "!A" VALID (mTiposep2$"ME")
READ
IF LASTKEY() == K_ESC // Cancelar operacion
RETURN F.
ENDIF

IF mTiposep2 == "M" // Si es verdadero indicara que se trata de SEP M,xico
mTiposep := .T. // Si es falso indicara que se trata de SEP Estado
ELSE
mTiposep := .F.
ENDIF

mNomesc := LTRIM(mNomesc)
IF LEN(mNomesc) < 55 // Se valida la long. de la variable
DO WHILE LEN(mNomesc) < 55 // y que no tenga espacios a la izq.
mNomesc += SPACE(1)
ENDDO
@ 14,19 GET mNomesc PICTURE "@!"
CLEAR GETS
ENDIF

mDirect := cDir_esc := SUBSTR(mNomEsc,1,5)
IF mDirect == "LICEN".OR. mDirect == "DIPLO".OR. mDirect == "MAEST".OR. mDirect == "PREPA"
TONE(600)
TONE(600)
ALERT(" !!! E R R O R !!! No se debe especificar nuevamente el Nivel de la escuela, especifique unicamente el
NOMBRE DE LA ESCUELA")
RETURN .T.
ENDIF

IF tipo
DBSEEK(mNomesc)
IF FOUND()
DO WHILE !EOF().AND. mNomesc == escuela->nomesc
IF escuela->tipo == mTipo
ALERT("ESCUELA YA EXISTENTE, NO PUEDE REALIZARSE ESTA OPERACION")
RETURN .T.
ENDIF
DBSKIP()
ENDDO
ELSE
/* En este bloque se checa que no exista el directorio, si este existe
se le reducira una letra y se volvera a buscar. Esta operacion se
realizara 3 veces. */
mDirect := cDir_esc := SUBSTR(mNomEsc,1,5)
LOCATE FOR ALLTRIM(escuela->direct) $UPPER(mDirect)
IF FOUND()
FOR i = 1 to 3
mDirect := cDir_esc := SUBSTR(mNomEsc,1,5-i)
GOTO TOP
LOCATE FOR ALLTRIM(escuela->direct) $UPPER(mDirect)
IF !FOUND()
i = 3
ENDIF
IF FOUND().AND. i == 3
TONE(650)
ALERT(" ESCUELA NO SE PUEDE DAR DE ALTA CONSULTA CON LOS DISEÑADORES DEL SISTEMA")
RETURN .T.
ENDIF
NEXT
ENDIF
ENDIF
ENDIF

DO WHILE .T.
@ 20,5 SAY "Son correctos los datos (S/N) : " GET correcto PICTURE "!A" VALID (correcto$"SN")
READ
SET CURSOR OFF
cColor := SETCOLOR(C_BORRADO)
@ 20, 5 SAY REPLICATE("Ú",35)

```

```

SETCOLOR (cColor)
IF LASTKEY () == K_ESC
RETURN .T.
ENDIF
IF correcto == "S"
IF tipo
IF mTipo == "L"
ntipo_esc := 1
ELSEIF mTipo == "P"
ntipo_esc := 2
ELSEIF mTipo == "G"
ntipo_esc := 3
ELSEIF mTipo == "D"
ntipo_esc := 4
ENDIF
cDirectorio := cDir_esc
IF !mkdir (cDirectorio)
TONE (1200)
ALERT ("NO SE HA PODIDO CREAR AL DIRECTORIO, NO SE HA DADO DE ALTA LA ESCUELA")
RETURN .F.
ENDIF
IF !chdir (cDirectorio)
TONE (1200)
ALERT ("NO SE HA PODIDO CAMBIAR AL DIRECTORIO CREADO, NO SE HA DADO DE ALTA LA ESCUELA")
RETURN .F.
ENDIF
crea_bases (nTipo_esc)
IF !chdir ("..")
TONE (1200)
ALERT ("NO SE HA PODIDO CAMBIAR AL DIRECTORIO ORIGINAL, ES NECESARIO INICIAR NUEVAMENTE EL SISTEMA
PARA EVITAR PERDIDA DE INFORMACION")
CLS
QUIT
ENDIF
SET CURSOR ON
ok_almacen := grabar(1)
IF !ok_almacen
TONE (1200)
ALERT ("HA OCURRIDO UN ERROR, NO SE HA DADO DE ALTA LA ESCUELA EN FORMA CORRECTA, PONGASE EN
CONTACTO CON LOS DESARROLLADORES DEL SISTEMA")
RETURN .F.
ELSE
RETURN .T.
ENDIF
ELSE
IF lockr(30)
ok_almacen := grabar(2)
IF ok_almacen
SET CURSOR ON
RETURN .T.
ELSE
RETURN .F.
ENDIF
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
SET CURSOR ON
RETURN .T.
ENDIF
ENDIF
ELSE
SET CURSOR ON
RETURN .T.
ENDIF
ENDDO
RETURN .F.

```

/* En el modulo de inscripciones esta funcion realiza la busqueda del alumno por su nombre, pulsando la tecla F1 */

```

FUNCTION busca_nombre(cTipo)
LOCAL cColor, cMatricula,y, mNomalun := SPACE(35)
SET KEY K_F1 TO

```

```

CLEAR GETS
pic(1)

*..... Esto es para posicionar en pantalla en el lugar correcto
*      dependiendo de si es SCC o SCE.
IF cTipo == "PAGO_COLEG"
  y := 19
ELSE
  y := 17
  cColor := SETCOLOR (C_BORRADO)
  @ 11, 17 SAY REPLICATE ("■",7)
  SETCOLOR (cColor)
ENDIF

@ 12, y GET mNomalum  PICTURE "@!"
READ
IF LASTKEY() == K_ESC    // Cancelar operacion
  RETURN
ENDIF
mNomalum := ALLTRIM(mNomalum)
existe_nombre := .T.
alumnos->(DBCLOSEAREA())
ok_bases := abre_bases (9) // Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  RETURN
ENDIF

LOCATE FOR mNomalum $UPPER(alumnos->Nomalum) WHILE mNomalum $UPPER(alumnos->Nomalum)
DO WHILE !EOF().OR. FOUND()
  IF FOUND()
    cMatricula := ALLTRIM(alumnos->Matriculacion)
    alumnos->(DBCLOSEAREA()) // Cierra la base de alumnos no indexada y la abre indexada
    ok_bases := abre_bases (7) // Base de alumnos
    IF !ok_bases
      TONE (750)
      ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
      RETURN
    ENDIF
    RETURN cMatricula
  ELSE
    CONTINUE
  ENDIF
ENDDO
alumnos->(DBCLOSEAREA()) // Cierra la base de alumnos no indexada y la abre indexada
ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  RETURN
ENDIF
RETURN SPACE(7)

```

SCE3.PRG

```

/* Programa      : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje      : CLIPPER 5.2
   Enlazador     : RTLINK 3.14B
*/

#xcommand REPEAT => while .t
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "caos.ch"    // Incluir las definiciones para los archivos temporales
#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla

/*
   Función para abrir ficheros en red con control de reintentos
   en función de un tiempo determinado.
*/
FUNCTION CsOpenDbf2( cFichero,lModo,nTiempo,lArea,cAlias,bBlock )
LOCAL lControl, cPantal, nVpos, cuadro, color
LOCAL lValor := FALSE // Fichero COMMON.CH

// Comprobación de par metros. Si no se pasa el nombre
// de fichero devuelve .F.

IF ValType( cFichero ) != "C"
RETURN lValor
END IF

// Si no se pasa el segundo par metro asume el modo de
// apertura que tenga _SET_EXCLUSIVE
IF ValType( lModo ) != "L"
lModo := Set( _SET_EXCLUSIVE )
END IF

// Si no se pasa el nombre de alias asume que no existe
IF ValType( cAlias ) != "C"
cAlias := ""
END IF

// Valores por defecto.
// Si se ignora el rea de trabajo asume que se realizar
// en la primera que se encuentre libre, y si se omite el
// tiempo lo reintentar un tiempo indefinido.

DEFAULT lArea TO TRUE // Fichero COMMON.CH
DEFAULT nTiempo TO 0

// Comprueba la extensión del fichero. Si se omite por
// defecto asume extensión .DBF

IF Right( cFichero, 4 ) != ".DBF"
cFichero += ".DBF"
END IF

// Comprueba la existencia del fichero y si no existe
// devuelve un valor falso

IF !File( cFichero )
RETURN lValor
END IF

// Para diferenciar si es tiempo indefinido
lControl := If( nTiempo == 0, TRUE, FALSE )

cuadro := SAVESCREEN (9,18,16,63) // Se guarda el area de pantalla donde se
// desplegaría en caso necesario el mensaje
// de acceso

```

```

// ( Punto 1 )
WHILE nTiempo > 0 .OR. !Control

DbUseArea( !Area,,cFichero,cAlias,lModo )

IF !NetErr() // Si no hay ningun tipo de error
  IValor := .T.
  EXIT
ELSE
  DISPBOX (9,18,15,62," |J-L| ",C_REYENO3) // Se dibuja la ventana de captura
  color := SETCOLOR ("R+N")
  @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
  SETCOLOR (color)
  color := SETCOLOR ("GR+*/N")
  @ 12,23 SAY "SE ESTA INTENTANDO ACCEDER A LA RED"
  SETCOLOR (color)
END IF

INKEY( 1 )

nTiempo--
END WHILE

RESTSCREEN (9,18,16,63,cuadro)
RETURN IValor

FUNCTION iniefranja(clavegen) //Determina el primer registro de la franja
  DBSEEK(clavegen)
RETURN NIL

FUNCTION finfranja(clavegen) //DETERMINA EL ULTIMO
  ** Liga SOFTSEEK (PARA CLAVE MAYOR DE 1 POSICION) y aumenta el valor de la
  **clave aumentando el valor de su último byte. Por ejemplo, si la última letra
  **fuera "a", se transformaría en "B"
  IF LEN(clavegen)=1
    DBSEEK(ultimaclave(clavegen), .F.)
    DBSKIP(-1)
    RETURN ( (| DBSEEK(ultimaclave(clavegen), .F.), DBSKIP(-1)) )
  ELSE
    DBSEEK(ultimaclave(clavegen), .T.)
    DBSKIP(-1)
    RETURN ( (| DBSEEK(ultimaclave(clavegen),.T.),DBSKIP(-1)) )
  ENDIF
RETURN

FUNCTION ultimaclave(clavegen)
  LOCAL ultimoreg
  ultimoreg := STUFF(clavegen, LEN(clavegen), 1, ;
  CHR( ASC(RIGHT(clavegen,1)) + 1 ))
RETURN (ultimoreg)

FUNCTION muevereg( nParamover, clavegen )
  LOCAL nMovidos
  nMovidos := 0
  IF nParamover == 0 .OR. LASTREC() == 0
    DBSKIP(0)
  ELSEIF nParamover > 0 .AND. RECNO() != LASTREC() + 1
    DO WHILE nMovidos <= nParamover .AND. ! EOF() .AND. &(INDEXKEY(0)) == clavegen
      //AVANZA REGISTROS
      DBSKIP(1)
      nMovidos++
    ENDDO
    DBSKIP(-1) //SIEMPRE DEVUELVE UN REGISTRO, PUES EL BUCLE
    nMovidos-- //WHILE TERMINA CUANDO ESTAMOS UN REGISTRO ADELANTE
  ELSEIF nParamover < 0 //FUERA DE LA FRANJA MOVER PARA TRAS (PARA INICIO)
    DO WHILE nMovidos >= nParamover .AND. ! BOF() .AND. &(INDEXKEY(0)) == clavegen
      DBSKIP(-1)
      nMovidos--
    ENDDO
    IF ! BOF() //SIEMPRE DEVUELVE UN REGISTRO, PUES EL BUCLE WHILE
      DBSKIP(1) //TERMINA CUANDO ESTAMOS FUERA DE LA FRANJA EN EL

```

```

    ENDIF          //CASO DE NO SER INICIO DE ARCHIVO
    nMovidos++
  ENDIF
RETURN (nMovidos)

// **** IGUALA VARIABLES CON CAMPOS"
PROCEDURE igual_var_cam()
  A := FCOUNT()
  FOR I = 1 TO A
    IF TYPE(FIELD(I))<>"M"
      NOMECAMPO=FIELD(I)
      NOMEVAR="M"+NOMECAMPO
      &NOMEVAR. = &NOMECAMPO
    ENDIF
  NEXT
RETURN

// ***** INICIALIZA VARIABLES CON EL MISMO TIPO Y TAM. DE CAMPO
FUNCTION var_ini()
  A := FCOUNT()
  FOR I = 1 TO A
    IF TYPE(FIELD(I))<>"M"
      NOMECAMPO=FIELD(I)
      NOMEVAR="M"+NOMECAMPO
      PUBLIC &NOMEVAR
      DO CASE
        CASE TYPE(FIELD(I))="C"
          &NOMEVAR. = SPACE(LEN(&NOMECAMPO))
        CASE TYPE(FIELD(I))="N"
          &NOMEVAR. = 0
        CASE TYPE(FIELD(I))="D"
          &NOMEVAR=CTOD("")
        CASE TYPE(FIELD(I))="L"
          &NOMEVAR. =.F.
      ENDCASE
    ENDIF
  NEXT
RETURN

// *****CREACION DE REPLACE PARA CADA CAMPO"
FUNCTION grabar (tipo_grabar)
  A := FCOUNT()
  IF tipo_grabar == 1
    APPEND BLANK
    IF !NETERR()
      FOR I=1 TO A
        IF TYPE(FIELD(I))<>"M"
          NOMECAMPO=FIELD(I)
          NOMEVAR="M"+NOMECAMPO
          REPLACE &NOMECAMPO. WITH &NOMEVAR.
        ENDIF
      NEXT
      COMMIT
      RETURN .T.
    ELSE
      TONE (550)
      ALERT ("NO SE PUEDE GUARDAR LA INFORMACION, ESPERE UN MOMENTO Y VUELVA A INTENTAR")
      RETURN .F.
    ENDIF
  ELSEIF tipo_grabar == 2
    IF !NETERR()
      FOR I=1 TO A
        IF TYPE(FIELD(I))<>"M"
          NOMECAMPO=FIELD(I)
          NOMEVAR="M"+NOMECAMPO
          REPLACE &NOMECAMPO. WITH &NOMEVAR.
        ENDIF
      NEXT
      COMMIT

```

```

RETURN .T.
ELSE
  TONE (550)
  ALERT ("NO SE PUEDE GUARDAR LA INFORMACION, ESPERE UN MOMENTO Y VUELVA A INTENTAR")
  RETURN .F.
ENDIF
ENDIF
RETURN

```

// Procedimiento que es invocado al presionar la tecla F1

PROCEDURE help (p,l,v)

LOCAL pantalla := SAVESCREEN (13,1,maxrow(),maxcol()-1)

SET CURSOR OFF

pie(4)

DO CASE

CASE p == "GRUPO_ALT"

DO CASE

CASE v == "MCLAVMAT"

ayuda_mat()

CASE v == "MCLAVMAE"

maes_ayuda ()

ENDCASE

CASE p == "CONCEN_MAT"

DO CASE

CASE v == "MCLAVMAT"

ayuda_mat()

ENDCASE

CASE p == "CALIF_GRUP"

DO CASE

CASE v == "MCLAVMAT"

ayuda_mat()

ENDCASE

CASE p == "MATS_LISTA"

DO CASE

CASE v == "MCLAVMAT"

ayuda_mat()

ENDCASE

ENDCASE

RESTSCREEN (13,1,maxrow(),maxcol()-1,pantalla)

SET CURSOR ON

// Despliega todas las materias dadas de alta en la escuela que este en uso

FUNCTION ayuda_mat()

LOCAL reg := materias->(RECNO()), area := ALIAS()

DISPBOX (13,1,maxrow()-3,maxcol()-1,"ÜÄÏ *ÜÄÄ*Ü",C_REYENO) // Se dibuja la ventana de captura

SELECT materias

materias->(DBGOTOP())

// Objeto de las materias existentes

otb:= TBROWSEDB (14,2,maxrow()-4,maxcol()-2)

otb:ADDCOLUMN(TBCOLUMNNEW("Clave",{materias->clavmat}))

otb:ADDCOLUMN(TBCOLUMNNEW("Materia",{materias->nommat}))

otb:COLSEP:=" |"

otb:HEADSEP:="_"

otb:RIGHT()

SET CURSOR OFF

DO WHILE .T.

DO WHILE !otb:STABILIZE()

ENDDO

/* El reempinkey con estos parametros es exclusivo para los

Tbrowses, esto es para que se realice un refresco cada

10 segundos. Esto es sobre todo para cuando se trabaja

en ambiente de red. */

nKey := reempinkey (0, otb)

```

DO CASE
CASE nKey == K_ENTER
  mclavmat := materias->clavmat
  SET CURSOR ON
  SELECT &area
  RETURN
CASE nKey == K_ESC
  SET CURSOR ON
  materias->(DBGOTO (reg))
  SELECT &area
  RETURN
CASE nKey==K_UP           ;otb:UP()
CASE nKey==K_DOWN        ;otb:DOWN()
CASE nKey==K_LEFT        ;otb:LEFT()
CASE nKey==K_CTRL_LEFT   ;otb:PANLEFT()
CASE nKey==K_RIGHT       ;otb:RIGHT()
CASE nKey==K_CTRL_RIGHT  ;otb:PANRIGHT()
CASE nKey==K_PGUP        ;otb:PAGEUP()
CASE nKey==K_CTRL_PGUP   ;otb:GOTOP()
CASE nKey==K_PGDN        ;otb:PAGEDOWN()
CASE nKey==K_CTRL_PGDN   ;otb:GOBOTTOM()
CASE nKey==K_HOME        ;otb:HOME()
CASE nKey==K_CTRL_HOME   ;otb:PANHOME()
CASE nKey==K_END         ;otb:END()
CASE nKey==K_CTRL_END    ;otb:PANEND()
OTHERWISE
  TONE(125) ; TONE(300)
ENDCASE
ENDDO
RETURN NIL

// Despliega todos los maestros dados de alta en la escuela que este en uso
FUNCTION maes_ayuda ()
  LOCAL reg := maestros->(RECNO()), area := ALIAS()

  DISPBOX (13,1,maxrow()-3,maxcol()-1,"ÚÁ: ÚÄÄ'Ú",C_REYENO) // Se dibuja la ventana de captura
  SELECT maestros
  maestros->(DBGOTOP())

  // Objeto de las materias existentes
  otb:= TBROWSEDB (14,2,maxrow()-4,maxcol()-2)
  otb:ADDCOLUMN(TBCOLUMNNEW("Clave",{||maestros->clavmae}))
  otb:ADDCOLUMN(TBCOLUMNNEW("Maestros",{||maestros->nommae}))
  otb:COLSEP:="|"
  otb:HEADSEP:="--"
  otb:RIGHT()

  SET CURSOR OFF
  DO WHILE .T.
    DO WHILE !otb:STABILIZE()
    ENDDO

  /* El reempinkey con estos parametros es exclusivo para los
  Tbrowseres, esto es para que se realice un refresco cada
  10 segundos. Esto es sobre todo para cuando se trabaja
  en ambiente de red. */

  nKey := reempinkey (0, otb)

  DO CASE
  CASE nKey == K_ENTER
    mclavmae := maestros->clavmae
    SET CURSOR ON
    SELECT &area
    RETURN
    CASE nKey == K_ESC
      SET CURSOR ON
      maestros->(DBGOTO (reg))
      SELECT &area
      RETURN
    CASE nKey==K_UP           ;otb:UP()

```



```

CASE nKey==K_DOWN      ;otb:DOWN()
CASE nKey==K_LEFT      ;otb:LEFT()
CASE nKey==K_CTRL_LEFT ;otb:PANLEFT()
CASE nKey==K_CTRL_RIGHT ;otb:PANRIGHT()
CASE nKey==K_PGUP      ;otb:PAGEUP()
CASE nKey==K_CTRL_PGUP ;otb:GOTOP()
CASE nKey==K_PGDN      ;otb:PAGEDOWN()
CASE nKey==K_CTRL_PGDN ;otb:GOBTOTOM()
CASE nKey==K_HHOME     ;otb:HHOME()
CASE nKey==K_CTRL_HOME ;otb:PANHOME()
CASE nKey==K_END       ;otb:END()
CASE nKey==K_CTRL_END  ;otb:PANEND()
OTHERWISE
  TONE(125); TONE(300)

```

```

ENDCASE
ENDDO
RETURN NIL

```

// Función que edita las columnas de Tbrowse en los módulos de calificaciones

```

FUNCTION edita(oTab)
  LOCAL oColumna ,oGet

  DO WHILE .NOT. oTab:STABILIZE()
    ENDDO

    oColumna := oTab:getColumn( oTab:colPos )
    oGet := GetNew(ROW(), COL(), oColumna:block, oColumna:heading)
    SET CURSOR ON
    ReadModal({oGet})
    oTAB:REFRESHCURRENT()
    SET CURSOR OFF
  RETURN

```

// Bloquea y/o intenta bloquear el registro de trabajo

```

FUNCTION lockr(según, tipo)
  LOCAL ciclo, aNored := {"Seguir","Cancelar"},conti,cuadro

  IF VALTYPE(tipo) == "U"
    tipo := 0
  ENDIF

  IF tipo # 0 .AND. tipo # 1
    tipo := 0
  ENDIF

  IF RLOCK()
    RETURN .T.
  ENDIF
  cuadro := SAVESCREEN (9,18,15,62)
  DO WHILE .T.
    segundos := según
    ciclo :=(segundos=0)
    DO WHILE (ciclo .OR. segundos>0)
      DISPBOX (9,18,15,62," | | - | " ,C_REYENO3) // Se dibuja la ventana de captura
      color := SETCOLOR ("R+/N")
      @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
      SETCOLOR (color)
      color := SETCOLOR ("GR+*/N")
      @ 12,21 SAY "ESTOY TRATANDO DE TENER ACCESO A LA RED"
      SETCOLOR (color)

      INKEY (.5)
      segundos--.5
      IF RLOCK()
        RESTSCREEN (9,18,15,62,cuadro)
        RETURN .T.
      ENDIF
    ENDIF
  ENDIF

```

```

ENDDO
IF tipo == 0
  RESTSCREEN (9,18,15,62,cuadro)
  TONE (550)
  conti := ALERT ("NO SE LOGRO EL ACCESO A LA RED",aNored)
  IF conti == 2
    RETURN .F.
  ENDDIF
ENDIF
ENDDO
RETURN .F.

// Función para bloquear un registro con control de reintentos
// en función del tiempo y conservar los bloqueos de registros
// ya existentes (MULTIBLOQUEO)
FUNCTION CsDbLockRec(nTiempo, nRec, bBlock )
LOCAL IControl, cPantal
LOCAL IValor := FALSE // Fichero COMMON.CH
LOCAL ciclo, aNored := {"Seguir","Cancelar"},conti,cuadro

// Se comprueba si existe fichero .DBF activo antes de
// realizar el bloqueo del registro

IF Empty( Alias() )
  RETURN IValor
END IF

// Si se omite el par metro tiempo se asume 0 por defecto
DEFAULT nTiempo TO 0 // Fichero COMMON.CH

// Comprobar si se pasa registro a bloquear
DEFAULT nRec TO RecNo()

// Realizar la operación de bloqueo del registro y comprobar tiene o no ,xito
IF DbRLock( nRec )
  IValor := TRUE
ELSE
  // Para diferenciar si es tiempo indefinido
  IControl := IIf( nTiempo == 0 , TRUE , FALSE )

  cuadro := SAVESCREEN (9,18,15,62)

  DO WHILE .T.
    segundos := nTiempo
    // Bucle que reintentar el bloqueo
    DO WHILE segundos > 0 .OR. IControl
      DISPBOX (9,18,15,62," |J-L| ",C_REYENO3) // Se dibuja la ventana de captura
      color := SETCOLOR ("R+N")
      @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
      SETCOLOR (color)
      color := SETCOLOR ("GR+*N")
      @ 12,21 SAY "ESTOY TRATANDO DE TENER ACCESO A LA RED"
      SETCOLOR (color)

      InKey( 1 )

      segundos-- // Decrementamos el par metro de tiempo pasado

      // Si el bloqueo no tiene ,xito se reintentar nuevamente

    IF DbRLock( nRec )
      IValor := TRUE
      RESTSCREEN (9,18,15,62,cuadro)
      EXIT
    END IF
  ENDDO
  IF !IValor

```

```

RESTSCREEN (9,18,15,62,cuadro)
TONE (530)
conti := ALERT ("NO SE LOGRO EL ACCESO A LA RED",aNored)
IF conti == 2
    RETURN IValor
ENDIF
ELSE
    RETURN IValor
ENDIF
ENDDO
ENDIF
RETURN IValor

```

/* Función que encuentra un nombre unico para el archivo temporal.

Se crea la base de datos con el nombre unico de archivo obtenido. Esta base se crea con una estructura de acuerdo al tipo de escuela en el que se este trabajando y/o de acuerdo al proceso que la llame */

FUNCTION nomtemp(ruta,tipo)

LOCAL cTemp

cTemp := CsTmpName(ruta) // Esta es la función que obtiene el nombre del archivo

aAsigna := {}

IF tipo == 1

AADD(aAsigna,{"clavmat","C",5,0})

AADD(aAsigna,{"nommat","C",55,0})

AADD(aAsigna,{"area","C",25,0})

ENDIF

IF tipo == 1 .OR. tipo == 4

IF tipo == 4

AADD(aAsigna,{"clavmat","C",5,0})

ENDIF

DO CASE // Se crean los campos para contener las calificaciones de cada cada materia.

CASE cTipo == "L" // Licenciatura

AADD(aAsigna,{"P1","N",4,1})

AADD(aAsigna,{"I1","N",2,0})

AADD(aAsigna,{"P2","N",4,1})

AADD(aAsigna,{"I2","N",2,0})

AADD(aAsigna,{"F3","N",4,1})

AADD(aAsigna,{"I3","N",2,0})

CASE cTipo == "P" // Preparatoria

AADD(aAsigna,{"P1","N",4,1})

AADD(aAsigna,{"I1","N",2,0})

AADD(aAsigna,{"P2","N",4,1})

AADD(aAsigna,{"I2","N",2,0})

AADD(aAsigna,{"P3","N",4,1})

AADD(aAsigna,{"I3","N",2,0})

AADD(aAsigna,{"S","N",4,1})

AADD(aAsigna,{"IS","N",2,0})

AADD(aAsigna,{"P4","N",4,1})

AADD(aAsigna,{"I4","N",2,0})

AADD(aAsigna,{"P5","N",4,1})

AADD(aAsigna,{"I5","N",2,0})

AADD(aAsigna,{"P6","N",4,1})

AADD(aAsigna,{"I6","N",2,0})

AADD(aAsigna,{"F","N",4,1})

AADD(aAsigna,{"IF","N",2,0})

CASE cTipo == "G" // Doctorado

AADD(aAsigna,{"U","N",4,1})

AADD(aAsigna,{"I","N",2,0})

CASE cTipo == "D" // Diplomado

AADD(aAsigna,{"U","N",4,1})

AADD(aAsigna,{"I","N",2,0})

ENDCASE

ELSEIF tipo == 2

AADD(aAsigna,{"nommat","C",55,0})

AADD(aAsigna,{"parcial","N",4,1})

AADD(aAsigna,{"faltas","N",2,0})

ELSEIF tipo == 3

AADD(aAsigna,{"campo","N",2,0})

AADD(aAsigna,{"Matricula","C",7,0})

```

AADD(aAsigna,{ "nombre","C",35,0})
AADD(aAsigna,{ "parcial","N",4,1})
AADD(aAsigna,{ "faltas","N",2,0})
ENDIF
DBCREATE(cTemp,aAsigna)
RETURN cTemp

```

/* Función que obtiene y devuelve un nombre de archivo unico, para lograrlo hace uso de la función _FTmpName que esta hecha en "C" e incluida en la libreria CAOS */

```

FUNCTION CsTmpName(cPath)
LOCAL cRet := "
DEFAULT cPath TO CHR(CURDRIVE()+65)+ "\ + CURDIR()

cPath := ALLTRIM (cPath) + CHR(0) + SPACE(12)
IF _FTmpName (cPath) > 0
cRet := ALLTRIM (StrTran(cPath,CHR(0),"))
FERASE (cRet)
ENDIF
RETURN cRet

```

// Limpia la pantalla de los valores que hayan tenidos las variables.

```

FUNCTION limpia_vars(tipo_vars)
LOCAL cColor
cColor := SETCOLOR (C_BORRADO)
IF tipo_vars == 1
@ 12,16 SAY mClavmat
@ 14,16 SAY mNommat
@ 16,16 SAY mSeriemat
@ 16,37 SAY mArea
ELSEIF tipo_vars == 2
@ 11,17 SAY mClavmae
@ 11,40 SAY mNommae
@ 13,17 SAY mRfcmac+""
@ 13,48 SAY mNivmac
@ 13,74 SAY mSexmae
@ 15,17 SAY mDomumae
@ 15,70 SAY mCpmae
@ 17,17 SAY mCiudmae
@ 17,55 SAY mColmae
@ 19,17 SAY mTelmae
@ 19,67 SAY mTeingmae
ELSEIF tipo_vars == 3
@ 11, 42 SAY mSemesimalum
@ 11, 61 SAY mGrupo+""
@ 12, 17 SAY mNomalunum
@ 12, 61 SAY mSexalunum
@ 13, 17 SAY mDomalunum
@ 14, 17 SAY mColalunum
@ 14, 61 SAY mCpalunum
@ 15, 17 SAY mCiudalunum
@ 15, 61 SAY mTelalunum1
@ 16, 28 SAY mEscproalunum
@ 17, 28 SAY mNomtutalunum
@ 18, 28 SAY mTelalunum2
@ 19, 28 SAY mTeinsalunum
@ 19, 75 SAY ""
ELSEIF tipo_vars == 4
@ 11,13 SAY mGrupo
ENDIF
IF tipo_vars == 5
@ 11, 37 SAY mClavmat
@ 11, 65 SAY mClavmae
@ 16, 10 SAY mHora1+""
@ 16, 20 SAY mHora2+""
@ 16, 35 SAY mSalon1
@ 16, 50 SAY mEdifi1
@ 16, 60 SAY mDia1
@ 17, 10 SAY mHora3+""
@ 17, 20 SAY mHora4+""
@ 17, 35 SAY mSalon1

```

```
@ 17, 50 SAY mEdifi1
@ 17, 60 SAY mDia2
@ 18, 10 SAY mHora5+
@ 18, 20 SAY mHora6+
@ 18, 35 SAY mSalon1
@ 18, 50 SAY mEdifi1
@ 18, 60 SAY mDia3
@ 19, 10 SAY mHora7+
@ 19, 20 SAY mHora8+
@ 19, 35 SAY mSalon1
@ 19, 50 SAY mEdifi1
@ 19, 60 SAY mDia4
ENDIF
SETCOLOR (cColor)
RETURN
```

ALUMNOS.PRG

```

/* Programa      : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje      : CLIPPER 5.2
   Enlazador     : RTLINK 3.14B
*/

#xcommand REPEAT => do while t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla
#include "caos.ch"

/* Procedimiento que abre las bases de datos necesarias para poder dar de alta
   o reinscribir alumnos. A su vez llama a la función que realmente realiza
   la captura de la información */
PROCEDURE alta_alumnos()
LOCAL pantalla, cColor
LOCAL cColor2, ok:=.T., ok_bases
PUBLIC consec := -1, cYear

cColor2 := SETCOLOR()
SET PATH TO &cPath
IF !FILE("&cPath\alumno1.ntx")
  ok_bases := abre_bases(8) //Base de alumnos
  IF ok_bases
    INDEX ON Matrialum TO &cPath\alumno1
    INDEX ON Nomalum TO &cPath\alumno2
    INDEX ON Grupo TO &cPath\alumno3
    INDEX ON grupo+nomalum TO &cPath\alumno4
    DBCLOSEAREA()
  ELSE
    TONE(750)
    ALERT("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
    SETCOLOR("W/N")
    CLS
    QUIT
  ENDIF
ENDIF

SET CURSOR ON
pantalla := SAVESCREEN(10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menú
pantallacampos(4) // Despliega los campos a capturar

// En el siguiente bloque se calcula el numero consecutivo
cYear := STR(YEAR(DATE()))
cYear := SUBSTR(cYear,4,2)
ok_bases := abre_bases(9) // Base de alumnos
IF !ok_bases
  TONE(750)
  ALERT("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
  SETCOLOR("W/N")
  CLS
  QUIT
ENDIF
LOCATE ALL FOR ALLTRIM(cYear)+ALLTRIM(cClave) $UPPER(alumnos->Matrialum) WHILE ALLTRIM(cYear)+ALLTRIM(cClave)
$UPPER(alumnos->Matrialum)
DO WHILE !EOF().OR. FOUND()
  IF FOUND()
    consec ++
                                ENDIF
                                CONTINUE
  ENDDO
  DBCLOSEAREA()
  consec ++
  // Aqui termina el bloque para el calculo del consecutivo

```

```

ok_bases := abre_bases (7) //Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
  SETCOLOR ("W/N")
  CLS
  QUIT
ENDIF

DO WHILE ok
  SELECT alumnos
  pie(5)
  GOTO TOP
  var_ini()
  limpia_vars(3)
  ok := alum_alt(1,consec)
ENDDO
SET KEY K_F1 TO
DBCLOSEAREA()
SET CURSOR OFF
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor2)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL

/* Función que realiza la captura de la información de los alumnos que van
a ser reinscritos o dados de alta */
FUNCTION alum_alt(tipo)
LOCAL ok_bases,ok_almacen := 2, correcto := "S", flag := 0,cMatricula
PUBLIC existe_nombre := .F.

IF tipo == 1
  mMatrialum := ALLTRIM(cYear) + ALLTRIM(cClave) + PADL(LTRIM(STR(consec)),3,"0")
ENDIF

SET KEY K_F1 TO mMatrialum := busca_nombre()

@ 11,17 GET mMatrialum PICTURE "@K" VALID (!EMPTY(mMatrialum))
READ
IF LASTKEY() == K_ESC // Cancelar operacion
  RETURN .F.
ENDIF
SET KEY K_F1 TO
mMatrialum := LTRIM(mMatrialum)
IF LEN(mMatrialum) < 5 // Se valida la long. de la variable
  DO WHILE LEN (mMatrialum) < 5 // y que no tenga espacios a la izq.
    mMatrialum += SPACE(1)
  ENDDO
  @ 11,17 GET mMatrialum PICTURE "@!"
  CLEAR GETS
ENDIF
IF tipo == 1
  DBSEEK (mMatrialum)
  IF FOUND()
    flag := 1
    igual_var_cam()
    IF EMPTY(Alumnos->baja)
      @ 11, 61 GET mGrupo PICTURE "@!"
      @ 12, 17 GET mNomalum PICTURE "@!"
      @ 12, 61 GET mSexalum PICTURE "@!"
      @ 13, 17 GET mDomalum PICTURE "@!"
      @ 14, 17 GET mColalum PICTURE "@!"
      @ 14, 61 GET mCpalum PICTURE "@!"
      @ 15, 17 GET mCiudalum PICTURE "@!"
      @ 15, 61 GET mTelalum1 PICTURE "@!"
      @ 16, 28 GET mEscproalum PICTURE "@!"
      @ 17, 28 GET mNomtutalum PICTURE "@!"
      @ 18, 28 GET mTelalum2 PICTURE "@!"
      @ 19, 28 GET mFeinsalum PICTURE "@E"
      CLEAR GETS
    ELSE

```

```

TONE(550)
ALERT ("EL ALUMNO ESTA DADO DE BAJA TEMPORALMENTE")
RETURN .T.
ENDIF
ELSEIF !FOUND() .AND. existe_nombre == .T.
    existe_nombre := .F.
    TONE(550)
    @ 12,17 SAY "NO EXISTE"+SPACE(26)
    INKEY (1)
    cColor := SETCOLOR (C_BORRADO)
    @ 12,17 SAY REPLICATE ("█",35)
    SETCOLOR (cColor)
    RETURN .T.
ENDIF
ENDIF
pie (1)
IF flag <> 1
    mFeinsalum := DATE ()
    cColor := SETCOLOR ("GR+/G")
    @ 19, 28 SAY mFeinsalum PICTURE "@E"
    SETCOLOR (cColor)
ENDIF
@ 11, 42 GET mSemesalum PICTURE "@K@R@! X9" VALID (!EMPTY(mSemesalum))
@ 11, 61 GET mGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(mGrupo))
@ 12, 17 GET mNomalum PICTURE "@!" VALID (!EMPTY(mNomalum))
@ 12, 61 GET mSexalum PICTURE "@!" VALID (mSexalum$"FM ")
@ 13, 17 GET mDomalum PICTURE "@!" VALID (!EMPTY(mDomalum))
@ 14, 17 GET mColalum PICTURE "@!"
@ 14, 61 GET mCpalum PICTURE "@!"
@ 15, 17 GET mCiudalum PICTURE "@!"
@ 15, 61 GET mTelalum1 PICTURE "@!"
@ 16, 28 GET mEscproalum PICTURE "@!"
@ 17, 28 GET mNomtutalum PICTURE "@!"
@ 18, 28 GET mTelalum2 PICTURE "@!"
READ
IF LASTKEY() == K_ESC // Cancelar operacion
    cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
    @ MAXROW()-1,1 SAY SPACE (78)
    SETCOLOR (cColor)
    RETURN .T.
ENDIF
DO WHILE .T.
    @ 20,5 SAY "Son correctos los datos (S/N) : " GET correcto PICTURE "!A" VALID (correcto$"SN")
    READ
    cColor := SETCOLOR (C_BORRADO)
    @ 20,5 SAY SPACE (35)
    SETCOLOR (cColor)
    IF LASTKEY () == K_ESC
        RETURN .T.
    ENDIF
    IF correcto == "S"
        IF tipo == 1 .AND. flag == 0
            cMatricula := mMatrialum
            mInscrito := .T.
            ok_almacen := grabar(1)
            IF !ok_almacen
                RETURN .F.
            ENDIF
            consec ++
        ELSEIF tipo == 2 .OR. flag == 1
            flag := 0
            IF lockr(30)
                mInscrito := .T.
                ok_almacen := grabar(2)
                IF !ok_almacen
                    RETURN .F.
                ENDIF
            ELSE
                ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION
                EN ESTE MOMENTO")
                RETURN .T.
            ENDIF
        ENDIF
    ENDIF

```



```

ENDIF
RETURN .T.
ELSE
RETURN .T.
ENDIF
ENDDO
RETURN .F.

```

* Procedimiento que realiza la baja de alumnos

```

PROCEDURE baja_alum()
LOCAL pantalla, cColor, ok_almacen := 2, existe := 0
LOCAL cColor2, cColor3, ok := "N", ok_bases

cColor3 := SETCOLOR ()
SET PATH TO &cPath
IF !FILE("&cPath\alumno1.ntx")
ok_bases := abre_bases (8) // Base de alumnos
IF ok_bases
INDEX ON Matrialum TO &cPath\alumno1
INDEX ON Nomalum TO &cPath\alumno2
INDEX ON Grupo TO &cPath\alumno3
INDEX ON grupo+nomalum TO &cPath\alumno4
DBCLOSEAREA()
ELSE
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
SETCOLOR ("W/N")
CLS
QUIT
ENDIF
ENDIF

SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menL
pantallacampos (4) // Despliega los campos a capturar

ok_bases := abre_bases (7) //Base de alumnos
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
SETCOLOR ("W/N")
CLS
QUIT
ENDIF

DO WHILE .T.
alumnos->(DBGOTOP ())
pie(2)
DO WHILE !FOUND()
alumnos->(DBGOTOP ())
var_ini()
limpia_vars (3)

@ 11,17 GET mMatrialum PICTURE "@K" VALID (!EMPTY(mMatrialum))
READ

IF LASTKEY() == K_ESC // Cancelar operacion
DBCLOSEAREA()
SELECT 1
DBCLOSEAREA()
SET CURSOR OFF
RESTSCRFEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)

@ MAXROW()-1,1 SAY SPACE (78)
RETURN
ENDIF

mMatrialum := LTRIM(mMatrialum)
IF LEN(mMatrialum) < 5 // Se valida la long. de la variable

```

```

DO WHILE LEN (mMatrIalUm) < 35 // y que no tenga espacios a la izq.
mMatrIalUm += SPACE(1)
ENDDO

@ 11,17 GET mMatrIalUm PICTURE "@K"
CLEAR GETS
ENDIF
/* ***** Se busca el nombre (o parte de el) en la base de datos.
Si existe se posiciona el apuntador en el registro donde se encuentre */
SET FILTER TO ALLTRIM(mMatrIalUm) $UPPER(alumnos->MatrIalUm)
GOTO TOP
LOCATE FOR ALLTRIM(mMatrIalUm) $UPPER(alumnos->MatrIalUm)

/* Si no se encuentra el nombre en la base de datos se piden los
datos necesarios para darlo de alta */
IF .NOT. FOUND ()
TONE (350)
@ 11,17 SAY SPACE(5)
@ 11,17 SAY "NO EXISTE"
INKEY (1)
@ 11,17 SAY " "
ENDIF
ENDDO

// Si se encuentra el nombre (o parte de el) en la base ...
// se leen los campos de la base de datos
DO WHILE (FOUND () .OR. LASTKEY() <> K_ESC)
pie(3)
IF EOF () // Se verifica que no sea el fin de archivo
TONE(450)
cColor2 := SETCOLOR ("W+/R")
@ 20,5 SAY "Es el ultimo"
SETCOLOR (cColor2)
INKEY (1)
@ 20,5 SAY " "
DBSKIP(-1)
ELSE
IF EMPTY(Alumnos->baja)
@ 11, 17 GET alumnos->MatrIalUm PICTURE "@K"
@ 11, 42 GET alumnos->SemesalUm PICTURE "@K@R@! 99"
@ 11, 61 GET alumnos->Grupo PICTURE "@!"
@ 12, 17 GET alumnos->NomalUm PICTURE "@!"
@ 12, 61 GET alumnos->SexalUm PICTURE "@!"
@ 13, 17 GET alumnos->DomalUm PICTURE "@!"
@ 14, 17 GET alumnos->ColalUm PICTURE "@!"
@ 14, 61 GET alumnos->CpalUm PICTURE "@!"
@ 15, 17 GET alumnos->CiudalUm PICTURE "@!"
@ 15, 61 GET alumnos->TelalUm1 PICTURE "@!"
@ 16, 28 GET alumnos->EscproalUm PICTURE "@!"
@ 17, 28 GET alumnos->NomtutalUm PICTURE "@!"
@ 18, 28 GET alumnos->TelalUm2 PICTURE "@!"
@ 19, 28 GET alumnos->FeinsalUm PICTURE "@E"
CLEAR GETS
ELSE
TONE(550)
ALERT ("EL ALUMNO ESTA DADO DE BAJA TEMPORALMENTE")
EXIT
ENDIF
ENDIF
INKEY (0)
DO CASE
CASE LASTKEY () == K_LEFT // Si la tecla pulsada fue <-
flechas (1)
CASE LASTKEY () == K_RIGHT // Si la tecla pulsada fue ->
flechas (2)
CASE LASTKEY () == K_ENTER // Se a seleccionado una escuela
pie(1)
igual_var_cam()
@ 20,5 SAY SPACE(57)
@ 19,38 SAY "Tipo baja: (T)emporal (D)efinitiva "
@ 19,75 GET mBaja PICTURE "!A" VALID (mBaja$"TD")
READ

```

```

IF LASTKEY () == K_ESC
  TONE (550)
  cColor2 := SETCOLOR ("R/B+")
  @ 20,5 SAY "El Registro NO a sido dado de baja"
  SETCOLOR (cColor2)
  inkey (2)
  @ 20,5 SAY SPACE(55)
  alumnos->(DBCLEARFILTER ())
  cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
  @ MAXROW()-1,1 SAY SPACE (78)
  SETCOLOR (cColor)
  cColor := SETCOLOR (C_BORRADO)
  @ 19,38 SAY REPLICATE (" ",39)
  SETCOLOR (cColor)
  EXIT
ENDIF

IF mBaja == "T"
  cColor2 := SETCOLOR ("R/B+")
  @ 20,5 SAY "Esta seguro en DARLO DE BAJA :."
  SETCOLOR (cColor2)
  @ 20,36 GET ok PICTURE "IA" VALID (ok$"SN")
  READ
  IF ok == "N".OR. LASTKEY () == K_ESC
    TONE (550)
    cColor2 := SETCOLOR ("R/B+")
    @ 20,5 SAY "El Registro NO a sido dado de baja"
    SETCOLOR (cColor2)
    inkey (2)
    @ 20,5 SAY SPACE(55)
    alumnos->(DBCLEARFILTER ())
    cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
    @ MAXROW()-1,1 SAY SPACE (78)
    SETCOLOR (cColor)
    cColor := SETCOLOR (C_BORRADO)
    @ 19,38 SAY REPLICATE (" ",39)
    SETCOLOR (cColor)
    EXIT
  ELSEIF ok == "S"
    DBSEEK (mMatrIalumno)
    IF FOUND()
      ok:="N"
      IF lockr (30)
        ok_almacen := grabar(2)
        IF !ok_almacen
          DBCLOSEAREA()
          SELECT 1
          DBCLOSEAREA()
          SET CURSOR OFF
          RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
          SETCOLOR (cColor3)
          @ MAXROW()-1,1 SAY SPACE (78)
          RETURN .F.
        ENDIF
        DBUNLOCK ()
        TONE (550)
        cColor2 := SETCOLOR ("W+/R")
        @ 20,5 SAY "El alumno ha sido dado de BAJA"
        SETCOLOR (cColor2)
        inkey (1)
        @ 20,5 SAY SPACE(55)
        alumnos->(DBCLEARFILTER ())
        GOTO TOP
        cColor := SETCOLOR (C_BORRADO)
        @ 19,38 SAY REPLICATE (" ",39)
        SETCOLOR (cColor)
        EXIT
      ELSE
        ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN
ESTE MOMENTO")
        EXIT
      ENDIF
    ENDIF
  ENDIF

```

```

ENDIF
ELSE
ALERT (" EL ALUMNO ACABA DE SER DADO DE BAJA EN ALGUNA OTRA ESTACION DE LA RED ")
(@ 20,5 SAY SPACE(55)
alumnos->(DBCLEARFILTER ())
ENDIF
ENDIF
ELSE
cColor2 := SETCOLOR ("R/B+")
(@ 20,5 SAY "Esta seguro en DARLO DE BAJA :")
SETCOLOR (cColor2)
@ 20,36 GET ok PICTURE "!A" VALID (ok$"SN")
READ
IF ok == "N" .OR. LASTKEY () == K_ESC
TONE (550)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "El Registro NO a sido dado de baja"
SETCOLOR (cColor2)
inkey (2)
@ 20,5 SAY SPACE(55)
alumnos->(DBCLEARFILTER ())
cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
cColor := SETCOLOR (C_BORRADO)
@ 19,38 SAY REPLICATE ("█",39)
SETCOLOR (cColor)
EXIT
ELSE
ok := "N"
IF lockr (30)
DELETE
DBUNLOCK ()
SELECT 1
IF lockr (120)
DELETE
DBUNLOCK ()
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
EXIT
ENDIF
SELECT 2
TONE (550)
cColor2 := SETCOLOR ("W+/R")
@ 20,5 SAY "El alumno ha sido dado de BAJA"
SETCOLOR (cColor2)
inkey (1)
@ 20,5 SAY SPACE(55)
alumnos->(DBCLEARFILTER ())
alumnos->(DBGOTOP ())
cColor := SETCOLOR (C_BORRADO)
@ 19,38 SAY REPLICATE ("█",39)
SETCOLOR (cColor)
EXIT
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE
MOMENTO")
EXIT
ENDIF
ENDIF
ENDIF
CASE LASTKEY () == K_ESC // Se a cancelado la operacion
@ 20, 5 SAY SPACE (55)
alumnos->(DBCLEARFILTER ())
EXIT
ENDCASE

ENDDO
ENDDO
DBCLOSEAREA()
SELECT 1
DBCLOSEAREA()
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)

```

```
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL
```

* Procedimiento encargado de recuperar los alumnos dados de baja temporalmente

```
PROCEDURE Recup_alum()
LOCAL pantalla, cColor, ok_almacen := 2, correcto := "N", existe := 0
LOCAL cColor2, cColor3, ok := "N", ok_bases

cColor3 := SETCOLOR ()
SET PATH TO &cPath
IF !FILE ("%cPath\alumno1.ntx")
ok_bases := abre_bases (8) //Base de alumnos
IF ok_bases
INDEX ON Matrialum TO &cPath\alumno1
INDEX ON Nomalum TO &cPath\alumno2
INDEX ON Grupo TO &cPath\alumno3
INDEX ON grupo+nomalum TO &cPath\alumno4
DBCLOSEAREA()
ELSE
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
SETCOLOR ("W/N")
CLS
QUIT
ENDIF
ENDIF

SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menf
pantallacampos (4) // Despliega los campos a capturar
ok_bases := abre_bases (7) //Base de alumnos
IF lok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
SETCOLOR ("W/N")
CLS
QUIT
ENDIF

DO WHILE .T.
GOTO TOP
pie(2)
DO WHILE !FOUND()
GOTO TOP
correcto := "N"
var_ini()
limpia_vars (3)

@ 11,17 GET mMatrialum PICTURE "@K" VALID (!EMPTY(mMatrialum))
READ

IF LASTKEY() == K_ESC // Cancelar operacion
DBCLOSEAREA()
SET CURSOR OFF
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN
ENDIF

mMatrialum := LTRIM(mMatrialum)
IF LEN(mMatrialum) < 5 // Se valida la long. de la variable
DO WHILE: LEN (mMatrialum) < 35 // y que no tenga espacios a la izq.
mMatrialum += SPACE(1)
ENDDO

@ 11,17 GET mMatrialum PICTURE "@K"
CLEAR GETS
ENDIF
```

```

/* ***** Se busca el nombre (o parte de el) en la base de datos.
Si existe se posiciona el apuntador en el registro donde se encuentre */
SET FILTER TO ALLTRIM(mMatrialum) $UPPER(alumnos->Matrialum)
GOTO TOP
LOCATE FOR ALLTRIM(mMatrialum) $UPPER(alumnos->Matrialum)

/* Si no se encuentra el nombre en la base de datos se piden los
datos necesarios para darlo de alta */
IF .NOT. FOUND ()
  TONE (350)
  @ 11,17 SAY SPACE(5)
  @ 11,17 SAY "NO EXISTE"
  INKEY (1)
  @ 11,17 SAY " "
ENDIF
ENDDO

// Si se encuentra el nombre (o parte de el) en la base ...
// se leen los campos de la base de datos
DO WHILE (.FOUND () .OR. LASTKEY() <> K_ESC)
  pie (3)
  IF EOF () // Se verifica que no sea el fin de archivo
    TONE(450)
    cColor2 := SETCOLOR ("W+/R")
    @ 20,5 SAY "Es el ultimo"
    SETCOLOR (cColor2)
    INKEY (1)
    @ 20,5 SAY " "
    DBSKIP(-1)
  ELSE
    IF !EMPTY(Alumnos->baja)
      @ 11, 17 GET alumnos->Matrialum PICTURE "@K"
      @ 11, 42 GET alumnos->Semesalum PICTURE "@K@R@! 99"
      @ 11, 61 GET alumnos->Grupo PICTURE "@!"
      @ 12, 17 GET alumnos->Nomalum PICTURE "@!"
      @ 12, 61 GET alumnos->Sexalum PICTURE "@!"
      @ 13, 17 GET alumnos->Domalum PICTURE "@!"
      @ 14, 17 GET alumnos->Colalum PICTURE "@!"
      @ 14, 61 GET alumnos->Cpalum PICTURE "@!"
      @ 15, 17 GET alumnos->Ciudalum PICTURE "@!"
      @ 15, 61 GET alumnos->Telalum1 PICTURE "@!"
      @ 16, 28 GET alumnos->Escproalum PICTURE "@!"
      @ 17, 28 GET alumnos->Nomtutalum PICTURE "@!"
      @ 18, 28 GET alumnos->Telalum2 PICTURE "@!"
      @ 19, 28 GET alumnos->Feinsalum PICTURE "@E"
      CLEAR GETS
    ELSE
      TONE(550)
      ALERT ("EL ALUMNO NO ESTA DADO DE BAJA")
      EXIT
    ENDIF
  ENDIF
  INKEY (0)
  DO CASE
    CASE LASTKEY () == K_LEFT // Si la tecla pulsada fue <-
      flechas (1)
    CASE LASTKEY () == K_RIGHT // Si la tecla pulsada fue ->
      flechas (2)
    CASE LASTKEY () == K_ENTER // Se a seleccionado una escuela
      pie(1)
      igual_var_cam()
      @ 20,5 SAY SPACE(57)
      cColor2 := SETCOLOR ("R/B+")
      @ 20,5 SAY "Esta seguro en RECUPERARLO : "
      SETCOLOR (cColor2)
      @ 20,36 GET ok PICTURE "!A" VALID (ok$"SN")
      READ
      cColor := SETCOLOR (C_BORRADO)
      @ 20,5 SAY REPLICATE ("█",35)
      SETCOLOR (cColor)

      IF ok == "N" .OR. LASTKEY () == K_ESC

```

```

TONE (550)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "El Registro NO a sido Recuperado"
SETCOLOR (cColor2)
inkey (2)
@ 20,5 SAY SPACE(55)
DBCLEARFILTER ()
cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
EXIT
ELSEIF ok == "S"
DBSEEK (mMatrerialum)
IF FOUND()
ok:="N"
IF lockr (30)
mBaja := SPACE(1)
ok_almacen := grabar(2)
IF !ok_almacen
DBCLOSEAREA()
SET CURSOR OFF
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN .F.
ENDIF
DBUNLOCK ()
TONE (550)
cColor2 := SETCOLOR ("W+/R")
@ 20,5 SAY "El alumno ha sido Recuperado"
SETCOLOR (cColor2)
inkey (1)
@ 20,5 SAY SPACE(55)
DBCLEARFILTER ()
GOTO TOP
cColor := SETCOLOR (C_BORRADO)
@ 19,38 SAY REPLICATE ("█",39)
SETCOLOR (cColor)
EXIT
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
EXIT
ENDIF
ELSE
ALERT (" EL ALUMNO ACABA DE SER RECUPERADO EN ALGUNA OTRA ESTACION DE LA RED ")
@ 20,5 SAY SPACE(55)
DBCLEARFILTER ()
ENDIF
ENDIF

CASE LASTKEY () == K_ESC // Se a cancelado la operacion
@ 20, 5 SAY SPACE (55)
DBCLEARFILTER ()
EXIT
ENDCASE
ENDDO
ENDDO
DBCLOSEAREA()
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL

```

// ***** FUNCIONES PARA LA CAPTURA DE CALIFICACIONES *****

/* Función que se encarga de capturar las CALIFICACIONES de todas las materias asignadas a un ALUMNO en particular. Se hace uso de un archivo temporal
También sirve para consultar en pantalla las calificaciones por alumno para lograr esto basta con hacer .F. el parametro consulta */

```

FUNCTION cal_alumno (consulta)
  LOCAL ok_bases, cColor2 := SETCOLOR (), pantalla, pantalla2, aux2
  LOCAL cuadro, cColor3, flag3 := .F., temp, lModifico := .F.
  LOCAL cGrupo := SPACE(3), cAlum := SPACE(7), cNomAlum, flag4 := .F.
  PUBLIC cMaxEvals := 0, aux

  ok_bases := abre_bases (10) // Base de grupos
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
    SETCOLOR ("W/N")
    CLS
    QUIT
  ENDIF

  ok_bases := abre_bases (7) // Base de alumnos
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
    SETCOLOR ("W/N")
    CLS
    QUIT
  ENDIF

  ok_bases := abre_bases (3) // Base de materias
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
    SETCOLOR ("W/N")
    CLS
    QUIT
  ENDIF

  // Dependiendo si es licenciatura, preparatoria, o postgrado, es el nLmero
  // Maximo de evaluaciones
  DO CASE
    CASE cTipo == "L" // Licenciatura
      cMaxEvals := 3
    CASE cTipo == "P" // Preparatoria
      cMaxEvals := 8
    CASE cTipo == "G" // Doctorado
      cMaxEvals := 1
    CASE cTipo == "D" // Diplomado
      cMaxEvals := 1
  ENDCASE

  pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

  limpiamenus (2)
  @ 9, 2 SAY REPLICATE ("-",70)
  DO WHILE .T.
    pie (2)
    DISPBOX (10,1,MAXROW()-3,maxcol()-1,"┌──┴──┐",C_REYENO) // Se dibuja la ventana de captura
    pantallacampos (11)
    DO WHILE .T.
      cGrupo := SPACE(3)
      SET CURSOR ON
      @ 11, 20 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
      READ
      SET CURSOR OFF
      IF LASTKEY () == K_ESC
        RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
        SETCOLOR (cColor2)
        @ MAXROW()-1,1 SAY SPACE (78)
        alumnos->(DBUNLOCK())
        DBCLOSEALL()
        RETURN -1
      ELSE
        grupos->(DBGOTOP ())
        grupos->(DBSEEK (cGrupo))
        IF grupos->(FOUND())

```



```

alumnos->(DBSETORDER(3))
alumnos->(DBSEEK (cGrupo))
IF alumnos->(FOUND())
  EXIT
ELSE
  TONE (800)
  ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
ENDIF
ELSE
  TONE (800)
  ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
ENDIF
ENDIF
ENDDO

pantalla2 := SAVESCREEN (7,1,MAXROW(),MAXCOL()-1) // Se resguarda el area de pantalla de captura
// Se busca el alumno especificado
DO WHILE .T.
  // Se pide la clave del alumno al que se van a asignar las materias
  cAlum := SPACE(7)
  SET CURSOR ON
  @ 13, 20 GET cAlum PICTURE "@!" VALID (!EMPTY(cAlum))
  READ
  SET CURSOR OFF
  IF LASTKEY () == K_ESC
    EXIT
  ELSE
    alumnos->(DBSETORDER(3))
    alumnos->(DBSEEK (cGrupo))
    REPEAT
      IF alumnos->matrerialum == cAlum
        cNomAlum := alumnos->(NomAlum)
        flag4 := .T.
        EXIT
      ENDIF
    alumnos->(DBSKIP())
  UNTIL alumnos->(EOF()) .OR. alumnos->grupo # cGrupo

  IF !flag4 .OR. !alumnos->inscrito .OR. !EMPTY(alumnos->baja)
    TONE (750)
    ALERT ("El alumno no Existe, no pertenece a este Grupo, no ha sido Reinscrito, o ha sido dado de Baja Temporal")
  LOOP
  ENDIF
  IF EMPTY(alumnos->mat1)
    TONE (750)
    ALERT ("El alumno aun no tiene asignadas materias")
  LOOP
  ENDIF
  ENDIF
  flag4 := .F.

  // Se pide el número de evaluación
  DO WHILE .T.
    cNumEval := SPACE(1)
    SET CURSOR ON
    @ 15, 20 GET cNumEval PICTURE "9"
    READ
    aux := cNumEval
    cNumEval := VAL(cNumEval) // Se pasa de caracter a número
    SET CURSOR OFF
    IF LASTKEY () == K_ESC
      flag := .T.
      cColor := SETCOLOR (C_BORRADO)
      @ 15,20 SAY REPLICATE ("█",2)
      SETCOLOR (cColor)
      flag3 := .T.
    ELSE
      IF cNumEval > cMaxEvals
        TONE (800)
        ALERT ("No existe esta evaluación en esta escuela")
      LOOP
    ENDIF
  ENDIF

```

```

ENDIF
EXIT
ENDIF
ENDDO
IF flag3 == .T.
  flag3 := .F.
  LOOP
ENDIF

IF alumnos->(lockr (30)) // Si se logra bloquear el registro

// Se crea el archivo temporal en el que se va a trabajar
arch := nontemp(cPath,2)
OPEN FILE AUX &arch TIME 30 EXCLUSIVE ALIAS auxiliar

FOR i = 1 TO 13
  temp1 := "mat" + ALLTRIM(STR (i))
  temp2 := alumnos->(&temp1.)
  IF !EMPTY(temp2) //En caso de que ya no haya mas materias, se sale del ciclo
    auxiliar->(DBAPPEND())
    REPLACE auxiliar->nommat WITH regresa_materias(temp2,1,0)
    DO CASE
      // Los siguientes dos casos son para licenciaturas
      CASE cTipo == "L" .AND. cNumEval <= 2
        temp1 := "P" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
        temp2 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
      CASE cTipo == "L" .AND. cNumEval == 3
        temp1 := "F" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
        temp2 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))

      // Los siguientes tres casos son para PREPARATORIAS
      CASE cTipo == "P" .AND. cNumEval <= 3
        temp1 := "P" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
        temp2 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
      CASE cTipo == "P" .AND. cNumEval >= 5 .AND. cNumEval <= 7
        temp1 := "P" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval-1)))
        temp2 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval-1)))
      CASE cTipo == "P" .AND. cNumEval == 4
        temp1 := "S" + ALLTRIM(STR (i))
        temp2 := "IS" + ALLTRIM(STR (i))
      CASE cTipo == "P" .AND. cNumEval == 8
        temp1 := "F" + ALLTRIM(STR (i))
        temp2 := "IF" + ALLTRIM(STR (i))

      // El siguiente caso es para postgrados
      CASE cTipo == "G" .OR. cTipo == "D" .AND. cNumEval == 1
        temp1 := "U" + ALLTRIM(STR (i))
        temp2 := "I" + ALLTRIM(STR (i))

    ENDCASE

    REPLACE auxiliar->parcial WITH alumnos->(&temp1.)
    REPLACE auxiliar->faltas WITH alumnos->(&temp2.)
  ELSE
    EXIT
  ENDIF
NEXT
auxiliar->(DBCOMMIT())
ELSE
  ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
  LOOP
ENDIF

IF consulta
  pie (10)
ELSE
  pie (12)
ENDIF
cColor3 := SETCOLOR ("N/N")
@ 7, 1 SAY SPACE (78)
SETCOLOR (cColor3)
nCentra := (78-LEN (ALLTRIM(cNomAlum)))/2

```

```

cColor3 := SETCOLOR (C_TITULOS)
@ 7, nCentra SAY ALLTRIM(cNomAlum)
SETCOLOR (cColor3)
DISPBOX (8,1,maxrow()-3,maxcol()-1," |J-L|",C_REYENO) // Se dibuja la ventana de captura

SELECT auxiliar
GOTO TOP

otb:=TBROWSEDB (9,2,maxrow()-4,maxcol()-2)
otb:ADDCOLUMN(TBCOLUMNNEW("Materia",FIELDWBLOCK("nommat",SELE())))
otb:ADDCOLUMN(TBCOLUMNNEW("Calificaci3n",FIELDWBLOCK("parcial",SELE())))
/* En el concentrado de calificaciones de licenciatura y postgrados se
necesitan las asistencias y para las boletas de preparatoria se
necesitan las inasistencias. */
IF cTipo == "P"
otb:ADDCOLUMN(TBCOLUMNNEW("Inasistencias",FIELDWBLOCK("faltas",SELE())))
ELSE
otb:ADDCOLUMN(TBCOLUMNNEW("Asistencias",FIELDWBLOCK("faltas",SELE())))
ENDIF
columna=otb:GETCOLUMN(2)
columna:DEFcolor:={6,5}
columna=otb:GETCOLUMN(3)
columna:DEFcolor:={6,5}

otb:COLSEP:=" |"
otb:HEADSEP:="-"

otb:FREEZE:=1
DO WHILE .T.

DO WHILE NEXTKEY()=0 .AND. !otb:STABILIZE()
SET ESCAPE OFF
IF !alumnos->(CsDbLockRec(20))
ALERT ("Algun(os) registro(s), estan siendo utilizados por otra(s) estacion(es)")
alumnos->(DBUNLOCK())
DBCLOSEALL()
RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
RETURN
ELSE
SET ESCAPE ON
ENDIF
ENDDO

/* El reempinkey con estos parametros es exclusivo para los
Throwses, esto es para que se realice un refresco cada
10 segundos. Esto es sobre todo para cuando se trabaja
en ambiente de red. */

nKey := reempinkey (0, otb)

SET CURSOR OFF

DO CASE

CASE nKey == K_ENTER // Se a seleccionado una calificaci3n
IF otb:FREEZE = 1 .AND. otb:COLPOS=1
otb:RIGHT()
ELSEIF consulta
IModifico := .T.
EDITA(otb)
ENDIF

CASE nKey == K_ESC // Se a cancelado la operacion
IF consulta
IF IModifico
IModifico := .F.

SET ESCAPE OFF
cuadro := SAVESCREEN (9,18,16,63)
DISPBOX (9,18,15,62," |J-L|",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")
@ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)

```

```

color := SETCOLOR ("OR+*N")
@ 12,23 SAY "SE ESTA ACTUALIZANDO LA INFORMACION"
SETCOLOR (color)

alumnos->(DBSETORDER(1))
alumnos->(DBGOTOP ())
alumnos->(DBSEEK (cAlum))
IF alumnos->(FOUND()) // Si todavia existe el registro del alumno
  auxiliar->(DBGOTOP())
  FOR i = 1 TO 13
    IF !EMPTY(auxiliar->(parcial)) .OR. auxiliar->(IEOF()) //En caso de que ya no haya mas materias, se sale del ciclo

    DO CASE // Se construyen los nombres de los campos adecuados en la base de alumnos
      // Los siguientes dos casos son para licenciaturas
      CASE cTipo == "L" .AND. cNumEval <= 2
        temp1 := "P" + ALLTRIM(STR(i)) + aux
        temp2 := "I" + ALLTRIM(STR(i)) + aux
      CASE cTipo == "L" .AND. cNumEval == 3
        temp1 := "F" + ALLTRIM(STR(i)) + aux
        temp2 := "I" + ALLTRIM(STR(i)) + aux

      // Los siguientes tres casos son para PREPARATORIAS
      CASE cTipo == "P" .AND. cNumEval <= 3
        temp1 := "P" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
        temp2 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
      CASE cTipo == "P" .AND. cNumEval >= 5 .AND. cNumEval <= 7
        temp1 := "P" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval-1)))
        temp2 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval-1)))
      CASE cTipo == "P" .AND. cNumEval == 4
        temp1 := "S" + ALLTRIM(STR (i))
        temp2 := "IS" + ALLTRIM(STR (i))
      CASE cTipo == "P" .AND. cNumEval == 8
        temp1 := "F" + ALLTRIM(STR (i))
        temp2 := "IF" + ALLTRIM(STR (i))

      // El siguiente caso es para postgrados
      CASE cTipo == "G" .OR. cTipo == "D" .AND. cNumEval == 1
        temp1 := "U" + ALLTRIM(STR (i))
        temp2 := "I" + ALLTRIM(STR (i))
    ENDCASE

    // Se guardan las calificaciones y las faltas en la base de alumnos
    REPLACE alumnos->&temp1. WITH auxiliar->parcial
    REPLACE alumnos->&temp2. WITH auxiliar->faltas
  ELSE
    EXIT
  ENDF
  auxiliar->(DBSKIP())
NEXT
alumnos->(DBUNLOCK())
RESTSCREEN (9,18,16,63,cuadro)
RESTSCREEN (7,1,MAXROW(),MAXCOL()-1,pantalla2)
auxiliar->(DBCLOSEAREA())
FERASE (arch+".DBF") // Se borra el archivo temporal
SET ESCAPE ON
EXIT
ELSE
  TONE(750)
  TONE(850)
  ALERT ("EL ALUMNO A SIDO DADO DE BAJA EN OTRA ESTACION DE LA RED")
  EXIT
ENDIF
ELSE
  RESTSCREEN (7,1,MAXROW(),MAXCOL()-1,pantalla2)
  alumnos->(DBUNLOCK())
  auxiliar->(DBCLOSEAREA())
  FERASE (arch+".DBF") // Se borra el archivo temporal
  EXIT
  ENDF
ELSE
  RESTSCREEN (7,1,MAXROW(),MAXCOL()-1,pantalla2)
  auxiliar->(DBCLOSEAREA())

```

```

        FERASE (arch+*.DBF") // Se borra el archivo temporal
    EXIT
ENDIF

CASE nKey==K_UP ;otb:UP()
CASE nKey==K_DOWN ;otb:DOWN()
CASE nKey==K_LEFT ;otb:LEFT()
CASE nKey==K_CTRL_LEFT ;otb:PANLEFT()
CASE nKey==K_RIGHT ;otb:RIGHT()
CASE nKey==K_CTRL_RIGHT ;otb:PANRIGHT()
CASE nKey==K_PGUP ;otb:PAGEUP ()
CASE nKey==K_CTRL_PGUP ;otb:GOTOP()
CASE nKey==K_PGDN ;otb:PAGEDOWN ()
CASE nKey==K_CTRL_PGDN ;otb:GOBOTTOM()
CASE nKey==K_HOME ;otb:HOME()
CASE nKey==K_CTRL_HOME ;otb:PANHOME()
CASE nKey==K_END ;otb:END()
CASE nKey==K_CTRL_END ;otb:PANEND()
OTHERWISE
    TONE(125); TONE(300)
ENDCASE
ENDDO
ENDDO
ENDDO
RETURN
    
```

/* Función que se encarga de capturar las CALIFICACIONES de todos los alumnos asignados a un GRUPO en particular. Se hace uso de un archivo temporal También sirve para consultar en pantalla las calificaciones por grupo para lograr esto basta con hacer .F. el parametro consulta */

```

FUNCTION calif_grupo (consulta)
LOCAL nKey, ok_bases, cColor2 := SETCOLOR ()
LOCAL pantalla, pantalla2, flag := .F.
LOCAL cColor3, lModifico := .F.
PRIVATE aux,cNumEval := SPACE(1), cMaxEvals := 0
PUBLIC cGrupo := SPACE(3), mClavMat := SPACE(5)

IF !FILE("grupo1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
    RETURN
ENDIF
ok_bases := abre_bases (10) // Base de grupos
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
    SETCOLOR (cColor2)
    RETURN
ENDIF

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
    SETCOLOR (cColor2)
    RETURN
ENDIF

ok_bases := abre_bases (3) // Base de materias
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
    SETCOLOR (cColor2)
    RETURN
ENDIF

// Dependiendo si es licenciatura, preparatoria, o postgrado, es el número
// Máximo de evaluaciones
DO CASE
    
```

```

CASE cTipo == "L" // Licenciatura
  cMaxEvals := 3
CASE cTipo == "P" // Preparatoria
  cMaxEvals := 8
CASE cTipo == "O" // Doctorado
  cMaxEvals := 1
CASE cTipo == "D" // Diplomado
  cMaxEvals := 1
ENDCASE

pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

limpiamenus (2)
@ 9, 2 SAY REPLICATE (" ",70)

DO WHILE .T.
  pie (2)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1,"┌──┐┌──┐",C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (10)
  DO WHILE .T.
    cGrupo := SPACE(3)
    SET CURSOR ON
    @ 11, 18 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
    READ
    SET CURSOR OFF
    IF LASTKEY () == K_ESC
      RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
      SETCOLOR (cColor2)
      @ MAXROW()-1,1 SAY SPACE (78)
      alumnos->(DBUNLOCK())
      DBCLOSEALL()
      RETURN -1
    ELSE
      grupos->(DBGOTOP ())
      grupos->(DBSEEK (cGrupo))
      IF grupos->(FOUND())
        alumnos->(DBSETORDER(3))
        alumnos->(DBSEEK (cGrupo))
        IF alumnos->(FOUND())
          EXIT
        ELSE
          TONE (800)
          ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
        ENDIF
      ELSE
        TONE (800)
        ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
      ENDIF
    ENDIF
  ENDDO

pantalla2 := SAVESCREEN (7,1,MAXROW(),MAXCOL()-1) // Se resguarda el area de pantalla de captura
// Se pide y busca la materia
DO WHILE .T.
  pie (9)
  SELECT materias
  SET KEY K_F1 TO help
  // Se pide la clave de la materia
  mClavMat := SPACE(5)
  SET CURSOR ON
  @ 13, 18 GET mClavMat PICTURE "@!" VALID (!EMPTY(mClavMat))
  READ
  SET KEY K_F1 TO
  pie(1)
  SET CURSOR OFF
  IF LASTKEY () == K_ESC
    EXIT
  ELSE
    materias->(DBSEEK(mClavMat))
    IF materias->(FOUND())
      cNomMat := materias->(NomMat)
    ELSE

```

```

TONE (350)
ALERT ("LA MATERIA NO ESTA DADA DE ALTA, NO ESTA ASIGNADA A NINGUN GRUPO Y POR LO TANTO NO ESTA
ASIGNADA A NINGUN ALUMNO")
LOOP
ENDIF
ENDIF

// Se tiene que checar que la materia este asigna a algun alumno
// de este grupo
grupos->(DBGOTOP ())
grupos->(DBSEEK (cGrupo))
IF grupos->(FOUND())
  alumnos->(DBSETORDER(3))
  alumnos->(DBSEEK (cGrupo))
  IF alumnos->(FOUND())
    REPEAT
      FOR i = 1 to 13
        aux := "mat" + ALLTRIM(STR(i))
        IF alumnos->(&aux) == mClavMat
          flag := .T.
          EXIT
        ENDIF
      NEXT
      IF flag
        EXIT
      ENDIF
      alumnos->(DBSKIP())
    UNTIL alumnos->(EOF()) .OR. alumnos->grupo # cGrupo
    IF !flag
      TONE (800)
      ALERT ("LA MATERIA NO ESTA ASIGNADA A NINGUN ALUMNO DE ESTE GRUPO")
      LOOP
    ELSE
      flag := .F.
    ENDIF
  ELSE
    TONE (800)
    ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO O ACABAN DE SER DADOS DE BAJA EN ALGUNA OTRA
ESTACION DE LA RED")
    LOOP
  ENDIF
ELSE
  TONE (800)
  ALERT ("EL GRUPO ACABA DE SER DADO DE BAJA EN ALGUNA OTRA ESTACION DE LA RED")
  LOOP
ENDIF

// Se pide el número de evaluación
DO WHILE .T.
  cNumEval := SPACE(1)
  SET CURSOR ON
  @ 15, 18 GET cNumEval PICTURE "9"
  READ
  aux := cNumEval
  cNumEval := VAL(cNumEval) // Se pasa de caracter a número
  SET CURSOR OFF
  IF LASTKEY () == K_ESC
    cColor := SETCOLOR (C_BORRADO)
    @ 15,18 SAY REPLICATE (" ",2)
    SETCOLOR (cColor)
    flag := .T.
    EXIT
  ELSE
    IF cNumEval > cMaxEvals
      TONE (800)
      ALERT ("No existe esta evaluación en esta escuela")
      LOOP
    ENDIF
  ENDIF
  EXIT
ENDIF
ENDDO
IF flag

```

```

flag := .F.
LOOP
ENDIF

alumnos->(DBSETORDER(4))
alumnos->(DBSEEK (cGrupo))
IF alumnos->(FOUND())

/* Se despliega un cuadro donde se indica la accion que se esta
realizando. Esto se hace, porque si se esta ejecutando el sistema
en una maquina lenta o cuando la red este muy congestionada, est,
proceso puede ser extremadamente lento y es necesario que el
usuario no piense que se a bloqueado el sistema */

cuadro := SAVESCREEN (9,18,15,62)
DISPBOX (9,18,15,62," |  | ^-L | ",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")
@ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)
color := SETCOLOR ("GR+*/N")
@ 12,23 SAY "SE ESTA RECUPERANDO LA INFORMACION"
SETCOLOR (color)

// Se crea el archivo temporal en el que se va a trabajar
arch := nomtemp(cPath,3)
OPEN FILE AUX &arch TIME 30 EXCLUSIVE ALIAS auxiliar

DO WHILE alumnos->grupo == cGrupo
FOR i = 1 TO 13
temp1 := "mat" + ALLTRIM(STR(i))
IF alumnos->(&temp1.) == mClavMat .AND. alumnos->inscrito .AND. EMPTY(alumnos->baja)
IF alumnos->(CsDbLockReq(20))
DO CASE // Se construyen los nombres de los campos adecuados en la base de alumnos
// Los siguientes dos casos son para licenciaturas
CASE cTipo == "L" .AND. cNumEval <= 2
temp2 := "P" + ALLTRIM(STR(i)) + aux
temp3 := "I" + ALLTRIM(STR(i)) + aux
CASE cTipo == "L" .AND. cNumEval == 3
temp2 := "F" + ALLTRIM(STR(i)) + aux
temp3 := "I" + ALLTRIM(STR(i)) + aux

// Los siguientes tres casos son para PREPARATORIAS
CASE cTipo == "P" .AND. cNumEval <= 3
temp2 := "P" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
temp3 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval)))
CASE cTipo == "P" .AND. cNumEval >= 5 .AND. cNumEval <= 7
temp2 := "P" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval-1)))
temp3 := "I" + ALLTRIM(STR (i)+ALLTRIM(STR(cNumEval-1)))
CASE cTipo == "P" .AND. cNumEval == 4
temp2 := "S" + ALLTRIM(STR (i))
temp3 := "IS" + ALLTRIM(STR (i))
CASE cTipo == "P" .AND. cNumEval == 8
temp2 := "F" + ALLTRIM(STR (i))
temp3 := "IF" + ALLTRIM(STR (i))

// El siguiente caso es para postgrados
CASE cTipo == "G" .OR. cTipo == "D" .AND. cNumEval == 1
temp2 := "U" + ALLTRIM(STR (i))
temp3 := "I" + ALLTRIM(STR (i))
ENDCASE
auxiliar->(DBAPPEND(.F.))
REPLACE auxiliar->campo WITH i
REPLACE auxiliar->matricula WITH alumnos->matialum
REPLACE auxiliar->nombre WITH alumnos->nomalum
REPLACE auxiliar->parcial WITH alumnos->(&temp2.)
REPLACE auxiliar->faltas WITH alumnos->(&temp3.)
auxiliar->(DBCOMMIT())
ELSE
ALERT ("Algún(os) registro(s), estan siendo utilizados por otra(s) estacion(es)")

```



```

alumnos-=(DBUNLOCK())
DBCLOSEALL()
RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
SETCOLOR (cColor2)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN -1
ENDIF
ENDIF
NEXT
alumnos-=(DBSKIP())
ENDDO
ELSE
TONE (750)
ALERT ("EL GRUPO ACABA DE SER DADO DE BAJA EN ALGUNA OTRA ESTACION DE LA RED")
DBCLOSEALL()
RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
SETCOLOR (cColor2)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN -1
ENDIF

RESTSCREEN (9,18,15,62,cuadro) // Se restaura el area del cuadro de dialogo
DISPBOX (8,1,maxrow()-3,maxcol()-1,"ÚÄ:ÚÄÄ:Ú",C_REYENO) // Se dibuja la ventana de captura
// Se despliega en la parte superior de la pantalla, el nombre de la
// materia que se selecciono
cColor3 := SETCOLOR ("N/N")
@ 7, 1 SAY SPACE (78)
SETCOLOR (cColor3)
nCentra := (78-LEN (ALLTRIM(cNomMat)))/2
cColor3 := SETCOLOR (C_TITULOS)
@ 7, nCentra SAY ALLTRIM(cNomMat)
SETCOLOR (cColor3)
IF consulta
pie (10)
ELSE
pie (12)
ENDIF

SELECT auxiliar
GOTO TOP

otb:=TBROWSEDB (9,2,maxrow()-4,maxcol()-2)
otb:ADDCOLUMN(TBCOLUMNNEW("Alumno",FIELDWBLOCK("nombre",SELE())))
otb:ADDCOLUMN(TBCOLUMNNEW("Calificación",FIELDWBLOCK("parcial",SELE())))
/* En el concentrado de calificaciones de licenciatura y postgrados se
necesitan las asistencias y para las boletas de preparatoria se
necesitan las inasistencias. */
IF cTipo == "P"
otb:ADDCOLUMN(TBCOLUMNNEW("Inasistencias",FIELDWBLOCK("faltas",SELE())))
ELSE
otb:ADDCOLUMN(TBCOLUMNNEW("Asistencias",FIELDWBLOCK("faltas",SELE())))
ENDIF
columna=otb:GETCOLUMN(2)
columna:DEFcolor:={6,5}
columna=otb:GETCOLUMN(3)
columna:DEFcolor:={6,5}

otb:COLSEP:="|"
otb:HEADSEP:="-"

otb:FREEZE:=1
DO WHILE .T.

DO WHILE NEXTKEY()=0 .AND. !otb:STABILIZE()
ENDDO

/* El reempinkey con estos parametros es exclusivo para los
Tbrowses, esto es para que se realice un refresco cada
10 segundos. Esto es sobre todo para cuando se trabaja
en ambiente de red. */

nKey := reempinkey (0, otb)

```



```

        RETURN -1
    ENDIF
ELSE
    TONE(750)
    ALERT ("    --- PRECAUCION  !!!!           Alguno de los alumnos a sido dado de baja en alguna otra estacion
de la red. Las calificaciones de este alumno se han perdido")
    auxiliar->(DBSKIP())
ENDIF
ENDDO
alumnos->(DBUNLOCK())
RESTSCREEN(7,1,MAXROW(),MAXCOL()-1,pantalla2)
auxiliar->(DBCLOSEAREA())
ERASE (arch+".DBF") // Se borra el archivo temporal
SET ESCAPE ON
EXIT
ELSE
alumnos->(DBUNLOCK())
RESTSCREEN(7,1,MAXROW(),MAXCOL()-1,pantalla2)
auxiliar->(DBCLOSEAREA())
ERASE (arch+".DBF") // Se borra el archivo temporal
EXIT
ENDIF
ELSE
RESTSCREEN(7,1,MAXROW(),MAXCOL()-1,pantalla2)
auxiliar->(DBCLOSEAREA())
ERASE (arch+".DBF") // Se borra el archivo temporal
EXIT
ENDIF

CASE nKey==K_UP           ;otb:UP()
CASE nKey==K_DOWN        ;otb:DOWN()
CASE nKey==K_LEFT        ;otb:LEFT()
CASE nKey==K_CTRL_LEFT   ;otb:PANLEFT()
CASE nKey==K_RIGHT       ;otb:RIGHT()
CASE nKey==K_CTRL_RIGHT  ;otb:PANRIGHT()
CASE nKey==K_PGUP        ;otb:PAGEUP()
CASE nKey==K_CTRL_PGUP   ;otb:GOTOP()
CASE nKey==K_PGDN        ;otb:PAGEDOWN()
CASE nKey==K_CTRL_PGDN   ;otb:GOBOTTOM()
CASE nKey==K_HOME        ;otb:HOME()
CASE nKey==K_CTRL_HOME   ;otb:PANHOME()
CASE nKey==K_END         ;otb:END()
CASE nKey==K_CTRL_END    ;otb:PANEND()
OTHERWISE
    TONE(125); TONE(300)
ENDCASE
ENDDO
ENDDO
ENDDO
RETURN

/* Función que regresa el nombre de la materia seleccionada */
FUNCTION regresa_materia (claves, tipo, regnum)
LOCAL orden
IF tipo == 1
    materias->(DBSEEK (claves))
    IF materias->(FOUND ())
        RETURN materias->nommat
    ENDIF
ELSEIF tipo == 2
    IF asigna->grupo == cGrupo
        materias->(DBSEEK (claves))
        IF materias->(FOUND ())
            RETURN materias->nommat
        ENDIF
    ENDIF
ELSEIF tipo == 3
    IF asigna->grupo == cGrupo
        materias->(DBSEEK (claves))
        IF materias->(FOUND ())
            RETURN materias->area
        ENDIF
    ENDIF
ENDIF

```

```
    ENDIF
  ENDIF
ELSEIF tipo = 4
  materias->(DBSEEK (claves))
  IF materias->(FOUND ())
    RETURN materias->area
  ENDIF
ENDIF
RETURN SPACE (55)
```

MAESTROS.PRG

```

/* Programa   : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje     : CLIPPER 5.2
   Enlazador    : RLINK 3.14B
*/

#xcommand REPEAT => do while .t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla
#include "caos.ch"

PROCEDURE alta_maes()
LOCAL pantalla, cColor, cColor2, ok_base:= 2, ok_bases
LOCAL ok := .T., direct := SPACE(5), ok_dir := 1, arch, nTipo_esc

cColor2 := SETCOLOR()
SET PATH TO &cPath
IF !FILE("&cPath\maestro1.ntx")
  ok_bases := abre_bases(6) // Base de maestros
  IF ok_bases
    INDEX ON clavmae TO &cPath\maestro1
    INDEX ON Nommae TO &cPath\maestro2
    DBCLOSEAREA()
  ELSE
    TONE(750)
    ALERT("NO SE PUEDE ABRIR LA BASE DE MAESTROS")
    SETCOLOR(cColor2)
    RETURN
  ENDIF
ENDIF

SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN(10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menf.
pantallacampo(3) // Despliega los campos a capturar
ok_bases := abre_bases(5) // Base de maestros
IF !ok_bases
  TONE(750)
  ALERT("NO SE PUEDE ABRIR LA BASE DE MAESTROS")
  SETCOLOR(cColor2)
  RETURN
ENDIF

DO WHILE ok
  pie(2)
  GOTO TOP
  var_ini()
  limpia_vars(2)
  ok := maes_alt(1)
ENDDO
DBCLOSEAREA()
SET CURSOR OFF
RESTSCREEN(10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR(cColor2)
@ MAXROW()-1,1 SAY SPACE(78)
RETURN NIL

FUNCTION maes_alt(tipo)
  LOCAL ok_almacen := 2, correcto := "S"

  @ 11,17 GET mClavmae PICTURE "@" VALID (!EMPTY(mClavmae))
  READ

```

```

IF LASTKEY() == K_ESC // Cancelar operacion
RETURN .F.
ENDIF
mClavmae := LTRIM(mClavmae)
IF LFN(mClavmae) < 5 // Se valida la long. de la variable
DO WHILE LFN(mClavmae) < 5 // y que no tenga espacios a la izq.
mClavmae += SPACE(1)
ENDDO
cColor := SETCOLOR ("GR+/G")
@ 11,17 SAY mClavmae PICTURE "@!"
SETCOLOR (cColor)
ENDIF
IF tipo == 1
DBSEEK (mClavmae)
IF FOUND ()
TONE (550)
ALERT("MAESTRO YA EXISTENTE, NO PUEDE REALIZARSE ESTA OPERACION")
RETURN .T.
ENDIF
ENDIF
pie (1)
@ 11,40 GET mNommae PICTURE "@!" VALID (!EMPTY(mNommae))
@ 13,17 GET mRfcmmae PICTURE "@R@! AAAA-999999-XXX"
@ 13,48 GET mNivmae PICTURE "@!"
@ 13,74 GET mSexmae PICTURE "@!" VALID (mSexmae$ "FM ")
@ 15,17 GET mDommae PICTURE "@!" VALID (!EMPTY(mDommae))
@ 15,70 GET mCpmae PICTURE "@!"
@ 17,17 GET mCiudmae PICTURE "@!"
@ 17,55 GET mColmae PICTURE "@!"
@ 19,17 GET mTelmae PICTURE "@!"
@ 19,67 GET ml'cingmae PICTURE "@e"
READ
IF LASTKEY() == K_FSC // Cancelar operacion
cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
RETURN .T.
ENDIF
DO WHILE .T.
@ 20,5 SAY "Son correctos los datos (S/N) : " GET correcto PICTURE "!A" VALID (correcto$ "SN")
READ
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("U",35)
SETCOLOR (cColor)
IF LASTKEY () == K_ESC
RETURN .T.
ENDIF
IF correcto == "S"
IF tipo == 1
ok_almacen := grabar(1)
IF !ok_almacen
RETURN .F.
ENDIF
ELSEIF tipo == 2
IF lockr(30)
ok_almacen := grabar(2)
IF !ok_almacen
RETURN .F.
ENDIF
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
RETURN .T.
ENDIF
ENDIF
RETURN .T.
ELSE
RETURN .T.
ENDIF
ENDDO
RETURN .F.

```

* Procedimiento que permite modificar los datos de la información de maestros que ya existen en la base de datos */

```

PROCEDURE modif_mae()
LOCAL pantalla, cColor, ok_almacen := 2, existe := 0
LOCAL cColor2, cColor3, ok := "N", salir := 0, ok_bases

cColor3 := SETCOLOR ()
SET PATH TO &cPath
IF !FILE("&cPath\maestro1.ntx")
ok_bases := abre_bases (6) // Base de maestros
IF ok_bases
INDEX ON clavmae TO &cPath\maestro1
INDEX ON Nommae TO &cPath\maestro2
DBCLOSEAREA()
ELSE
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
SETCOLOR ("W/N")
CLS
QUIT
ENDIF
ENDIF

SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menú
pantallacampos (3) // Despliega los campos a capturar
ok_bases := abre_bases (5) // Base de maestros
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
SETCOLOR ("W/N")
CLS
QUIT
ENDIF

DO WHILE .T.
GOTO TOP
salir := 0
cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
pie(2)
DO WHILE !FOUND()
GOTO TOP
var_ini()
limpia_vars (2)

@@ 11,17 GET mClavmae PICTURE "@!" VALID (!EMPTY(mClavmae))
READ

IF LASTKEY() == K_ESC // Cancelar operacion
DBCLOSEAREA()
SET CURSOR OFF
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN
ENDIF

mClavmae := LTRIM(mClavmae)
IF LEN(mClavmae) < 5 // Se valida la long. de la variable
DO WHILE LEN (mClavmae) < 35 // y que no tenga espacios a la izq.
mClavmae += SPACE(1)
ENDDO
@@ 11,17 GET mClavmae PICTURE "@!"
CLEAR GETS
ENDIF
/* ***** Se busca el nombre (o parte de el) en la base de datos.
Si existe se posiciona el apuntador en el registro donde se encuentre */
SET FILTER TO ALLTRIM(mClavmae) $UPPER(maestros->Clavmae)
GOTO TOP

```

```
LOCATE FOR ALLTRIM(mClavmae) $UPPER(maestros->Clavmae)
```

```
/* Si no se encuentra el nombre en la base de datos se piden los
datos necesarios para darlo de alta */
```

```
IF NOT FOUND()
  TONE(350)
  @ 11,17 SAY SPACE(5)
  @ 11,17 SAY "NO EXISTE"
  INKEY(1)
  @ 11,17 SAY "    "
ENDIF
ENDDO
```

```
// Si se encuentra el nombre (o parte de el) en la base ...
// se leen los campos de la base de datos
DO WHILE (FOUND().OR. LASTKEY() <> K_ESC)
```

```
  IF salir == 1
    salir := 0
    EXIT
  ENDIF
  pie(3)
  IF EOF() // Se verifica que no sea el fin de archivo
    TONE(450)
    cColor2 := SETCOLOR("W+/R")
    @ 20,5 SAY "Es el ultimo"
    SETCOLOR(cColor2)
    INKEY(1)
    cColor := SETCOLOR(C_BORRADO)
    @ 20, 5 SAY REPLICATE("■",13)
    SETCOLOR(cColor)
    DBSKIP(-1)
  ELSE
    @ 11,17 GET maestros->Clavmae PICTURE "@!"
    @ 11,40 GET maestros->Nommae PICTURE "@!"
    @ 13,17 GET maestros->Rfcmae PICTURE "@R@! AAAA-999999-XXX"
    @ 13,48 GET maestros->Nivmae PICTURE "@!"
    @ 13,74 GET maestros->Sexmae PICTURE "@!"
    @ 15,17 GET maestros->Dommae PICTURE "@!"
    @ 15,70 GET maestros->Cpmae PICTURE "@!"
    @ 17,17 GET maestros->Ciudmae PICTURE "@!"
    @ 17,55 GET maestros->Colmae PICTURE "@!"
    @ 19,17 GET maestros->Telmae PICTURE "@!"
    @ 19,67 GET maestros->Feingmae PICTURE "@e"
    CLEAR GETS
  ENDIF
  INKEY(0)
  DO CASE
    CASE LASTKEY() == K_LEFT // Si la tecla pulsada fue <-
      flechas(1)
    CASE LASTKEY() == K_RIGHT // Si la tecla pulsada fue ->
      flechas(2)
    CASE LASTKEY() == K_ENTER // Se a seleccionado una escuela
      pie(1)
      igual_var_cam()
      maes_alt(2)
      EXIT
    CASE LASTKEY() == K_ESC // Se a cancelado la operacion
      cColor := SETCOLOR(C_BORRADO)
      @ 20, 5 SAY REPLICATE("■",55)
      SETCOLOR(cColor)
      DBCLEARFILTER()
      EXIT
  ENDCASE
ENDDO
ENDDO

  DBCLOSEAREA()
  RESTSCREEN(10,1,maxrow()-3,maxcol()-1,pantalla)
  SETCOLOR(cColor3)
  @ MAXROW()-1,1 SAY SPACE(78)
  RETURN NIL
```



```

* Procedimiento que borra la información de la base de maestros
PROCEDURE baja_mae()
LOCAL pantalla, cColor, ok_almacen := 2, correcto := "N", existe := 0
LOCAL cColor2, cColor3, ok := "N", ok_bases, panta, ok_cambia, mClavmae2 := SPACE(5)
LOCAL flag := 0

cColor3 := SETCOLOR ()
SET PATH TO &cPath
IF !FILE("&cPath\maestro1.ntx")
    ok_bases := abre_bases (6) // Base de maestros
    IF ok_bases
        INDEX ON Clavmae TO &cPath\maestro1
        INDEX ON Nommae TO &cPath\maestro2
        DBCLOSEAREA()
    ELSE
        TONE (750)
        ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS")
        SETCOLOR (cColor3)
        RETURN
    ENDIF
ENDIF

SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menú
pantallacampos (6) // Despliega los campos a capturar

ok_bases := abre_bases (10) // Base de grupos
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
    SETCOLOR (cColor3)
    RETURN
ENDIF

ok_bases := abre_bases (5) // Base de maestros
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS")
    SETCOLOR (cColor3)
    RETURN
ENDIF

DO WHILE .T.
    GOTO TOP
    pic(2)
    DO WHILE !FOUND()
        GOTO TOP
        correcto := "N"
        var_ini()
        limpia_vars (2)

        @ 11,17 GET mClavmae PICTURE "@!" VALID (!EMPTY(mClavmae))
        READ

        IF LASTKEY() == K_ESC // Cancelar operacion
            DBCLOSEAREA()
            SELECT 1
            DBCLOSEAREA()
            SET CURSOR OFF
            RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
            SETCOLOR (cColor3)
            @ MAXROW()-1,1 SAY SPACE (78)
            RETURN
        ENDIF

        mClavmae := LTRIM(mClavmae)
        IF LEN(mClavmae) < 5 // Se valida la long. de la variable
            DO WHILE LEN (mClavmae) < 35 // y que no tenga espacios a la izq.
                mClavmae += SPACE(1)
            ENDDO
    
```

```

@ 11,17 GET mClavmae PICTURE "@!"
CLEAR GETS
ENDIF
/* ***** Se busca el nombre (o parte de el) en la base de datos.
Si existe se posiciona el apuntador en el registro donde se encuentre */
SET FILTER TO ALLTRIM(mClavmae) $UPPER(maestros->Clavmae)
GOTO TOP
LOCATE FOR ALLTRIM(mClavmae) $UPPER(maestros->Clavmae)

/* Si no se encuentra el nombre en la base de datos se piden los
datos necesarios para darlo de alta */
IF .NOT. FOUND ()
TONE (350)
@ 11,17 SAY SPACE(5)
@ 11,17 SAY "NO EXISTE"
INKEY (1)
@ 11,17 SAY " "
ENDIF
ENDDO

// Si se encuentra el nombre (o parte de el) en la base ...
// se leen los campos de la base de datos
DO WHILE (FOUND () .OR. LASTKEY() <> K_ESC)
pie (3)
IF EOF () // Se verifica que no sea el fin de archivo
TONE(450)
cColor2 := SETCOLOR ("W+/R")
@ 20,5 SAY "Es el ultimo"
SETCOLOR (cColor2)
INKEY (1)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATÉ ("■",13)
SETCOLOR (cColor)
DBSKIP(-1)
ELSE
@ 11,17 GET maestros->Clavmae PICTURE "@!"
@ 11,40 GET maestros->Nommae PICTURE "@!"
@ 13,17 GET maestros->Rfcmae PICTURE "@R@! AAAA-999999-XXXX"
@ 13,48 GET maestros->Nivmae PICTURE "@!"
CLEAR GETS
ENDIF
INKEY (0)
DO CASE
CASE LASTKEY () == K_LEFT // Si la tecla pulsada fue <-
flechas (1)
CASE LASTKEY () == K_RIGHT // Si la tecla pulsada fue ->
flechas (2)
CASE LASTKEY () == K_ENTER // Se a seleccionado una escuela
pie(1)
mClavmae := maestros->Clavmae
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro en DARLO DE BAJA :"
SETCOLOR (cColor2)
@ 20,36 GET ok PICTURE "!A" VALID (ok$"SN")
READ
IF ok == "N" .OR. LASTKEY () == K_ESC
TONE (550)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "El Registro NO a sido dado de baja"
SETCOLOR (cColor2)
inkey (2)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATÉ ("■",55)
SETCOLOR (cColor)
DBCLEARFILTER ()

cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
EXIT
ELSEIF ok == "S"
DBCLEARFILTER ()

```

```

SELECT 1 // archivo de grupos
DBSETORDER (3)
DBSEK (mClavmae)
IF !FOUND()
  SELECT 2
  DBSEK (mClavmae)
  IF FOUND()
    ok:="N"
    IF lockr (30)
      DELETE
      DBUNLOCK ()
      TONE (550)
      cColor2 := SETCOLOR ("W+/R")
      @ 20,5 SAY "El maestro ha sido dado de BAJA"
      SETCOLOR (cColor2)
      inkey (1)
      cColor := SETCOLOR (C_BORRADO)
      @ 20, 5 SAY REPLICATE ("█",55)
      SETCOLOR (cColor)
      DBCLEARFILTER ()
      GOTO TOP
    EXIT
  ELSE
    ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE
MOMENTO")
    EXIT
  ENDIF
  EXIT
ELSE
  ALERT (" EL MAESTRO ACABA DE SER DADO DE BAJA EN ALGUNA OTRA ESTACION DE LA RED
")
  cColor := SETCOLOR (C_BORRADO)
  @ 20, 5 SAY REPLICATE ("█",55)
  SETCOLOR (cColor)
  DBCLEARFILTER ()
  EXIT
ENDIF
ELSE
  cColor := SETCOLOR (C_BORRADO)
  @ 20, 5 SAY REPLICATE ("█",55)
  SETCOLOR (cColor)
  DBSETORDER (1) // Orden de indices de grupos
  SELECT 2 // Archivo de maestros
  TONE (600)
  TONE (600)
  ALERT ("ESTE MAESTRO TIENE ASIGNADAS MATERIA(s), SE DEBERA ASIGNAR OTRO MAESTRO PARA DICHA(s)
MATERIA(s)")
  panta := SAVESCREEN (1,1,maxrow(),maxcol())
  pantallacampos (7)
  cColor := SETCOLOR ("R+*/N")
  @ MAXROW()-2,28 SAY "REEMPLAZANDO MAESTRO"
  SETCOLOR (cColor)
  DO WHILE .T.
    @ 17,17 GET mClavmae2 PICTURE "@!" VALID (!EMPTY(mClavmae))
    READ
    IF mClavmae2 == mClavmae
      TONE (800)
      cColor := SETCOLOR (C_BORRADO)
      @ 17,17 SAY REPLICATE ("█",5)
      SETCOLOR (cColor)
      @ 17,17 SAY "E R R O R"
      INKEY (1)
      @ 17,17 SAY " "
      LOOP
    ENDIF
    IF LASTKEY () == K_ESC
      TONE (550)
      cColor2 := SETCOLOR ("R/B+")
      @ 20,5 SAY "El Registro NO a sido dado de baja"
      SETCOLOR (cColor2)
      inkey (2)
      cColor := SETCOLOR (C_BORRADO)

```

```

@ 20,5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
RESTSCREEN (1,1,maxrow(),maxcol(),panta)
flag := 1
EXIT
ENDIF
DBSEEK (mClavmae2)
IF !FOUND ()
  TONE (800)
  cColor := SETCOLOR (C_BORRADO)
  @ 17,17 SAY REPLICATE ("█",5)
  SETCOLOR (cColor)
  @ 17,17 SAY "NO EXISTE"
  INKEY (1)
  @ 17,17 SAY " "
ELSE
  @ 17,40 GET maestros->Nonmae PICTURE "@!"
  @ 19,17 GET maestros->Rfcmac PICTURE "@R@! AAAA-999999-XXX"
  @ 19,48 GET maestros->Nivmae PICTURE "@!"
  CLEAR GETS
  cColor2 := SETCOLOR ("R/B+")
  @ 20,5 SAY "Esta seguro del reemplazo ."
  SETCOLOR (cColor2)
  ok := "N"
  @ 20,33 GET ok PICTURE "!A" VALID (ok$"SN")
  READ
  IF ok == "S"
    EXIT
  ELSE
    cColor := SETCOLOR (C_BORRADO)
    @ 17,40 SAY REPLICATE ("█",36)
    @ 19,17 SAY REPLICATE ("█",15)
    @ 19,48 SAY REPLICATE ("█",15)
    @ 20,5 SAY REPLICATE ("█",39)
    SETCOLOR (cColor)
  ENDIF
ENDIF
ENDDO
IF flag == 1 // si se presiono ESC al pedir la clave de reemplazo
  flag := 0
  exit
ENDIF
SET ESCAPE OFF // Evita que el programa pueda se abortado con Alt-C
DBSEEK (mClavmae)
IF FOUND()
  ok: "N"
  SELECT 1 // Archivo de grupos
  DBSETORDER (3)
  DO WHILE .T.
    GOTO TOP
    DBSEEK (mclavmae)
    IF FOUND ()
      IF lockr (120)
        REPLACE clavmae WITH mclavmae2
      ENDIF
    ELSE
      EXIT
    ENDIF
  ENDDO
  DBSETORDER (1)
  SELECT 2
  IF lockr (30)
    DELETE
  DBUNLOCK ()
  TONE (550)
  cColor2 := SETCOLOR ("W+/R")
  @ 20,5 SAY "El maestro ha sido dado de BAJA"
  SETCOLOR (cColor2)
  inkey (1)

```

```

cColor := SETCOLOR (C_BORRADO)
@@ 20, 5 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
E.I.SE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE
MOMENTO")
SET ESCAPE ON
EXIT
ENDIF
ELSE
ALERT (" EL MAESTRO ACABA DE SER DADO DE BAJA EN ALGUNA OTRA ESTACION DE LA RED ")
cColor := SETCOLOR (C_BORRADO)
@@ 20, 5 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
RESTSCREEN (1,1,maxrow(),maxcol(),panta)
SET ESCAPE ON
EXIT
ENDIF
SET ESCAPE ON
RESTSCREEN (1,1,maxrow(),maxcol(),panta)
EXIT
ENDIF
ENDIF

CASE LASTKEY () == K_FSC // Se a cancelado la operacion
cColor := SETCOLOR (C_BORRADO)
@@ 20, 5 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
EXIT
ENDCASE
ENDDO
ENDDO
DBCLOSEAREA()
SELECT 1
DBCLOSEAREA()
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL
    
```

MATERIAS.PRG

```

/* Programa   : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje     : CLIPPER 5.2
   Enlazador    : RTLINK 3.14B
*/

#xcommand REPEAT => do while .t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla
#include "caos.ch"

// Procedimiento para dar de alta al sistema las materias de cada escuela
PROCEDURE alta_mats()
LOCAL pantalla, cColor
LOCAL cColor2, ok:=.T., ok_bases

cColor2 := SETCOLOR()
SET PATH TO &cPath
IF !FILE("&cPath\materia1.ntx")
  ok_bases := abre_bases(4) // Base de materias
  IF ok_bases
    INDEX ON clavmat TO &cPath\materia1
    INDEX ON Nommat TO &cPath\materia2
    DBCLOSEAREA()
  ELSE
    TONE(750)
    ALERT("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
    SETCOLOR(cColor2)
    RETURN
  ENDIF
ENDIF

SET CURSOR ON
pantalla := SAVESCREEN(10,1,maxrow()-3,maxcol()-1)
limpianenus(2) // Apaga el men
pantallacampos(2) // Despliega los campos a capturar
ok_bases := abre_bases(3) // Base de materias
IF !ok_bases
  TONE(750)
  ALERT("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  SETCOLOR(cColor2)
  RETURN
ENDIF
DO WHILE ok
  pi(2)
  GOTO TOP
  var_ini()
  limpia_vars(1)
  ok := mats_alt(1)
ENDDO
DBCLOSEAREA()
SET CURSOR OFF
RESTSCREEN(10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR(cColor2)
@ MAXROW()-1,1 SAY SPACE(78)
RETURN NIL

FUNCTION mats_alt(tipo)
LOCAL ok_almacen := 2, correcto := "S"

@ 12,16 GET mClavmat PICTURE "@!" VALID (!EMPTY(mClavmat))
READ
IF LASTKEY() == K_ESC // Cancelar operacion

```

```

RETURN .F.
ENDIF
mClavmat := LTRIM(mClavmat)
IF LEN(mClavmat) < 5 // Se valida la long. de la variable
DO WHILE LEN(mClavmat) < 5 // y que no tenga espacios a la izq.
mClavmat += SPACE(1)
ENDDO
cColor := SETCOLOR ("GR+G")
@ 12,16 SAY mClavmat PICTURE "@!"
SETCOLOR (cColor)
ENDIF
IF tipo == 1
DBSEEK (mClavmat)
IF FOUND ()
TONE (550)
ALERT("MATERIA YA EXISTENTE, NO PUEDE REALIZARSE ESTA OPERACION")
RETURN .T.
ENDIF
ENDIF
pic (1)
@ 14,16 GET mNommat PICTURE "@!" VALID (!EMPTY(mNommat))
@ 16,16 GET mSerimat PICTURE "@!"
@ 16,37 GET mArea PICTURE "@!"
READ
IF LASTKEY() == K_ESC // Cancelar operacion
cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
RETURN .T.
ENDIF
DO WHILE .T.
@ 18,5 SAY "Son correctos los datos (S/N) : " GET correcto PICTURE "!A" VALID (correcto$"SN")
READ
cColor := SETCOLOR (C_BORRADO)
@ 18,5 SAY REPLICATE ("■",35)
SETCOLOR (cColor)
IF LASTKEY () == K_ESC
RETURN .T.
ENDIF
IF correcto == "S"
IF tipo == 1
ok_almacen := grabar(1)
IF !ok_almacen
RETURN .F.
ENDIF
ELSEIF tipo == 2
IF lockr (30)
ok_almacen := grabar(2)
IF !ok_almacen
RETURN .F.
ENDIF
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
RETURN .T.
ENDIF
ENDIF
RETURN .T.
ELSE
RETURN .T.
ENDIF
ENDDO
RETURN .F.

```

*Procedimiento que permite dar de baja materias.

```

PROCEDURE baja_mats()
LOCAL pantalla, cColor, ok_almacen := 2, correcto := "N", existe := 0
LOCAL cColor2, cColor3, ok := "N", ok_basas

cColor3 := SETCOLOR ()
SET PATH TO &cPath
IF !FILE ("&cPath\material.ntx")
ok_basas := abre_basas (4) // Base de materias

```

```

IF ok_bases
  INDEX ON clavmat TO &cPath\materia1
  INDEX ON Nommat TO &cPath\materia2
  DBCLOSEAREA()
ELSE
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS")
  SETCOLOR (cColor2)
  RETURN
ENDIF
ENDIF

SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menf
pantallacampos (2) // Despliega los campos a capturar

ok_bases := abre_bases (10) // Base de grupos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
  SETCOLOR (cColor2)
  RETURN
ENDIF

ok_bases := abre_bases (3) // Base de materias
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  SETCOLOR (cColor2)
  RETURN
ENDIF

DO WHILE .T.
  GOTO TOP
  cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
  @ MAXROW()-1,1 SAY SPACE (78)
  pie (2)
  SETCOLOR (cColor)
  DO WHILE !FOUND()
    GOTO TOP
    correcto := "N"
    var_ini()
    limpia_vars (1)

    @ 12,16 GET mClavmat PICTURE "@!" VALID (!EMPTY(mClavmat))
    READ

    IF LASTKEY() == K_ESC // Cancelar operacion
      grupos->(DBCLOSEAREA())
      materias->(DBCLOSEAREA())
      SET CURSOR OFF
      RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
      SETCOLOR (cColor3)
      @ MAXROW()-1,1 SAY SPACE (78)
      RETURN
    ENDIF

    mClavmat := LTRIM(mClavmat)
    IF LEN(mClavmat) < 5 // Se valida la long. de la variable
      DO WHILE LEN (mClavmat) < 5 // y que no tenga espacios a la izq.
        mClavmat += SPACE(1)
      ENDDO
      @ 12,16 GET mClavmat PICTURE "@!"
      CLEAR GETS

      ENDIF
      /* ***** Se busca el nombre (o parte de el) en la base de datos.
      Si existe se posiciona el apuntador en el registro donde se encuentre */
      SET FILTER TO ALLTRIM(mClavmat) $UPPER(materias->Clavmat)
      GOTO TOP
      LOCATE FOR ALLTRIM(mClavmat) $UPPER(materias->Clavmat)

```



```
/* Si no se encuentra el nombre en la base de datos se piden los
datos necesarios para darlo de alta */
```

```
IF .NOT. FOUND ()
  TONE (350)
  cColor := SETCOLOR (C_BORRADO)
  @ 12,16 SAY REPLICATE ("█",55)
  SETCOLOR (cColor)
  @ 12,16 SAY "NO EXISTE"
  INKEY (1)
  @ 12,16 SAY " "
```

```
ENDIF
ENDDO
```

```
// Si se encuentra el nombre (o parte de el) en la base ...
```

```
// se leen los campos de la base de datos
```

```
DO WHILE (FOUND () .OR. LASTKEY() <> K_ESC)
```

```
  pie (3)
```

```
  IF EOF () // Se verifica que no sea el fin de archivo
```

```
    TONE(450)
```

```
    cColor2 := SETCOLOR ("W+/R")
```

```
    @ 20,5 SAY "Es el ultimo"
```

```
    SETCOLOR (cColor2)
```

```
    INKEY (1)
```

```
    cColor := SETCOLOR (C_BORRADO)
```

```
    @ 20, 5 SAY REPLICATE ("█",13)
```

```
    SETCOLOR (cColor)
```

```
    DBSKIP(-1)
```

```
  ELSE
```

```
    @ 12,16 GET materias->Clavmat PICTURE "@!"
```

```
    @ 14,16 GET materias->Nommat PICTURE "@!"
```

```
    @ 16,16 GET materias->Serimat PICTURE "@!"
```

```
    @ 16,37 GET materias->Area PICTURE "@!"
```

```
    CLEAR GETS
```

```
  ENDIF
```

```
  INKEY (0)
```

```
  DO CASE
```

```
    CASE LASTKEY () == K_LEFT // Si la tecla pulsada fue <-
```

```
      flechas (1)
```

```
    CASE LASTKEY () == K_RIGHT // Si la tecla pulsada fue ->
```

```
      flechas (2)
```

```
    CASE LASTKEY () == K_ENTER // Se a seleccionado una escuela
```

```
      mClavmat := materias->Clavmat
```

```
      grupos->(DBSETORDER (2))
```

```
      grupos->(DBSEEK (mclavmat))
```

```
      IF grupos->(FOUND())
```

```
        TONE (650)
```

```
        ALERT ("No se puede dar de baja, ya que a sido asignada a algun grupo. Elimine la asignación y reintente la operación")
```

```
        grupos->(DBSETORDER (1))
```

```
        GO TO TOP
```

```
        TONE (550)
```

```
        cColor2 := SETCOLOR ("R/B+")
```

```
        @ 20,5 SAY "El Registro NO a sido dado de baja"
```

```
        SETCOLOR (cColor2)
```

```
        inkey (2)
```

```
        cColor := SETCOLOR (C_BORRADO)
```

```
        @ 20, 5 SAY REPLICATE ("█",55)
```

```
        SETCOLOR (cColor)
```

```
        DBCLEARFILTER ()
```

```
        cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
```

```
        @ MAXROW()-1,1 SAY SPACE (78)
```

```
        SETCOLOR (cColor)
```

```
        EXIT
```

```
      ENDIF
```

```
      grupos->(DBSETORDER (1))
```

```
      pie (1)
```

```
      cColor2 := SETCOLOR ("R/B+")
```

```
      @ 20,5 SAY "Esta seguro en DARLO DE BAJA :"
```

```
      SETCOLOR (cColor2)
```

```
      @ 20,36 GET ok PICTURE "!A" VALID (ok$"SN")
```

```
      READ
```

```
      IF ok == "N" .OR. LASTKEY () == K_ESC
```



```

TONE (550)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "El Registro NO a sido dado de baja"
SETCOLOR (cColor2)
inkey (2)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColor)
EXIT
ELSEIF ok == "S"
DBSEEK (mClavmat)
IF FOUND()
ok:="N"
IF lockr (30)
DELETE
DBUNLOCK ()
SELECT materias
TONE (550)
cColor2 := SETCOLOR ("W+R")
@ 20,5 SAY "La materia a sido dada de BAJA"
SETCOLOR (cColor2)
inkey (1)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
GOTO TOP
EXIT
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE
MOMENTO")
EXIT
ENDIF
ELSE
ALERT (" LA MATERIA ACABA DE SER DADA DE BAJA EN ALGUNA OTRA ESTACION DE LA RED ")
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
ENDIF
ENDIF

CASE LASTKEY () == K_ESC // Se a cancelado la operacion
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("█",55)
SETCOLOR (cColor)
DBCLEARFILTER ()
EXIT
ENDCASE

ENDDO
ENDDO
DBCLOSEAREA()
SELECT 1
DBCLOSEAREA()
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL

*Procedimiento para modificar los datos de las materias existentes
PROCEDURE modif_mats()
LOCAL pantalla, cColor, ok_almacen := 2, correcto := "N", existe := 0
LOCAL cColor2, cColor3, ok := "N", salir := 0, ok_bases

cColor3 := SETCOLOR ()
SET PATH TO &cPath
IF !FILE ("&cPath\materia1.txt")
ok_bases := abre_bases (4) // Base de materias en exclusiva

```

```

IF ok_bases
  INDEX ON clavmat TO &cPath\materia1
  INDEX ON Nommat TO &cPath\materia2
  DBCLOSEAREA()
ELSE
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  SETCOLOR (cColor2)
  RETURN
ENDIF
ENDIF

SET CURSOR ON
cColor := SETCOLOR()
pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
limpiamenus(2) // Apaga el menú
pantallacampos (2) // Despliega los campos a capturar
ok_bases := abre_bases (3) //Base de materias compartida
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  SETCOLOR (cColor2)
  RETURN
ENDIF

DO WHILE .T.
  GOTO TOP
  salir := 0
  cColor := SETCOLOR ("N/N") // Cambian los mensajes de pie de pantalla
  @ MAXROW()-1,1 SAY SPACE (78)
  SETCOLOR (cColor)
  pie(2)
  DO WHILE !FOUND()
    GOTO TOP
    correcto := "N"
    var_ini()
    limpia_vars (1)
    @ 12,16 GET mClavmat PICTURE "@!" VALID (!EMPTY(mClavmat))
    READ

    IF LASTKEY() == K_ESC // Cancelar operacion
      DBCLOSEAREA()
      SET CURSOR OFF
      RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
      SETCOLOR (cColor3)
      @ MAXROW()-1,1 SAY SPACE (78)
      RETURN
    ENDIF

    mClavmat := LTRIM(mClavmat)
    IF LEN(mClavmat) < 5 // Se valida la long. de la variable
      DO WHILE LEN (mClavmat) < 5 // y que no tenga espacios a la izq.
        mClavmat += SPACE(1)
      ENDDO
      @ 12,16 GET mClavmat PICTURE "@!"
      CL.FAR GETS
    ENDIF

    /* ***** Se busca el nombre (o parte de el) en la base de datos.
    Si existe se posiciona el apuntador en el registro donde se encuentre */
    SET FILTER TO ALLTRIM(mClavmat) $UPPER(materias->Clavmat)
    GOTO TOP
    LOCATE FOR ALLTRIM(mClavmat) $UPPER(materias->Clavmat)

    /* Si no se encuentra el nombre en la base de datos se piden los
    datos necesarios para darlo de alta */
    IF .NOT. FOUND ()
      TONE (350)
      cColor := SETCOLOR (C_BORRADO)
      @ 12,16 SAY REPLICATE ("■",55)
      SETCOLOR (cColor)
      @ 12,16 SAY "NO EXISTE"
      INKEY (1)

```

```

    @ 12,16 SAY "    "
ENDIF
ENDDO

// Si se encuentra el nombre (o parte de el) en la base ...
// se leen los campos de la base de datos
DO WHILE (FOUND() .OR. LASTKEY() <> K_ESC)
  IF salir == 1
    salir := 0
    EXIT
  ENDIF
  pie(3)
  IF EOF() // Se verifica que no sea el fin de archivo
    TONE(450)
    cColor := SETCOLOR ("W+/R")
    @ 20, 5 SAY "Es el ultimo"
    SETCOLOR (cColor)
    INKEY (1)
    cColor := SETCOLOR (C_BORRADO)
    @ 20, 5 SAY REPLICATE ("■",13)
    SETCOLOR (cColor)
    DBSKIP(-1)
  ELSE
    @ 12,16 GET materias->Clavmat PICTURE "@!"
    @ 14,16 GET materias->Nommat PICTURE "@!"
    @ 16,16 GET materias->Seriemat PICTURE "@!"
    @ 16,37 GET materias->Area PICTURE "@!"
    CLEAR GETS
  ENDIF
  INKEY (0)
  DO CASE
    CASE LASTKEY () == K_LEFT // Si la tecla pulsada fue <-
      flechas (1)
    CASE LASTKEY () == K_RIGHT // Si la tecla pulsada fue ->
      flechas (2)
    CASE LASTKEY () == K_ENTER // Se a seleccionado una escuela
      pie (1)
      igual_var_cam()
      mats_alt(2)
      EXIT
    CASE LASTKEY () == K_ESC // Se a cancelado la operacion
      cColor := SETCOLOR (C_BORRADO)
      @ 20, 5 SAY REPLICATE ("■",55)
      SETCOLOR (cColor)
      DBCLEARFILTER ()
      EXIT
    ENDCASE
  ENDDO
ENDDO
DBCLOSEAREA()
RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
SETCOLOR (cColor3)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN NIL

```

GRUPOS.PRG

```

/* Programa   : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje     : CLIPPER 5.2
   Enlazador    : RTLINK 3.14B
*/

#xcommand REPEAT => do while .t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla
#include "caos.ch"

/*****/
PROCEDURE alta_grupos()
  LOCAL pantalla, cColor
  LOCAL cColor2, ok := .T., ok_bases
  PUBLIC consec := -1, cYear

  cColor2 := SETCOLOR()
  SET PATH TO &cPath

  IF !FILE("&cPath\maestro1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA MAESTROS PARA ESTA ESCUELA")
    SETCOLOR (cColor2)
    RETURN
  ELSE
    ok_bases := abre_bases (5) // Abre la base de maestros
    IF !ok_bases
      TONE (750)
      ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS")
      SETCOLOR (cColor2)
      RETURN
    ENDIF
  ENDIF

  SET CURSOR ON
  pantalla := SAVESCREEN (10,1,maxrow()-3,maxcol()-1)
  limpiamenus(2) // Apaga el menL
  pantallacampos (5) // Despliega los campos a capturar

  DO WHILE ok
    SET KEY K_F2 TO
    pic(2)
    cColor := SETCOLOR (C_BORRADO)
    @ 12, 21 SAY REPLICATE ("█",5)
    @ 13, 21 SAY REPLICATE ("█",5)
    @ 12, 43 SAY REPLICATE ("█",35)
    @ 13, 43 SAY REPLICATE ("█",35)
    SETCOLOR (cColor)
    ok := grupo_alt()
  ENDDO
  DBCLOSEALL()
  SET CURSOR OFF
  RESTSCREEN (10,1,maxrow()-3,maxcol()-1,pantalla)
  SETCOLOR (cColor2)
  @ MAXROW()-1,1 SAY SPACE (78)
  RETURN NIL

FUNCTION grupo_alt()
  LOCAL ok_almacen := 2, correcto := "S", flag := 0
  LOCAL cGrupo := SPACE(3), tipo, orden
  PUBLIC reg := 0, flag2 := .F.

  // Checa que existan los indices de grupos y si existen los crea

```

```

IF !FILE ("%cPath\grupo1.ntx") .OR. !FILE ("%cPath\grupo2.ntx") .OR. !FILE ("%cPath\grupo3.ntx")
ok_bases := abre_bases (11) // Base de grupos
IF ok_bases
INDEX ON grupo TO &cPath\grupo1
INDEX ON clavmat TO &cPath\grupo2
INDEX ON clavmac TO &cPath\grupo3
INDEX ON grupo+CLAVMAT TO &cPath\grupo4
DBCLOSEAREA()
ELSE
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
RETURN .F.
ENDIF
ENDIF

ok_bases := abre_bases (10) // Abre la base de grupos
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
RETURN .F.
ENDIF
grupos->(DBGOTOP())
var_ini()
limpia_vars(4)

@ 11,13 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
READ
IF LASTKEY() == K_ESC // Cancelar operacion
grupos->(DBCLOSEAREA())
RETURN .F.
ENDIF
DO WHILE .T.
IF !FILE ("%cPath\alumno1.ntx")
TONE (700)
ALERT ("NO SE HAN DADO DE ALTA ALUMNOS PARA ESTA ESCUELA")
SETCOLOR (cColor2)
DBCLOSEALL()
RETURN
ELSE
ok_bases := abre_bases (7) // Abre la base de alumnos
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
RETURN .F.
ENDIF
ENDIF

tipo := 1
orden := alumnos->(DBSETORDER(3))
alumnos->(DBSEEK (cGrupo))
IF alumnos->(FOUND ())
TONE (650)
ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
grupos->(DBCLOSEAREA())
alumnos->(DBCLOSEAREA())
RETURN .T.
ENDIF
alumnos->(DBCLOSEAREA())

IF !FILE ("%cPath\material.ntx")
TONE (700)
ALERT ("NO SE HAN DADO DE ALTA MATERIAS PARA ESTA ESCUELA")
SETCOLOR (cColor2)
RETURN .F.
ELSE
ok_bases := abre_bases (3) // Abre la base de materias
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
RETURN .F.
ENDIF
ENDIF
ENDIF

```

```

SELECT grupos
DBSEEK (cGrupo)
IF FOUND()
  reg := RECNO()
ENDIF
pie (9)
cColor := SETCOLOR (C_BORRADO)
@ 12, 16 SAY REPLICATE ("■",55) // Borra informacion existente en el nombre
@ 13, 16 SAY REPLICATE ("■",35) // de materias y profesores
SETCOLOR (cColor)

var_ini()
mGrupo := cGrupo
limpia_vars(5)
flag := 0

SET KEY K_F1 TO help
@ 11, 37 GET mClavmat PICTURE "@!" VALID (!EMPTY(mClavmat))
READ
IF LASTKEY () == K_ESC
  cColor := SETCOLOR (C_BORRADO)
  @ 11, 37 SAY REPLICATE ("■",5)
  SETCOLOR (cColor)
  grupos->(DBCLOSEAREA())
  materias->(DBCLOSEAREA())
  SET KEY K_F1 TO
  RETURN .T.
ENDIF
SELECT materias
DBSEEK (mClavmat)
IF !FOUND()
  TONE (550)
  @ 11,37 SAY "NO EXISTE"
  INKEY (1)
  cColor := SETCOLOR (C_BORRADO)
  @ 11,37 SAY REPLICATE ("■",9)
  SETCOLOR (cColor)
  materias->(DBCLOSEAREA())
  LOOP
ELSE
  SELECT grupos
  IF reg > 0
    GOTO reg
    DO WHILE grupos->grupo == mGrupo .AND. grupos->(!EOF())
      IF grupos->clavmat == mClavmat // Ya existe y se va a editar
        igual_var_cam() // Se leen los campos
        SELECT maestros
        DBSEEK (mClavmae)
        @ 13, 16 GET maestros->nommae PICTURE "@!"
        pie (1)
        horario()
        CLEAR GETS
        tipo := 2
        pie (6)
        SET KEY K_F2 TO grupo_baja()
        EXIT
      ELSE
        SET KEY K_F2 TO
        DBSKIP()
      ENDIF
    ENDDO
  ENDIF
  SELECT materias
  @ 12, 16 GET materias->nommat PICTURE "@!"
  CLEAR GETS

  ENDIF
  DO WHILE .T.
    @ 11, 65 GET mClavmae PICTURE "@!" VALID (!EMPTY(mClavmae))
    READ

    SET KEY K_F1 TO

```

```

IF LASTKEY () == K_ESC .OR. LASTKEY () == K_F2 .OR. flag2 == .T.
  mClavmae := SPACE(5)
  flag2 := .F.
  flag := 1
  EXIT
ENDIF
SELECT maestros
DBSEEK (mClavmae)
IF !FOUND()
  TONE (550)
  @ 11,65 SAY "NO EXISTE"
  INKEY (1)
  cColor := SETCOLOR (C_BORRADO)
  @ 11,65 SAY REPLICATE ("█",9)
  SETCOLOR (cColor)
  LOOP
ELSE
  SELECT maestros
  @ 13, 16 GET maestros->nommae PICTURE "@!"
  CLEAR GETS
  EXIT
ENDIF
ENDDO
IF flag == 1
  flag := 0
  materias->(DBCLOSEAREA())
  LOOP
ENDIF
SELECT grupos
color := SETCOLOR ("GR+/B")
@ 16,65 SAY "( Junes"
@ 17,65 SAY "( artes"
@ 18,65 SAY "m( Jercules"
@ 19,65 SAY "( Jueves"
@ 20,65 SAY "( Jiermes"
SETCOLOR (color)
color := SETCOLOR ("R+/B")
@ 16,66 SAY "L"
@ 17,66 SAY "M"
@ 18,67 SAY "I"
@ 19,66 SAY "J"
@ 20,66 SAY "V"
SETCOLOR (color)
pie (1)
horario()
READ
color := SETCOLOR (C_BORRADO)
@ 16,65 SAY REPLICATE ("█",11)
@ 17,65 SAY REPLICATE ("█",11)
@ 18,65 SAY REPLICATE ("█",11)
@ 19,65 SAY REPLICATE ("█",11)
@ 20,65 SAY REPLICATE ("█",11)
SETCOLOR (color)

IF LASTKEY () == K_ESC .OR. LASTKEY () == K_F2 .OR. flag2 == .T.
  flag2 := .F.
  mClavmae := SPACE(5)
  materias->(DBCLOSEAREA())
  LOOP
ENDIF
@ 20,5 SAY "Son correctos los datos (S/N) : " GET correcto PICTURE "!A" VALID (correcto$"SN")
READ
cColor := SETCOLOR (C_BORRADO)
@ 20,5 SAY REPLICATE ("█",50)
SETCOLOR (cColor)
IF LASTKEY () == K_ESC .OR. LASTKEY () == 78 .OR. LASTKEY () == 110
  materias->(DBCLOSEAREA())
  LOOP
ENDIF
IF correcto == "S"
  IF tipo == 1
    ok_almacen := grabar(1)
  
```



```

materias->(DBCLOSEAREA())
IF !ok_almacen
    RETURN .F.
ENDIF
ELSEIF tipo == 2
    flag := 0
    flag := busca_grupo () // checa si aun existe el grupo
    IF flag == 1
        IF lockr(30)
            ok_almacen := grabar(2)
            DBUNLOCK ()
            materias->(DBCLOSEAREA())
            IF !ok_almacen
                RETURN .F.
            ENDIF
        ELSE
            ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
            materias->(DBCLOSEAREA())
            grupos->(DBCLOSEAREA())
            RETURN .T.
        ENDIF
    ELSE
        ok_almacen := grabar(1)
        materias->(DBCLOSEAREA())
        IF !ok_almacen
            RETURN .F.
        ENDIF
    ENDIF
ENDIF
RETURN .F.

```

* Función que permite dar de baja grupos

```

FUNCTION grupo_baja()
    LOCAL ok := "N", anterior:= ALLAS(), flag := 0

    CLEAR GETS

    // Se abre la base de asigna para comprobar si la materia que se
    // pretende dar de baja no a sido asignada ya; de ser así; no se
    // permitira dar de baja la materia, hasta que se "desasigne"

    IF !FILE ("%cPath\asigna1.ntx").OR. !FILE ("%cPath\asigna2.ntx")
        ok_bases := abre_bases (16) // Abre la base de materias asignadas
        IF !ok_bases
            TONE (750)
            ALERT ("NO SE PUEDE ABRIR LA BASE DE ASIGNACION")
            RETURN
        ENDIF
        INDEX ON grupo TO asigna1
        INDEX ON clavmat TO asigna2
        DBCLOSEAREA ()
    ENDIF
    ok_bases := abre_bases (17) // Abre la base de asigna
    IF !ok_bases
        TONE (750)
        ALERT ("NO SE PUEDE ABRIR LA BASE DE ASIGNACION")
        RETURN
    ENDIF

    asigna->(DBSEEK (mgrupo))
    DO WHILE asigna->grupo == mGrupo .AND. asigna->(!EOF())
        IF asigna->clavmat == mClavmat // Ya existe y se va a editar
            TONE (650)
            ALERT ("No se puede dar de baja, porque ya a sido asignada. Elimine la asignacion y reintente la operacin")
            asigna->(DBCLOSEAREA())
            TONE (550)
            cColor2 := SETCOLOR ("R/B+")
            @ 20,5 SAY "La Asignacin NO a sido dado de baja"
            SETCOLOR (cColor2)
            inkey (2)

```

```

cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
RETURN
ELSE
SET KEY K_F2 TO
DBSKIP()
ENDIF
ENDDO
asigna->(DBCLOSEAREA())

TONE (600)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro en DARLO DE BAJA :)"
SETCOLOR (cColor2)
@ 20,37 GET ok PICTURE "!A" VALID (ok$"SN")
READ
IF ok == "N".OR. LASTKEY () == K_ESC
TONE (550)
cColor2 := SETCOLOR ("R/B+")
@ 20,5 SAY "La Asignación NO a sido dado de baja"
SETCOLOR (cColor2)
inkey (2)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
RETURN
ELSEIF ok == "S"
flag := busca_grupo () // checa si aun existe
IF flag == 1
ok:="N"
IF lockr (30)
DELETE
DBUNLOCK ()
flag2 := .T.
TONE (550)
cColor2 := SETCOLOR ("W+/R")
@ 20,5 SAY "La asignación a sido dada de BAJA"
SETCOLOR (cColor2)
inkey (1)
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
GOTO TOP
REG := 0
SELECT &anterior
RETURN
ELSE
cColor := SETCOLOR (C_BORRADO)
@ 20,5 SAY REPLICATE ("■",50)
SETCOLOR (cColor)
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
GOTO TOP
SELECT &anterior
RETURN
ENDIF
ELSE
ALERT ("LA ASIGNACION ACABA DE SER DADA DE BAJA EN ALGUNA OTRA ESTACION DE LA RED")
cColor := SETCOLOR (C_BORRADO)
@ 20, 5 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
ENDIF
ENDIF
GOTO TOP
SELECT &anterior
RETURN

```

FUNCTION horario()



```

@ 16, 10 GET mHora1 PICTURE "@R 99:99"
@ 16, 20 GET mHora2 PICTURE "@R 99:99"
@ 16, 35 GET mSalon1 PICTURE "@"
@ 16, 50 GET mEdifi1 PICTURE "@"
@ 16, 60 GET mDia1 PICTURE "@R@! A" VALID (mDia1$"LMIJV ")

@ 17, 10 GET mHora3 PICTURE "@R 99:99"
@ 17, 20 GET mHora4 PICTURE "@R 99:99"
@ 17, 35 GET mSalon2 PICTURE "@"
@ 17, 50 GET mEdifi2 PICTURE "@"
@ 17, 60 GET mDia2 PICTURE "@R@! A" VALID (mDia1$"LMIJV ")

@ 18, 10 GET mHora5 PICTURE "@R 99:99"
@ 18, 20 GET mHora6 PICTURE "@R 99:99"
@ 18, 35 GET mSalon3 PICTURE "@"
@ 18, 50 GET mEdifi3 PICTURE "@"
@ 18, 60 GET mDia3 PICTURE "@R@! A" VALID (mDia1$"LMIJV ")

@ 19, 10 GET mHora7 PICTURE "@R 99:99"
@ 19, 20 GET mHora8 PICTURE "@R 99:99"
@ 19, 35 GET mSalon4 PICTURE "@"
@ 19, 50 GET mEdifi4 PICTURE "@"
@ 19, 60 GET mDia4 PICTURE "@R@! A" VALID (mDia1$"LMIJV ")
RETURN

```

```
FUNCTION busca_grupo ()
```

```

LOCAL flag := 0
SELECT grupos
IF reg > 0
  GOTO reg
DO WHILE grupos->grupo == mGrupo .OR. !EOF()
  IF grupos->clavmat == mClavmat // Aun no se a dado de baja
    flag := 1
    EXIT
  ELSE
    DBSKIP()
  ENDF
ENDDO
ENDIF
RETURN flag

```

```
// ***** FUNCIONES DE ASIGNACION DE MATERIAS *****
```

```
FUNCTION alumno_asigna()
```

```

LOCAL nKey, ok_bases, reg, cColor2 := SETCOLOR (), flag3 := .F.
LOCAL flag := 0, pantalla, pantalla2, aux, aux2, cuadro
LOCAL cColor3, arch, pantalla3
PUBLIC regnum := 0, cGrupo := SPACE(3), cAlum := SPACE(7)
PUBLIC cNomAlum, flag2 := .F., asig := {}, calif := {}, faltas := {}

IF !FILE("grupo1.ntx")
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA POR LO TANTO NO PUEDEN REALIZARSE
ASIGNACIONES")
  RETURN
ENDIF
ok_bases := abre_bases (10) // Base de grupos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
  RETURN
ENDIF

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  RETURN
ENDIF

```

```

ok_bases := abre_bases (3) // Base de materias
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  RETURN
ENDIF

pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

limpiamenus (2)
@ 9, 14 SAY REPLICATE ("-",45)
pantalla2 := SAVESCREEN (7,1,9,MAXCOL()-1) // Se resguarda el area de pantalla de captura
DO WHILE .T.
  pic (2)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1," |J-L|",C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (9)
  DO WHILE .T.
    cGrupo := SPACE(3)
    SET CURSOR ON
    @ 11, 20 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
    READ
    SET CURSOR OFF
    IF LASTKEY () == K_ESC
      RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
      SETCOLOR (cColor2)
      @ MAXROW()-1,1 SAY SPACE (78)
      DBCLOSEALL()
      RETURN -1
    ELSE
      grupos->(DBGOTOP ())
      grupos->(DBSEEK (cGrupo))
      IF grupos->(FOUND())
        alumnos->(DBSETORDER(3))
        alumnos->(DBSEEK (cGrupo))
        IF alumnos->(FOUND())
          EXIT
        ELSE
          TONE (800)
          ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
        ENDIF
      ELSE
        TONE (800)
        ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
      ENDIF
    ENDIF
  ENDDO
  pantalla3 := SAVESCREEN (7,1,MAXCOL(),MAXCOL()-1) // Se resguarda el area de pantalla de captura

  // Se busca el alumno especificado
  DO WHILE .T.
    flag2 := .F.

    // Se pide la clave del alumno al que se van a asignar las materias
    cAlum := SPACE(7)
    SET CURSOR ON
    @ 13, 20 GET cAlum PICTURE "@!" VALID (!EMPTY(cAlum))
    READ
    SET CURSOR OFF
    IF LASTKEY () == K_ESC
      EXIT
    ELSE
      alumnos->(DBSETORDER(1))
      alumnos->(DBGOTOP ())
      alumnos->(DBSEEK (cAlum))
      IF alumnos->(FOUND())
        IF alumnos->grupo == cGrupo
          cNomAlum := alumnos->(NomAlum)
        ELSE
          ALERT ("ESTE ALUMNO NO ESTA DADO DE ALTA EN ESTE GRUPO")
          LOOP
        ENDIF
      ELSE

```

```

TONE (350)
cColor := SETCOLOR (C_BORRADO)
@ 13,20 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
@ 13, 20 SAY "No existe"
INKEY (1)
@ 13,20 SAY " "
LOOP
ENDIF
ENDIF
ASIZE (asig,0)
ASIZE (calif,0)
ASIZE (faltas,0)

IF alumnos->(lockr (20))
  arch := nomtemp(cPath,1)
  OPEN FILE AUX &arch TIME 30 EXCLUSIVE ALIAS auxiliar
  INDEX ON clavmat TO &arch

  FOR i = 1 TO 13
    temp1 := "mat" + ALLTRIM(STR (i))
    IF EMPTY (alumnos->(&temp1.))
      EXIT
    ENDIF
    auxiliar->(DBAPPEND())
    REPLACE clavmat WITH alumnos->(&temp1.)
    materias->(DBSEEK (alumnos->(&temp1.)))
    REPLACE nommat WITH materias->nommat
    REPLACE area WITH materias->area
    DO CASE
      CASE cTipo == "L" // Licenciatura
        FOR x = 1 to 2
          temp2 := "P"+ALLTRIM(STR(x)) // Campo de base auxiliar
          temp3 := "P"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE &temp2. WITH alumnos->(&temp3.)
          temp2 := "I"+ALLTRIM(STR(x)) // Campo de base auxiliar
          temp3 := "I"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE &temp2. WITH alumnos->(&temp3.)
        NEXT
        temp2 := "F3" // Campo de base auxiliar
        temp3 := "F"+ALLTRIM(STR(i))+"3" // Campo de base alumnos
        REPLACE &temp2. WITH alumnos->(&temp3.)
        temp2 := "I3" // Campo de base auxiliar
        temp3 := "I"+ALLTRIM(STR(i))+"3" // Campo de base alumnos
        REPLACE &temp2. WITH alumnos->(&temp3.)
      CASE cTipo == "P" // Preparatoria
        FOR x = 1 to 3
          temp2 := "P"+ALLTRIM(STR(x)) // Campo de base auxiliar
          temp3 := "P"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE &temp2. WITH alumnos->(&temp3.)
          temp2 := "I"+ALLTRIM(STR(x)) // Campo de base auxiliar
          temp3 := "I"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE &temp2. WITH alumnos->(&temp3.)
        NEXT
        temp2 := "S" // Campo de base auxiliar
        temp3 := "S"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE &temp2. WITH alumnos->(&temp3.)
        temp2 := "IS" // Campo de base auxiliar
        temp3 := "IS"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE &temp2. WITH alumnos->(&temp3.)
        FOR x = 4 to 6
          temp2 := "P"+ALLTRIM(STR(x)) // Campo de base auxiliar
          temp3 := "P"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE &temp2. WITH alumnos->(&temp3.)
          temp2 := "I"+ALLTRIM(STR(x)) // Campo de base auxiliar
          temp3 := "I"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE &temp2. WITH alumnos->(&temp3.)
        NEXT
        temp2 := "F" // Campo de base auxiliar
        temp3 := "F"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE &temp2. WITH alumnos->(&temp3.)
    END CASE
  END FOR

```

```

temp2 := "IF" // Campo de base auxiliar
temp3 := "IF"+ALLTRIM(STR(i)) // Campo de base alumnos
REPLACE &temp2. WITH alumnos->(&temp3.)

CASE cTipo == "G" // Doctorado
temp2 := "U" // Campo de base auxiliar
temp3 := "U"+ALLTRIM(STR(i)) // Campo de base alumnos
REPLACE &temp2. WITH alumnos->(&temp3.)
temp2 := "I" // Campo de base auxiliar
temp3 := "I"+ALLTRIM(STR(i)) // Campo de base alumnos
REPLACE &temp2. WITH alumnos->(&temp3.)

CASE cTipo == "D" // Diplomado
temp2 := "U" // Campo de base auxiliar
temp3 := "U"+ALLTRIM(STR(i)) // Campo de base alumnos
REPLACE &temp2. WITH alumnos->(&temp3.)
temp2 := "I" // Campo de base auxiliar
temp3 := "I"+ALLTRIM(STR(i)) // Campo de base alumnos
REPLACE &temp2. WITH alumnos->(&temp3.)
ENDCASE

NEXT

auxiliar->(DBCOMMIT())
ELSE
ALERT ("EL REGISTRO ESTA SIENDO OCUPADO POR OTRA ESTACION DE LA RED, NO SE PUEDE REALIZAR LA
OPERACION EN ESTE MOMENTO")
LOOP
ENDIF

pie (7)
cColor3 := SETCOLOR ("N/N")
@ 7, 1 SAY SPACE (78)
SETCOLOR (cColor3)
nCentra := (78-LEN (ALLTRIM(cNomAlum)))/2
cColor3 := SETCOLOR (C_TITULOS)
@ 7, nCentra SAY ALLTRIM(cNomAlum)
SETCOLOR (cColor3)
DISPBOX (8,1,MAXROW()-10,maxcol()-1," |J-L|",C_REYENO) // Se dibuja la ventana de materias existentes
DISPBOX (MAXROW()-10,1,MAXROW()-3,maxcol()-1," |J-L|",C_REYENO) // Se dibuja la ventana de captura

grupos->(DBSEEK (cGrupo))
// Objeto de las materias asignadas
otb2:= TBROWSEDB (MAXROW()-9,2,maxrow()-4,maxcol()-2)
otb2:ADDCOLUMN(TBCOLUMNNEW("Grupo",{||cGrupo}))
otb2:ADDCOLUMN(TBCOLUMNNEW("Materia Asignada",{||auxiliar->nommat}))
otb2:ADDCOLUMN(TBCOLUMNNEW("Area",{||auxiliar->area}))
otb2:COLSEP:="|"
otb2:HEADSEP:="-"
otb2:freeze:=2

// Objeto de las materias existentes
otb:=TBROWSEDB (9,2,maxrow()-11,maxcol()-2)
otb:ADDCOLUMN(TBCOLUMNNEW("Grupo",{||grupos->grupo}))
otb:ADDCOLUMN(TBCOLUMNNEW("Materia Registrada",{||regresa_materia(grupos->clavmat,1,0)})
otb:ADDCOLUMN(TBCOLUMNNEW("Area",{||regresa_materia(grupos->clavmat,4,regnum)})
otb:COLSEP:="|"
otb:HEADSEP:="-"
otb:GOTOPBLOCK={||inicfranja(cGrupo)}
otb:GOBOTOMBLOCK:={||finfranja(cGrupo)}
otb:SKIPBLOCK:={||nParamover|muevereg(nParamover,cGrupo)}
otb:freeze:=2

oObj:=otb2
SELECT auxiliar
GOTO TOP

DO WHILE !oObj:STABILIZE()
ENDDO
oObj:DEHILITE()
oObj:RIGHT()

SELECT grupos

```

```

oObj:=otb
oObj:RIGHT()
DO WHILE .T.
  DO WHILE !oObj:STABILIZE()
  ENDDO

nKey := reempinkey (0)

SET CURSOR OFF

oObj:= otb2
SELECT auxiliar
DBGOTOP()
DO WHILE !oObj:STABILIZE()
ENDDO
oObj:REFRESHALL()
SELECT grupos
oObj:= otb

DO CASE
CASE nKey == K_F6
  reg := RECNO()
  oObj:DEHILITE()
  materias_asignadas(.F.)
  oObj:HILITE()
  GOTO reg
CASE nKey == K_ENTER

  auxiliar->(DBGOTOP())
  IF auxiliar->(RECCOUNT()) == 13
    TONE (750)
    ALERT ("SE HAN INTENTADO ASIGNAR MAS MATERIAS DE LAS PERMITIDAS PARA UN SOLO ALUMNO, EL
MAXIMO NUMERO DE MATERIAS ES DE 13 ")
  ELSE
    DO WHILE .T.
      IF auxiliar->clavmat # clavmat .AND. auxiliar->(EOF())
        auxiliar_guarda()
        flag2 := .T.
        EXIT
      ELSEIF auxiliar->clavmat == clavmat
        EXIT
      ENDIF
      auxiliar->(DBSKIP())
    ENDDO
  ENDDIF
  oObj:= otb2
  SELECT auxiliar
  DBGOTOP ()
  DO WHILE !oObj:STABILIZE()
  ENDDO
  oObj:REFRESHALL()
  SELECT grupos
  oObj:= otb

CASE nKey == K_ESC
  IF flag2
    SET ESCAPE OFF
    auxiliar->(DBGOTOP())
    REPEAT
      AADD (asig , auxiliar->clavmat)
      DO CASE
        CASE cTipo == "L" // Licenciatura
          FOR i = 1 TO 2
            temp1 := "P" + ALLTRIM(STR(i))
            temp2 := "I" + ALLTRIM(STR(i))
            AADD (calif , auxiliar->(&temp1.))
            AADD (faltas , auxiliar->(&temp2.))
          NEXT
            temp1 := "F" + ALLTRIM(STR(i))
            temp2 := "I" + ALLTRIM(STR(i))
            AADD (calif , auxiliar->(&temp1.))
            AADD (faltas , auxiliar->(&temp2.))

```

```

    auxiliar->(DBSKIP())
CASE cTipo == "P" // Preparatoria
  FOR i = 1 TO 3
    temp1 := "P" + ALLTRIM(STR(i))
    temp2 := "I" + ALLTRIM(STR(i))
    AADD (calif, auxiliar->(&temp1.))
    AADD (faltas, auxiliar->(&temp2.))
  NEXT
  temp1 := "S"
  temp2 := "IS"
  AADD (calif, auxiliar->(&temp1.))
  AADD (faltas, auxiliar->(&temp2.))
  FOR i = 4 TO 6
    temp1 := "P" + ALLTRIM(STR(i))
    temp2 := "I" + ALLTRIM(STR(i))
    AADD (calif, auxiliar->(&temp1.))
    AADD (faltas, auxiliar->(&temp2.))
  NEXT
  temp1 := "F"
  temp2 := "IF"
  AADD (calif, auxiliar->(&temp1.))
  AADD (faltas, auxiliar->(&temp2.))

  auxiliar->(DBSKIP())
CASE cTipo == "G" // Postgrado
  temp1 := "U"
  temp2 := "I"
  AADD (calif, auxiliar->(&temp1.))
  AADD (faltas, auxiliar->(&temp2.))
  auxiliar->(DBSKIP())

CASE cTipo == "D" // Diplomado
  temp1 := "U"
  temp2 := "I"
  AADD (calif, auxiliar->(&temp1.))
  AADD (faltas, auxiliar->(&temp2.))
  auxiliar->(DBSKIP())

END CASE
UNTIL auxiliar->(EOF())

FOR i = IIF(LEN(asig)= 0, LEN(asig)+1,LEN(asig)) TO 12
  AADD (asig, SPACE(5))
NEXT

DO CASE
CASE cTipo == "L" // Licenciatura
  FOR i = IIF(LEN(calif)= 0, LEN(calif)+1,LEN(calif)) TO 39
    AADD (calif, 0) // El 39 resulta de 13 (materias) * 3 (evaluaciones)
    AADD (faltas,0)
  NEXT
CASE cTipo == "P" // Preparatoria
  FOR i = IIF(LEN(calif)= 0, LEN(calif)+1,LEN(calif)) TO 104
    AADD (calif, 0) // El 104 resulta de 13 (materias) * 8 (evaluaciones)
    AADD (faltas,0)
  NEXT
CASE cTipo == "G" // Postgrado
  FOR i = IIF(LEN(calif)= 0, LEN(calif)+1,LEN(calif)) TO 13
    AADD (calif, 0) // El 13 resulta de 13 (materias) * 1 (evaluaciones)
    AADD (faltas,0)
  NEXT
CASE cTipo == "D" // Diplomado
  FOR i = IIF(LEN(calif)= 0, LEN(calif)+1,LEN(calif)) TO 13
    AADD (calif, 0) // El 13 resulta de 13 (materias) * 1 (evaluaciones)
    AADD (faltas,0)
  NEXT

END CASE

cuadro := SAVESCREEN (9,18,16,63)
DISPBOX (9,18,15,62,"┌──┴──┐",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")

```



```

@ 11,21 SAY "   ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)
color := SETCOLOR ("GR+*/N")
@ 12,23 SAY "SE ESTA ACTUALIZANDO LA INFORMACION"
SETCOLOR (color)

z:= 0 // Contador de los arreglos
alumnos->(DBSEEK(cAlum))
IF alumnos->(FOUND())
  // Se restaura la informaci3n de los arreglos a la base de Alumnos
  FOR i = 1 TO LEN(asig)
    aux2 := "mat"+ALLTRIM(STR(i))
    REPLACE alumnos->&aux2 WITH asig[i]
    DO CASE
      CASE cTipo == "L" // Licenciatura
        z++
        FOR x = 1 to 2
          temp2 := "P"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE alumnos->&temp2. WITH calif[z]
          temp2 := "I"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE alumnos->&temp2. WITH faltas[z]
        z++
        NEXT
        temp2 := "F"+ALLTRIM(STR(i))+*3" // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH calif[z]
        temp2 := "I"+ALLTRIM(STR(i))+*3" // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH faltas[z]
      CASE cTipo == "P" // Preparatoria
        z++
        FOR x = 1 to 3
          temp2 := "P"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE alumnos->&temp2. WITH calif[z]
          temp2 := "I"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE alumnos->&temp2. WITH faltas[z]
        z++
        NEXT
        temp2 := "S"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH calif[z]
        temp2 := "IS"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH faltas[z]
        z++
        FOR x = 4 to 6
          temp2 := "P"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE alumnos->&temp2. WITH calif[z]
          temp2 := "I"+ALLTRIM(STR(i))+ALLTRIM(STR(x)) // Campo de base alumnos
          REPLACE alumnos->&temp2. WITH faltas[z]
        z++
        NEXT
        temp2 := "F"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH calif[z]
        temp2 := "IF"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH faltas[z]

      CASE cTipo == "G" // Postgrado
        temp2 := "U"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH calif[i]
        temp2 := "I"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH faltas[i]

      CASE cTipo == "D" // Diplomado
        temp2 := "U"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH calif[i]
        temp2 := "I"+ALLTRIM(STR(i)) // Campo de base alumnos
        REPLACE alumnos->&temp2. WITH faltas[i]

    ENDCASE

    NEXT
    alumnos->(DBCOMMIT())
  ENDF
  RESTSCREEN (9,18,16,63,cuadro)
  SET ESCAPE ON
ENDIF

```



```

READ
SET CURSOR OFF
IF LASTKEY () == K_ESC
  RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
  SETCOLOR (cColor2)
  @ MAXROW()-1,1 SAY SPACE (78)
  DBCLOSEALL()
  RETURN -1
ELSE
  ok_bases := abre_bases (7) // Base de alumnos
  IF lok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS, NO SE PUEDE CONTINUAR EJECUTANDO EL SISTEMA")
    SETCOLOR ("W/N")
    CLS
    QUIT
  ENDIF
  grupos->(DBGOTOP ())
  grupos->(DBSEEK (cGrupo))
  IF grupos->(FOUND())
    alumnos->(DBSETORDER(3))
    alumnos->(DBSEEK (cGrupo))
    IF alumnos->(FOUND())
      TONE (800)
      ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
      alumnos->(DBCLOSEAREA())
    ENDIF
  ELSE
    TONE (800)
    ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
    alumnos->(DBCLOSEAREA())
  LOOP
ENDIF
ENDIF

cuadro := SAVESCREEN (9,18,16,63)
DISPBOX (9,18,15,62," | | - | ",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")
@ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)
color := SETCOLOR ("GR+/N")
@ 12,23 SAY "SE ESTA COMPROBANDO LA INFORMACION"
SETCOLOR (color)

DO CASE
CASE cTipo == "L"
  REPEAT
    FOR i = 1 to 1
      FOR x = 1 to 2
        temp1 := "P" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
        temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
        flag5 := imposible (temp1,temp2)
        IF flag5
          EXIT
        ENDIF
      NEXT
    IF flag5
      EXIT
    ENDIF

    temp1 := "F" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
    temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
    flag5 := imposible (temp1,temp2)
    IF flag5
      EXIT
    ENDIF

    NEXT
  IF flag5
    EXIT
  ENDIF
  alumnos->(DBSKIP())
UNTIL alumnos->(EOF()) .OR. alumnos->grupo # cGrupo

```

```

CASE cTipo == "P" // Preparatoria
REPEAT
  FOR i = 1 to 13
    FOR x = 1 to 2
      temp1 := "P" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
      temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
      flag5 := imposible (temp1,temp2)
      IF flag5
        EXIT
      ENDIF
    NEXT
    IF flag5
      EXIT
    ENDIF
    temp1 := "S" + ALLTRIM(STR(i))
    temp2 := "IS" + ALLTRIM(STR(i))
    flag5 := imposible (temp1,temp2)
    IF flag5
      EXIT
    ENDIF
    FOR x = 4 to 6
      temp1 := "P" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
      temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
      flag5 := imposible (temp1,temp2)
      IF flag5
        EXIT
      ENDIF
    NEXT
    IF flag5
      EXIT
    ENDIF
    temp1 := "F" + ALLTRIM(STR(i))
    temp2 := "IF" + ALLTRIM(STR(i))
    flag5 := imposible (temp1,temp2)
    IF flag5
      EXIT
    ENDIF
  NEXT
  IF flag5
    EXIT
  ENDIF
  alumnos->(DBSKIP())
UNTIL alumnos->(EOF()) .OR. alumnos->grupo # cGrupo
CASE cTipo == "G" .OR. cTipo == "D"
REPEAT
  FOR i = 1 to 13
    temp1 := "U" + ALLTRIM(STR(i))
    temp2 := "I" + ALLTRIM(STR(i))
    flag5 := imposible (temp1,temp2)
    IF flag5
      EXIT
    ENDIF
  NEXT
  IF flag5
    EXIT
  ENDIF
  alumnos->(DBSKIP())
UNTIL alumnos->(EOF()) .OR. alumnos->grupo # cGrupo

END CASE
IF flag5
  flag5 := .F.
  alumnos->(DBCLOSEAREA())
  RESTSCREEN (9,18,16,63,cuadro)
LOOP
  ELSE
    EXIT
  ENDIF
ENDDO

RESTSCREEN (9,18,16,63,cuadro)

```

```

alumnos->(DBCLOSEAREA())
IF !FILE ("%&cPath\asigna1.ntx").OR. !FILE ("%&cPath\asigna2.ntx")
  ok_bases := abre_bases (16) // Abre la base de materias asignadas
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ASIGNACION")
    RETURN
  ENDF
  INDEX ON grupo TO asigna1
  INDEX ON clavmat TO asigna2
  DBCLOSEAREA ()
ENDIF
ok_bases := abre_bases (17) // Abre la base de asigna
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ASIGNACION")
  RETURN
ENDIF

pie (7)
DISPBOX (7,1,MAXROW()-10,maxcol()-1," |J-L| █",C_REYENO) // Se dibuja la ventana de materias existentes
DISPBOX (MAXROW()-10,1,MAXROW()-3,maxcol()-1," |J-L| █",C_REYENO) // Se dibuja la ventana de captura

// Objeto de las materias asignadas
otb2:= TBROWSEDB (MAXROW()-9,2,maxrow()-4,maxcol()-2)
otb2:ADDCOLUMN(TBCOLUMNNEW("Grupo",{||cGrupo}))
otb2:ADDCOLUMN(TBCOLUMNNEW("Materia Asignada",{||regresa_materia(asigna->clavmat,2,regnum)}))
otb2:ADDCOLUMN(TBCOLUMNNEW("Area",{||regresa_materia(asigna->clavmat,3,regnum)}))
otb2:COLSEP:="|"
otb2:HEADSEP:="-"
otb2:GOTOPBLOCK={|| inicfranja(cGrupo)}
otb2:GOBOTOMBLOCK:={|| finfranja(cGrupo)}
otb2:SKIPBLOCK:={|| nParamover| muevereg(nParamover,cGrupo)}
otb2:freeze:=2

// Objeto de las materias existentes
otb:=TBROWSEDB (8,2,maxrow()-11,maxcol()-2)
otb:ADDCOLUMN(TBCOLUMNNEW("Grupo",{||grupos->grupo}))
otb:ADDCOLUMN(TBCOLUMNNEW("Materia Registrada",{||regresa_materia(grupos->clavmat,1,0)}))
otb:ADDCOLUMN(TBCOLUMNNEW("Area",{||regresa_materia(grupos->clavmat,4,regnum)}))
otb:COLSEP:="|"
otb:HEADSEP:="-"
otb:GOTOPBLOCK={|| inicfranja(cGrupo)}
otb:GOBOTOMBLOCK:={|| finfranja(cGrupo)}
otb:SKIPBLOCK:={|| nParamover| muevereg(nParamover,cGrupo)}
otb:freeze:=2

oObj:=otb2
SELECT asigna
GOTO TOP
DBSEEK (cGrupo)
DO WHILE !oObj:STABILIZE()
ENDDO
oObj:DEHILITE()
oObj:RIGHT()

SELECT grupos
oObj:=otb
oObj:RIGHT()
DO WHILE .T.
  DO WHILE !oObj:STABILIZE()
  ENDDO

/* El reempinkey con estos parametros es exclusivo para los
  Tbrowses, esto es para que se realice un refresco cada
  10 segundos. Esto es sobre todo para cuando se trabaja
  en ambiente de red. */

SELECT asigna
oObj:= otb2
DBSEEK (cGrupo)

```

```
nKey := reampinkey (0, oObj, T.)
```

```
SET CURSOR OFF
```

```
SELECT asigna
oObj:= otb2
DBSEEK (cGrupo)
DO WHILE !oObj:STABILIZE()
ENDDO
oObj:REFRESHALL()
SELECT grupos
oObj:= otb
```

```
DO CASE
```

```
  CASE nKey == K_F6
    reg := RECNO()
    oObj:DEHILITE()
    materias_asignadas(.T.)
    oObj:HILITE()
    GOTO reg
  CASE nKey == K_ENTER
```

```
  asigna->(DBSEEK(grupos->grupo))
```

```
  IF asigna->(FOUND())
```

```
    z:= 0
```

```
    DO WHILE asigna->grupo == cGrupo .AND. asigna->(EOF())
```

```
      z++
```

```
      asigna->(DBSKIP())
```

```
    ENDDO
```

```
    IF z >= 13
```

```
      TONE (750)
```

```
      ALERT ("SE HAN INTENTADO ASIGNAR MAS MATERIAS DE LAS PERMITIDAS PARA UN SOLO ALUMNO, EL  
MAXIMO NUMERO DE MATERIAS ES DE 13 ")
```

```
    ELSE
```

```
      asigna->(DBGOTOP())
```

```
      asigna->(DBSEEK(grupos->grupo))
```

```
      DO WHILE asigna->grupo == cGrupo .AND. asigna->(EOF())
```

```
        IF asigna->clavmat == grupos->clavmat
```

```
          flag := 0
```

```
          EXIT
```

```
        ELSE
```

```
          flag := 1
```

```
          asigna->(DBSKIP())
```

```
        ENDIF
```

```
      ENDDO
```

```
    ENDIF
```

```
  ELSE
```

```
    flag2 := .T.
```

```
    asigna_guarda()
```

```
    flag := 0
```

```
  ENDIF
```

```
  IF flag == 1
```

```
    flag2 := .T.
```

```
    asigna_guarda()
```

```
    flag := 0
```

```
  ENDIF
```

```
  DBCOMMIT()
```

```
  SELECT asigna
```

```
  oObj:= otb2
```

```
  DBSEEK (cGrupo)
```

```
  DO WHILE !oObj:STABILIZE()
```

```
  ENDDO
```

```
  oObj:REFRESHALL()
```

```
  SELECT grupos
```

```
  oObj:= otb
```

```
  CASE nKey == K_ESC
```

```
  IF flag2
```

```
    SET ESCAPE OFF
```

```
    asigna->(DBSEEK (cGrupo))
```

```
    IF asigna->(FOUND())
```

```
      REPEAT
```

```
        AADD (asig, asigna->clavmat)
```

```

    asigna->(DBSKIP())
  UNTIL asigna->grupo # cGrupo .OR. asigna->(EOF())
ENDIF

// Se ordena el arreglo que contiene las claves de las materias
ASORT (asig)

FOR i = IIF(LEN(asig)= 0, LEN(asig)+1,LEN(asig)) TO 12
  AADD (asig, SPACE(5))
NEXT
materias->(DBCLOSEAREA())

cuadro := SAVESCREEN (9,18,16,63)
DISPBOX (9,18,15,62," | | | | | " ,C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")
@ 11,21 SAY "  ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)
color := SETCOLOR ("GR+*/N")
@ 12,23 SAY "SE ESTA ACTUALIZANDO LA INFORMACION"
SETCOLOR (color)

ok_bases := abre_bases (7) //Base de alumnos
IF !ok_bases
  TONE (660)
  - ALERT ("!!! ERROR FATAL !!! LA ASIGNACIÓN FALLO, CONSULTE INMEDIATAMENTE CON LOS
DISEÑADORES DEL SISTEMA")
  RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
  SETCOLOR (cColor2)
  @ MAXROW()-1,1 SAY SPACE (78)
  DBCLOSEALL()
  RETURN -1
ENDIF
alumnos->(DBSETORDER (3))
alumnos->(DBSEEK(cGrupo))
IF alumnos->(FOUND())
  tiempo := SECONDS()
  REPEAT
    IF lockr (30,1)
      FOR i = 1 TO LEN(asig)
        aux:= asig [i]
        aux2 := "mat"+ALLTRIM(STR(i))
        REPLACE alumnos->&aux2 WITH asig[i]
        IF SECONDS () - tiempo >= 20
          SETCOLOR (color)
          color := SETCOLOR ("G+/N")
          @ 14,27 SAY "NO DESESPERE YA CASI TERMINO"
          SETCOLOR (color)
        ENDIF
      NEXT
    ELSE
      TONE (660)
      ALERT ("!!! ERROR FATAL !!! FALLO LA ASIGNACIÓN, CONSULTE INMEDIATAMENTE CON LOS
DISEÑADORES DEL SISTEMA")
      RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
      SETCOLOR (cColor2)
      @ MAXROW()-1,1 SAY SPACE (78)
      DBCLOSEALL()
      RETURN -1
    ENDIF
  alumnos->(DBCOMMIT())
  alumnos->(DBSKIP())
  UNTIL alumnos->(EOF()) .OR. alumnos->grupo # cGrupo
ENDIF
RESTSCREEN (9,18,16,63,cuadro)
alumnos->(DBCLOSEAREA())
SET ESCAPE ON

ENDIF
IF !flag2
  materias->(DBCLOSEAREA())
ENDIF
asigna->(DBCLOSEAREA())
RESTSCREEN (7,1,9,MAXCOL()-1,pantalla2) // Se resguarda el area de pantalla de captura

```

```

EXIT
CASE nKey==K_UP ;otb:UP()
CASE nKey==K_DOWN ;otb:DOWN()
CASE nKey==K_LEFT ;otb:LEFT()
CASE nKey==K_CTRL_LEFT ;otb:PANLEFT()
CASE nKey==K_RIGHT ;otb:RIGHT()
CASE nKey==K_CTRL_RIGHT ;otb:PANRIGHT()
CASE nKey==K_PGUP ;otb:PAGEUP()
CASE nKey==K_CTRL_PGUP ;otb:GOTOP()
CASE nKey==K_PGDN ;otb:PAGEDOWN()
CASE nKey==K_CTRL_PGDN ;otb:GOBOTTOM()
CASE nKey==K_HOME ;otb:HOME()
CASE nKey==K_CTRL_HOME ;otb:PANHOME()
CASE nKey==K_END ;otb:END()
CASE nKey==K_CTRL_END ;otb:PANEND()
OTHERWISE
TONE(125); TONE(300)
ENDCASE
ENDDO
RETURN NIL
    
```

// Función que es llamada al presionarse la tecla F6, sirve para navegar dentro
// de la ventana que contiene las materias ya asignadas previamente, tanto a
// un grupo como a un alumno en especial (para esto es el parametro).

FUNCTION materias_asignadas (tipo)

```

LOCAL nKey
oObj:= otb2
IF tipo
SELECT asigna
DBSEEK (oGrupo)
ELSE
SELECT auxiliar
DBGOTOP()
ENDIF
pie (8)
DO WHILE .T.
DO WHILE !oObj:STABILIZE()
ENDDO
    
```

/* El reempinkey con estos parametros es exclusivo para los
Tbrowsers, esto es para que se realice un refresco cada
10 segundos. Esto es sobre todo para cuando se trabaja
en ambiente de red. */

nKey := reempinkey (0, oObj)

SET CURSOR OFF

```

DO CASE
CASE nKey == K_F6
SELECT grupos
pie (7)
oObj:DEHILITE()
oObj:= otb
oObj:HILITE()
RETURN
CASE nKey==K_UP ;oObj:UP()
CASE nKey==K_DOWN ;oObj:DOWN()
CASE nKey==K_LEFT ;oObj:LEFT()
CASE nKey==K_CTRL_LEFT ;oObj:PANLEFT()
CASE nKey==K_RIGHT ;oObj:RIGHT()
CASE nKey==K_CTRL_RIGHT ;oObj:PANRIGHT()
CASE nKey==K_PGUP ;oObj:PAGEUP()
CASE nKey==K_CTRL_PGUP ;oObj:GOTOP()
CASE nKey==K_PGDN ;oObj:PAGEDOWN()
CASE nKey==K_CTRL_PGDN ;oObj:GOBOTTOM()
CASE nKey==K_HOME ;oObj:HOME()
CASE nKey==K_CTRL_HOME ;oObj:PANHOME()
CASE nKey==K_END ;oObj:END()
    
```



```

CASE nKey==K_CTRL_END ;oObj: PANEND()
CASE nKey == K_DEL
  IF lockr (30)
    DELETE
    DBUNLOCK()
    DBCOMMIT()
    IF tipo
      DBSEEK (oGrupo)
    ELSE
      DBGOTOP ()
    ENDIF
    DO WHILE !oObj:STABILIZE()
    ENDDO
    oObj:REFRESHALL()
    flag2 := .T.
  ELSE
    TONE (800)
    ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
  ENDIF
  OTHERWISE
    TONE(125) ; TONE(300)
  ENDCASE
ENDDO
RETURN

```

```

// Funcion que guarda en disco las materias asignadas
FUNCTION asigna_guarda()
asigna--(DBAPPEND())
IF !NETERR ()
  IF lockr (30)
    REPLACE asigna->grupo WITH grupos->grupo
    REPLACE asigna->clavmat WITH grupos->clavmat
    asigna--(DBCOMMIT())
    asigna--(DBUNLOCK())
  ELSE
    TONE (800)
    ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
    RETURN -1
  ENDIF
ELSE
  TONE (800)
  ALERT ("NO SE PUEDE REALIZAR LA OPERACION, ESPERE UN MOMENTO E INTENTE NUEVAMENTE")
ENDIF
RETURN

```

```

// Funcion que guarda en el archivo temporal las materias asignadas POR ALUMNO
FUNCTION auxiliar_guarda()
auxiliar--(DBAPPEND())
materias--(DBSEEK (grupos->clavmat))
IF !NETERR ()
  IF lockr (30)
    REPLACE auxiliar->clavmat WITH grupos->clavmat
    REPLACE auxiliar->nommat WITH materias->nommat
    auxiliar--(DBCOMMIT())
    auxiliar--(DBUNLOCK())
  ELSE
    TONE (800)
    ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
    RETURN -1
  ENDIF
ELSE
  TONE (800)
  ALERT ("NO SE PUEDE REALIZAR LA OPERACION, ESPERE UN MOMENTO E INTENTE NUEVAMENTE")
ENDIF
RETURN

```

/* Función que determina si los campos de calificaciones y faltas estan vacios. Esto se hace para que no se pueda modificar la asignación de materias por grupos. */
 FUNCTION imposible (temp1, temp2)

```
IF alumnos->(&temp1.) # 0 .OR. alumnos->(&temp2.) # 0
  TONE (800)
  TONE (650)
  ALERT ("YA SE HAN ASIGNADO CALIFICACIONES EN ESTE GRUPO, POR LO TANTO ESTA OPERACION NO SE PUEDE
REALIZAR DE ESTA FORMA, ES NECESARIO REALIZARLA ALUMNO POR ALUMNO")
  RETURN .T.
ELSE
  RETURN .F.
ENDIF
RETURN .F.
```

REPORTES.PRG

```

/* Programa   : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
Lenguaje     : CLIPPER 5.2
Enlazador    : RTLINK 3.14B
*/

#xcommand REPEAT => do while .t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end

#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla
#include "caos.ch"
#include "memoedit.ch"

/***** FUNCIONES PARA LA IMPRESION DE LISTAS *****/

// Función que imprime las listas por Alumno
FUNCTION mats_listas ()
  LOCAL nPmActive, cPuerto, aPuertos, aCapCfg, npm := Nmlmp;
  pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1);
  cColor2, cNomAlum, cSemestre,ok := "N";
  aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F., cuadro
  PRIVATE impresora, mClavMat := SPACE(5),nHoja := 0,materia, maestro,cGrupo
/* Vars. que indican la posición donde se desplegara el nombre de la
impresora si est es cambiada */
  PUBLIC x := 14, y := 32

  IF !FILE("grupo1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
    RETURN
  ENDIF
  ok_bases := abre_bases (10) // Abre la base de grupos
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
    RETURN
  ENDIF

  IF !FILE("alumno1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
    RETURN
  ENDIF
  ok_bases := abre_bases (7) // Base de alumnos compartida
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
    RETURN
  ENDIF

  ok_bases := abre_bases (19) // Base de impresoras
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
    RETURN
  ENDIF

  ok_bases := abre_bases (20) // Base de codigos de escapes
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
    RETURN
  ENDIF

```

```

impresora := CsPmName2(nPm) // Inicializa la impresora
impres->(DBCLOSEAREA())

limpiamenus (2)
@ 9, 15 SAY REPLICATE (" ",60)
cColor2 := SETCOLOR(C_CAP)
DO WHILE .T.
  pic (13)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1," | | | | |",C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (13)
DO WHILE .T.

  SET KEY K_F2 TO cambia impresora ()
  @@ x,y SAY impresora COLOR C_CONFIG4

  cGrupo : SPAC(3)
  SET CURSOR ON
  @ 11, 13 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
  READ
  IF LASTKEY () == K_ESC
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    @ MAXROW()-1,1 SAY SPACE (78)
    DBCLOSEALL()
    RETURN -1
  ELSE
    grupos->(DBGOTOP ())
    grupos->(DBSEEK (cGrupo))
    IF grupos->(FOUND())
      alumnos->(DBSETORDER(3))
      alumnos->(DBSEEK (cGrupo))
      IF alumnos->(FOUND())
        EXIT
      ELSE
        TONE (800)
        ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
        LOOP
      ENDIF
    ELSE
      TONE (800)
      ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
      LOOP
    ENDIF
  ENDIF
ENDIF
ENDDO

DO WHILE .T.
  ok_bases := abre_bases (3) // Abre la base de materias
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
    RETURN
  ENDIF

  // Se limpia la pantalla de informaci3n anterior
  cColor := SETCOLOR (C_BORRADO)
  @ 11,37 SAY REPLICATE (" ",9)
  @ 12,15 SAY REPLICATE (" ",60)
  @ 20,5 SAY REPLICATE (" ",30)
  SETCOLOR (cColor)

  SET CURSOR ON
  pic (9)
  SET KEY K_F1 TO help
  SET KEY K_F2 TO

  mClavmat := SPACE (5)
  @ 11, 37 GET mClavmat PICTURE "@!" VALID (!EMPTY(mClavmat))
  READ
  SET KEY K_F1 TO
  IF LASTKEY () == K_ESC
    cColor := SETCOLOR (C_BORRADO)

```

```

(@ 11,37 SAY REPLICATE ("■",9)
SETCOLOR (cColor)
materias->(DBCLOSEAREA())
EXIT
ENDIF
materias->(DBSEEK (mClavmat))
IF materias->(FOUND())
  TONE: (550)
  (@ 11,37 SAY "NO EXISTE"
  INKEY (1)
  cColor := SETCOLOR (C_BORRADO)
  (@ 11,37 SAY REPLICATE ("■",9)
  SETCOLOR (cColor)
  materias->(DBCLOSEAREA())
  LOOP
ELSE
  (@ 12, 15 GET materias->nommat PICTURE "@!"
  CLEAR GETS
  materias->(DBCLOSEAREA())
  SET KEY K_F2 TO cambia_impresora ()
ENDIF

pie (13)
cColor := SETCOLOR ("R/B+")
(@ 20,5 SAY "Esta seguro de IMPRIMIR : "
SETCOLOR (cColor)
(@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN")
READ
IF ok == "N" .OR. LASTKEY () == K_ESC
  LOOP
ENDIF
ok : "N"
SET KEY K_F1 TO

pie (1)
SET CURSOR OFF
DO WHILE .T.
  IF ISPRINTER()
    cuadro := SAVESCREEN (9,18,16,63)
    DISPBOX (9,18,15,62,"┌──┴──┐",C_REYENO3) // Se dibuja la ventana de captura
    color := SETCOLOR ("R+/N")
    (@ 11,21 SAY "  ESPERE UN MOMENTO POR FAVOR"
    SETCOLOR (color)
    color := SETCOLOR ("GR+*/N")
    (@ 12,25 SAY "SE ESTAN IMPRIMIENDO LAS LISTAS"
    SETCOLOR (color)
    color := SETCOLOR ("GR+/N")
    (@ 14,25 SAY "PRESIONE " COLOR C_CONFIG2
    (@ 14,34 SAY " <ESC> " COLOR C_CONFIG4
    (@ 14,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
    SETCOLOR (color)

    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET ( _SET_DEVICE, "PRINTER")

    // Se vuelve a checar que la materia este asignada al grupo especificado
    grupos->(DBSETORDER (2))
    grupos->(DBSEEK(mClavMat))
    grupos->(DBSETORDER (1))
    IF grupos->(FOUND())
      // Se checa que halla alumnos inscritos en el grupo especificado
      alumnos->(DBSEEK (cGrupo))
      IF alumnos->(FOUND())
        ok bases := abre bases (5) // Abre la base de maestros
        IF !ok bases
          TONE: (750)

          ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS")
          RETURN
        ENDIF
        maestros->(DBSEEK (grupos->clavmae))
        maestro := maestros->nommae
        maestros->(DBCLOSEAREA())

```

```

ok_bases := abre_bases (3) // Abre la base de materias
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  RETURN
ENDIF
materias->(DBSEEK (grupos->clavmat))
materia : materias->nommat
materias->(DBCLOSEAREA())
bandera := .F.

/* Se hace una pre-busqueda para saber si la materia
esta asignada a por lo menos un alumno dentro del
grupo, de no ser asi, esta materia no se imprimira */

REPEAT
  FOR i = 1 TO 13
    temp1 := "mat"+ ALLTRIM(STR(i))
    IF alumnos->(&temp1.) == grupos->clavmat
      bandera := .T.
      EXIT
    ENDIF
  NEXT
  IF bandera
    EXIT
  ELSE
    alumnos->(DBSKIP())
  ENDIF
UNTIL alumnos->grupo # cGrupo .OR. alumnos->(EOF())

IF bandera
  bandera := .F.
  alumnos->(DBSETORDER (4)) // Ordenar por nombre del alumno y por grupo
  alumnos->(DBSEEK (cGrupo)) // Posicionar en el primer alumno del grupo
  (@ PROW(), PCOL() SAY CsSendPm3 ("+COMPRIMIDA", m->NmImp)
  nl := 1
  nHoja++
  ISalir := .F.
  ISalir := encabezado()
  IF ISalir
    EXIT
  ENDIF
  REPEAT
    /* Se imprimen unicamente los alumnos que tengan
    asignada la materia actual de trabajo, para saber
    esto se hace una busqueda dentro de las materias
    asignadas a cada alumno */
    bandera := .F.
    FOR i = 1 TO 13
      temp1 := "mat"+ ALLTRIM(STR(i))
      IF alumnos->(&temp1.) == grupos->clavmat .AND. alumnos->inscrito .AND. EMPTY (alumnos->baja)
        bandera := .T.
        EXIT
      ENDIF
    NEXT
    // Si el alumno tiene asignada la materia, se imprime.
    IF bandera
      @ 10+nl,1 SAY ALLTRIM(STR(nl))
      @ 10+nl,6 SAY alumnos->matrrialum
      @ 10+nl,20 SAY alumnos->nomalum
      @ 10+nl,57 SAY REPLICATE(" | ",25)+" | "
      nl++
      IF nl > 48
        /* Se imprime el encabezado de la lista. Si dentro de
        esta función se presiona "ESC" se aborta el proceso */
        ISalir := .F.
        nHoja++
        ISalir := encabezado()
        IF ISalir
          EXIT
        ENDIF
      ENDIF
    ENDIF
  ENDIF

```

```

        ENDIF
    ENDIF
    alumnos->(DBSKIP())
    INKEY (.2) // Permite abortar la impresi3n
    IF LASTKEY () == K_ESC
        (@ 0, PCOL()) SAY CHR(27)
        EXIT
    ENDIF
    UNTIL alumnos->grupo # cGrupo .OR. alumnos->(EOF())
    (@ PROW(), PCOL()) SAY CsSendPm3 ("-COMPRIMIDA", m->NmImp)
    EJECT
ELSE
    TONE (750)
    ALERT ("ESTA MATERIA NO ESTA ASIGNADA A NINGUN ALUMNO DENTRO DE ESTE GRUPO")
ENDIF
ELSE
    TONE (750)
    ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    (@ MAXROW()-1,1 SAY SPACE (78)
ENDIF
ELSE
    TONE (750)
    ALERT ("LA MATERIA ACABA DE SER DESASIGNADA EN ALGUNA OTRA ESTACION DE LA RED")
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    (@ MAXROW()-1,1 SAY SPACE (78)
ENDIF
EXIT
ELSE
    TONE (350)
    nOp := ALERT ("La impresora no esta lista",aOpcion)
    IF nOp == 2
        EXIT
    ENDIF
ENDIF
ENDDO
alumnos->(DBSETORDER (3)) // Orden por grupo
RESTSCREEN (9,18,16,63, cuadro)
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
ENDDO
ENDDO
RETURN

// Funci3n que imprime las listas por Alumno
FUNCTION listas_grupo ()
    LOCAL nPmActive, cPuerto, aPuertos, aCapCfg, nprn := NmImp,;
    pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1),;
    cColor2, cNomAlum, cSemestre, lSalir := .F., ok := "N",;
    aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F., cuadro
    PRIVATE impresora, nHoja := 1, materia, maestro, cGrupo := SPACE(3)
    /* Vars. que indican la posici3n donde se desplegara el nombre de la
    impresora si est es cambiada */
    PUBLIC x := 14, y := 32

    IF !FILE("grupo1.ntx")
        TONE (700)
        ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
        RETURN
    ENDIF
    ok_bases := abre_bases (10) // Abre la base de grupos
    IF !ok_bases
        TONE (750)
        ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
        RETURN
    ENDIF

    IF !FILE("alumno1.ntx")
        TONE (700)

```

```

ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
RETURN
ENDIF
ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
    RETURN
ENDIF

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
    RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
ENDIF

impresora := CsPmName2(nPm) // Inicializa la impresora
impres- (DBCLOSEAREA())

limpiamenus (2)
(@ 9, 15 SAY REPLICATE (" ",60)
cColor2 := SETCOLOR(C CAP)
DO WHILE .T.

pic (13)
DISPBOX (10,1,MAXROW()-3,maxcol()-1," | | | | |",C_REYENO) // Se dibuja la ventana de captura
pantallacampus (12)
DO WHILE .T.
    SET KEY K_F2 TO cambia_impresora ()
    (@ 14,32 SAY impresora COLOR C_CONFIG4

cGrupo := SPACE(3)
SET CURSOR ON
(@ 11, 13 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
READ
IF LASTKEY () == K_ESC
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    (@ MAXROW()-1,1 SAY SPACE (78)
    DBCLOSEALL()
    RETURN -1
ELSE
    grupos- (DBSETORDER(1))
    grupos- (DBGOTOP ())
    grupos- (DBSEFK (cGrupo))
    IF grupos- (FOUND())
        grupos- (DBSETORDER(4))
        alumnos- (DBSETORDER(3))
        alumnos- (DBSEFK (cGrupo))
        IF alumnos- (FOUND())
            EXIT
        ELSE
            TONE (800)
            ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
        ENDIF
    ELSE
        TONE (800)
        ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
    ENDIF
ENDIF

ENDIF

ENDIF
ENDIF
cColor := SETCOLOR ("R/B+")
(@ 20,5 SAY "Esta seguro de IMPRIMIR :"
SETCOLOR (cColor)
(@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN")

```



```

READ
IF ok = "N" .OR. LASTKEY () == K_ESC
    LOOP
ENDIF
SET KEY K_F2 TO
ok := "N"

SET CURSOR OFF
DO WHILE .T.
    IF ISPRINTER()
        cuadro := SAVESCREEN (9,18,16,63)
        DISPBOX (9,18,15,62," |  | - | " ,C_REYENO3) // Se dibuja la ventana de captura
        color := SETCOLOR ("R+/N")
        @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
        SETCOLOR (color)
        color := SETCOLOR ("GR+*/N")
        @ 12,25 SAY "SE ESTAN IMPRIMIENDO LAS LISTAS"
        SETCOLOR (color)
        color := SETCOLOR ("GR+/N")
        @ 14,25 SAY "PRESIONE " COLOR C_CONFIG2
        @ 14,34 SAY " <ESC> " COLOR C_CONFIG4
        @ 14,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
        SETCOLOR (color)

        SET ( SET PRINTFILE, ALLTRIM(cPuerto))
        SET ( SET DEVICE, "PRINTER")
        @ PROW(), PCOL() SAY CsScndPm3 (" + COMPRIMIDA", m->NmImp)

        *..... Se busca el primer registro que cumpla con el grupo
        grupos->(DBGOTOP())
        grupos->(DBSEEK(cGrupo))

        *..... Comienza el proceso de impresion
        DO WHILE grupos->grupo == cGrupo
            alumnos->(DBSETORDER (4))
            alumnos->(DBGOTOP())
            alumnos->(DBSEEK (cGrupo))
            IF alumnos->(FOUND())
                ok_bases := abre_bases (5) // Abre la base de maestros
                IF !ok_bases
                    TONE (750)
                    .ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS")
                    RETURN
                ENDIF
                maestros->(DBSEEK (grupos->clavmae))
                maestro := maestros->nommae
                maestros->(DBCLOSEAREA())

                ok_bases := abre_bases (3) // Abre la base de materias
                IF !ok_bases
                    TONE (750)
                    .ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
                    RETURN
                ENDIF
                materias->(DBSEEK (grupos->clavmat))
                materia := materias->nommat
                materias->(DBCLOSEAREA())
                bandera := .F.

                /* Se hace una pre-búsqueda para saber si las materias
                estan asignadas a por lo menos un alumno dentro del
                grupo, de no ser así, la materia no se imprimira */

                REPEAT
                    FOR i = 1 TO 13
                        temp1 := "mat"+ ALLTRIM(STR(i))
                        IF alumnos->(&temp1.) == grupos->clavmat .AND. alumnos->inscrito .AND. EMPTY(alumnos->baja)
                            bandera := .T.
                        EXIT
                    ENDIF
                NEXT
            ENDIF
        ENDIF
    ENDIF
ENDWHILE
    
```



```

@ PROW(), PCOL() SAY CsSend!m3 ("-COMPRIMIDA", m->NmImp)
EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET ( SET_DEVICE, "SCREEN")
RESTSCREEN (9,18,16,63, cuadro)
EXIT
ELSE
TONE (350)
nOp : ALERT ("La impresora no esta lista",aOpcion)
IF nOp == 2
EXIT
ENDIF
ENDIF
ENDDO
ENDDO
RETURN
// Función que imprime el encabezado de las listas de asistencia.
FUNCTION encabezado ()
LOCAL cCiclo

// Se determina el ciclo al que corresponde la lista a ser impresa.
IF MONTH (DATE()) < 9
cCiclo := "CICLO "+ALLTRIM(STR(YEAR(DATE()-1)))+ "-" +ALLTRIM(STR(YEAR(DATE())))
ELSE
cCiclo := "CICLO "+ALLTRIM(STR(YEAR(DATE())))+ "-" +ALLTRIM(STR(YEAR(DATE()+1)))
ENDIF

@ 1,1 SAY "SISTEMA DE CONTROL ESCOLAR"
@ 1,125 SAY "HOJA : "+ALLTRIM(STR(nHoja))
@ 2,1 SAY "LISTA DE ASISTENCIA" + SPACE(44) + cCiclo + SPACE(44) + "FECHA : " + ALLTRIM(DTOC(DATE()))
@ 3,1 SAY esc_selec
@ 4,1 SAY REPLICATE("I",140)
@ 5,1 SAY "CLAVE MATERIA GRUPO CLAVE PROFESOR"
@ 5,105 SAY "SALON DIAS DE CLASE A LA SEMANA"
@ 6,1 SAY grupos->clavmat
@ 6,8 SAY materia
@ 6,51 SAY cGrupo
@ 6,58 SAY grupos->clavmae
@ 6,68 SAY maestro
@ 6,105 SAY grupos->salon1
@ 6,115 SAY grupos->dia1
@ 6,120 SAY grupos->dia2
@ 6,125 SAY grupos->dia3
@ 6,130 SAY grupos->dia4
@ 7,1 SAY REPLICATE("=",140)
@ 8,1 SAY "NL MATRICULA NOMBRE DEL ALUMNO"
@ 8,57 SAY REPLICATE(" | ",25) + " | CAL | FAL | "
@ 9,1 SAY REPLICATE("=",140)
INKEY (2) // Permite abortar la impresion
IF LASTKEY () == K ESC
RETURN .T.
ENDIF
RETURN .F.

```

***** FUNCIONES PARA LA IMPRESION DE CREDENCIALES *****

```

// Función que imprime las credenciales de los Alumnos por grupo
FUNCTION credenciales grupo ()
LOCAL nPmActive, cPuerto, aPuertos, aCapCig, npm := NmImp,;
pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1);
cColor2, cNomAlum, cSemestre, avance1, avance2,;
aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F., cuadro.ok := "N"
PRIVATE impresora
/* Vars. que indican la posición donde se desplegara el nombre de la
impresora si est es cambiada */
PUBLIC x := 14, y := 32

IF !FILE("grupo1.ntx")
TONE (700)
ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")

```

```

RETURN
ENDIF
ok_bases := abre_bases (10) // Abre la base de grupos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
RETURN
ENDIF

IF !FILE("alumno1.ntx")
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
RETURN
ENDIF
ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
RETURN
ENDIF

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
RETURN
ENDIF

impresora := CsPmName2(nPm) // Inicializa la impresora
impres->(DBCLOSEAREA())

limpiamenus (2)
(@ 9, 15 SAY REPLICATE ("-",60)
cColor2 := SETCOLOR(C_CAP)
DO WHILE .T.
  pic (13)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1," | | | | |",C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (15)
  DO WHILE .T.
    SET KEY K_F2 TO cambia_impresora ()
    (@ 14,32 SAY impresora COLOR C_CONFIG4
    cGrupo := SPACE(3)
    cColor := SETCOLOR (C_BORRADO)
    (@ 12,37 SAY REPLICATE ("■",30)
    (@ 16,13 SAY REPLICATE ("■",30)
    SETCOLOR (cColor)
    SET CURSOR ON
    (@ 12, 13 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
    READ

  IF LASTKEY () == K_ESC
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    (@ MAXROW()-1,1 SAY SPACE (78)
    DBCLOSEALL()
    RETURN -1
  ENDIF

  cVigencia := SPACE(30)
  cFirma := SPACE(30)
  SET CURSOR ON
  (@ 12, 37 GET cVigencia PICTURE "@!" VALID (!EMPTY(cVigencia))
  (@ 16, 13 GET cFirma PICTURE "@!" VALID (!EMPTY(cFirma))
  READ

```

```

IF LASTKEY () == K_ESC
  LOOP
ELSE
  grupos->(DBGOTOP ())
  grupos->(DBSEFK (cGrupo))
  IF grupos->(FOUND())
    alumnos->(DBSETORDER(3))
    alumnos->(DBSEFK (cGrupo))
    IF alumnos->(FOUND())
      EXIT
    ELSE
      TONE (800)
      ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
    ENDIF
  ELSE
    TONE (800)
    ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
  ENDIF
ENDIF
ENDIF
ENDIF
ENDDO
cColor := SETCOLOR ("R/B+")
(@ 20,5 SAY "Esta seguro de IMPRIMIR :")
SETCOLOR (cColor)
(@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN"))
READ
IF ok = "N" .OR. LASTKEY () == K_ESC
  LOOP
ENDIF
SET KEY K_F2 TO
ok := "N"

SET CURSOR OFF
DO WHILE .T.
  IF ISPRINTER()
    cuadro := SAVESCREEN (9,18,16,63)
    DISPBOX (9,18,15,62," |J-L| ",C_REYENO3) // Se dibuja la ventana de captura
    color := SETCOLOR ("R+/N")
    (@ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR")
    SETCOLOR (color)
    color := SETCOLOR ("GR+*/N")
    (@ 12,22 SAY "SE ESTAN IMPRIMIENDO LAS CREDENCIALES")
    SETCOLOR (color)
    color := SETCOLOR ("GR+/N")
    (@ 14,25 SAY "PRESIONE " COLOR C_CONFIG2)
    (@ 14,34 SAY "<ESC> " COLOR C_CONFIG4)
    (@ 14,41 SAY " PARA CANCELAR" COLOR C_CONFIG2)
    SETCOLOR (color)

    SET ( SET PRINTFILE, ALLTRIM(cPuerto))
    SET ( SET DEVICE, "PRINTER")

  DO WHILE grupos->grupo == cGrupo
    alumnos->(DBSEEK (cGrupo))
    IF alumnos->(FOUND())
      (@ PROW(), PCOL()) SAY CsSendPm3 ("+COMPRIMIDA", m->NmImp)
      INKEY (.2) // Permite abortar la impresion
      IF LASTKEY () == K_ESC
        EXIT
      ENDIF

    avance1 := 6
    avance2 := 4
    REPEAT
      IF alumnos->inscrito .AND. EMPTY(alumnos->baja)
        DO CASE
          CASE cTipo == "P"
            (@ avance1, 26 SAY alumnos->nomalum
            avance1 += 2
            (@ avance1, 43 SAY alumnos->matricialum
            avance1 += 2
            (@ avance1, 35 SAY cVigencia
            avance1 += 2

```

```

    (@ PROW(), PCOL()) SAY CsSendPm3 ("+COMPRIMIDA", m->NmImp)
    (@ avance1, 35 SAY cfirma
CASE cTipo == "L"
    (@ avance2, 28 SAY alumnos->nomalum
    avance2 += 2
    (@ avance2, 35 SAY esc_selec
    avance2 += 2
    (@ avance2, 45 SAY alumnos->matrerialum
    avance2 += 2
    (@ avance2, 40 SAY cVigencia
    avance2 += 2
    (@ avance2, 42 SAY cfirma
CASE cTipo == "G" .OR. cTipo == "D"
    TONE (750)
    ALERT ("NO ESTA DISPONIBLE")
ENDCASE
avance1 += 7
avance2 += 9
alumnos->(DBSKIP())
INKEY (2) // Permite abortar la impresiøn
IF LASTKEY () == K_ESC
    EXIT
ENDIF
ELSE
    alumnos->(DBSKIP())
ENDIF
UNTIL alumnos->grupo # cGrupo .OR. alumnos->(EOF())
IF LASTKEY () == K_ESC
    (@ PROW(), PCOL()) SAY CHR(27)
    EXIT
ENDIF
grupos->(DBSKIP())
ELSE
    TONE (750)
    ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    (@ MAXROW()-1,1 SAY SPACE (78)
    DBCLOSEALL()
    RETURN -1
ENDIF
ENDDO
(@ PROW(), PCOL()) SAY CsSendPm3 ("-COMPRIMIDA", m->NmImp)
EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
RESTSCREEN (9,18,16,63, cuadro)
EXIT
ELSE
    TONE (350)
    nOp := ALERT ("La impresora no esta lista",aOpcion)
    IF nOp == 2
        EXIT
    ENDIF
ENDIF
ENDDO
ENDDO
RETURN

// Funciøn que imprime las credenciales individuales por Alumno
FUNCTION alumno_credenciales ()
    LOCAL nPmActive, cPuerto, aPuertos, aCapCf, npr := NmImp,;
    pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1),;
    cColor2 := SETCOLOR(), cNomAlum, cSemestre,;
    aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F., cuadro

    IF !FILE("alumno1.ntx")
        TONE (700)
        ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
        RETURN
    ENDIF

```

```

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  RETURN
ENDIF

limpiamenu (2)
(@ 9, 15 SAY REPLICATE (" ",60)
DO WHILE .T.
  IF !FILE("grupo1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
    RETURN
  ENDIF
ok_bases := abre_bases (10) // Abre la base de grupos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
  RETURN
ENDIF

pie (2)
DISPBOX (10,1,MAXROW()-3,maxcol()-1," | | | | |",C_REYENO) // Se dibuja la ventana de captura
pantallacampo (16)
DO WHILE .T.
  cGrupo := SPACE(3)
  cColor := SETCOLOR (C_BORRADO)
  @ 15, 20 SAY REPLICATE (" ",30)
  @ 17, 20 SAY REPLICATE (" ",30)
  SETCOLOR (cColor)
  SET CURSOR ON
  @ 11, 20 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
  READ
  SET CURSOR OFF

  IF LASTKEY () == K_ESC
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    @ MAXROW()-1,1 SAY SPACE (78)
    DBCLOSEALL()
    RETURN -1
  ELSE
    grupos->(DBGOTOP ())
    grupos->(DBSEEK (cGrupo))
    IF grupos->(FOUND())
      alumnos->(DBSETORDER(3))
      alumnos->(DBSEEK (cGrupo))
      IF alumnos->(FOUND())
        EXIT
      ELSE
        TONE (800)
        ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
      ENDIF
    ELSE
      TONE (800)
      ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
    ENDIF
  ENDIF
ENDIF
ENDDO

// Se busca el alumno especificado
DO WHILE .T.

// Se pide la clave del alumno al que se va a imprimir la credencial
cAlum := SPACE(7)

  SET CURSOR ON
  @ 13, 20 GET cAlum PICTURE "@!" VALID (!EMPTY(cAlum))
  READ

  cVigencia := SPACE(30)
  cFirma := SPACE(30)

```

```

SET CURSOR ON
@ 15, 20 GET cVigencia PICTURE "@!" VALID (!EMPTY(cVigencia))
@ 17, 20 GET cFirma PICTURE "@!" VALID (!EMPTY(cFirma))
READ

SET CURSOR OFF
IF LASTKEY () == K_ESC
  IF !flag
    grupos->(DBCLOSEAREA())
  ELSE
    flag := .F.
  ENDIF
EXIT
ELSE
  alumnos->(DBSETORDER(1))
  alumnos->(DBGOTOP ())
  alumnos->(DBSEEK (cAlum))
  IF alumnos->(FOUND())
    IF alumnos->grupo == cGrupo .AND. alumnos->inscrito .AND. EMPTY(alumnos->baja)
      cNomAlum := alumnos->(NomAlum)
      cSemestre := alumnos->(semesalum)
    ELSE
      TONE (750)
      ALERT ("ESTE ALUMNO NO ESTA DADO DE ALTA EN ESTE GRUPO O NO HA SIDO REINSCRITO")
      LOOP
    ENDIF
  ELSE
    TONE (350)
    cColor := SETCOLOR (C_BORRADO)
    @ 13,20 SAY REPLICATE ("█",55)
    SETCOLOR (cColor)
    @ 13, 20 SAY "No existe"
    INKEY (1)
    @ 13,20 SAY " "
    LOOP
  ENDIF
ENDIF

IF !flag
  grupos->(DBCLOSEAREA())
ENDIF
flag := .T.
ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
  RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
  RETURN
ENDIF

CsPmName2(nPm) // Inicializa la impresora

DO WHILE .T.
  IF ISPRINTER()
    cuadro := SAVESCREEN (9,18,16,63)
    DISPBOX (9,18,15,62," |J-L| ",C_REYENO3) // Se dibuja la ventana de captura
    color := SETCOLOR ("R+N")
    @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
    SETCOLOR (color)
    color := SETCOLOR ("GR+*N")
    @ 12,24 SAY "SE ESTA IMPRIMIENDO LA CREDENCIAL"
    SETCOLOR (color)

    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "PRINTER")
  
```



```

@ PROW(), PCOL() SAY CsSendPm2 (" +COMPRIMIDA")

avance1 := 6
avance2 := 4

DO CASE
CASE cTipo == "P" // Preparatoria
@ avance1, 26 SAY alumnos->nomalum
avance1 += 2
@ avance1, 43 SAY alumnos->matrrialum
avance1 += 2
@ avance1, 35 SAY cVigencia
avance1 += 2
@ PROW(), PCOL() SAY CsSendPm2 (" +COMPRIMIDA")
@ avance1, 35 SAY cfirma
CASE cTipo == "L" // Licenciatura
@ avance2, 28 SAY alumnos->nomalum
avance2 += 2
@ avance2, 35 SAY esc_selec
avance2 += 2
@ avance2, 45 SAY alumnos->matrrialum
avance2 += 2
@ avance2, 40 SAY cVigencia
avance2 += 2
@ avance2, 42 SAY cfirma
CASE cTipo == "G" .OR. cTipo == "D"
TONE (750)
ALERT ("NO ESTA DISPONIBLE")
ENDCASE
avance1 += 7
avance2 += 9

EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
REFSTSCREEN (9,18,16,63, cuadro)
EXIT
ELSE
TONE (350)
nOp := ALERT ("La impresora no esta lista",aOpcion)
IF nOp == 2
EXIT
ENDIF
ENDIF
ENDDO
escapes->(DBCLOSEAREA())
impres->(DBCLOSEAREA())
ENDDO
ENDDO
RETURN

/***** FUNCIONES PARA LA IMPRESION DE BOLETAS *****/

// Función que imprime las boletas por Alumno
FUNCTION boletas_alumno ()
LOCAL nPmActive, cPuerto, aPuertos, aCapCig, nPm := NmImp,;
pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1),;
cColor2 := SETCOLOR(), cNomAlum, cSemestre, ok := "N",;
aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F., cuadro,;
temp1,temp2,temp3,cal1,cal2,cal3,prom, mate, fall,;
nPTotal := 0.0,aProm1 := {}, aProm2 := {}, aFall := {},aAux := {};
cFirma := SPACE (30),cCargo := SPACE (40),cComen1 := SPACE (50),;
cComen2 := SPACE (50),cCiclo,IFlagProm := .F.

IF !FILE("alumno1.ntx")
TONE (700)

ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
RETURN
ENDIF

ok_bases := abre_bases (7) // Base de alumnos
If !ok_bases

```

```

TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
RETURN
ENDIF

limpiamenu (2)
@ 9, 15 SAY REPLICATE ("ä",60)
DO WHILE .T.
  IF !FILE("grupo1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
    RETURN
  ENDIF
  ok_bases := abre_bases (10) // Abre la base de grupos
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
    RETURN
  ENDIF

  pie (2)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1," |J-L|",C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (14)
  DO WHILE .T.
    cGrupo := SPACE(3)
    SET CURSOR ON
    @ 11, 20 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
    READ
    SET CURSOR OFF
    IF LASTKEY () == K_ESC
      RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
      SETCOLOR (cColor2)
      @ MAXROW()-1,1 SAY SPACE (78)
      DBCLOSEALL()
      RETURN -1
    ELSE
      grupos->(DBGOTOP ())
      grupos->(DBSEEK (cGrupo))
      IF grupos->(FOUND())
        alumnos->(DBSETORDER(3))
        alumnos->(DBSEEK (cGrupo))
        IF alumnos->(FOUND())
          EXIT
        ELSE
          TONE (800)
          ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
        ENDIF
      ELSE
        TONE (800)
        ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
      ENDIF
    ENDIF
  ENDIF
ENDDO

PIE (1)
// Se busca el alumno especificado
DO WHILE .T.
  // Se pide la clave del alumno
  cAlum := SPACE(7)
  // Se borra el nombre anteriormente utilizado
  cColor := SETCOLOR (C_BORRADO)
  @ 13,20 SAY REPLICATE ("■",35)
  SETCOLOR (cColor)

  SET CURSOR ON
  @ 12, 20 GET cAlum PICTURE "@!" VALID (!EMPTY(cAlum))
  READ
  SET CURSOR OFF
  IF LASTKEY () == K_ESC
    IF !flag
      grupos->(DBCLOSEAREA())
    ELSE

```

```

flag := .F.
ENDIF
EXTI
ELSE
alumnos->(DBSETORDER(1))
alumnos->(DBGOTOP ())
alumnos->(DBSEEK (cAlum))
IF alumnos->(FOUND())
IF alumnos->grupo == cGrupo .AND. alumnos->inscrito .AND. EMPTY(alumnos->baja)
cNomAlum := alumnos->(NomAlum)
cSemestre := alumnos->(semesalum)
ELSE
TONE (750)
ALERT ("ESTE ALUMNO NO ESTA DADO DE ALTA EN ESTE GRUPO O NO HA SIDO REINSCRITO")
LOOP
ENDIF
ELSE
TONE (350)
cColor := SETCOLOR (C_BORRADO)
@ 12,20 SAY REPLICATE ("■",55)
SETCOLOR (cColor)
@ 12, 20 SAY "No existe"
INKEY (1)
@ 12,20 SAY " "
LOOP
ENDIF
ENDIF
@ 13,20 GET cNomAlum //Se despliega el nombre del alumno para verificacion
CLEAR GETS
SET CURSOR ON
@ 15, 20 GET cFirma PICTURE "@!" VALID (!EMPTY(cFirma))
@ 16, 20 GET cCargo PICTURE "@!" VALID (!EMPTY(cCargo))
@ 17, 20 GET cComen1 PICTURE "@!"
@ 18, 20 GET cComen2 PICTURE "@!"
READ

cColor := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro de IMPRIMIR :."
SETCOLOR (cColor)
@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN")
READ
cColor := SETCOLOR (C_BORRADO)
@@ 20,5 SAY REPLICATE ("■",30)
SETCOLOR (cColor)

SET CURSOR OFF
IF ok == "N" .OR. LASTKEY () == K_ESC
LOOP
ENDIF
ok := "N"

IF !flag
grupos->(DBCLOSEAREA())
ENDIF

flag := .T.

ok_bases := abre_bases (3) // Abre la base de materias
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
RETURN
ENDIF

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes

```

```

IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
  RETURN
ENDIF

CsPmName2(nPm)           // Inicializa la impresora

DO WHILE .T.
  IF ISPRINTER()
    cuadro := SAVESCREEN (9,18,16,63)
    DISPBOX (9,18,15,62,"┌──┴──┐ |└──┬──┘ |",C_REYENO3) // Se dibuja la ventana de captura
    color := SETCOLOR ("R+N")
    @ 11,21 SAY "  ESPERE UN MOMENTO POR FAVOR"
    SETCOLOR (color)
    color := SETCOLOR ("GR+*N")
    @ 12,26 SAY "SE ESTA IMPRIMIENDO LA BOLETA"
    SETCOLOR (color)
    // Calcula el ciclo escolar de vigencia de la boleta
    IF MONTH (DATE()) < 9
      cCiclo := "CICLO "+ALLTRIM(STR(YEAR(DATE()-1))+"-"+ALLTRIM(STR(YEAR(DATE()))))
    ELSE
      cCiclo := "CICLO "+ALLTRIM(STR(YEAR(DATE()))+"-"+ALLTRIM(STR(YEAR(DATE()+1)))
    ENDIF

    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "PRINTER")
    // Se imprime el encabezado de la boleta
    @ PROW(), PCOL() SAY CsSendPm2 ("+GRUESA")
    @ 1,6 SAY "UNIVERSIDAD VASCO DE QUIROGA"
    @ 4,6 SAY "constancia de calificaciones"
    @ PROW(), PCOL() SAY CsSendPm2 ("-GRUESA")
    @ 7,55 SAY ALLTRIM(STR(DAY(DATE()))+" DE "+UPPER(CMONTH(DATE()))+" DE "+ALLTRIM(STR(YEAR(DATE()))))
    @ 8,1 SAY "TAXIMAROA # 62 COL. FELIX IRETA"
    @ 8,55 SAY cCiclo
    @ 10,1 SAY cAlum+ SPACE(5) + cNomAlum
    @ 10,55 SAY esc_selcc

  DO CASE
    CASE cTipo == "L" // Licenciatura
      @ 11,55 SAY "SEMESTRE "+cSemestre
      @ 13,1 SAY "┌" + REPLICATE("-",77) + "┐"
      @ 14,1 SAY "└" + REPLICATE("-",77) + "┘"
      @ 15,1 SAY "┌" + REPLICATE("-",77) + "┐"
      x := 1
      nPTotal := 0.0
      temp1 := "mat" + ALLTRIM(STR(x))
      mate := alumnos->(&temp1.)
      REPEAT
        materias->(DBSEEK(mate))
        temp2 := "p"+ALLTRIM(STR(x))+1"
        temp3 := "p"+ALLTRIM(STR(x))+2"
        temp4 := "p"+ALLTRIM(STR(x))+3"
        cal1 := alumnos->(&temp2.)
        cal2 := alumnos->(&temp3.)
        cal3 := alumnos->(&temp4.)
        prom := ((cal1+cal2)/2)+cal3/2

        /* Checa si la escuela esta dada de alta en SEP Mexico
           o en SEP Edo. y dependiendo de esto redondea el promedio */
        IF cTipoSep .AND. prom % 1 == .5
          prom := (prom % 1)
        ELSE
          prom := ROUND (prom,0)
        ENDIF

        nPTotal += prom
        @ 15+x,1 SAY ""+materias->nommat
        @ 15+x,79 SAY ""
        @ 15+x,64 SAY PADL(ALLTRIM(STR(prom)),2," ")
        x++
        // Se evita el error ya que no existiria el campo "MAT14"

```

```

IF x == 14
  EXIT
ENDIF
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
UNTIL EMPTY(mate)
nPTotal /= (x-1)

@ 15+x,1 SAY " " + REPLICATE("=",77) + " "
@ 16+x,7 SAY cFirma
@ 17+x,1 SAY cCargo
@ 17+x,57 SAY "PROM.: " + ALLTRIM(STR(nPTotal))
@ PROW(), PCOL() SAY CsSendPm2 (" + COMPRIMIDA")
@ 18+x,92 SAY "Constancia de Control Interno"
@ PROW(), PCOL() SAY CsSendPm2 (" - COMPRIMIDA")
@ 19+x,1 SAY cComen1
@ 20+x,1 SAY cComen2
CASE cTipo == "P" // Preparatoria
aProm1 := {0,0,0,0,0,0,0}
aFall := {0,0,0,0,0,0}
aAux := {0,0,0,0,0,0}
ASIZE (aProm2, 13)
FOR x := 1 TO 13
  aProm2 [x] := 0.0
NEXT
@ 11,55 SAY "GRUPO : " + cGrupo
@ 13,1 SAY REPLICATE("=",84)
@ PROW(), PCOL() SAY CsSendPm2 (" + COMPRIMIDA")
@ 14,1 SAY "MATERIA
EVALUACIONES"
@ 15,39 SAY "PRIMER SEGUNDO TERCER PROME EXAMEN PROME CUARTO QUINTO SEXTO PROME
EXAMEN CALIF RESULTADO"
@ 16,39 SAY "CAL INA CAL INA CAL INA MENS. 1 SEM. SEMES CAL INA CAL INA CAL INA MENS. 2 SEM.
SEMES"
@ 17,1 SAY REPLICATE("=",145)
x := 1
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
REPEAT
  IFlagProm := .F.
  materias->(DBSEEK(mate))
  @ 17+x,1 SAY materias->nommat
  y := 6
  aProm1 [1] := 0.0 // Se inicializa para cada materia
  call := 0.0
  // Imprime las primeras 3 evaluaciones
  FOR i = 1 TO 3
    temp2 := "p" + ALLTRIM(STR(x)) + ALLTRIM(STR(i))
    temp3 := "i" + ALLTRIM(STR(x)) + ALLTRIM(STR(i))
    call := alumnos->(&temp2.)
    fall := alumnos->(&temp3.) // Se imprimen las faltas
    aProm1 [1] += call // Promedios horizontales
    aFall [i] += fall // Suma de faltas
    aProm2 [i] += call
    IF call # 0
      @ 17+x,32+y SAY STR(call) + SPACE(1) + STR(fall)
    ELSE
      IFlagProm := .T.
    ENDIF
    y += 9
  NEXT
  /* Imprime el primer promedio mensual, solo
  que ya se hayan capturado las primeras
  tres calificaciones */
  IF IFlagProm
    aProm1 [1] /= 3
    aProm2 [4] += aProm1 [1]
    @ 17+x,33+y SAY PADL(ALLTRIM(STR(aProm1 [1])),4," ")
    // Imprime la 1ª calificación semestral y el 1er. promedio semestral
    temp2 := "s" + ALLTRIM(STR(x))
    call := alumnos->(&temp2.)
    aProm1 [2] := (aProm1 [1] + call) / 2
    aProm2 [5] += call
  
```

```

aProm2 [6] += aProm1 [2]
@ 17+x,40+y SAY STR(cal1)+SPACE(3)+PADL(ALLTRIM(STR(aProm1 [2])),4," ")
ENDIF
y+=22
aProm1 [3] := 0.0
FOR i = 4 TO 6
temp2 := "p"+ALLTRIM(STR(x))+ALLTRIM(STR(i))
temp3 := "t"+ALLTRIM(STR(x))+ALLTRIM(STR(i))
cal1 := alumnos->(&temp2.)
fal1 := alumnos->(&temp3.) // Se imprimen las faltas
aProm1 [3] += cal1
aFall [i] += fal1
aProm2 [i+3] += cal1
IF cal1 # 0
@ 17+x,32+y SAY STR(cal1)+SPACE(1)+STR(fal1)
ELSE
IFlagProm := .T.
ENDIF
y+=9
NEXT
IF !IFlagProm
aProm1 [3] /= 3
aProm2 [10] += aProm1 [3]
@ 17+x,32+y SAY ALLTRIM(STR(aProm1 [3]))
//Imprime la 2ª calificación semestral y el 2º promedio semestral
temp2 := "t"+ALLTRIM(STR(x))
cal1 := alumnos->(&temp2.)
aProm1 [4] := (aProm1 [3]+cal1)/2
aProm2 [11] += cal1
aProm2 [12] += aProm1 [4]
aprom2 [13] += (aProm1 [2] + aProm1 [4]) / 2
@ 17+x,119 SAY STR(cal1)+SPACE(4)+PADL(ALLTRIM(STR(aProm1 [4])),4," ")
@ 17+x,137 SAY PADL(ALLTRIM(STR((aProm1 [2]+aProm1 [4])/2)),4," ")
ENDIF
x++
// Se evita el error ya que no existiría el campo "MAT14"
IF x == 14
EXIT
ENDIF
// SE avanza a la siguiente materia
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
IFlagProm := .F.
UNTIL EMPTY(mate) // hasta que no haya mas materias
@ 17+x,1 SAY REPLICATE("!",145)
@ 18+x,1 SAY "INASISTENCIAS"
@ 18+x,34 SAY aFall [1]
@ 18+x,43 SAY aFall [2]
@ 18+x,52 SAY aFall [3]
@ 18+x,83 SAY aFall [4]
@ 18+x,92 SAY aFall [5]
@ 18+x,101 SAY aFall [6]
@ 19+x,1 SAY "PROMEDIOS"
y := 6
FOR i = 1 TO 3
IF aprom2 [i] # 0
@ 19+x,24+y SAY aProm2 [i] / (x-1)
ENDIF
y += 9
NEXT
y += 1
// Si el promedio es 0 (cero), no se imprime
IF aprom2 [4] # 0
@ 19+x,57 SAY aProm2 [4] / (x-1)
ENDIF
IF aprom2 [5] # 0
@ 19+x,64 SAY aProm2 [5] / (x-1)
ENDIF
IF aprom2 [6] # 0
@ 19+x,71 SAY aProm2 [6] / (x-1)
ENDIF
IF aprom2 [7] # 0

```

```

    @ 19+x,78 SAY aProm2 [7] / (x-1)
ENDIF
IF aProm2 [8] # 0
    @ 19+x,87 SAY aProm2 [8] / (x-1)
ENDIF
IF aProm2 [9] # 0
    @ 19+x,96 SAY aProm2 [9] / (x-1)
ENDIF
IF aProm2 [10] # 0
    @ 19+x,104 SAY aProm2 [10] / (x-1)
ENDIF
IF aProm2 [11] # 0
    @ 19+x,111 SAY aProm2 [11] / (x-1)
ENDIF
IF aProm2 [12] # 0
    @ 19+x,119 SAY aProm2 [12] / (x-1)
ENDIF
IF aProm2 [13] # 0
    @ 19+x,129 SAY aProm2 [13] / (x-1)
ENDIF
@ 24+x,7 SAY cFirma
@ 25+x,1 SAY cCargo
@ 26+x,1 SAY cComen1
@ 27+x,1 SAY cComen2
@ PROW(), PCOL() SAY CsSendPm2 ("-COMPRIMIDA")
CASE cTipo == "G" .OR. cTipo == "D"
@ 11,55 SAY "SEMESTRE " + cSemestre
@ 13,1 SAY "r" + REPLICATE("=",77) + "q"
@ 14,1 SAY " | " + MATERIA
@ 15,1 SAY " | " + REPLICATE("=",77) + "q"
x := 1
nPTotal := 0.0
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
REPEAT
    materias->(DBSEEK(mate))
    temp2 := "u" + ALLTRIM(STR(x))
    temp3 := "i" + ALLTRIM(STR(x))
    call := alumnos->(&temp2.)
    cal2 := alumnos->(&temp3.)

    /* Checa si la escuela esta dada de alta en SEP México
    o en SEP Edo. y dependiendo de esto redondea el promedio */
    IF cTipoSep .AND. call % 1 == .5
        call := call - (call % 1)
    ELSE
        call := ROUND (call,0)
    ENDIF

    nPTotal += call
    @ 15+x,1 SAY "" + materias->nommat
    @ 15+x,79 SAY ""
    @ 15+x,64 SAY PADL(ALLTRIM(STR(call)),2," ")
    x++
    // Se evita el error ya que no existiria el campo "MAT14"
    IF x = 14
        EXIT
    ENDIF
    temp1 := "mat" + ALLTRIM(STR(x))
    mate := alumnos->(&temp1.)
UNTIL EMPTY(mate)
nPTotal /= (x-1)

@ 15+x,1 SAY "k" + REPLICATE("=",77) + "j"
@ 16+x,7 SAY cFirma
@ 17+x,1 SAY cCargo
@ 17+x,57 SAY "PROM.: " + ALLTRIM(STR(nPTotal))
@ PROW(), PCOL() SAY CsSendPm2 ("COMPRIMIDA")
@ 18+x,92 SAY "Constancia de Control Interno"
@ PROW(), PCOL() SAY CsSendPm2 ("-COMPRIMIDA")
@ 19+x,1 SAY cComen1
@ 20+x,1 SAY cComen2
    
```

CALIFICACION | "

```

    OTHERWISE
      TONE (800)
      ALERT ("TIPO DE ESCUELA NO VALIDO")
    END CASE
  EJECT
  SET ( _SET_PRINTFILE, ALLTRIM(cPuerto))
  SET ( _SET_DEVICE, "SCREEN")
  RESTSCREEN (9,18,16,63, cuadro)
  EXIT
ELSE
  TONE (350)
  nOp := ALERT ("La impresora no esta lista",aOpcion)
  IF nOp == 2
    EXIT
  ENDIF
ENDIF
ENDDO
materias->(DBCLOSEAREA())
escapes->(DBCLOSEAREA())
impres->(DBCLOSEAREA())
ENDDO
ENDDO
RETURN

// Función que imprime las boletas por grupo
FUNCTION gpo_boletas ()
  LOCAL nPmActive, cPuerto, aPuertos, aCapCfg, npm := NmImp;,
  pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1);,
  cColor2 := SETCOLOR(), cNomAlum, cSemestre, ok := "N";,
  cCiclo, aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F., cuadro;,
  temp1,temp2,temp3,cal1,cal2,cal3,prom, mate, fall;,
  nPTotal := 0.0,aProm1 := {}, aProm2 := {}, aFall := {},aAux := {};,
  cFirma := SPACE (30),cCargo := SPACE (40),cComen1 := SPACE (50);,
  cComen2 := SPACE (50),lFlagProm := .F.

  IF !FILE("alumno1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
    RETURN
  ENDIF

  ok_bases := abre_bases (7) // Base de alumnos
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
    RETURN
  ENDIF

  limpiamcenas (2)
  @@ 9, 15 SAY REPLICATE ("-",60)
  DO WHILE .T.
    IF !FILE("grupo1.ntx")
      TONE (700)
      ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
      RETURN
    ENDIF
    ok_bases := abre_bases (10) // Abre la base de grupos
    IF !ok_bases
      TONE (750)
      ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
      RETURN
    ENDIF

    pic (2)
    DISPBOX (10,1,MAXROW()-3,maxcol()-1," |J-L|",C_REYENO) // Se dibuja la ventana de captura
    pantallacampos (14.5)
    DO WHILE .T.
      cGrupo := SPACE(3)
      SET CURSOR ON
      @ 11, 20 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
      READ
      SET CURSOR OFF

```



```

IF LASTKEY () == K_ESC
  RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
  SETCOLOR (cColor2)
  @ MAXROW()-1,1 SAY SPACE (78)
  DBCLOSEALL()
  RETURN -1
ELSE
  grupos->(DBGOTOP ())
  grupos->(DBSEEK (cGrupo))
  IF grupos->(FOUND())
    alumnos->(DBSF:ORDER(3))
    alumnos->(DBSEEK (cGrupo))
    IF alumnos->(FOUND())
      EXIT
    ELSE
      TONE (800)
      ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
      LOOP
    ENDIF
  ELSE
    TONE (800)
    ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
    LOOP
  ENDIF
ENDIF
ENDDO
grupos->(DBCLOSEAREA())

PIE (1)
SET CURSOR ON
@ 13, 20 GET cFirma PICTURE "@!" VALID (!EMPTY(cFirma))
@ 14, 20 GET cCargo PICTURE "@!" VALID (!EMPTY(cCargo))
@ 15, 20 GET cComen1 PICTURE "@!"
@ 16, 20 GET cComen2 PICTURE "@!"
READ
IF LASTKEY () == K_ESC
  LOOP
ENDIF

cColor := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro de IMPRIMIR : "
SETCOLOR (cColor)
@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN")
READ
cColor := SETCOLOR (C_BORRADO)
@ 20,5 SAY REPLICATE ("█",30)
SETCOLOR (cColor)

SET CURSOR OFF
IF ok == "N" .OR. LASTKEY () == K_ESC
  LOOP
ENDIF
ok := "N"

ok_bases := abre_bases (3) // Abre la base de materias
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  RETURN
ENDIF

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
  RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")

```

```

RETURN
ENDIF

CsPmName2(nPm)           // Inicializa la impresora

DO WHILE .T.
  IF ISPRINTER()
    cuadro := SAVESCREEN (9,18,16,63)
    DISPBOX (9,18,15,62," | | - | ",C_REYENO3) // Se dibuja la ventana de captura
    color := SETCOLOR ("R+N")
    @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
    SETCOLOR (color)
    color := SETCOLOR ("GR+N")
    @ 12,24 SAY "SE ESTAN IMPRIMIENDO LAS BOLETAS"
    SETCOLOR (color)
    color := SETCOLOR ("GR+N")
    @ 14,25 SAY "PRESIONE " COLOR C_CONFIG2
    @ 14,34 SAY "<ESC>" COLOR C_CONFIG4
    @ 14,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
    SETCOLOR (color)

    // Calcula el ciclo escolar de vigencia de la boleta
    IF MONTH (DATE()) < 9
      cCiclo := "CICLO "+ALLTRIM(STR(YEAR(DATE()-1))+"-"+ALLTRIM(STR(YEAR(DATE()))))
    ELSE
      cCiclo := "CICLO "+ALLTRIM(STR(YEAR(DATE()))+"-"+ALLTRIM(STR(YEAR(DATE()+1)))
    ENDIF

    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "PRINTER")
    REPEAT
      IF alumnos->inscrito .AND. EMPTY(alumnos->baja)
        // Se imprime el encabezado de la boleta
        @ PROW(), PCOL() SAY CsSendPm2 ("GRUESA")
        @ 1,6 SAY "UNIVERSIDAD VASCO DE QUIROGA"
        @ 4,6 SAY "constancia de calificaciones"
        @ PROW(), PCOL() SAY CsSendPm2 ("-GRUESA")
        @ 7,55 SAY ALLTRIM(STR(DAY(DATE())))+" DE "+UPPER(CMONTH(DATE()))+" DE "+ALLTRIM(STR(YEAR(DATE()))))
        @ 8,1 SAY "TAXIMAROA # 62 COL. FELIX IRETA"
        @ 8,55 SAY cCiclo
        @ 10,1 SAY alumnos->matrerialum+ SPACE(5)+ alumnos->NomAlum
        @ 10,55 SAY esc_selec
        /* Este bloque sirve para detectar si se pulso la tecla "ESC"
           aqui es donde se cancela la impresion */
        INKEY (.2)
        IF LASTKEY () == K_ESC
          EJECT
          EXIT
        ENDIF

      DO CASE
        CASE cTipo = "I" // Licentatura
          @ 11,55 SAY "SEMESTRE: " + alumnos->(csemalumno)
          @ 13,1 SAY " " + REPLICATE("=",77)+" "
          @ 14,1 SAY " | " + MATERIA + " | " + CALIFICACION + " | "
          @ 15,1 SAY " | " + REPLICATE("=",77)+" | "
          x := 1
          nPTotal := 0.0
          temp1 := "mat" + ALLTRIM(STR(x))
          mate := alumnos->(&temp1)
          /* Este bloque sirve para detectar si se pulso la tecla "ESC"
             aqui es donde se cancela la impresion */
          INKEY (.2)
          IF LASTKEY () == K_ESC
            EJECT
            EXIT
          ENDIF

        REPEAT
          materias->(DBSEEK(mate))
          temp2 := "p"+ALLTRIM(STR(x))+ "1"
          temp3 := "p"+ALLTRIM(STR(x))+ "2"

```

```

temp4 := "I"+ALLTRIM(STR(x))+3"
cal1 := alumnos->(&temp2.)
cal2 := alumnos->(&temp3.)
cal3 := alumnos->(&temp4.)
prom := (((cal1+cal2)/2)+cal3)/2

/* Checa si la escuela esta dada de alta en SEP M,xico
   o en SEP Edo. y dependiendo de esto redondea el promedio */
IF cTipoSep .AND. prom % 1 == .5
  prom -= (prom % 1)
ELSE
  prom := ROUND (prom,0)
ENDIF

nPTotal += prom
@ 15+x,1 SAY "I"+materias->nommat
@ 15+x,79 SAY "I"
@ 15+x,64 SAY PADL(ALLTRIM(STR(prom)),2," ")
/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_ESC
  EJECT
  EXIT
ENDIF

x++
// Se evita el error ya que no existiria el campo "MAT14"
IF x == 14
  EXIT
ENDIF
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
UNTIL EMPTY(mate)
nPTotal /= (x-1)

@ 15+x,1 SAY "I"+REPLICATE("-",77)+"I"
@ 16+x,7 SAY cFirma
@ 17+x,1 SAY cCargo
@ 17+x,57 SAY "PROM.: "+ALLTRIM(STR(nPTotal))
@ PROW(), PCOL() SAY CsSendPm2 ("COMPRIMIDA")
@ 18+x,92 SAY "Constancia de Control Interno"
@ PROW(), PCOL() SAY CsSendPm2 ("-COMPRIMIDA")
@ 19+x,1 SAY cComen1
@ 20+x,1 SAY cComen2
/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_ESC
  EJECT
  EXIT
ENDIF
CASE cTipo == "P" // Preparatoria
aProm1 := {0,0,0,0,0,0,0}
aFall := {0,0,0,0,0,0}
aAux := {0,0,0,0,0,0}
ASIZE (aProm2, 13)
FOR x := 1 TO 13
  aProm2 [x] := 0.0
NEXT
@ 11,55 SAY "GRUPO : "+cGrupo
@ 13,1 SAY REPLICATE("-",84)
@ PROW(), PCOL() SAY CsSendPm2 ("COMPRIMIDA")
@ 14,1 SAY "MATERIA EVALUACIONES"
@ 15,39 SAY "PRIMER SEGUNDO TERCER PROME EXAMEN PROME CUARTO QUINTO SEXTO PROME
EXAMEN CALIF RESULTADO"
@ 16,39 SAY "CAL INA CAL INA CAL INA MENS. 1 SEM. SEMES CAL INA CAL INA
CAL INA MENS. 2 SEM. SEMES"
@ 17,1 SAY REPLICATE("-",145)
x := 1
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)

```

```

/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_ESC
  EJECT
  EXIT
ENDIF

REPEAT
  IFlagProm := .F.
  materias->(DBSEFK(mate))
  @@ 17+x,1 SAY materias->nommat
  y := 6
  aProm1 [1] := 0.0 // Se inicializa para cada materia
  // Imprime las primeras 3 evaluaciones
  FOR i = 1 TO 3
    temp2 := "p"+ALLTRIM(STR(x))+ALLTRIM(STR(i))
    temp3 := "i"+ALLTRIM(STR(x))+ALLTRIM(STR(i))
    call := alumnos->(&temp2.)
    fall := alumnos->(&temp3.) // Se imprimen las faltas
    aProm1 [1] += call // Promedios horizontales
    aFall [i] += fall // Suma de faltas
    aProm2 [i] += call
    IF call # 0
      @@ 17+x,32+y SAY STR(call)+SPACE(1)+STR(fall)
    ELSE
      IFlagProm := .T.
    ENDIF
    y+=9
  NEXT
  /* Imprime el primer promedio mensual, solo
     que ya se hayan capturado las primeras
     tres calificaciones */
  IF !IFlagProm
    aProm1 [1] /= 3
    aProm2 [4] += aProm1 [1]
    @@ 17+x,33+y SAY PADL(ALLTRIM(STR(aProm1 [1])),4," ")
    //Imprime la 1ª calificación semestral y el 1er. promedio semestral
    temp2 := "s"+ALLTRIM(STR(x))
    call := alumnos->(&temp2.)
    aProm1 [2] := (aProm1 [1]+call)/2
    aProm2 [5] += call
    aProm2 [6] += aProm1 [2]
    @@ 17+x,40+y SAY STR(call)+SPACE(3)+PADL(ALLTRIM(STR(aProm1 [2])),4," ")
  ENDIF
  y+=22
  aProm1 [3] := 0.0
  FOR i = 4 TO 6
    temp2 := "p"+ALLTRIM(STR(x))+ALLTRIM(STR(i))
    temp3 := "i"+ALLTRIM(STR(x))+ALLTRIM(STR(i))
    call := alumnos->(&temp2.)
    fall := alumnos->(&temp3.) // Se imprimen las faltas
    aProm1 [3] += call
    aFall [i] += fall
    aProm2 [i+3] += call
    IF call # 0
      @@ 17+x,32+y SAY STR(call)+SPACE(1)+STR(fall)
    ELSE
      IFlagProm := .T.
    ENDIF
    y+=9
  NEXT
  IF !IFlagProm
    aProm1 [3] /= 3
    aProm2 [10] += aProm1 [3]
    @@ 17+x,32+y SAY ALLTRIM(STR(aProm1 [3]))
    //Imprime la 2ª calificación semestral y el 2º. promedio semestral
    temp2 := "F"+ALLTRIM(STR(x))
    call := alumnos->(&temp2.)
    aProm1 [4] := (aProm1 [3]+call)/2
    aProm2 [11] += call
    aProm2 [12] += aProm1 [4]
  ENDIF

```

```

aprom2 [13] += (aProm1 [2] + aProm1 [4]) / 2
@ 17+x,119 SAY STR(cal1)+SPACE(4)+PADL(ALLTRIM(STR(aProm1 [4])),4," ")
@ 17+x,137 SAY PADL(ALLTRIM(STR((aProm1 [2]+aProm1 [4])/2),4," ")
/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_ESC
  EJECT
  EXIT
ENDIF
ENDIF
x++
// Se evita el error ya que no existiria el campo "MAT14"
IF x == 14
  EXIT
ENDIF
// SE avanza a la siguiente materia
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
IFlagProm := .F.
UNTIL EMPTY(mate) // hasta que no haya mas materias
@ 17+x,1 SAY REPLICATE("!",145)
@ 18+x,1 SAY "INASISTENCIAS"
@ 18+x,34 SAY aFall [1]
@ 18+x,43 SAY aFall [2]
@ 18+x,52 SAY aFall [3]
@ 18+x,83 SAY aFall [4]
@ 18+x,92 SAY aFall [5]
@ 18+x,101 SAY aFall [6]
@ 19+x,1 SAY "PROMEDIOS"
y := 6
FOR i = 1 TO 3
  @ 19+x,24+y SAY aProm2 [i] / (x-1)
  y += 9
NEXT
y += 1
@ 19+x,57 SAY aProm2 [4] / (x-1)
@ 19+x,64 SAY aProm2 [5] / (x-1)
@ 19+x,71 SAY aProm2 [6] / (x-1)
@ 19+x,78 SAY aProm2 [7] / (x-1)
@ 19+x,87 SAY aProm2 [8] / (x-1)
@ 19+x,96 SAY aProm2 [9] / (x-1)
@ 19+x,104 SAY aProm2 [10] / (x-1)
@ 19+x,111 SAY aProm2 [11] / (x-1)
@ 19+x,119 SAY aProm2 [12] / (x-1)
@ 19+x,129 SAY aProm2 [13] / (x-1)
@ 24+x,7 SAY cFirma
@ 25+x,1 SAY cCargo
@ 26+x,1 SAY cComen1
@ 27+x,1 SAY cComen2
@ PROW(), PCOL() SAY CsSendPm2 ("-COMPRIMIDA")
/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_ESC
  EJECT
  EXIT
ENDIF
CASE cTipo == "G" .OR. cTipo == "D"
@ 11,55 SAY "SEMESTRE " + alumnos->(semesalum)
@ 13,1 SAY "r" + REPLICATE("-",77) + "r"
@ 14,1 SAY " | " + MATERIA + " | " + CALIFICACION + " | "
@ 15,1 SAY "f" + REPLICATE("-",77) + "f"
x := 1
nPTotal := 0.0
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_ESC
  EJECT

```

```

EXIT
ENDIF

REPEAT
materias->(DBSEFK(mate))
temp2 := "u"+ALLTRIM(STR(x))
temp3 := "i"+ALLTRIM(STR(x))
call := alumnos->(&temp2.)
cal2 := alumnos->(&temp3.)

/* Checa si la escuela esta dada de alta en SEP México
   o en SEP Edo. y dependiendo de esto redondea el promedio */
IF cTipoSep .AND. call % 1 == .5
  call := (call % 1)
ELSE
  call := ROUND (call,0)
ENDIF

nPTotal += call
@ 15+x,1 SAY "I"+materias->nommat
@ 15+x,79 SAY "I"
@ 15+x,64 SAY PADL(ALLTRIM(STR(call)),2," ")
/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_FSC
  EJECT
  EXIT
ENDIF

x++
// Se evita el error ya que no existiria el campo "MAT14"
IF x == 14
  EXIT
ENDIF
temp1 := "mat" + ALLTRIM(STR(x))
mate := alumnos->(&temp1.)
UNTIL EMPTY(mate)
nPTotal /= (x-1)

@ 15+x,1 SAY "I"+REPLICATE("-",77)+"I"
@ 16+x,7 SAY cFirma
@ 17+x,1 SAY cCargo
@ 17+x,57 SAY "PROM.: "+ALLTRIM(STR(nPTotal))
@ PROW(), PCOL() SAY CsScndPm2 ("+COMPRIMIDA")
@ 18+x,92 SAY "Constancia de Control Interno"
@ PROW(), PCOL() SAY CsScndPm2 ("-COMPRIMIDA")
@ 19+x,1 SAY cComen1
@ 20+x,1 SAY cComen2
/* Este bloque sirve para detectar si se pulso la tecla "ESC"
   aqui es donde se cancela la impresi3n */
INKEY (.2)
IF LASTKEY () == K_ESC
  EJECT
  EXIT
ENDIF

OTHERWISE
TONE (800)
ALERT ("TIPO DE ESCUELA NO VALIDO")
END CASE
EJECT
alumnos->(DBSKIP())
ELSE
alumnos->(DBSKIP())
ENDIF

UNTIL alumnos->grupo # cGrupo .OR. alumnos->(EOF())
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
RESTSCREEN (9,18,16,63, cuadro)
EXIT
ELSE

```

```

TONE (350)
nOp := ALERT ("La impresora no esta lista",aOpcion)
IF nOp == 2
    EXIT
ENDIF
ENDIF
ENDDO
materias->(DBCLOSEAREA())
escapes->(DBCLOSEAREA())
impres->(DBCLOSEAREA())
ENDDO
RETURN

***** FUNCIONES PARA LA IMPRESION DE CONSTANCIAS *****

// La siguiente funcion permite al usuario elaborar constancias de
// estudio e imprimirlas a traves de un editor propio del SCE
FUNCTION editor()
    LOCAL nPmActive, cPuerto, aPuertos, aCapCfg, nPm := NmImp;;
    pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1);;
    cColor := SETCOLOR (), cColor2, ok := "N";;
    aOpcion := {"REINTENTAR","CANCELAR"}, cuadro, opcion := {"REINTENTAR", "CANCELAR"}
PRIVATE impresora, consta, lTermina := .F.
/* Vars. que indican la posición donde se desplegara el nombre de la
impresora si est es cambiada */
PUBLIC x := 20,y := 32

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
    RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
    RETURN
ENDIF

ok_bases := abre_bases (23) // Base del editor
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DEL EDITOR")
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1, pantalla)
    RETURN
ENDIF

pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1) // Se resguarda el area de pantalla de captura
SET CURSOR ON
impresora := CsPmName2(nPm) // Inicializa la impresora
impres->(DBCLOSEAREA())
@ 9, 15 SAY REPLICATE ("-",60)

pie(14)
DISPBOX (MAXROW()-5,1,MAXROW()-3,maxcol()-1," ┌───┴───┐",C_REYENO) // Se dibuja la ventana de captura
DISPBOX (10,1,MAXROW()-6,maxcol()-1," ┌───┴───┐",C_REYENO) // Se dibuja la ventana de captura
pantallacampos(17)
cColor2 := SETCOLOR(C_CAP)

SET KEY K_F2 TO cambia_impresora ()
@ 20, 32 SAY impresora COLOR C_CONFIG4

SET KEY K_F11 TO imprime ()
DO WHILE !lTermina
    consta := memoedit(edit->edit,11,3,17,77,T,,"teclas",73)
    IF !LASTKEY () # K_ESC
        IF lockr (10)
            REPLACE edit WITH consta
            DBCOMMIT()
        
```

```

        DBUNLOCK()
    ELSE
        TONE (750)
        ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO POR OTRA ESTACION", opcion)
        IF opcion == 2
            EXIT
        ENDIF
    ENDIF
ENDIF
ENDDO
SET KEY K_F11 TO
escapes->(DBCLOSEAREA())
edit->(DBCLOSEAREA())
RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla) // Se resguarda el area de pantalla de captura
SETCOLOR(cColor)
RETURN

```

```

FUNCTION teclas(modo, fila, columna)
    LOCAL tecla := LASTKEY(), salir, ok := {"NO","SI"}
    DO CASE
        CASE modo == ME_UNKEYX
            DO CASE
                CASE tecla == K_ESC
                    TONE (750)
                    salir := ALERT (" FINALIZAR SIN GUARDAR CAMBIOS ?",ok)
                    IF salir == 1
                        RETURN ME_IGNORE
                    ELSE
                        !Termina := .T.
                        RETURN ME_DEFAULT
                    ENDIF
                CASE tecla == K_F10
                    RETURN 23
            ENDCASE
        CASE modo == ME_UNKEY
            DO CASE
                CASE tecla == K_ESC
                    !Termina := .T.
            ENDCASE
    ENDCASE
RETURN

```

```

FUNCTION imprime()
    LOCAL nPmActive, cPuerto, aPuertos, aCapCfg, nprn := Nmlmp
    LOCAL aOpcion := {"REINTENTAR","CANCELAR"}, cuadro
    LOCAL cColor, color, nOp, ok := "N"

```

```

    DO WHILE .T.
        cColor := SETCOLOR ("R/B+")
        @ 20,65 SAY "Imprimir :"
        SETCOLOR (cColor)
        @ 20,76 GET ok PICTURE "!A" VALID (ok$"SN")
        READ
        cColor := SETCOLOR (C_BORRADO)
        @ 20,61 SAY REPLICATE ("█",16)
        SETCOLOR (cColor)

```

```

    IF ok == "N" .OR. LASTKEY () == K_ESC
        RETURN
    ENDIF
    ok := "N"

```

```

    DO WHILE .T.
        edit-:(DBGOTOP())
        IF ISPRINTER()

```

```

            cuadro := SAVESCREEN (9,18,16,63)
            DISBOX (9,18,15,62,"┌──┴──┐",C_REYENO3) // Se dibuja la ventana de captura
            color := SETCOLOR ("R+/N")
            @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
            SETCOLOR (color)
            color := SETCOLOR ("GR+*/N")

```



```

@ 12,21 SAY " SE ESTA IMPRIMIENDO LA CONSTANCIA"
SETCOLOR (color)
color := SETCOLOR ("GR+N")
@ 14,25 SAY "PRESIONE " COLOR C_CONFIG2
@ 14,34 SAY "<ESC>" COLOR C_CONFIG4
@ 14,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
SETCOLOR (color)

SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "PRINTER")
@ 5, 1 SAY edit->(edit)
EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")

RESTSCREEN (9,18,16,63, cuadro)
RETURN
ELSE
TONE (350)
nOp := ALERT ("La impresora no esta lista",aOpcion)
IF nOp == 2
EXIT
ENDIF
ENDIF
ENDDO
ENDDO
RETURN

/** FUNCIONES PARA LA IMPRESION DE CONCENTRADOS DE CALIFICACIONES **/

// Función que imprime los concentrados de calificaciones por materia
FUNCTION concen_mats ()
LOCAL ok_bases, cColor2 := SETCOLOR ();
pantalla, pantalla2, flag := .F., ok := "N", nprn := NmImp;
aOpcion := {"REINTENTAR","CANCELAR"}, cuadro, cSemestre, nSemestre;
cMacstro, temp1,temp2, temp3, temp4, prom, renglon;
cJefe := SPACE (35), cDirec := SPACE(35);
sigue,aOpcion2 := {"Continuar","Cancelar"},flag2 := .F. // Temporales no hay formas continuas
PRIVATE impresora
PUBLIC cGrupo := SPACE(3), mClavMat := SPACE(5)
/* Vars. que indican la posición donde se desplegara el nombre de la
impresora si est. es cambiada */
PUBLIC x := 18, y := 32

IF !FILE("grupo1.ntx")
TONE (700)
ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
RETURN
ENDIF

ok_bases := abre_bases (10) // Base de grupos
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
RETURN
ENDIF

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
RETURN
ENDIF

ok_bases := abre_bases (3) // Base de materias
IF !ok_bases
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")

```



```

SET CURSOR ON
@ 13, 18 GET mClavMat PICTURE "@!" VALID (!EMPTY(mClavMat))
READ
SET KEY K_F1 TO
SET CURSOR OFF
IF LASTKEY () == K_ESC
  EXIT
ELSE
  materias->(DBSEEK(mClavMat))
  IF materias->(FOUND())
    cNomMat := materias->(NomMat)
  ELSE
    TONE (350)
    ALERT ("LA MATERIA NO ESTA DADA DE ALTA, NO ESTA ASIGNADA A NINGUN GRUPO Y POR LO TANTO NO ESTA
ASIGNADA A NINGUN ALUMNO")
    LOOP
  ENDIF
ENDIF

// Se tiene que checar que la materia este asigna a algun alumno
// de este grupo
grupos->(DBGOTOP ())
grupos->(DBSEEK (cGrupo))
IF grupos->(FOUND())
  alumnos->(DBSETORDER(3))
  alumnos->(DBSEEK (cGrupo))
  IF alumnos->(FOUND())
    REPEAT
      FOR i = 1 to 13
        aux := "mat" + ALLTRIM(STR(i))
        IF alumnos->(&aux) == mClavMat
          flag := .T.
          EXIT
        ENDIF
      NEXT
    IF flag
      EXIT
    ENDIF
    alumnos->(DBSKIP())
  UNTIL alumnos->(EOF()) .OR. alumnos->grupo # cGrupo
  IF !flag
    TONE (800)
    ALERT ("LA MATERIA NO ESTA ASIGNADA A NINGUN ALUMNO DE ESTE GRUPO")
    LOOP
  ELSE
    flag := .F.
  ENDIF
ELSE
  TONE (800)
  ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO O ACABAN DE SER DADOS DE BAJA EN ALGUNA OTRA
ESTACION DE LA RED")
  LOOP
ENDIF
ELSE
  TONE (800)
  ALERT ("EL GRUPO ACABA DE SER DADO DE BAJA EN ALGUNA OTRA ESTACION DE LA RED")
  LOOP
ENDIF
SET CURSOR ON
@a 15, 24 GET cJefe PICTURE "@!" VALID (!EMPTY(cJefe))
@a 16, 24 GET cDues PICTURE "@!" VALID (!EMPTY(cDues))
READ
SET CURSOR OFF
IF LASTKEY () == K_ESC
  LOOP
ENDIF

pie(13)
SET CURSOR ON
cColor := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro de IMPRIMIR :'"
SETCOLOR (cColor)

```

```

@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN")
READ
SET CURSOR OFF
cColor := SETCOLOR (C_BORRADO)
@ 20,5 SAY REPLICATE ("0",30)
SETCOLOR (cColor)

IF ok == "N" .OR. LASTKEY () == K_FSC
  LOOP
ENDIF
ok := "N"
SET KEY K_F1 TO

pic (1)

// Se determina en que semestre estan inscritos los alumnos de este grupo
nSemestre := VAL (ALLTRIM(alumnos->semesalum))
DO CASE
  CASE nSemestre == 1
    cSemestre := "Primer Semestre"
  CASE nSemestre == 2
    cSemestre := "Segundo Semestre"
  CASE nSemestre == 3
    cSemestre := "Tercer Semestre"
  CASE nSemestre == 4
    cSemestre := "Cuarto Semestre"
  CASE nSemestre == 5
    cSemestre := "Quinto Semestre"
  CASE nSemestre == 6
    cSemestre := "Sexto Semestre"
  CASE nSemestre == 7
    cSemestre := "Septimo Semestre"
  CASE nSemestre == 8
    cSemestre := "Octavo Semestre"
  CASE nSemestre == 9
    cSemestre := "Noveno Semestre"
  CASE nSemestre == 10
    cSemestre := "Decimo Semestre"
  OTHERWISE
    cSemestre := STR(nSemestre)
END CASE

/* En el siguiente bloque se recupera el nombre del maestro que
  imparte la materia seleccionada. Para ello se checa que todo
  este correcto con respecto a la clave del maestro */
grupos->(DBSETORDER(4))
grupos->(DBSEFK(cGrupo+mClavMat))
IF grupos->(FOUND())
  ok_bases := abre_bases (5) // Abre la base de maestros
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS NO PUEDE SEGUIR EJECUTANDO ESTE MODULO")
    RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    @ MAXROW()-1,1 SAY SPACE (78)
    DBCLOSEALL()
    RETURN -1
  ENDIF
  DBSEFK(maestros->(grupos->clavmae))
  IF maestros->(FOUND())
    cMaestro := maestros->nommae
    maestros->(DBCLOSEAREA())
  ELSE
    TONE (750)
    ALERT ("ERROR : El maestro asignado a esta materia no esta dado de alta, NO SE PUEDE CONTINUAR CON ESTA MATERIA")
    maestros->(DBCLOSEAREA())
    LOOP
  ENDIF
ELSE
  TONE (750)
  ALERT ("LA MATERIA NO FORMA PARTE DE LA CARGA DE MATERIAS DE ESTE GRUPO")
  LOOP
ENDIF

```

```

ENDIF
// Aqui termina el bloque de recuperaci3n de nombre del maestro

DO WHILE .T.
  IF ISPRINTER()
    cuadro := SAVESCREEN (12,18,19,63)
    DISPBOX (12,18,18,62,"┌──┴──┐",C_REYENO3) // Se dibuja la ventana de captura
    color := SETCOLOR ("R+/N")
    @ 14,21 SAY "  ESPERE UN MOMENTO POR FAVOR"
    SETCOLOR (color)
    color := SETCOLOR ("GR+*/N")
    @ 15,24 SAY "SE ESTA IMPRIMIENDO EL CONCENTADO"
    SETCOLOR (color)
    color := SETCOLOR ("GR+/N")
    @ 17,25 SAY "PRESIONE " COLOR C_CONFIG2
    @ 17,34 SAY " <ESC> " COLOR C_CONFIG4
    @ 17,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
    SETCOLOR (color)

    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "PRINTER")
    @ PROW(), PCOL() SAY CsSendPm3 ("INICIO", m->NmImp)
    INKEY (.2) // Permite abortar la impresi3n
    IF LASTKEY () == K_ESC
      @ 0, PCOL() SAY CHR(27)
      EXIT
    ENDIF
    @ 4,30 SAY cSemestre
    @ 4,68 SAY cGrupo
    @ PROW(), PCOL() SAY CsSendPm3 ("COMPRIMIDA", m->NmImp)
    @ 7,27 SAY cNomMat
    @ 9,35 SAY cMaestro
    @ 9,101 SAY esc_selec
    renglon := 13
    REPEAT
      // Se busca la materia para recuperar su informaci3n
      FOR n = 1 TO 13
        temp1 := "mat" + ALLTRIM(STR(n))
        IF alumnos->&temp1. == mClavMat .AND. EMPTY(alumnos->baja) .AND. alumnos->inscrito
          // Se imprime el nombre del alumno

          @ renglon,6 SAY PADL(ALLTRIM(STR(renglon - 12)),2," ")
          @ renglon,12 SAY alumnos->nomalum

          // Se imprimen las asistencias
          temp2 := "i" + ALLTRIM(STR(n)) + "1"
          temp3 := "i" + ALLTRIM(STR(n)) + "2"
          temp4 := "i" + ALLTRIM(STR(n)) + "3"
          @ renglon,62 SAY alumnos->&temp2. // asistencias primer periodo
          @ renglon,65 SAY alumnos->&temp3. // asistencias primer periodo
          @ renglon,68 SAY alumnos->&temp4. // asistencias primer periodo
          prom := alumnos->&temp2. + alumnos->&temp3. + alumnos->&temp4.
          @ renglon,78 SAY PADL(ALLTRIM(STR(prom)),2," ") // Total de asistencias

          // Ahora las calificaciones
          temp2 := "p" + ALLTRIM(STR(n)) + "1"
          temp3 := "p" + ALLTRIM(STR(n)) + "2"
          temp4 := "p" + ALLTRIM(STR(n)) + "3"
          @ renglon,81 SAY alumnos->&temp2. // Primer Parcial
          @ renglon,85 SAY alumnos->&temp3. // Segundo Parcial
          prom := (alumnos->&temp2. + alumnos->&temp3.) / 2
          @ renglon,91 SAY PADL(ALLTRIM(STR(prom)),2," ") // Promedio de Parciales
          @ renglon,99 SAY alumnos->&temp4. // Examen final

          prom += alumnos->&temp4.
        ENDIF
      ENDFOR
      prom /= 2

      /* Checa si la escuela esta dada de alta en SEP M,xico
      o en SEP Edo. y dependiendo de esto redondea el promedio */
      IF cTipoSep .AND. prom % 1 == .5
        prom -= (prom % 1)
      ENDIF
    ENDREPEAT
  ENDIF
ENDWHILE

```

```

ELSE
  prom := ROUND (prom,0)
ENDIF
(@ renglon,109 SAY PADI(ALLTRIM(STR(prom)),2," ") // Calificación definitiva
renglon ++
ENDIF
NEXT

INKEY (.2) // Permite abortar la impresión
IF LASTKEY () == K_ESC
  @ 0, PCOL() SAY CHR(27)
  EXIT
ENDIF
alumnos->(DBSKIP())
IF renglon > 62
  @ 66,11 SAY cMaestro
  @ 66,85 SAY cJefe
  @ 69,11 SAY cDirec
  @ 70,90 SAY DAY(DATE())
  @ 70,100 SAY CMONTH(DATE())
  @ 70,126 SAY YEAR(DATE())
  EJECT
  // Se realiza el salto de pagina
  TONE (750)
  sigue := ALERT ("INSERTE NUEVA HOJA", aOpcion2)
  IF sigue == 2
    flag2 := .T.
    exit
  ENDIF
  @ PROW(), PCOL() SAY CsSendPm3 ("-COMPRIMIDA", m->NmImp)
  @ 4,30 SAY cSemestre
  @ 4,68 SAY cGrupo
  @ PROW(), PCOL() SAY CsSendPm3 ("COMPRIMIDA", m->NmImp)
  @ 7,27 SAY cNomMat
  @ 9,35 SAY cMaestro
  @ 9,101 SAY esc_selec
  renglon := 13
ENDIF
UNTIL alumnos->grupo # cGrupo .OR. alumnos->(EOF())
IF !flag2
  @ 66,11 SAY cMaestro
  @ 66,85 SAY cJefe
  @ 69,11 SAY cDirec
  @ 70,90 SAY DAY(DATE())
  @ 70,100 SAY CMONTH(DATE())
  @ 70,126 SAY YEAR(DATE())
ELSE
  flag2 := .F.
ENDIF
@ PROW(), PCOL() SAY CsSendPm3 ("-COMPRIMIDA", m->NmImp)
EXIT
ELSE
  TONE (350)
  nOp := ALERT ("La impresora no esta lista",aOpcion)
  IF nOp == 2
    EXIT
  ENDIF
ENDIF
ENDIF
ENDDO
EJECT
// Se restablece la salida a pantalla
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
RESTSCREEN (12,18,19,63,cuadro) // Se borra el cuadro de dialogo
ENDDO
ENDDO

RETURN

// Función que imprime los concentrados de calificaciones por grupo
FUNCTION concen_grupo ()
  LOCAL ok_bases, cColor2 := SETCOLOR (,);
  pantalla, pantalla2, flag := .F., ok := "N",npm := NmImp;

```

```

aOpcion := {"REINTENTAR","CANCELAR"}, cuadro, cSemestre, nSemestre;;
cMaestro, temp1, temp2, temp3, temp4, prom, renglon, flag3 := .F.;;
cLefe := SPACE(35), cDirec := SPACE(35), bandera := .F.;;
sigue, aOpcion2 := {"Continuar","Cancelar"}, flag2 := .F. // Temporales no hay formas continuas
PRIVATE impresora
PUBLIC cGrupo := SPACE(3), mClavMat := SPACE(5)
/* Vars. que indican la posición donde se desplegara el nombre de la
  impresora si est es cambiada */
PUBLIC x := 15, y := 32

IF !FILE("grupo1.ntx")
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
  RETURN
ENDIF

ok_bases := abre_bases (10) // Base de grupos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
  RETURN
ENDIF

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  RETURN
ENDIF

ok_bases := abre_bases (3) // Base de materias
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE MATERIAS")
  RETURN
ENDIF

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
  RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
  RETURN
ENDIF

pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

limpiamenus (2)
@ 9, 2 SAY REPLICATE ("-",70)

impresora := CsPmName2(nPm) // Inicializa la impresora
impres->(DHCLOSEAREA())

DO WHILE .T.
  pie (13)
  DISBOX (10,1,MAXROW()-3,maxcol()-1,"┌──┴──┐",C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (19)
DO WHILE .T.

SET KEY K_F2 TO cambia_impresora ()
@ x,y SAY impresora COLOR C_CONFIG4

cGrupo := SPACE(3)
SET CURSOR ON
@ 11, 18 GET cGrupo PICTURE "(a)R(a) X9-A" VALID (!EMPTY(cGrupo))

```



```

READ
SET CURSOR OFF
IF LASTKEY () == K_ESC
  RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
  SETCOLOR (cColor2)
  @ MAXROW()-1,1 SAY SPACE (78)
  DBCLOSEALL()
  RETURN -1
ELSE
  grupos->(DBGOTOP ())
  grupos->(DBSEEK (cGrupo))
  IF grupos->(FOUND())
    alumnos->(DBSETORDER(3))
    alumnos->(DBSEEK (cGrupo))
    IF alumnos->(FOUND())
      TONE (800)
      ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
      LOOP
    ENDIF
  ELSE
    TONE (800)
    ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
    LOOP
  ENDIF
ENDIF

pie (15)
SET CURSOR ON
@ 12, 23 GET cJefe PICTURE "@!" VALID (!EMPTY(cJefe))
@ 13, 23 GET cDirec PICTURE "@!" VALID (!EMPTY(cDirec))
READ
SET CURSOR OFF
IF LASTKEY () == K_ESC
  cColor := SETCOLOR (C_BORRADO)
  @ 12,23 SAY REPLICATE ("█",40)
  @ 13,23 SAY REPLICATE ("█",40)
  SETCOLOR (cColor)
  cJefe := SPACE (35)
  cDirec := SPACE (35)
  LOOP
ENDIF

pie (13)
SET CURSOR ON
cColor := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro de IMPRIMIR :"
SETCOLOR (cColor)
@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN")
READ
SET CURSOR OFF
cColor := SETCOLOR (C_BORRADO)
@ 20,5 SAY REPLICATE ("█",30)
SETCOLOR (cColor)

IF ok == "N" .OR. LASTKEY () = K_ESC
  LOOP
ELSE
  EXIT
ENDIF
ok := "N"
ENDDO
pie (1)

// Se determina en que semestre estan inscritos los alumnos de este grupo
nSemestre := VAL (ALLTRIM(alumnos->semesalum))
DO CASE
  CASE nSemestre == 1
    cSemestre := "Primer Semestre"
  CASE nSemestre == 2
    cSemestre := "Segundo Semestre"
  CASE nSemestre == 3
    cSemestre := "Tercer Semestre"

```



```

CASE nSemestre == 4
  cSemestre := "Cuarto Semestre"
CASE nSemestre == 5
  cSemestre := "Quinto Semestre"
CASE nSemestre == 6
  cSemestre := "Sexto Semestre"
CASE nSemestre == 7
  cSemestre := "Septimo Semestre"
CASE nSemestre == 8
  cSemestre := "Octavo Semestre"
CASE nSemestre == 9
  cSemestre := "Noveno Semestre"
CASE nSemestre == 10
  cSemestre := "Decimo Semestre"
OTHERWISE
  cSemestre := STR(nSemestre)
END CASE

cuadro := SAVESCREEN (12,18,19,63)
DISPBOX (12,18,18,62," |J-L| ",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")
@ 14,21 SAY "   ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)
color := SETCOLOR ("GR+*/N")
@ 15,24 SAY "SE ESTA IMPRIMIENDO EL CONCENTADO"
SETCOLOR (color)
color := SETCOLOR ("GR+/N")
@ 17,25 SAY "PRESIONE " COLOR C_CONFIG2
@ 17,34 SAY "<ESC> " COLOR C_CONFIG4
@ 17,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
SETCOLOR (color)

// Se repite hasta que ya no haya materias en este grupo
REPEAT
  alumnos->(DBSETORDER(3))
  alumnos->(DBSEEK (cGrupo))
  IF alumnos->(FOUND())
    IF ISPRINTER ()
      renglon := 13
      /* Se hace una pre-busqueda para saber si la materia
      esta asignada a por lo menos un alumno dentro del
      grupo, de no ser asi, esta materia no se imprimira */

      REPEAT
        FOR i = 1 TO 13
          temp1 := "mat"+ ALLTRIM(STR(i))
          IF alumnos->(&temp1) == grupos->clavmat .AND. EMPTY(alumnos->baja) .AND. alumnos->inscrito
            mClavmat := grupos->clavmat
            bandera := .T.
            EXIT
          ENDIF
        NEXT
        IF bandera
          EXIT
        ELSE
          alumnos->(DBSKIP())
        ENDIF
      UNTIL alumnos->grupo # cGrupo .OR. alumnos->(EOF())

      IF bandera
        bandera := .F.
        /* En el siguiente bloque se recupera el nombre del maestro que
        imparte la materia seleccionada. Para ello se checa que todo
        este correcto con respecto a la clave del maestro */

        ok_bases := abre_bases (5) // Abre la base de maestros
        IF !ok_bases
          TONE (750)
          ALERT ("NO SE PUEDE ABRIR LA BASE DE MAESTROS NO PUEDE SEGUIR EJECUTANDO
          ESTE MODULO")
          RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
          SETCOLOR (cColor2)

```

```

@ MAXROW()-1,1 SAY SPACE (78)
DBCLOSEALL()
RETURN -1
ENDIF
DBSEEK(maestros->(grupos->clavmae))
IF maestros->(FOUND())
  cMaestro := maestros->nommae
  maestros->(DBCLOSEAREA())
ELSE
  TONE (750)
  ALERT ("ERROR : El maestro asignado a esta materia no esta dado de alta, NO SE PUEDE CONTINUAR CON ESTA
MATERIA")
  maestros->(DBCLOSEAREA())
  I.OOP
ENDIF
// Aquí termina el bloque de recuperación de nombre del maestro

materias->(DBSEEK(mClavMat))
IF materias->(FOUND())
  cNomMat := materias->(NomMat)
ELSE
  TONE (350)
  ALERT ("ERROR: LA MATERIA NO ESTA DADA DE ALTA")
  grupos->(DBSKIP())
  I.OOP
ENDIF
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "PRINTER")
@ PROW(), PCOL() SAY CsSendPm3 ("INICIO", m->NmImp)
INKEY (.2) // Permite abortar la impresión
IF LASTKEY () == K_ESC
  @ 0, PCOL() SAY CHR(27)
  EXIT
ENDIF
@ 4,30 SAY cSemestre
@ 4,68 SAY cGrupo
@ PROW(), PCOL() SAY CsSendPm3 ("+ COMPRIMIDA", m->NmImp)
@ 7,27 SAY cNomMat
@ 9,35 SAY cMaestro
@ 9,101 SAY esc_selec

REPEAT
  // Se busca la materia para recuperar su información
  FOR n = 1 TO 13
    temp1 := "mat" + ALLTRIM(STR(n))
    INKEY (.2) // Permite abortar la impresión
    IF LASTKEY () == K_ESC
      @ 0, PCOL() SAY CHR(27)
      flag3 := .T.
      EXIT
    ENDIF
    IF alumnos->&temp1. == mClavMat .AND. EMPTY(alumnos->baja) .AND. alumnos->inscrito
      flag2 := .T. // Indica que se va a imprimir el alumno
      // Se imprime el nombre del alumno

      @ renglon,6 SAY PADL(ALLTRIM(STR(renglon - 12)),2," ")
      @ renglon,12 SAY alumnos->nomalum

      // Se imprimen las asistencias
      temp2 := "i" + ALLTRIM(STR(n)) + "1"
      temp3 := "i" + ALLTRIM(STR(n)) + "2"
      temp4 := "i" + ALLTRIM(STR(n)) + "3"
      @ renglon,62 SAY alumnos->&temp2. // asistencias primer periodo
      @ renglon,65 SAY alumnos->&temp3. // asistencias primer periodo
      @ renglon,68 SAY alumnos->&temp4. // asistencias primer periodo
      prom := alumnos->&temp2.+alumnos->&temp3.+alumnos->&temp4.
      @ renglon,78 SAY PADL(ALLTRIM(STR(prom)),2," ") // Total de asistencias

      // Ahora las calificaciones
      temp2 := "p"+ALLTRIM(STR(n))+ "1"
      temp3 := "p"+ALLTRIM(STR(n))+ "2"
      temp4 := "f"+ALLTRIM(STR(n))+ "3"

```

```

@ renglon,81 SAY alumnos->&temp2. // Primer Parcial
@ renglon,85 SAY alumnos->&temp3. // Segundo Parcial
prom := (alumnos->&temp2. + alumnos->&temp3.) / 2
@ renglon,91 SAY PADL(ALLTRIM(STR(prom)),2," ") // Promedio de Parciales
@ renglon,99 SAY alumnos->&temp4. // Examen final

prom += alumnos->&temp4.
prom /= 2

/* Checa si la escuela esta dada de alta en SEP M,xico
o en SEP Edo. y dependiendo de esto redondea el promedio */
IF cTipoSep .AND. prom % 1 == .5
    prom = (prom % 1)
ELSE
    prom := ROUND (prom,0)
ENDIF
@ renglon,109 SAY PADL(ALLTRIM(STR(prom)),2," ") // Calificación definitiva
EXIT
ENDIF
NEXT

INKEY (.2) // Permite abortar la impresión
IF LASTKEY () == K_ESC
    @ 0, PCOL() SAY CHR(27)
    flag3 := .T.
    EXIT
ENDIF
alumnos->(DBSKIP())
IF flag2
    renglon ++
    flag2 := .F.
ENDIF
IF renglon > 62
    @ 66,11 SAY cMaestro
    @ 66,85 SAY cJefe
    @ 69,11 SAY cDirec
    @ 70,90 SAY DAY( DATE() )
    @ 70,100 SAY CMONTH( DATE() )
    @ 70,126 SAY YEAR( DATE() )
    EJECT
    // Se realiza el salto de pagina
    TONE (750)
    sigue := ALERT ("INSERTE NUEVA HOJA", aOpcion2)
    IF sigue == 2
        flag2 := .T.
        exit
    ENDIF
    @ PROW(), PCOL() SAY CsSendPm3 ("-COMPRIMIDA", m->NmImp)
    @ 4,30 SAY cSemestre
    @ 4,68 SAY cGrupo
    @ PROW(), PCOL() SAY CsSendPm3 (" +COMPRIMIDA", m->NmImp)
    @ 7,27 SAY cNomMat
    @ 9,35 SAY cMaestro
    @ 9,101 SAY esc_selec
    renglon := 13
ENDIF
UNTIL alumnos->grupo # cGrupo .OR. alumnos->(EOF())
ENDIF

IF !flag2
    @ 66,11 SAY cMaestro
    @ 66,85 SAY cJefe
    @ 69,11 SAY cDirec
    @ 70,90 SAY DAY( DATE() )
    @ 70,100 SAY CMONTH( DATE() )
    @ 70,126 SAY YEAR( DATE() )
    ELSE
        flag2 := .F.
    ENDIF
    @ PROW(), PCOL() SAY CsSendPm3 ("-COMPRIMIDA", m->NmImp)
    EJECT // Termina el presente concentrado
ELSE

```

```

    TONE (350)
    nOp := ALERT ("La impresora no esta lista",aOpcion)
    IF nOp == 2
        EXIT
    ENDIF
ENDIF
ELSE
    TONE (350)
    nOp := ALERT ("EL GRUPO ACABA DE SER DADO DE BAJA EN ALGUNA OTRA ESTACION DE LA RED")
    EXIT
ENDIF
grupos->(DBSKIP()) // Pasa a la siguiente materia
UNTIL grupos->Grupo # cGrupo .OR. grupos->(EOF()) .OR. flag3
flag3 := .F.

// Se restablece la salida a pantalla
SET (_SET_PRINTFILE, AJ.LTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
RESTSCREEN (12,18,19,63,cuadro) // Se borra el cuadro de dialogo
ENDDO
RETURN

*Función que actualiza la información del kardex
FUNCTION actualiza_kardex ()
    LOCAL cColor := SETCOLOR(), cColor2,cuadro, aSigue := {"No","Si"}, ok,;
    ok_bases, temp1, temp2, temp3, temp4, cal1, cal2, cal3, prom

/* Se abren las bases de datos necesarias y se crean los indices del
kardex (en caso necesario) */

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
    SETCOLOR (cColor)
    RETURN
ENDIF

ok_bases := abre_bases (21) // Base de INICIO
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE INICIO")
    SETCOLOR (cColor)
    alumnos->(DBCLOSEAREA())
    RETURN
ENDIF

IF inicio->kardex == .T.
    TONE (800)
    TONE (1000)
    TONE (1200)
    ALERT ("EL CARDEX YA HA SIDO ACTUALIZADO PARA ESTE PERIODO")
    SETCOLOR (cColor)
    alumnos->(DBCLOSEAREA())
    inicio->(DBCLOSEAREA())
    RETURN
ENDIF

IF !FILE("kardex1.ntx")
    ok_bases := abre_bases (24) // Base de kardex en exclusiva
    IF !ok_bases
        TONE (750)
        ALERT ("NO SE PUEDE ABRIR LA BASE DE KARDEX")
        SETCOLOR (cColor)
        alumnos->(DBCLOSEAREA())
        inicio->(DBCLOSEAREA())
        RETURN
    ENDIF
    INDEX ON Matrialum+semestre TO &cPath\kardex1
    INDEX ON Grupo+ClavMat+MatriAlum TO &cPath\kardex2
    INDEX ON Actualiza TO &cPath\kardex3

```

```

INDEX ON grupo TO &cPath\kardex4
kardex->(DBCLOSEAREA())
ok_bases := abre_bases (25) // Base de kardex compartida
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE KARDEX")
  SETCOLOR (cColor)
  alumnos->(DBCLOSEAREA())
  inicio->(DBCLOSEAREA())
  RETURN
ENDIF
ELSE
  ok_bases := abre_bases (25) // Base de kardex compartida
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE KARDEX")
    SETCOLOR (cColor)
    alumnos->(DBCLOSEAREA())
    inicio->(DBCLOSEAREA())
    RETURN
  ENDIF
ENDIF

// Se pide al usuario confirme la decisi3n de realizar este proceso
TONE (800)
TONE (1000)
TONE (1200)
ok := ALERT ("Esta totalmente seguro de continuar",aSigue)
IF ok == 1
  alumnos->(DBCLOSEAREA())
  kardex->(DBCLOSEAREA())
  inicio->(DBCLOSEAREA())
  SETCOLOR (cColor)
  RETURN
ENDIF

DISPBEGIN ()
cuadro := SAVESCREEN (9,18,16,63)
DISPBOX (9,18,15,62,"┌──┐├─┤",C_REYENO3) // Se dibuja la ventana de captura
cColor2 := SETCOLOR ("R+N")
@ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (cColor2)
cColor2 := SETCOLOR ("GR+*/N")
@ 12,26 SAY "SE ESTA REALIZANDO EL PROCESO"
SETCOLOR (cColor2)
cColor2 := SETCOLOR ("GR+*/N")
@ 14,25 SAY "PRESIONE " COLOR C_CONFIG2
@ 14,34 SAY "<ESC>" COLOR C_CONFIG4
@ 14,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
SETCOLOR (cColor2)
DISPEND ()

kardex->(var_ini()) // Se inicializan y declaran las variables

// Se repite este proceso para todos los alumnos inscritos
alumnos->(DBGOTOP()) // Para asegurarse de que sean todos
DO WHILE alumnos->(!EOF())
  INKEY (.1) // Permite abortar el proceso
  IF LASTKEY () == K_ESC
    SETCOLOR (cColor)
    RESTSCREEN (9,18,16,63, cuadro)
    inicio->(DBCLOSEAREA())
    alumnos->(DBCLOSEAREA())
    kardex->(DBCLOSEAREA())
    TONE (350)
    TONE (700)
    TONE (1000)
    ALERT ("EL PROCESO A SIDO ABORTADO POR EL USUARIO")
    RETURN
  ENDIF

  IF alumnos->inscrito .AND. EMPTY(alumnos->baja)

```

```

mMatrIalUm := alumnos->matrIalUm
mGrupo := alumnos->grupo
mSemestre := alumnos->SemesAlum
mActualiza := DATE()
FOR x = 1 TO 13
  temp1 := "mat"+ALLTRIM(STR(x))
  INKEY (.1) // Permite abortar el proceso
  IF LASTKEY() = K_ESCAPE
    SETCOLOR (cColor)
    RESTSCREEN (9,18,16,63, cuadro)
    inicio->(DBCLOSEAREA())
    alumnos->(DBCLOSEAREA())
    kardex->(DBCLOSEAREA())
    TONE (350)
    TONE (700)
    TONE (1000)
    ALERT ("EL PROCESO A SIDO ABORTADO POR EL USUARIO")
    RETURN
  ENDIF

  IF !EMPTY(alumnos->(&temp1.))
    mClavmat := alumnos->(&temp1.)
    DO CASE
      CASE cTipo == "L" // Licenciatura
        temp2 := "p"+ALLTRIM(STR(x))+ "1"
        temp3 := "p"+ALLTRIM(STR(x))+ "2"
        temp4 := "p"+ALLTRIM(STR(x))+ "3"
        cal1 := alumnos->(&temp2.)
        cal2 := alumnos->(&temp3.)
        cal3 := alumnos->(&temp4.)
        prom := (((cal1+cal2)/2)+cal3)/2

        /* Checa si la escuela esta dada de alta en SEP M,xico
           o en SEP Edo. y dependiendo de esto redondea el promedio */
        IF cTipoSep .AND. prom % 1 == .5
          prom -= (prom % 1)
        ELSE
          prom := ROUND (prom,0)
        ENDIF

        mCal := prom
        /*
        mExtra :=
        mRegula :=
        mExam := */

      CASE cTipo == "P" // Preparatoria
        nProm2 := 0
        nProm3 := 0
        // Imprime las primeras 3 evaluaciones
        temp2 := "p"+ALLTRIM(STR(x))+ "1"
        mCalM1 := alumnos->(&temp2.)
        temp2 := "p"+ALLTRIM(STR(x))+ "2"
        mCalM2 := alumnos->(&temp2.)
        temp2 := "p"+ALLTRIM(STR(x))+ "3"
        mCalM3 := alumnos->(&temp2.)
        mPromM := (mCalM1 + mCalM2 + mCalM3)/3 // Promedio mensual 1
        temp2 := "s"+ALLTRIM(STR(x))
        mCalS := alumnos->(&temp2.) // Calific semestral 1
        mPromS := (mCalS + mPromM)/2 // Promedio semestral 1

        temp2 := "p"+ALLTRIM(STR(x))+ "4"
        mCalM4 := alumnos->(&temp2.)
        temp2 := "p"+ALLTRIM(STR(x))+ "5"
        mCalM5 := alumnos->(&temp2.)
        temp2 := "p"+ALLTRIM(STR(x))+ "6"
        mCalM6 := alumnos->(&temp2.)
        mPromM2 := (mCalM4 + mCalM5 + mCalM6)/3 // Promedio mensual 2
        temp2 := "p"+ALLTRIM(STR(x))
        mCalF := alumnos->(&temp2.) // Calific semestral 2
        mPromS2 := (mCalF + mPromM2)/2 // Promedio semestral 2
    
```

```

    mFinal := (mPromS+mPromS2)/2 // Promedio FINAL
    tabla(mFinal) // Se ajustan los promedio con la tabla
    CASE cTipo = "G" .OR. cTipo == "D"
        TONE (750)
        ALERT ("NO ESTA DISPONIBLE AUN")
    OTHERWISE
        TONE (800)
        ALERT ("ERROR : TIPO DE ESCUELA NO VALIDO")
    END CASE
ELSE
    EXIT
ENDIF

// Se almacena la información en disco
ok_almacen := kardex->(grabar(1))
IF !ok_almacen
    SETCOLOR (cColor)
    kardex->(DBCLOSEAREA())
    alumnos->(DBCLOSEAREA())
    inicio->(DBCLOSEAREA())
    TONE (350)
    TONE (700)
    TONE (1000)
    ALERT ("ATENCIÓN : EL PROCESO NO HA TERMINADO EXITOSAMENTE, ES NECESARIO REALIZARLO
NUEVAMENTE")
    RETURN
ENDIF
NEXT x
ENDIF
alumnos->(DBSKIP())
ENDDO

// Si se realiza en la totalidad el proceso se actualiza la base de inicio
IF inicio->(lockr (15))
    REPLACE inicio->kardex WITH .T.
    DBCOMMIT()
    DBUNLOCK()
ELSE
    TONE (750)
    ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO POR OTRA ESTACION", opcion)
    IF opcion == 2
        TONE (350)
        TONE (700)
        TONE (1000)
        ALERT ("ATENCIÓN : EL PROCESO NO HA TERMINADO EXITOSAMENTE, ES NECESARIO REALIZARLO NUEVAMENTE")
    ENDIF
ENDIF

SETCOLOR (cColor)
RESTSCREEN (9,18,16,63, cuadro)
inicio->(DBCLOSEAREA())
alumnos->(DBCLOSEAREA())
kardex->(DBCLOSEAREA())
TONE (350)
TONE (700)
TONE (1000)
ALERT ("EL PROCESO A TERMINADO EXITOSAMENTE")
RETURN

*Función que contiene la tabla para el ajuste de las calificaciones de preparatoria
FUNCTION tabla (nFinal)
    STATIC aAjuste := {{6,9,7,7,6,6,6,8,8,7,6,6,6},;
        {6,7,8,7,6,6,8,6,9,8,8,8,7},;
        {7,9,8,8,8,8,10,10,9,10,6,8,8},;
        {9,7,10,9,10,9,8,9,9,9,9,9},;
        {10,9,10,10,10,10,10,9,10,10,10,10},;
        {6,8,6,7,5,6,5,7,7,6,5,5,5},;
        }
    LOCAL x

    nFinal := ROUND (nFinal,0)

```

```

IF nFinal < 5
  nFinal := 5
ENDIF

DO CASE
  CASE nFinal == 5
    x := 6

  CASE nFinal == 6
    x := 1
  CASE nFinal == 7
    x := 2
  CASE nFinal == 8
    x := 3
  CASE nFinal == 9
    x := 4
  CASE nFinal == 10
    x := 5
ENDCASE

mCalM1 := aAjuste [x,1]
mCalM2 := aAjuste [x,2]
mCalM3 := aAjuste [x,3]
mPromM := aAjuste [x,4]
mCalS := aAjuste [x,5]
mPromS := aAjuste [x,6]

mCalM4 := aAjuste [x,7]
mCalM5 := aAjuste [x,8]
mCalM6 := aAjuste [x,9]
mPromM2 := aAjuste [x,10]
mCalF := aAjuste [x,11]
mPromS2 := aAjuste [x,12]

mFinal := aAjuste [x,13]
RETURN

*Función que genera e imprime la cfiya
FUNCTION cfiya ()
  LOCAL nPrnActive, cPuerto, aPuertos, aCapCfg, nPrn := NmImp;
  pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1);
  cColor2 := SETCOLOR(), cNomAlum, cSemestre, lSalir := .F., ok := "N";
  aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F., cuadro;
  cGrupo := SPACE(3), aMaterias := {}, getList := {}, i, nCol, nRow
PRIVATE impresora
/* Vars. que indican la posición donde se desplegará el nombre de la
impresora si está cambiada */
PUBLIC x : 14, y : 32

IF !FILE("kardex1.ntx")
  TONE (700)
  ALERT ("NO SE HA GENERADO EL KARDEX PARA ESTA ESCUELA")
  RETURN
ENDIF
ok_bases := abre_bases (25) // Abre la base de kardex
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE KARDEX")
  SETCOLOR (cColor2)
  RETURN
ENDIF

IF !FILE("grupo1.ntx")
  TONE (700)
  ALERT ("NO SE HAN FORMADO GRUPOS PARA ESTA ESCUELA")
  RETURN
ENDIF
ok_bases := abre_bases (10) // Abre la base de grupos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
  SETCOLOR (cColor2)

```



```

kardex->(DBCLOSEAREA())
RETURN
ENDIF

IF !FILE("alumno1.ntx")
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
  RETURN
ENDIF

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  SETCOLOR (cColor2)
  kardex->(DBCLOSEAREA())
  grupos->(DBCLOSEAREA())
  RETURN
ENDIF

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
  SETCOLOR (cColor2)
  kardex->(DBCLOSEAREA())
  grupos->(DBCLOSEAREA())
  alumnos->(DBCLOSEAREA())
  RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
  SETCOLOR (cColor2)
  kardex->(DBCLOSEAREA())
  grupos->(DBCLOSEAREA())
  alumnos->(DBCLOSEAREA())
  impres->(DBCLOSEAREA())
  RETURN
ENDIF

impresora := CsPmName2(nPm) // Inicializa la impresora
impres->(DBCLOSEAREA())

limpiamenus (2)
@ 9, 15 SAY REPLICATE ("A",60)
cColor2 := SETCOLOR(C_CAP)
DO WHILE .T.
  pie (13)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1,"┌───┴───┐",C_REYENO) // Se dibuja la ventana de captura
  pantallacampo (12)
  DO WHILE .T.
    SET KEY K_F2 TO cambia_impresora ()
    @ x,y SAY impresora COLOR C_CONFIG4

    cGrupo := SPACE(3)
    ASIZE (aMaterias, 0)
    SET CURSOR ON
    @ 11, 13 GET cGrupo PICTURE "@R@! X9-A" VALID (!EMPTY(cGrupo))
    READ
    IF LASTKEY () == K_ESC
      RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
      SETCOLOR (cColor2)
      @ MAXROW()-1,1 SAY SPACE (78)
      DBCLOSEALL()

      RETURN NIL
    ELSE
      grupos->(DBSETORDER(1))
      grupos->(DBGOTOP ())
      grupos->(DBSEEK (cGrupo))
      IF grupos->(FOUND())

```

```

grupos->(DBSETORDER(4))
alumnos->(DBSETORDER(4))
alumnos->(DBSEEK (cGrupo))
IF alumnos->(FOUND())
  EXIT
ELSE
  TONE (800)
  ALERT ("NO HAY ALUMNOS INSCRITOS EN ESTE GRUPO")
ENDIF
ELSE
  TONE (800)
  ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
ENDIF
ENDIF
ENDDO
cColor := SETCOLOR ("R/B+")
@ 20,5 SAY "Esta seguro de IMPRIMIR :)"
SETCOLOR (cColor)
@ 20,31 GET ok PICTURE "!A" VALID (ok$"SN")
READ
IF ok == "N" .OR. LASTKEY () == K_ESC
  LOOP
ENDIF
SET KEY K_F2 TO
ok := "N"
pie(1)
SET CURSOR OFF

*.....Se obtienen las materias
grupos->(DBSETORDER(1))
grupos->(DBSEEK(cGrupo))
DO WHILE grupos->grupo == cGrupo
  AADD (aMaterias,grupos->clavmat)
  grupos->(DBSKIP())
ENDIF
aMaterias := ASORT(aMaterias)

DO WHILE .T.
  IF ISPRINTER()
    cuadro := SAVESCREEN (9,18,16,63)
    DISPBOX (9,18,15,62,"┌──┴──┐",C_REYENO3) // Se dibuja la ventana de captura
    color := SETCOLOR ("R+/N")
    @ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
    SETCOLOR (color)
    color := SETCOLOR ("GR+*N")
    @ 12,25 SAY "SE ESTAN IMPRIMIENDO LAS LISTAS"
    SETCOLOR (color)
    color := SETCOLOR ("GR+/N")
    @ 14,25 SAY "PRESIONE " COLOR C_CONFIG2
    @ 14,34 SAY "<ESC>" COLOR C_CONFIG4
    @ 14,41 SAY " PARA CANCELAR" COLOR C_CONFIG2
    SETCOLOR (color)

    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "PRINTER")
    @ PROW(), PCOL() SAY CsSendPm3 ("+COMPRIMIDA", m->NmImp)
    kardex->(DBSETORDER(2)) // Indice de grupo + materia
    alumnos->(DBSEEK(cGrupo))
    nRow := 1
    REPEAT
      nCol := 3
      i := 1
      nCol += 7

      *//////////////////////////////////////
      /* Unicamente se imprimen los alumnos que esten inscrito y //
      no esten dados de baja temporal */ //
      DO WHILE (!alumnos->inscrito .OR. !EMPTY(ALLTRIM(alumnos->baja))) .AND.; //
        alumnos->grupo == cGrupo //
        alumnos->(DBSKIP()) //
      END DO //
      If alumnos->grupo # cGrupo //

```

```

EXIT //
ENDIF //
*////////////////////////////////////
@ nRow,3 SAY alumnos->nomalum
@ nRow,50 SAY alumnos->matrrialum

kardex- (DBGOTOP())

DO WHILE i<= LEN(aMaterias)
  kardex->(DBSEEK(cGrupo+aMaterias[i]+alumnos->matrrialum))
  IF kardex->(FOUND())
    @ nRow,50+nCol SAY kardex->cal
  ENDIF
  i++
  nCol+=3
ENDDO
alumnos->(DBSKIP())
nRow++
UNTIL alumnos->grupo # cGrupo

@ PROW(), PCOL() SAY CsSendPm3 ("-COMPRIMIDA", m->NmImp)
EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
EXIT
ELSE
TONE (350)
nOp := ALERT ("La impresora no esta lista",aOpcion)
IF nOp == 2
  EXIT
ENDIF
ENDDO
RESTSCREEN (9,18,16,63,cuadro)
ENDDO
RETURN NIL

```

UTILERIA.PRG

```
/* Programa   : SISTEMA DE CONTROL ESCOLAR UVAQ VER. 1.0
   Elaborado por : L.S.C. RAFAEL ROMERO DIAZ BARRIGA
   Lenguaje    : CLIPPER 5.2
   Enlazador   : RTLINK 3.14B
*/
```

```
#xcommand REPEAT => do while t.
#xcommand UNTIL <exp> => if <exp>;exit;end;end
```

```
#define B_LEN      9
#define B_TOP      1
#define B_LEFT     2
#define B_BOTTOM   3
#define B_RIGHT    4
#define B_BACKCOLOR 5
#define B_BARCOLOR 6
#define B_DISPLAYNUM 7
#define B_BARCHAR  8
#define B_PERCENT  9
#define B_BOXLINES "┌─┐├─┤└─┘"
```

```
#include "colores.ch" // Incluir las definiciones de todos los colores de pantalla
#include "caos.ch"
```

```
/****** FUNCIONES DE UTILERIAS DEL SISTEMA *****/
```

```
/* Función que inicializa un nuevo semestre. Para esto, pone .F. en todos los
registros en el campo "INSCRITO", además borra e inicializa a 0 (cero)
todas las calificaciones y las inasistencias */
```

```
FUNCTION inicia_sem ()
LOCAL ok_bases, cuadro, color := SETCOLOR () //, temp1, temp2, aSigue := {"No", "Si"}, ok
```

```
ok_bases := abre_bases (22) // Base de inicio en exclusiva
```

```
IF !ok_bases
```

```
  alumnos->(DBCLOSEAREA())
```

```
  asigna->(DBCLOSEAREA())
```

```
  TONE (750)
```

```
  TONE (1000)
```

```
  TONE (1200)
```

```
  ALERT ("ERROR: No se puede habrir en modo exclusivo la base de datos de inicio, esta siendo ocupada en alguna estacion de la red")
```

```
  SETCOLOR (color)
```

```
  RETURN
```

```
ENDIF
```

```
IF !inicio->kardex .AND. !inicio->cfiya
```

```
  inicio->(DBCLOSEAREA())
```

```
  TONE (750)
```

```
  TONE (1000)
```

```
  TONE (1200)
```

```
  ALERT ("ERROR: No se puede iniciar el semestre ya que no se han imprimido el KARDEX y/o la CFIYA o no se ha respaldado la
información, son necesarios estos procesos")
```

```
  SETCOLOR (color)
```

```
  RETURN
```

```
ENDIF
```

```
// Se pide al usuario confirme la decisión de realizar este proceso
```

```
TONE (800)
```

```
TONE (1000)
```

```
TONE (1200)
```

```
ok := ALERT ("Esta totalmente seguro de continuar", aSigue)
```

```
IF ok == 1
```

```
  inicio->(DBCLOSEAREA())
```

```
  SETCOLOR (color)
```

```
  RETURN
```



```

ENDIF

ok_bases := abre_bases (13) // Base de alumnos en exclusiva e indexada
IF !ok_bases
  TONE (750)
  TONE (1000)
  TONE (1200)
  ALERT (" ERROR: No se puede habrir en modo exclusivo la base de datos de alumnos, esta siendo ocupada en alguna estacion de la red")
  SETCOLOR (color)
  RETURN
ENDIF

ok_bases := abre_bases (14) // Base de grupos en exclusiva
IF !ok_bases
  alumnos->(DBCLOSEAREA())
  TONE (750)
  TONE (1000)
  TONE (1200)
  ALERT (" ERROR: No se puede habrir en modo exclusivo la base de datos de asignación, esta siendo ocupada en alguna estacion de la red")
  SETCOLOR (color)
  RETURN
ENDIF

SET ESCAPE OFF // Se evita que el proceso pueda ser abortado

cuadro := SAVESCREEN (9,18,16,63)
DISPBOX (9,18,15,62,"┌───┴───┐",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+N")
@ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)
color := SETCOLOR ("GR+*N")
@ 12,26 SAY "SE ESTA REALIZANDO EL PROCESO"
SETCOLOR (color)

// Pone como no inscritos a todos los alumnos
DO WHILE alumnos->(!EOF())
  REPLACE alumnos->inscrito WITH .F.
  alumnos->(DBSKIP())
ENDDO

alumnos->(DBGOTOP())
DO CASE
  CASE cTipo == "L" // Licenciatura
    DO WHILE alumnos->(!EOF())
      FOR i = 1 to 13 // Cals. y Fals. de la mat1 a la mat12
        temp1 := "mat" + ALLTRIM(STR(i))
        REPLACE alumnos->&temp1. WITH SPACE(5) // Se borra la materia asignada
        FOR x = 1 to 2 // Parciales
          temp1 := "P" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
          temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
          REPLACE alumnos->&temp1. WITH 0.0
          REPLACE alumnos->&temp2. WITH 0
        NEXT
        // Finales
        temp1 := "F" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
        temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
        REPLACE alumnos->&temp1. WITH 0.0
        REPLACE alumnos->&temp2. WITH 0
      NEXT
      alumnos->(DBSKIP())
    ENDDO
  CASE cTipo == "P" // Preparatoria
    DO WHILE alumnos->(!EOF())
      FOR i = 1 to 13 // Cals. y Fals. de la mat1 a la mat12
        temp1 := "mat" + ALLTRIM(STR(i))
        REPLACE alumnos->&temp1. WITH SPACE(5) // Se borra la materia asignada
        FOR x = 1 TO 3 // Primeros Parciales
          temp1 := "P" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
          temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
          REPLACE alumnos->&temp1. WITH 0.0
          REPLACE alumnos->&temp2. WITH 0
        NEXT
      NEXT
    ENDDO

```

```

NEXT
// Semestrales
temp1 := "S" + ALLTRIM(STR(i))
temp2 := "IS" + ALLTRIM(STR(i))
REPLACE alumnos->&temp1. WITH 0.0
REPLACE alumnos->&temp2. WITH 0
FOR x = 4 TO 6 // Segundos Parciales
  temp1 := "P" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
  temp2 := "I" + ALLTRIM(STR(i)) + ALLTRIM(STR(x))
  REPLACE alumnos->&temp1. WITH 0.0
  REPLACE alumnos->&temp2. WITH 0
NEXT
// Finales
temp1 := "F" + ALLTRIM(STR(i))
temp2 := "IF" + ALLTRIM(STR(i))
REPLACE alumnos->&temp1. WITH 0.0
REPLACE alumnos->&temp2. WITH 0
NEXT
alumnos->(DBSKIP())
ENDDO
CASE cTipo == "G" // Postgrado
DO WHILE alumnos->(!EOF())
FOR i = 1 to 13 // Cals. y Fals. de la mat1 a la mat12
temp1 := "mat" + ALLTRIM(STR(i))
REPLACE alumnos->&temp1. WITH SPACE(5) // Se borra la materia asignada

temp1 := "U" + ALLTRIM(STR(i))
temp2 := "I" + ALLTRIM(STR(i))
REPLACE alumnos->&temp1. WITH 0.0
REPLACE alumnos->&temp2. WITH 0
NEXT
alumnos->(DBSKIP())
ENDDO
CASE cTipo == "D" // Diplomado
DO WHILE alumnos->(!EOF())
FOR i = 1 to 13 // Cals. y Fals. de la mat1 a la mat12
temp1 := "mat" + ALLTRIM(STR(i))
REPLACE alumnos->&temp1. WITH SPACE(5) // Se borra la materia asignada

temp1 := "U" + ALLTRIM(STR(i))
temp2 := "I" + ALLTRIM(STR(i))
REPLACE alumnos->&temp1. WITH 0.0
REPLACE alumnos->&temp2. WITH 0
NEXT
alumnos->(DBSKIP())
ENDDO
OTHERWISE
TONE (700)
TONE (1000)
TONE (1200)
ALERT ("EL TIPO DE ESCUELA NO ES VALIDO, CONSULTE CON ALGUIEN ESPECIALIZADO")
END CASE
ZAP

RESTSCREEN (9,18,16,63,cuadro)
inicio->kardex := .F. // Se evita que se inicie el semestre a partir de
inicio->cliya := .F. // realizar este proceso y hasta que se realizen los
inicio->cuotas := .F. // los procesos necesarios para ello.

inicio->(DBCLOSEAREA())
alumnos->(DBCLOSEAREA())
asigna->(DBCLOSEAREA())
SET ESCAPE ON //Se restaura la tecla "ESC"
RETURN

```

/* En la siguiente funcion se realiza el respaldo de toda la informacion
contenida en las distintas bases de datos del SCE*/

```

FUNCTION respaldo()
LOCAL mResp := SPACE(1),cColor
LOCAL pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1), pantalla2

```



```

DISPBOX (10,1,MAXROW()-3,maxcol()-1," | | | | |", C_REYENO) // Se dibuja la ventana de captura
(@ 9, 15 SAY REPLICATE ("-",62)
cColor := SETCOLOR (C_CAP)
pic(2)
SET CURSOR ON

DO WHILE .T.
  (@ 12, 5 SAY "En que unidad desea realizar el respaldo A/B?"
  (@ 12, 53 GET mResp PICTURE "!A" VALID (mResp$"AB")
  READ

  IF LASTKEY() = K_ESC
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR(cColor)
    (@ MAXROW()-1,1 SAY SPACE (78)
    RETURN
  ELSE
    IF !chdir (cPath) // Funcion hecha en C
      TONE (1200)
      ALERT ("NO SE HA PODIDO CAMBIAR AL DIRECTORIO ESPECIFICADO, NO SE PUEDE REALIZAR EL RESPALDO")
      RESTSCREEN (0,0,MAXROW(), MAXCOL(),pantalla2)
      SETCOLOR(cColor)
      (@ MAXROW()-1,1 SAY SPACE (78)
      RETURN .F.
    ENDIF
    pantalla2 := SAVESCREEN (0,0,MAXROW(), MAXCOL())
    CLS
    IF mResp == "A"
      RUN c:\dos\BACKUP *.* A:
    ELSE
      RUN c:\dos\BACKUP *.* B:
    ENDIF
    IF !chdir ("..") // Funcion hecha en C
      TONE (1200)
      ALERT ("NO SE HA PODIDO CAMBIAR AL DIRECTORIO ORIGINAL, ES NECESARIO INICIAR NUEVAMENTE EL SISTEMA
      PARA EVITAR PERDIDA DE INFORMACION")
      CLS
      QUIT
    ENDIF
    RESTSCREEN (0,0,MAXROW(), MAXCOL(),pantalla2)
  ENDIF
ENDDO
RETURN

/* En la siguiente funcion se realiza el proceso de recuperacion
de toda la informacion contenida en los discos de respaldo */
FUNCTION restaura()
  LOCAL mResp := SPACE(1),cColor, pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1), pantalla2

  DISPBOX (10,1,MAXROW()-3,maxcol()-1," | | | | |", C_REYENO) // Se dibuja la ventana de captura
  (@ 9, 15 SAY REPLICATE ("-",62)
  cColor := SETCOLOR (C_CAP)
  pic(2)
  SET CURSOR ON

  DO WHILE .T.
    (@ 12, 5 SAY "En que unidad esta el disco del respaldo A/B?"
    (@ 12, 53 GET mResp PICTURE "!A" VALID (mResp$"AB")
    READ

    IF LASTKEY() = K_ESC
      RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
      SETCOLOR(cColor)
      (@ MAXROW()-1,1 SAY SPACE (78)
      RETURN
    ELSE
      pantalla2 := SAVESCREEN (0,0,MAXROW(), MAXCOL())
      CLS
      IF mResp == "A"

```

```

    RUN c:\dos\restore A: c: /s
ELSE
    RUN c:\dos\restore B: c: /s
ENDIF
*RUN CD..
RESTSCREEN (0,0,MAXROW(), MAXCOL(),pantalla2)
ENDIF
ENDDO
RETURN

/* Función que se encarga de regenerar los índices de todas las bases de datos
en el directorio de la escuela seleccionada. Si alguna de las bases no se
puede habrir en modo exclusivo se avisa al usuario y se aborta el proceso.
En esta función no se reindexan las bases de configuración, inicio, claves,
y escuelas, ya que estas son generales a todas las escuelas */
FUNCTION reindexa (tipo)
LOCAL cuadro := SAVESCREEN (9,18,16,63), aSeguir := {"No","Si"}
PUBLIC aHandle[ B_LEN ], archivo,cScreen

cScreen := SAVESCREEN (16, 20,21,60)
TONE (700)
TONE (1000)
ok := ALERT ("ESTA SEGURO DE QUERER REALIZAR ESTE PROCESO",aSeguir)
IF ok == 1
    RETURN
ENDIF

DISPBOX (9,18,15,62,"URL:" "DAA" ",C_REYENO3) // Se dibuja la ventana de captura
color := SETCOLOR ("R+/N")
@ 11,21 SAY "    ESPERE UN MOMENTO POR FAVOR"
SETCOLOR (color)
color := SETCOLOR ("GR+*/N")
@ 13,26 SAY "SE ESTA REALIZANDO EL PROCESO"
SETCOLOR (color)

IF tipo == 1 // Indexación de bases de alguna escuela
ok_bases := abre_bases (8) //Base de alumnos
IF !ok_bases
    RESTSCREEN (9,18,16,63,cuadro)
    TONE (800)
    TONE (1000)
    TONE (1200)
    ALERT ("NO SE PUDO HABRIR LA BASE DE ALUMNOS EN EXCLUSIVA, CIERRE EL SISTEMA EN TODOS LOS NODOS DE LA
RED")
    RETURN
ELSE
    archivo := "Alumnos"
    nHandle := GaugeNew (19, 20, 21, 60,"13","14","2")
    GaugeDisplay(nHandle)
    INDEX ON Matrialum TO &cPath\alumno1 EVAL gaugeUpdate()
    INDEX ON Nomalum TO &cPath\alumno2 EVAL gaugeUpdate()
    INDEX ON Grupo TO &cPath\alumno3 EVAL gaugeUpdate()
    INDEX ON grupo+nomalum TO &cPath\alumno4 EVAL gaugeUpdate()
    alumnos->(DBCLOSEAREA())
ENDIF

ok_bases := abre_bases (11) // Base de grupos
IF !ok_bases
    RESTSCREEN (9,18,16,63,cuadro)
    TONE (800)
    TONE (1000)
    TONE (1200)
    ALERT ("NO SE PUDO HABRIR LA BASE DE GRUPOS EN EXCLUSIVA, CIERRE EL SISTEMA EN TODOS LOS NODOS DE LA
RED")
    RETURN
ELSE
    archivo := "Grupos"
    nHandle := GaugeNew (19, 20, 21, 60,"13","14","3")
    GaugeDisplay(nHandle)
    INDEX ON grupo TO &cPath\grupo1 EVAL gaugeUpdate()
    INDEX ON clavmat TO &cPath\grupo2 EVAL gaugeUpdate()
    INDEX ON clavmae TO &cPath\grupo3 EVAL gaugeUpdate()

```



```

INDEX ON grupo+CLAVMAT TO &cPath\grupo4 EVAL gaugeUpdate()
grupos->(DBCLOSEAREA())
ENDIF

ok_bases := abre_bases (16) // Abre la base de asigna
IF !ok_bases
  RESTSCREEN (9,18,16,63,cuadro)
  TONE (800)
  TONE (1000)
  TONE (1200)
  ALERT ("NO SE PUDO HABRIR LA BASE DE ASIGNACION EN EXCLUSIVA, CIERRE EL SISTEMA EN TODOS LOS NODOS DE
LA RED")
RETURN
ELSE
  archivo := "Asigna"
  nHandle := GaugeNew (19, 20, 21, 60,"13","14","2")
  GaugeDisplay(nHandle)
  INDEX ON grupo TO asigna1 EVAL gaugeUpdate()
  INDEX ON clavmat TO asigna2 EVAL gaugeUpdate()
  asigna->(DBCLOSEAREA())
ENDIF

ok_bases := abre_bases (6) // Base de maestros
IF !ok_bases
  RESTSCREEN (9,18,16,63,cuadro)
  TONE (800)
  TONE (1000)
  TONE (1200)
  ALERT ("NO SE PUDO HABRIR LA BASE DE MAESTROS EN EXCLUSIVA, CIERRE EL SISTEMA EN TODOS LOS NODOS DE LA
RED")
RETURN
ELSE
  archivo := "Maestros"
  nHandle := GaugeNew (19, 20, 21, 60,"13","14","1")
  GaugeDisplay(nHandle)
  INDEX ON clavmac TO &cPath\maestro1 EVAL gaugeUpdate()
  INDEX ON Nommae TO &cPath\maestro2 EVAL gaugeUpdate()
  maestros->(DBCLOSEAREA())
ENDIF

ok_bases := abre_bases (4) // Base de materias
IF !ok_bases
  RESTSCREEN (9,18,16,63,cuadro)
  TONE (800)
  TONE (1000)
  TONE (1200)
  ALERT ("NO SE PUDO HABRIR LA BASE DE MATERIAS EN EXCLUSIVA, CIERRE EL SISTEMA EN TODOS LOS NODOS DE LA
RED")
RETURN
ELSE
  archivo := "Materias"
  nHandle := GaugeNew (19, 20, 21, 60,"13","14","2")
  GaugeDisplay(nHandle)
  INDEX ON clavmat TO &cPath\materia1 EVAL gaugeUpdate()
  INDEX ON Nommat TO &cPath\materia2 EVAL gaugeUpdate()
  materias->(DBCLOSEAREA())
ENDIF

ok_bases := abre_bases (24) // Base de kardex en exclusiva
IF !ok_bases
  RESTSCREEN (9,18,16,63,cuadro)
  TONE (800)
  TONE (1000)
  TONE (1200)
  ALERT ("NO SE PUDO HABRIR LA BASE DE MATERIAS EN EXCLUSIVA, CIERRE EL SISTEMA EN TODOS LOS NODOS DE LA
RED")
RETURN
ELSE
  archivo := "Kardex"
  nHandle := GaugeNew (19, 20, 21, 60,"13","14","1")
  GaugeDisplay(nHandle)
  INDEX ON Matrialum+semestre TO &cPath\kardex1 EVAL gaugeUpdate()

```

```

INDEX ON Grupo+ClavMat+MatriAlum TO &cPath\kardex2 EVAL gaugeUpdate()
INDEX ON Actualiza TO &cPath\kardex3 EVAL gaugeUpdate()
INDEX ON grupo TO &cPath\kardex4 EVAL gaugeUpdate()
kardex->(DBCLOSEAREA())
ENDIF
ELSE // Indexación de las bases principales del sistema
ok_bases := abre_bases (1) // Base de escuelas
IF ok_bases
archivo := "Escuela"
nHandle := GaugeNew (19, 20, 21, 60,"13","14","█")
GaugeDisplay(nHandle)
INDEX ON escuela->nomesc + escuela->tipo TO escuela EVAL gaugeUpdate()
DBCLOSEAREA()
ELSE
TONE (750)
ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS, NO SE A REALIZADO LA OPERACION")
ENDIF
ok_bases := abre_bases (20.5) // Base de escapes en exclusiva
IF !ok_bases
TONE (800)
TONE (1000)
TONE (1200)
ALERT ("NO SE PUDO HABRIR LA BASE DE ESCAPES EN EXCLUSIVA")
ELSE
archivo := "Escapes"
nHandle := GaugeNew (19, 20, 21, 60,"13","14","█")
GaugeDisplay(nHandle)
INDEX ON STR(nmimp,3,0)+escape TO escape1 EVAL gaugeUpdate() EVERY 50
escapes->(DBCLOSEAREA())
ENDIF
ok_bases := abre_bases (19.5) // Base de impresoras en exclusiva
IF !ok_bases
TONE (800)
TONE (1000)
TONE (1200)
ALERT ("NO SE PUDO HABRIR LA BASE DE IMPRESORAS EN EXCLUSIVA")
ELSE
archivo := "Impres"
nHandle := GaugeNew (19, 20, 21, 60,"13","14","█")
GaugeDisplay(nHandle)
INDEX ON nmimp TO impres1 EVAL gaugeUpdate()
INDEX ON dsimp TO impres2 EVAL gaugeUpdate()
impres->(DBCLOSEAREA())
ENDIF
ok_bases := abre_bases (15.5) // Base de CONFIG EN exclusiva
IF !ok_bases
TONE (800)
TONE (1000)
TONE (1200)
ALERT ("NO SE PUDO HABRIR LA BASE DE CONFIGURACION EN EXCLUSIVA")
ELSE
archivo := "Config"
nHandle := GaugeNew (19, 20, 21, 60,"13","14","█")
GaugeDisplay(nHandle)
INDEX ON clave TO config EVAL gaugeUpdate()
config->(DBCLOSEAREA())
ENDIF
ok_bases := abre_bases (28.5) // Base de PAGO EN exclusiva
IF !ok_bases
TONE (800)
TONE (1000)
TONE (1200)
ALERT ("NO SE PUDO HABRIR LA BASE DE FORMAS DE PAGO EN EXCLUSIVA")
ELSE
archivo := "Pago"
nHandle := GaugeNew (19, 20, 21, 60,"13","14","█")
GaugeDisplay(nHandle)
INDEX ON pago TO pago EVAL gaugeUpdate()
Pago->(DBCLOSEAREA())
ENDIF
ENDIF

```

```

RESTSCREEN (16, 20,21,60,cScreen) // Restaura la barra de avance
RESTSCREEN (9,18,16,63,cuadro) // Restaura el cuadro de dialogo
TONE (900)
TONE (1000)
ALERT ("PROCESO REALIZADO EXITOSAMENTE")
RETURN
    
```

* Crea la barra en la posición y con los colores especificados.

FUNCTION GaugeNew(nTop, nLeft, nBottom, nRight, cBackColor, cBarColor, cBarCharacter)

```

aHandle[ B_TOP ] := nTop
aHandle[ B_LEFT ] := nLeft
aHandle[ B_BOTTOM ] := nBottom
aHandle[ B_RIGHT ] := nRight
aHandle[ B_BACKCOLOR ] := "W/N"
aHandle[ B_BARCOLOR ] := "W+/N"
aHandle[ B_DISPLAYNUM ] := .T.
aHandle[ B_BARCHAR ] := CHR( 219 )
aHandle[ B_PERCENT ] := 0
    
```

```

IF cBackColor <> NIL
    aHandle[ B_BACKCOLOR ] := cBackColor
ENDIF
    
```

```

IF cBarColor <> NIL
    aHandle[ B_BARCOLOR ] := cBarColor
ENDIF
    
```

```

IF cBarCharacter <> NIL
    aHandle[ B_BARCHAR ] := cBarCharacter
ENDIF
    
```

```

IF aHandle[ B_RIGHT ] < aHandle[ B_LEFT ] + 4
    aHandle[ B_RIGHT ] := aHandle[ B_LEFT ] + 4
ENDIF
    
```

```

IF aHandle[ B_BOTTOM ] < aHandle[ B_TOP ] + 2
    aHandle[ B_BOTTOM ] := aHandle[ B_TOP ] + 2
ENDIF
    
```

```

// Determine if we can fit the bracketed number on top of the graph
IF aHandle[ B_RIGHT ] < aHandle[ B_LEFT ] + 9
    aHandle[ B_DISPLAYNUM ] := .F.
ENDIF
    
```

RETURN(aHandle)

* Despliega la barra en pantalla

FUNCTION GaugeDisplay(aHandle)

```

LOCAL nCenter := ROUND( (aHandle[B_RIGHT] - aHandle[B_LEFT]) / 2, 0) + 1
LOCAL cOldColor := SETCOLOR( aHandle[ B_BACKCOLOR ] )
    
```

```

@ aHandle[ B_TOP ]-3, aHandle[ B_LEFT ] CLEAR TO ;
aHandle[ B_BOTTOM ], aHandle[ B_RIGHT ]
    
```

```

@ aHandle[ B_TOP ]-3, aHandle[ B_LEFT ], ;
aHandle[ B_BOTTOM ]-3, aHandle[ B_RIGHT ] BOX B_BOXLINES
    
```

```

@ aHandle[ B_TOP ], aHandle[ B_LEFT ], ;
aHandle[ B_BOTTOM ], aHandle[ B_RIGHT ] BOX B_BOXLINES
    
```

```

IF aHandle[ B_DISPLAYNUM ]
    @ aHandle[ B_TOP ], aHandle[ B_LEFT ]+nCenter-4 SAY "[ ]"
ENDIF
    
```

```

SETCOLOR( cOldColor )
RETURN( aHandle )
    
```

* Actualiza el porcentaje

FUNCTION GaugeUpdate

LOCAL nCenter := ROUND((aHandle[B_RIGHT] - aHandle[B_LEFT]) / 2, 0) + 1;

cOldColor := SETCOLOR(aHandle[B_BARCOLOR]);

nBarRatio := (aHandle[B_RIGHT] - (aHandle[B_LEFT] + 1));

nRow := 0

LOCAL nCols := 0

nPercent := RECNO()/RECCOUNT()

IF aHandle[B_DISPLAYNUM]

@ aHandle[B_TOP]-2, aHandle[B_LEFT]+ 2 SAY "Indexando Archivo : " + UPPER(archivo)+".DBF"

ENDIF

IF aHandle[B_DISPLAYNUM]

@ aHandle[B_TOP], aHandle[B_LEFT]+ nCenter-2 SAY alltrim(STR(nPercent * 100, 3)) + "%"

ENDIF

IF nPercent > 1

nPercent := 1

ENDIF

IF nPercent < 0

nPercent := 0

ENDIF

nCols := ROUND(nPercent * nBarRatio, 0)

@ aHandle[B_TOP] + 1, aHandle[B_LEFT] + 1 CLEAR TO ;

aHandle[B_BOTTOM] - 1, aHandle[B_RIGHT] - 1

FOR nRow := 1 TO (aHandle[B_BOTTOM] - aHandle[B_TOP] - 1)

@ nRow + aHandle[B_TOP], aHandle[B_LEFT] + 1 SAY ;

REPLICATE(aHandle[B_BARCHAR], nCols)

NEXT

SETCOLOR(cOldColor)

RETURN .T.