

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

Sistemas de cuentas por cobrar para instituciones educativas

Autor: Eric Iribe Ruíz

**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.





M.R.

UNIVERSIDAD VASCO DE QUIROGA

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

**" SISTEMA DE CUENTAS POR COBRAR
PARA INSTITUCIONES EDUCATIVAS "**

T E S I S

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

PRESENTA:

ERIC IRIBE RUIZ

ASESOR DE TESIS

M. Y M. A. SERGIO FCO. BARRAZA IBARRA



No. ACUERDO 952006

CLAVE: 16PSU00 14Q

MORELIA, MICH., 28 DE JUNIO DE 1996.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS
AGRADECIMIENTOS

ÍNDICE

<i>Agradezco a Dios por permitirme llegar al lugar donde ahora me encuentro, también le agradezco infinitamente por haberme dado como madre a un ser inigualable pues gracias a su ejemplo soy lo que soy y me encuentro aquí alcanzando una meta más en mi vida. Como un atributo a su persistencia e inquebrantable fe de lucha.</i>		
1.-	Introducción	1
2.-	Antecedentes	5
3.-	Justificación	6
4.-	Objetivo general	7
4.1.-	Objetivos específicos	8
4.2.-	Metas propuestas	9
5.-	Análisis del proceso actual	10
5.1.-	Flujo de la información y principalmente por creer en mi.	16
5.2.-	Registro de la información	19
5.3.-	Actividades adicionales en el departamento de contabilidad <i>Tu hijo, Eric</i>	20
6.-	Transición del análisis al diseño del sistema	24
6.1.-	Especificación de los requerimientos de la aplicación	25
6.1.1.-	Revisión de los objetivos de la investigación de requerimientos	25
6.1.2.-	Percepción de requerimientos: Análisis de hechos	25
6.1.3.-	Identificación de requerimientos	30
6.1.4.-	Selección de estrategias para el cumplimiento total de los requerimientos	30
6.2.-	Objetivos al diseñar un sistema de información <i>Gracias a todos los profesores que contribuyeron con su granito de arena a lo largo de mi carrera para formarme como persona y como profesional, a todos ellos mi más sincero agradecimiento.</i>	32
6.2.1.-	Especificación de los elementos lógicos del diseño	32
6.2.2.-	Apoyo para las actividades para formarme como persona y como profesional,	32
6.2.3.-	Asegurar que las características de los requerimientos del usuario	32
6.2.4.-	Proporcionar un sistema <i>Un especial agradecimiento, a mi Director de Carrera y Director de Tesis Ing. y M.A. Sergio Francisco Barraza Ibarra por todo el apoyo incondicional que me ha brindado durante todos estos años.</i>	32
6.2.5.-	Ajustarse a los estándares	32

ÍNDICE

1.- Introducción	1
2.- Antecedentes	5
3.- Justificación	6
4.- Objetivo general	7
4.1.- Objetivos específicos	8
4.2.- Metas propuestas	9
5.- Análisis del proceso actual	10
5.1.- Flujo de la información	16
5.2.- Registro de la información	19
5.3.- Actividades adicionales en el departamento de caja y de contabilidad	20
6.- Transición del análisis al diseño del sistema	24
6.1.- Especificación de los requerimientos de la aplicación	25
6.1.1.- Revisión de los objetivos de la investigación de requerimientos	25
6.1.2.- Percepción de requerimientos: Análisis de hechos	25
6.1.3.- Identificación de requerimientos	30
6.1.4.- Selección de estrategias para el cumplimiento total de los requerimientos	30
6.2.- Objetivos al diseñar un sistema de información	32
6.2.1.- Especificación de los elementos lógicos del diseño	32
6.2.2.- Apoyo para las actividades de la empresa	33
6.2.3.- Asegurar que las características del sistema cumplan con los requerimientos del usuario	33
6.2.4.- Proporcionar un sistema que sea fácil de utilizar	34
6.2.5.- Ajustarse a los estándares de diseño	36

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

6.3.- Características con las que se debe diseñar un sistema de información	38
6.3.1.- Elementos de diseño	38
6.3.2.- Diseño de la salida	39
6.3.3.- Diseño de archivos	40
6.3.4.- Diseño de interacciones con la base de datos	41
6.3.5.- Diseño de la entrada	41
6.3.6.- Diseño de controles	43
6.3.7.- Diseño de procedimientos	43
6.3.8.- Diseño de especificaciones para programas	44
6.4.- Administración de bases de datos	46
6.4.1.- Diseño mediante un modelo de datos relacional	53
6.4.2.- Estructuras de las bases de datos utilizadas en el sistema de cuentas por cobrar	55
6.4.3.- Validación de las entradas	62
6.4.4.- Seguridad y confidencialidad	62
7.- Descripción del sistema propuesto	65
7.1.- Menú principal	65
7.1.1.- Módulo "Seleccionar"	66
7.1.1.1.- Módulo "Pagos"	68
7.1.1.2.- Módulo "Cuotas"	69
7.1.1.3.- Módulo "Catálogos"	70
7.1.1.4.- Módulo "Menú Anterior"	70
7.1.2.- Módulo "Reportes"	71
7.1.3.- Módulo "Utilerías"	71
7.1.4.- Módulo "Menú Anterior"	72
7.2.- Diagrama del sistema propuesto	73
8.- Conclusiones y recomendaciones	74
9.- Apéndices	79
9.A.- Manual del usuario	80
9.B.- Código fuente	173
10.- Bibliografía	275

ÍNDICE DE DIAGRAMAS DE FLUJO, TABLAS Y ESQUEMAS

5.1	Diagrama de flujo del proceso actual (nuevo ingreso)	11
5.2	Diagrama de flujo del proceso actual (reingreso)	13
5.3	Flujo de documentos de control interno para la solicitud de documentos oficiales y pagos diversos	15
5.4	Flujo de documentos de control interno para alumnos de nuevo ingreso y reingreso	18
5.5	Control de costos	22
6.1	Estructura de la base de datos EDOCUENT.DBF	56
6.2	Estructura de la base de datos ALUMNOS.DBF	57
6.3	Estructura de la base de datos DIARIO.DBF	58
6.4	Estructura de la base de datos PAGO.DBF	59
6.5	Estructura de la base de datos CONCEPTO.DBF	59
6.6	Estructura de la base de datos MES.DBF	59
6.7	Estructura de la base de datos INICIO.DBF	60
6.8	Estructura de la base de datos DMES/DIA/AÑO.DBF	60
6.9	Estructura de la base de datos IMPRES.DBF	60
6.10	Estructura de la base de datos ESCAPES.DBF	61
6.11	Estructura de la base de datos CONFIG.DBF	61
7.1	Menú Principal	65
7.2	Módulo Seleccionar (Escuela seleccionada)	67
7.3	Módulo de Pagos	69
7.4	Modulo de Cuotas	69
7.5	Módulo de Catálogos	70
7.6	Módulo de Menú Anterior	70
7.7	Módulo de Reportes	71
7.8	Módulo de Utilerías	72
7.9	Módulo de Menú Anterior	72
7.10	Diagrama del Sistema Propuesto	73

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 Cuentas por cobrar, con el cual se pueda lograr mayor eficiencia en lo que se refiere a la atención del alumnado y en lo que respecta al manejo de la información interna ligada entre el departamento de contabilidad de la Universidad Vasco de Quiroga y el departamento de control escolar.

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 de realizar la elaboración de un sistema de Cuentas por Cobrar, de tal forma que cubra explícitamente con las necesidades de la Universidad.

De esta forma se busca que la Universidad Vasco de Quiroga cubra todas aquellas necesidades internas a través de sistemas computarizados que faciliten el mejoramiento, fortalecimiento y superación de la Institución, brindando una mayor calidad en el servicio a la comunidad universitaria y un control interno mas efectivo y oportuno en cuanto a los procesos de registro de los ingresos de la institución.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

El contenido del presente trabajo de tesis tiene como finalidad fundamental servir y proporcionar una herramienta útil en el manejo de información y para el engrandecimiento de la universidad, pues se hace evidente que los sistemas de información o sistemas computarizados proporcionan en la actualidad un pilar importante para el apoyo de las diversas áreas de las organizaciones, obteniendo de los mismos, resultados satisfactorios y ofreciendo a la par servicios de primera calidad.

De tal manera, el presente trabajo esta clasificado en 10 capitulos en los cuales se contempla las fases de detección de necesidades, análisis, diseño y alcance del sistema como se mencionó al principio de este tema.

En el segundo capitulo, describo los antecedentes de la Universidad Vasco de Quiroga desde que se constituyo como Instituto de Estudios Superiores Vasco de Quiroga (IESVAQ) hasta la actualidad, tiempo durante el cual se han marcado grandes cambios en beneficio de los alumnos que en ella estudian, así, como en beneficio de la misma Institución.

En el tercer rubro se presenta la justificación de las razones por las cuales se propone la elaboración e implantación de un sistema de Cuentas por Cobrar que servirá de puntal para el mejoramiento del área administrativa y de operación.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

En el cuarto apartado se esclarece el objetivo principal por el cual se propone la implantación de un sistema de Cuentas por Cobrar en el área administrativa, se hace mención a los objetivos específicos que se pretenden cumplir con el diseño, elaboración e implantación del sistema propuesto, y se especifican las metas que se desean cumplir.

En el quinto capítulo se desarrolla y describe en forma concisa el proceso actual de los distintos departamentos que conforman la área administrativa. Se indica el manejo, flujo y registro de la información que se genera, así como el proceso de operación.

El sexto apartado hace referencia a los elementos que se proponen como pasos o puntos fundamentales a tomar en cuenta para llevar a cabo la elaboración de un sistema de información, lo que se refiere a la transición del análisis hacia el diseño del sistema.

En el séptimo capítulo se presenta la descripción del sistema que se propone, como se encuentra constituido y modulado, así como una breve descripción de las funciones que realiza cada una de la diversas opciones que ofrece el sistema.

El octavo rubro contiene las conclusiones y recomendaciones que se consideran de más trascendencia para alcanzar un óptimo funcionamiento y manejo del sistema.

El noveno tema esta constituido por dos apéndices de los cuales el primero contiene el manual del usuario en el cual se describe la operación del sistema, y el segundo apéndice muestra los listados del código fuente.

Por último, en el capítulo 10 se proporciona la bibliografía utilizada como apoyo para la elaboración del presente trabajo de tesis.

2.- ANTECEDENTES

La Universidad Vasco de Quiroga es una institución educativa de nivel medio superior y superior, que desde hace 16 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 con el nombre de Instituto de Estudios Superiores Vasco de Quiroga (IESVAQ), nombre que posteriormente fue cambiado ya que el nombre de IESVAQ ya no se ajustaba al tipo de educación que en la institución se ofrece. Hasta hace menos de tres años, el hasta entonces IESVAQ funcionaba en el edificio que actualmente ocupan la preparatoria propedeutica de Morelia y la tecnológica (Campus Taximaroa), pero dado el número cada vez mayor de estudiantes, éste edificio fue resultando poco a poco insuficiente; es por esta razón que se tomo la decisión de adquirir 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 las licenciaturas de la universidad a su edificio en Santa María, donde laboran actualmente 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.

3.- JUSTIFICACIÓN

El área administrativa, en lo que se refiere al departamento de contabilidad de la Universidad Vasco de Quiroga no cuenta actualmente con un sistema de Cuentas por Cobrar, por tal motivo se vio la necesidad de cubrir dichas necesidades llevando a cabo la implantación de un sistema de Cuentas por Cobrar.

Algunos de los aspectos que se pretenden cubrir son las siguientes:

- ♦ Actualmente en el departamento de contabilidad, en lo que respecta a los ingresos que se obtienen del cobro directo a los alumnos por concepto del pago de los servicios que la Universidad ofrece, no cuenta con un control oportuno.
- ♦ Actualmente no se lleva a cabo un control eficiente de los conceptos por los cuales ingresa el dinero a la caja.
- ♦ Se necesita conocer el nivel de operación económico de cada una de las escuelas de la Universidad.
- ♦ En lo referente a los becarios es necesario disponer de un sistema de cobro mas eficiente y confiable.
- ♦ En la actualidad todos éstos procesos se realizan en forma manual, haciéndolos lentos, pocos confiables y de alto costo.

4.- OBJETIVO GENERAL

Implantar un sistema de Cuentas por Cobrar para la Universidad Vasco de Quiroga que le permita eficientar los procesos económico-administrativos para alcanzar una mayor control en el proceso del ingreso y su registro oportuno, que mejorará la productividad y eficiencia del departamento de Contabilidad, para una efectiva toma de decisiones en el alto nivel directivo de la misma Institución.

4.1.- OBJETIVOS ESPECÍFICOS

- ♦ Agilizar el registro del pago de los alumnos por los servicios recibidos de la Universidad.
- ♦ Que el sistema funcione en un ambiente de Red de área local.
- ♦ Disponer de un sistema de control de la información por medio de llaves y/o claves de acceso.
- ♦ Agilizar el manejo interno de las procesos que se desarrollan en caja como son: inscripciones, reinscripciones, pago de colegiaturas, pago de memorándums, etc.
- ♦ Eliminar el carnet de pagos y reducir el número de formatos preimpresos que se manejan en caja. →
- ♦ Generar los reportes básicos necesarios, como son reporte de corte de caja, reporte para detallar los conceptos por los cuales ingresa el dinero a la caja, así como reportes de los alumnos que adeudan colegiaturas.

4.2.- METAS PROPUESTAS

- ♦ El desarrollo del sistema esta contemplado para llevarse a cabo en un periodo de dos meses a partir de la fecha en que se concluya la etapa de análisis y diseño del sistema, así como de la etapa de recolección de información.

- ♦ Una vez concluida la etapa de desarrollo del sistema, en lo se refiere a elaboración de código fuente, se procederá a llevar a cabo la etapa de depuración y prueba del sistema en un periodo de dos semanas.

- ♦ Liberada la etapa de depuración y prueba, se procederá a llevar a cabo la implantación del sistema en el área de caja y departamento de contabilidad, ésta etapa se realizará en un periodo de una semana.

- ♦ Ya instalado el sistema y funcionando en su totalidad, se procederá a llevar a cabo la capacitación del personal que utilizará el sistema, ésto a cumplirse en un periodo de una semana..

5.- ANÁLISIS DEL PROCESO ACTUAL (NUEVO INGRESO)

Para llevar a cabo la realización del pago de los diferentes servicios que ofrece la Universidad al alumnado se llevan a cabo dos tipos de procesos internos, ésto dependiendo del tipo de operación que van a efectuar, independientemente del nivel académico.

1º Si el tipo de transacción que se va a efectuar es el pago de inscripción a nuevo ingreso, el aspirante llega a la ventanilla de servicios escolares, ahí, se le facilita la información del plan de estudios en base al bachillerato o licenciatura que elija, se le proporciona la solicitud que debe de entregar debidamente requisitada y se le indica que pase a caja a realizar el pago del examen de admisión; una vez hecho esto, pasa nuevamente a ventanilla en donde se le entrega la documentación completa, la solicitud y la ficha para presentar el examen de admisión y se le indican las fechas de examen para que el aspirante escoja una de ellas. Una vez que el aspirante a realizado los exámenes correspondientes y ha sido admitido, regresa a la ventanilla de servicios escolares para solicitar cita para entrevista con el director de escuela (esto en caso de tratarse de un alumno de nuevo ingreso a nivel licenciatura), emitiéndose un dictamen de aprobado o rechazado, si ocurre lo primero, el aspirante debe concertar una cita para que le efectúen el examen médico; una vez que ha aprobado todo el proceso de admisión, se le solicitan los documentos finales y se le indica que pase a caja a efectuar el pago correspondiente a su inscripción; una vez que el alumno ha pagado, pasa de nuevo a ventanilla para que se le proporcione la información final.

DIAGRAMA DE FLUJO DEL PROCESO ACTUAL

(NUEVO INGRESO)

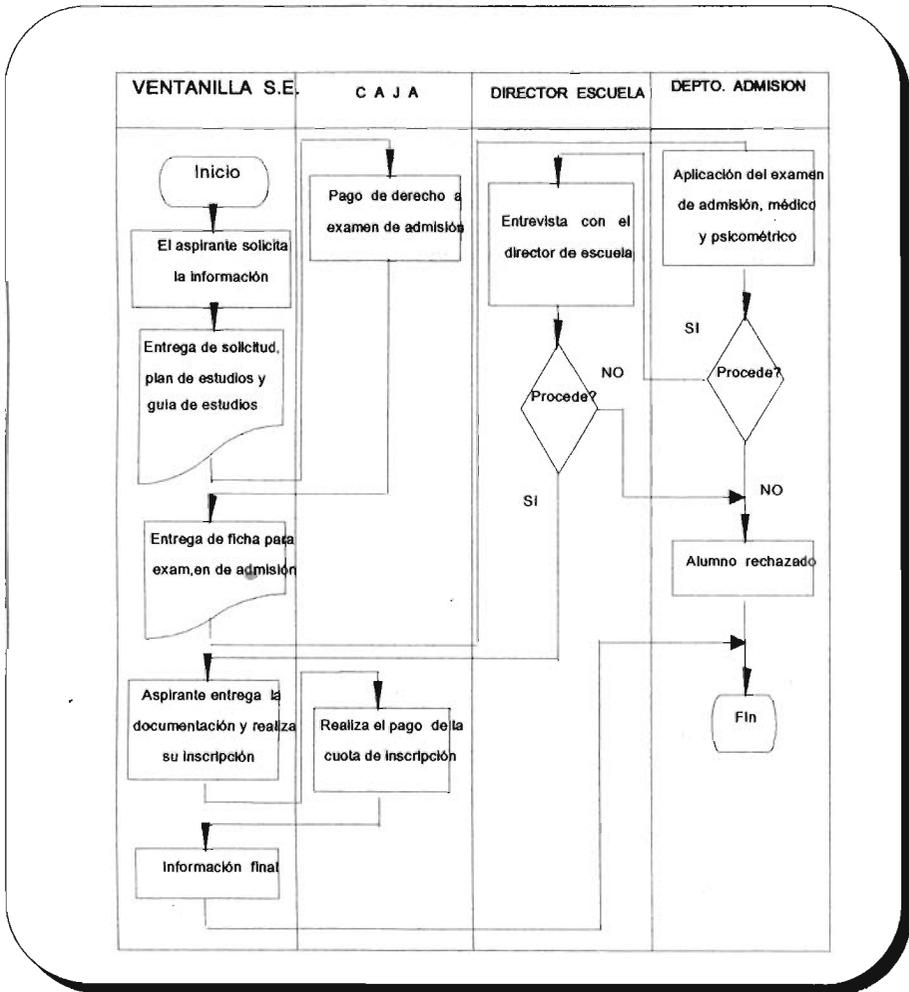


Diagrama de Flujo 5.1

(REINGRESO)

En caso de que el trámite que se va a efectuar sea el pago de una reinscripción, el alumno o la persona que va a llevar a cabo la operación, pasa a la ventanilla de información, ahí, se revisa que no tenga ningún de adeudo, tanto económico como de material, en caso de que así sea, se le da su ficha de reinscripción y se le solicita que pase a caja a realizar el pago correspondiente, pero en caso de que el alumno tenga algún tipo de adeudo, primero debe de pasar a caja a cubrir el saldo moroso y después regresar a la ventanilla a solicitar su ficha de reinscripción para finalmente pasar nuevamente a caja a cubrir el pago de la misma.

DIAGRAMA DE FLUJO DEL PROCESO ACTUAL

(REINGRESO)

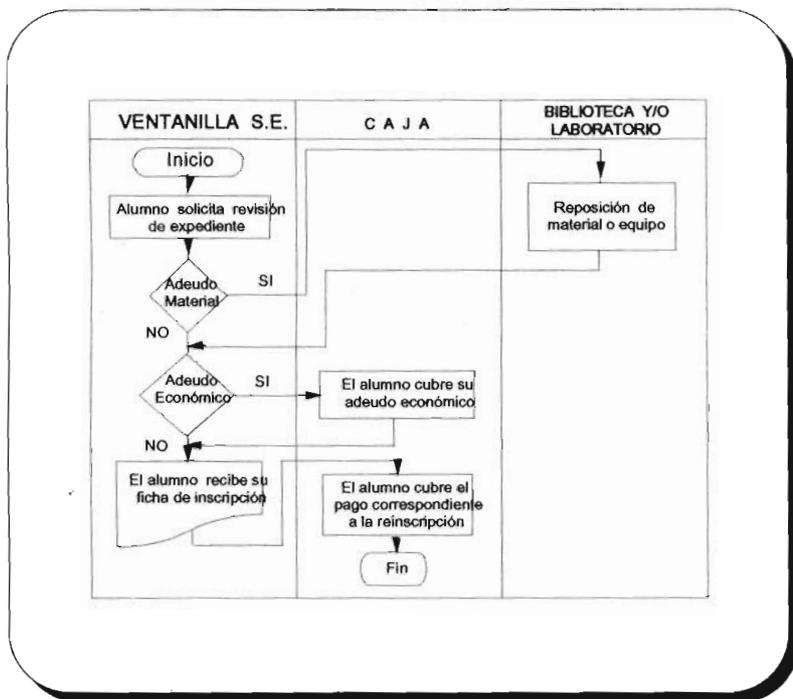


Diagrama de flujo 5.2

2º Si la operación que se va a efectuar se refiere al pago de colegiatura, memorándum de calificaciones, constancia de estudios, titulación, solicitud de certificado, carta de pasante, examen extraordinario o adicional el alumno tiene que presentarse directamente a la caja para realizar el pago que corresponda. En caso de que el alumno solicite algún tipo de documento oficial, se le indicará el día que tendrá que pasar a recogerlo, ese día, el alumno tendrá que presentar la copia color azul que le fue entregada en caja, dicha copia se anexará a su expediente y el alumno se quedará con la original.

FLUJO DE DOCUMENTOS DE CONTROL INTERNO PARA LA SOLICITUD DE DOCUMENTOS OFICIALES Y PAGOS DIVERSOS

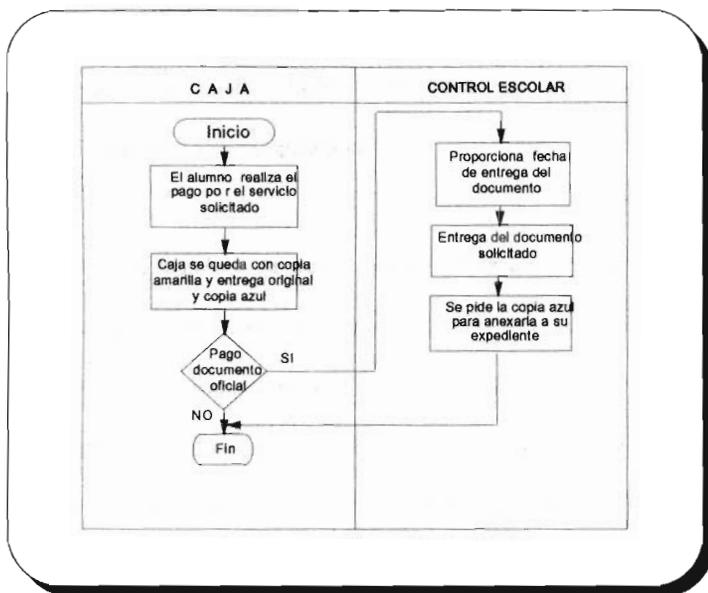


Diagrama de flujo 5.3

5.1.- FLUJO DE LA INFORMACIÓN

Habiendo detallado el seguimiento que se tiene que llevar a cabo para efectuar el pago de los diferentes servicios que ofrece la Universidad, se procederá a detallar en forma más específica como fluye la información dentro de los distintos departamentos que integran a la Universidad una vez que el alumno ha realizado alguna operación.

Cabe mencionar que los block de recibos que son utilizados en caja cuentan con un recibo original y dos copias, las cuales tienen en su parte inferior derecha la denominación de *alumno* para el caso de la original, *expediente* para el caso de la copia color azul y *contabilidad* para el caso de la copia amarilla; si la operación que se efectuó fue el pago de una colegiatura, la caja se queda con la copia amarilla y se le entrega al alumno la original y la copia azul. Al final del día se realiza el corte de caja (se ordenan los recibos por número de folio, ésto de menor a mayor, y se suma recibo por recibo sin importar el concepto por el cual entro dinero a caja, anotando en una hoja la cantidad total. Finalmente se desglosa la forma de pago, cuanto entró en cheque, cuanto en tarjeta de crédito y cuanto en efectivo para compararlo con la suma total de los recibos.) Se entrega el dinero y el recibo del corte al departamento de Contraloría donde se checa nuevamente el corte de caja para después hacer el depósito al banco.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

En cuanto se tienen las fichas de depósito se mandan al departamento de Cuentas por Cobrar (recibos y depósito). Ya estando en el departamento antes mencionado, se le hace el abono a cada alumno que pago exclusivamente colegiatura.

En éste mismo departamento se hace la clasificación de los recibos de acuerdo al concepto de pago realizado (colegiaturas, constancias, memorándums, inscripciones, reinscripciones, etc.) para posteriormente entregarlos al departamento de Contabilidad.

FLUJO DE DOCUMENTOS DE CONTROL INTERNO PARA ALUMNOS DE NUEVO INGRESO Y REINGRESO

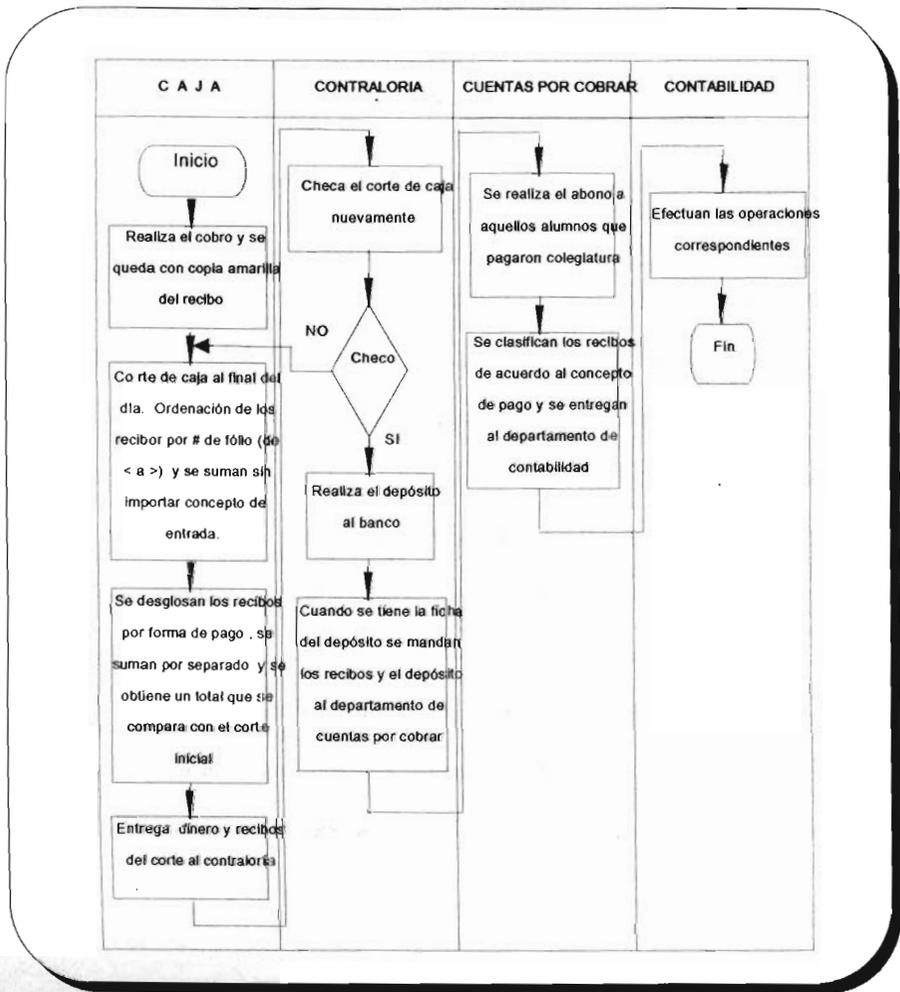


Diagrama de flujo 5.4

5.2.- REGISTRO DE LA INFORMACIÓN

Todos los alumnos de la Universidad independientemente del nivel académico que se trate (preparatorias, licenciaturas, diplomados, maestrías o doctorados), se encuentran registrados en distintas hojas de cálculo (QPRO); ahí es donde se realiza el abono o registro del pago de las colegiaturas exclusivamente anotando la fecha en que se hizo la operación y en caso de tener colegiaturas atrasadas, se marca que mes cubre el pago realizado y cuales faltan por cubrir.

Una vez que llegan los recibos al departamento de Contabilidad se codifica o contabiliza la póliza de ingresos por centro de costo, entendiéndose por centro costo como cada unidad operativa que genera una erogación que es indispensable para la operación de la universidad, adicionalmente, también cada uno de los programas que generan ingresos al realizar la actividad sustantiva y en consecuencia sus gastos de operación normal, a los cuales se mandan todo tipo de ingresos que les corresponde; posteriormente se manda a la subárea de archivo todos los recibos donde se pegan los mismos en hojas blancas tamaño carta anexándoles el depósito correspondiente al corte por recibo. Se pegan por orden de folio al concluir al mes. Concluido esto se procede a armar el legajo (todos los ingresos del mes) y se rotula la carátula del legajo donde lleva el nombre de la Universidad, que tipo de pólizas contiene y el periodo al que corresponde.

5.3.- ACTIVIDADES ADICIONALES EN EL DEPARTAMENTO DE CAJA Y DE CONTABILIDAD

Es importante mencionar que para poder llevar a cabo algunas de las actividades mencionadas anteriormente, como es el pago de reinscripciones o colegiaturas, es necesario realizar algunas actividades extra, que, aunque no requieren de mucho tiempo, en determinado momento llegan a ser un paso que evita cumplir y ofrecer un servicio eficiente y eficaz en aspectos de tiempo y de operatividad para la misma institución. Se hace mención a lo anterior ya que muchos de los alumnos que acuden a caja a realizar el pago de su reinscripción o el pago de su colegiatura y cuentan con beca o que vienen arrastrando morosidad en su saldo, tienen que esperar a que la persona encargada de caja primero cheque en el departamento de contabilidad cual es el porcentaje de su beca o su saldo moroso, para posteriormente, poder elaborar su recibo y realizar el cobro. Esto ocasiona que, aparte de hacer más tardía la atención a los alumnos o a los interesados, se desatiendan algunas otras actividades en los otros departamentos.

Cabe también mencionar que constantemente se hace imprescindible la necesidad de contar por parte del departamento de control escolar y contabilidad con un concentrado de alumnos que adeudan colegiaturas y que vienen arrastrando un saldo moroso, de alumnos que cuentan con beca así como sus porcentajes, y reportes de ingresos por escuela que cuenten con el concepto por el cual se generan ingresos para conocer su reutilidad o su rendimiento.

Actualmente éstos procesos se llevan a cabo en forma manual, lo cuál hace que se desatiendan otras actividades en ambos departamentos ya que se consume tiempo en la elaboración de los mismos durante la recabación de la información de cada una de las escuelas.

Otra de las actividades que se realizan en caja y que consumen tiempo durante su desarrollo, como se mencionó en el apartado “Flujo de la información”, es la elaboración del corte diario de caja, el cual provoca que se haga más tardío el chequeo de ingresos diario, así como su entrega al departamento de contabilidad, en el cuál se vuelve a realizar la misma operación ocasionando la misma pérdida de tiempo pues se tiene que realizar recibo por recibo y por forma de pago.

(*) UNIVERSIDAD VASCO DE QUIROGA, A.C.
"CONTROL DE COSTOS"

Nº	DESCRIPCIÓN
1	RECTORÍA
2	
3	COORDINACIÓN DE SERVICIOS UNIVERSITARIOS
4	CONTROL ESCOLAR
5	INTEGRACIÓN UNIVERSITARIA
6	PROMOCIÓN Y DIFUSIÓN CULTURAL
7	DEPORTES
8	SERVICIO MEDICO
9	BIBLIOTECA SANTA MARÍA
10	
11	
12	COORDINACIÓN ADMINISTRATIVA
13	ALMACÉN
14	RECURSOS MATERIALES
15	TESORERÍA Y FINANZAS
16	RECURSOS HUMANOS
17	CONTABILIDAD
18	
19	
20	COORDINACIÓN DE PREPARATORIAS
21	PREPARATORIA PROPEDEUTICA
22	PREPARATORIA BIVALENTE
23	PREPARATORIA CIUDAD HIDALGO
24	PREPARATORIA TACAMBARO
25	PREPARATORIA PATZCUARO
26	
27	
28	
29	
30	SECRETARIA ACADÉMICA

31	CONTABILIDAD Y ADMINISTRACIÓN
32	PSICOLOGÍA
33	ARQUITECTURA
34	CIENCIAS DE LA COMUNICACIÓN
35	SISTEMAS COMPUTARIZADOS
36	FILOSOFÍA
37	COMERCIO INTERNACIONAL
38	DISEÑO GRÁFICO
39	DERECHO
40	
41	
42	POSTGRADOS
43	IDIOMAS
44	INVESTIGACIÓN
45	
46	
47	
48	
49	
50	CAMPAÑA FINANCIERA

Tabla 5.5

6.- TRANSICIÓN DEL ANÁLISIS AL DISEÑO DEL SISTEMA

Para determinar los requerimientos de sistemas, es necesario analizar los hechos que se tienen a la mano. Las descripciones y la documentación desarrollada como resultado del esfuerzo de búsqueda de hechos, se estudian con la finalidad de evaluar el funcionamiento del sistema en uso y establecer los requerimientos que debe cumplir un nuevo diseño. Las conclusiones obtenidas durante esta actividad forman la base para la transición hacia el diseño así como de otras actividades de desarrollo.

El diseño es una solución: la conversión de los requerimientos en formas que los satisfagan, cómo diseñar un sistema de información para que cumpla con los requisitos previamente establecidos. El diseño determina el éxito del sistema.

Al considerar los objetivos de diseño, se observa que éstos van desde el satisfacer las necesidades de la empresa hasta el empleo eficaz de la tecnología de cómputo. La esencia del diseño de sistemas es la selección y especificación de las características de un sistema de información.

QUE ES UN SISTEMA DE INFORMACIÓN? Un sistema de información es un conjunto de personas, datos y procedimientos que interactúan entre sí para alcanzar un objetivo común para apoyar a las actividades de la organización.

6.1.- ESPECIFICACIÓN DE LOS REQUERIMIENTOS DE LA APLICACIÓN

La determinación de requerimientos incluye la anticipación, investigación y especificación de los mismos para satisfacer las necesidades del nuevo sistema.

6.1.1.- REVISIÓN DE LOS OBJETIVOS DE LA INVESTIGACIÓN DE REQUERIMIENTOS

Durante la fase de búsqueda de hechos de la investigación detallada, la atención se centra alrededor de los requerimientos básicos: comprensión de los procesos, incluyendo por qué y cómo se efectúan; identificación de los datos utilizados así como de las actividades de procesamiento de la información e identificación de los controles del sistema.

6.1.2.- PERCEPCIÓN DE REQUERIMIENTOS: ANÁLISIS DE HECHOS

El análisis de sistemas comprende la recopilación de hechos y el análisis de los mismos. En ésta etapa se proporciona un marco de referencia que es de gran utilidad para percibir las debilidades y requerimientos de un sistema. Así mismo, identifica las estrategias a seguir para satisfacer los requerimientos establecidos, como base para la transición hacia el diseño de sistemas. Un marco de referencia útil para examinar los hechos de un sistema y establecer los requerimientos de los usuarios, incluye las áreas de capacidad, control, acceso a la información y complejidad.

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. Desde el punto de vista de procesos, se debe determinar si la capacidad actual brinda soporte a todas las actividades generales de sistemas de información como son la aceptación de entradas, el procesamiento, almacenamiento y recuperación de datos, la producción de resultados o la transmisión de información.

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. Las soluciones más comunes son:

- ♦ **Aumentar las capacidades:** Añadir al sistema más personal, equipo u otros recursos para incrementar la carga de trabajo que se puede alcanzar.
- ♦ **Reducir las expectativas:** Disminuir los requerimientos de desempeño por medio de un cambio en los tiempos ya sea para dar inicio a una tarea o para finalizarla.
- ♦ **Volver a definir la naturaleza de la tarea:** Cambiar las actividades ya sea modificando las responsabilidades, ajustando los niveles de especialización o generalización, o cambiando los estándares de rendimiento.

CONTROL: La palabra *control* significa “mantener las cosas en su sitio”. En un sentido más formal, el 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. el control también permite detecte y notifique los casos donde estas actividades no se realizaron. También es esencial el manejo de eventos no anticipados. Un sistema bien diseñado debe tener la capacidad de notificarlos sin interrumpir las demás actividades.

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:

- ♦ **Diseños 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.

ACCESIBILIDAD DE LA INFORMACIÓN: Cuando es difícil el acceso a la información, se debe considerar que hay un problema de información, o sólo es cuestión de procedimientos. El análisis cuidadoso de los hechos puede revelar que, aunque exista la información necesaria, las personas tienden a confiar más en su memoria o corazonadas porque el procedimiento de recuperación es demasiado laborioso. el método de recuperación puede ser difícil o bien la información se presenta en forma equivocada.

Entre las estrategias más representativas para evitar los problemas anteriores, se encuentran las siguientes:

- ♦ **Eliminar la necesidad de información:** Cierta información puede ser recuperada y utilizada por un sistema automatizado sin necesidad de la intervención del usuario. En otros casos, el desarrollo de opciones preestablecidas, ejecutadas a menos que se presenten consideraciones poco comunes, eliminará un requerimiento de información.
- ♦ **Facilitar el acceso a la información:** Los sistemas se pueden diseñar para que respondan a mandatos y dependan de opciones preestablecidas que se ajusten a las situaciones que se presentan con mayor frecuencia.
- ♦ **Eliminar las necesidades de procesamiento:** Los detalles utilizados con mayor frecuencia pueden almacenarse en una forma que no requiera procesamiento o manejo. En este caso mejoran tanto la disponibilidad como la accesibilidad.
- ♦ **Cambiar el método de presentación:** Se pueden seleccionar el empleo de otros formatos, incluyendo gráficas, colores e incluso técnicas de animación.

COMPLEJIDAD: La elegancia de la sencillez es un atributo de los sistemas de información mejor diseñados. 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. La complejidad puede ser la base de los problemas de capacidad, control e información.

Para reducir la complejidad se deben considerar tres estrategias.

APLICADOS
A CTAS X COB.

- ♦ **Simplificación:** La simplificación puede obtenerse al eliminar los pasos que no son necesarios o el registro de información que no se utiliza. El uso de estándares preestablecidos simplificará los procesos. Proporcionar estándares más explícitos o lineamientos también puede ayudar a simplificar procesos que de otra manera serían complejos.
- ♦ **División:** Dividir un proceso complejo en tareas separadas disminuye con frecuencia la complejidad. Al dividir un proceso, se debe considerar qué tareas tienen que ser hechas por las personas y cuáles son más adecuadas para una computadora.
- ♦ **Cambios en la secuencia:** Cambiar el orden en el que ocurre un proceso puede disminuir la complejidad aparente. De esta manera es posible poner a disponibilidad información esencial y las respuestas a las preguntas difíciles se vuelven inmediatamente obvias.

6.1.3.- IDENTIFICACIÓN DE REQUERIMIENTOS

Los requerimientos de diseño se formulan a partir de los resultados del análisis. Los requerimientos de un nuevo sistema son aquellas características o detalles que deben incorporarse para producir las mejoras o cambios necesarios

Los requerimientos comunes de sistemas incluyen mejoras en la operatividad, tales como el aumento de volumen de trabajo o en tiempo menor para la recuperación de información. También son requerimientos comunes de sistemas la integración de datos o de varias áreas de las organizaciones. Asimismo, en el presente está aumentando el número de organizaciones que desarrollan sistemas de información con la finalidad de ganar ventaja competitiva en el mercado.

6.1.4.- SELECCIÓN DE ESTRATEGIAS PARA EL CUMPLIMIENTO TOTAL DE LOS REQUERIMIENTOS

El análisis hecho con el estudio de los datos sugiere, en general, varios caminos que conducen hacia el cambio o mejora deseada. Por consiguiente, se debe tratar de identificar aquéllos que son más factibles desde el punto de vista técnico, económico y operacional. Las opciones identificadas de esta manera son las *estrategias para satisfacer los requerimientos*. Ellas forman la base para el diseño de sistemas. Una de las estrategias es mejorar una situación mediante el desarrollo de mejores procedimientos de operación.

La revisión de procedimientos para eliminar formularios y documentos innecesarios o pasos duplicados, simplifica y acelera el trabajo. Lo anterior también puede disminuir el potencial de error. El papel de la computadora en un diseño se centra alrededor de sus capacidades de cálculo, almacenamiento y recuperación de datos, clasificación y comunicación de datos.

6.2.- OBJETIVOS AL DISEÑAR UN SISTEMA DE INFORMACIÓN

Los requerimientos se trasladan en especificaciones de diseño. Surge la pregunta ¿Qué se pretende alcanzar cuando se diseña un sistema de información?. Aquí se describen los objetivos del diseño de sistemas y se explora la naturaleza del diseño así como los resultados explícitos esperados de esta actividad.

6.2.1.- ESPECIFICACIÓN DE LOS ELEMENTOS LÓGICOS DEL DISEÑO

El diseño de sistemas tiene dos etapas: el diseño lógico y la construcción física del sistema. Cuando se formula el *diseño lógico*, se escriben las especificaciones detalladas del nuevo sistema, es decir aquellas que describen sus características: salidas, entradas archivos y bases de datos y los procedimientos, todo en una forma que satisfaga los requerimientos del proyecto. El conjunto formado por todas estas características recibe el nombre de *especificaciones de diseño* del sistema.

El diseño lógico de un sistema de información muestra las características más sobresalientes y la relación que guardan entre sí. Los procedimientos y datos se enlazan entre sí para poder producir un sistema que trabaja.

La *construcción física*, que es la siguiente actividad después del diseño lógico, produce el software, los archivos y un sistema que funciona.

Las especificaciones de diseño indican lo que el sistema debe hacer. a su vez, se escriben los programas que aceptan la entrada proporcionada por los usuarios, procesan los datos, producen reportes y guardan los datos en los archivos.

6.2.2.- APOYO PARA LAS ACTIVIDADES DE LA EMPRESA

Un objetivo fundamental en el diseño de un sistema de información es asegurar que éste brinde apoyo a la actividad de la empresa para la que fue desarrollado. En otras palabras, la tecnología de cómputo especificada en el diseño siempre debe de tener un papel secundario en relación con los resultados que se pretende que el sistema proporcione. De manera similar, el diseño tiene que ajustarse a la forma en que trabaja la empresa.

6.2.3.- 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 una forma aceptable y efectiva.
- ♦ Produce resultados exactos.
- ♦ Proporciona una interface y métodos de interacción aceptables.
- ♦ Es percibido por los usuarios como un sistema confiable.

El objetivo del diseño de sistemas es alcanzar los dos siguientes objetivos:

- Determinar el sistema correcto.
- Proporcionar el sistema correcto.

6.2.4.- PROPORCIONAR UN SISTEMA QUE SEA FÁCIL DE UTILIZAR

Muchas de las características técnicas de un sistema de información como son su confiabilidad, exactitud y velocidad de procesamiento, son secundarias en relación con los aspectos humanos del diseño de sistemas. Por consiguiente, se debe procurar diseñar el sistema con ingeniería dirigida hacia las personas y, en consecuencia, incluyen características ergonómicas.

INGENIERÍA HUMANA: Después que el sistema ha sido instalado, los usuarios comienzan a interactuar con el sistema. A medida que el entusiasmo y el miedo por la nueva aplicación toma su cauce normal y su uso se convierte en algo rutinario, los usuarios comienzan a examinar y probar sus características.

Es en este contexto donde las características de *ingeniería humana* a menudo tiene mayor importancia de las técnicas. Si los sistemas de información no son diseñados *para las personas* entonces no tendrán éxito.

En general, para que un sistema de información sea exitoso, se 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 o la falta de cuidado por parte de ellos.
- ♦ Evite fallas o procedimientos inapropiados que generen perjuicios o complicaciones para los usuarios o vayan en detrimento de la organización.
- ♦ Tenga suficiente flexibilidad para adaptarse a las necesidades de cada usuario.
- ♦ Se adapte a la creciente familiaridad del usuario con el sistema.
- ♦ Funcione, en general, en una forma que parezca natural al usuario.

A menudo se emplea el término *orientado hacia el usuario* para designar aquellos sistemas que exhiben excelentes características de ingeniería humana, desafortunadamente, el empleo frecuente de este término en la publicidad lo ha hecho tan popular que existe la tendencia a pasar por alto su significado.

→ Aplicado a nuestros usuarios?

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. El diseño de terminales, sillas y otro tipo de equipos, influye en la fatiga y tensión derivados del uso de dichos objetos. Estos factores afectan a su vez la tasa de errores cometidos por los usuarios cuando ingresan datos en el sistema, la eficiencia de los usuarios y hasta el ausentismo.

La ergonomía tiene relación con la selección de equipo y el diseño de las áreas de trabajo. Sin embargo, 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. En los sistemas más eficientes, se consideran con cuidado y desde diferentes perspectivas, los aspectos ergonómicos del diseño de sistemas.

6.2.5.- AJUSTARSE A LOS ESTÁNDARES DE DISEÑO

Los objetivos del diseño de sistemas son muy amplios y afectan aspectos tanto de la aplicación como de la organización en la que será utilizado el sistema.

Por consecuencia, no debe sorprender el hecho de que los grupos de sistemas de información mejor manejados, también mantengan estándares para el desarrollo de sistemas. Las especificaciones de diseño se establecen dentro del marco fijado por los estándares.

Los siguientes puntos son áreas incluidas en el diseño de estándares:

- ♦ **Estándares para datos:** Lineamientos para asignar nombres a los datos y especificar su longitud y tipo. Con frecuencia están contenidas en el diccionario de datos.

- ♦ **Estándares de codificación:** Abreviaturas y designaciones formales para describir actividades y entidades dentro de la organización.

- ♦ **Estándares estructurales:** Lineamientos sobre *cómo* estructurar el software y el sistema. Políticas para dividir el software en módulos, para la codificación estructurada y la relación existente entre los componentes del sistema.

- ♦ **Estándares de documentación:** Descripción de las características del diseño de sistema, de la relación entre componentes y de las características de operación que puedan ser revisadas para conocer los detalles de la aplicación.

6.3.- CARACTERÍSTICAS CON LAS QUE SE DEBE DISEÑAR UN SISTEMA DE INFORMACIÓN

Las especificaciones de diseño describen las características del sistema, los componentes o elementos del sistema y la forma en que éstos aparecerán ante los usuarios. Para muchos usuarios, el éxito de un sistema está relacionado con la creencia que tengan sobre sí el sistema tiene las características adecuadas.

6.3.1.- ELEMENTOS DE DISEÑO: Los componentes de un sistema de información descritos durante el análisis de requerimientos, son el punto focal del diseño de sistemas. Es necesario tomar en cuenta los siguientes elementos al realizar el diseño de un sistema de información.

- ♦ **Almacenes de datos:** Conjuntos temporales o permanentes de datos.
- ♦ **Procesos:** Actividades para aceptar, manejar y suministrar datos e información. Pueden ser manuales o basadas en computadoras.
- ♦ **Procedimientos:** Métodos y rutinas para utilizar el sistema de información y lograr con ellos los resultados esperados.

- ♦ **Controles:** Estándares y lineamientos para determinar si las actividades están ocurriendo en la forma anticipada o aceptada, es decir si se encuentran “bajo control”. Así mismo, debe especificar las acciones que tienen que emprenderse cuando ocurren problemas o se presentan circunstancias inesperadas.
- ♦ **Funciones del personal:** Las funciones de todas las personas que tienen que ver con el nuevo sistema. Abarca todo el espectro de componentes del sistema, incluso desde la entrada de datos hasta la distribución de salidas o resultados. A menudo, las funciones del personal se establecen en forma de procedimientos.

Estos elementos aparecen una y otra vez en muchas de las características de los sistemas de información. Por consiguiente, todos estos elementos tienen la misma importancia al estructurar el diseño.

→ se consulto a los usuarios

6.3.2.- DISEÑO DE SALIDA: El término 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. En la realidad, muchos usuarios no operan el sistema de información y tampoco ingresan datos en él, pero utilizan la salida generada por el sistema.

Cuando se diseña la salida, se debe realizar lo siguiente:

- ♦ Determinar que 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 se denomina distribución. Para llevar a cabo las actividades antes mencionadas, se requieren decisiones específicas tales como el empleo de formatos ya impresos cuando se preparan reportes o cuántas líneas planear sobre una página para emitir reportes impresos.

6.3.3.- DISEÑO DE ARCHIVOS: El diseño de archivos incluye decisiones con respecto a la naturaleza y contenido del propio archivo, como si se fuera a emplear para guardar detalles de las transacciones, datos de tipo histórico o información de referencia. entre las decisiones que se deben tomar 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. (La estructura de almacenamiento que puede ser secuencial, indexada o relativa.)

No todos los nuevos sistemas de información requieren el diseño de todos los archivos utilizados por la aplicación. Es probable que ya existan archivos maestros porque éstos son utilizados por otras aplicaciones existentes. Tal vez la nueva aplicación necesite hacer referencia sólo al archivo maestro. En este caso, los detalles del archivo se incluyen en las especificaciones de diseño de la aplicación, pero el archivo no vuelve a diseñarse.

6.3.4.- DISEÑO DE INTERACCIONES CON LA BASE DE DATOS: Muchos sistemas de información, ya sea implantados en sistemas de cómputo grandes o pequeños, interactúan con las bases de datos que abarcan varias aplicaciones. Dada la importancia que tienen las bases de datos en muchos sistemas, su diseño es establecido y vigilado por el desarrollador del sistema para determinar las interacciones más apropiadas con la base de datos.

6.3.5.- DISEÑO DE LA ENTRADA: Las decisiones de diseño para el manejo de entradas, especifican la forma en que serán aceptados los datos para su procesamiento por computadora. Al diseñar las entradas se debe decidir si los datos serán proporcionados directamente, quizá a través de una estación de trabajo, o por el uso de documentos donde los datos a su vez serán transferidos hacia la computadora para su procesamiento.

El diseño de la entrada también incluye la especificación de los medios por los que los usuarios darán instrucciones al sistema sobre las acciones que debe emprender.

Entre los detalles que se deben considerar para el diseño de entradas están los siguientes:

- 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 transacción para detectar errores.
- Métodos para llevar a cabo la validación de las entradas y los pasos a seguir cuando se presentan errores.

Los sistemas en línea incluyen un *diálogo o conversación* entre el usuario y el sistema. Por medio del diálogo el sistema le indica al usuario cuándo realizar cierta función. a menudo la naturaleza de la conversación en línea hace la diferencia entre un diseño exitoso y otro inaceptable. Un diseño inapropiado, que deja la pantalla en blanco, confunde al usuario con respecto a qué acción debe emprender.

La disposición de mensajes y comentarios en las *conversaciones en línea*, así como la ubicación de los datos, encabezados y títulos sobre las pantallas o documentos fuentes, también forman parte del diseño de entradas.

6.3.6.- DISEÑO DE CONTROLES: Otro de los aspectos en los que también se debe hacer hincapié, es el de anticipar los errores que se cometerán la 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 posibilidad 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.

6.3.7.- 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 los siguientes:

- ♦ **Procedimientos para entrada de datos:** Métodos para la captura de datos de las transacciones y su ingreso en el sistema de información.
- ♦ **Procedimientos durante la ejecución:** Pasos y acciones emprendidos por los usuarios finales que interactúan con el sistema para alcanzar los resultados esperados.

- ♦ **Procedimientos para el manejo de errores:** Acciones a seguir cuando se presentan resultados inesperados.
- ♦ **Procedimientos de seguridad y respaldo:** Acciones para proteger al sistema y sus recursos contra posibles daños.

6.3.8.- DISEÑO DE ESPECIFICACIONES PARA PROGRAMAS: Las especificaciones para programas son por sí mismas un diseño. Ellas describen cómo transformar las especificaciones de diseño del sistema como son: salidas, entradas, archivos, procesamiento y otras en software de computadora.

Un diseño de software en particular será diseñado sólo una vez, pero será usado repetidamente y es probable que evolucione en la medida que cambien las necesidades de los usuarios o de la misma organización. Estas observaciones añaden más importancia al diseño del software.

El diseño del software de computadora es importante para asegurar que:

- ♦ Los programas producidos llevan a cabo todas las tareas y lo hagan en forma establecida.
- ♦ La estructuración del software en módulos permita su prueba y validación para determinar si los procedimientos son correctos.
- ♦ Las modificaciones futuras se puedan realizar en forma eficiente y con un mínimo de interrupción en el diseño del sistema.

Los métodos para desarrollar el diseño o para especificar los detalles varían de acuerdo con las prácticas establecidas en cada organización. También serán diferentes como consecuencia de los lenguajes utilizados para escribir el software. Los lenguajes de tercera generación requieren una atención mucho mayor en lo que respecta a los detalles de los procedimientos en comparación con la dedicada cuando se emplean lenguajes de cuarta generación.

6.4.- ADMINISTRACIÓN DE BASES DE DATOS

El manejo acertado de datos es la clave para el apoyo exitoso de la información administrativa. Sin un manejo adecuado, datos vitales e información esencial no estarán disponibles para el respaldo de las operaciones de una empresa u organización. De igual modo, la administración deficiente de datos a menudo produce pérdida de la integridad y de la confiabilidad de los datos y de la información que están basados en dicha administración.

La administración de bases de datos consiste en la planeación, organización y control de las bases de datos de una empresa u organismo. Aquí mismo se definen las bases de datos y sus finalidades en los sistemas de información.

QUE ES UNA BASE DE DATOS? Una base de datos es una colección integrada de datos almacenados en diferentes tipos de registros. Los registros se interrelacionan por medio de relaciones propias de los datos, y no mediante su ubicación física en el almacenamiento.

Las bases de datos se distinguen de los archivos maestros comunes y de transacciones en cuatro formas significativas.

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

2^a.- Agregar registros a un archivo para agrandarlo no lo convierte en una base de datos. La existencia de una base de datos no esta determinada por el número de registros almacenados; no es cuestión de tamaño.

3^a.- Los registros referentes a las distintas entidades de interés pueden almacenarse dentro de una base de datos.

4^a.- Tener bases de datos no elimina la necesidad de archivos en un sistema de información. Los archivos de transacciones serán necesarios para capturar detalles de las actividades de la empresa. Los archivos maestros también pueden requerirse en virtud de que no todos los datos necesitan residir en la base de datos. Y, desde luego, los archivos de clasificación son esenciales cuando se deben reordenar los datos.

CUAL ES EL PROPÓSITO DE UNA BASE DE DATOS? El objetivo de almacenar y organizar datos en una base es representar las relaciones entre las entidades de interés para la empresa, un resultado que no se puede alcanzar con archivos maestros individuales. Organizar los datos de este modo facilita la integración de las áreas dentro de la empresa.

CUALES SON LOS OBJETIVOS AL ADMINISTRAR UNA BASE DE DATOS?

Muchas organizaciones tienen sistemas e información muy exitosos y no han tenido necesidad de invertir en administración de base de datos. Con todo, para la mayoría de las empresas la única forma eficaz y eficiente de lograr el nivel deseado de apoyo a los sistemas de información es mediante la administración de bases de datos. Existen siete objetivos para administrar una base de datos, en los cuales se muestran las ventajas que pueden obtenerse.

1° **Evitar la redundancia innecesaria.** A medida que las organizaciones aumentan la dependencia con respecto a los sistemas de información se implementan más aplicaciones. Con frecuencia cada aplicación se diseña para utilizar sus propios archivos maestros y de transacciones, tal vez debido a que los analistas no saben que los mismos datos existen en otros archivos, o porque los usuarios desean tener más control sobre sus propios archivos, o bien porque los archivos son un poco distintos. El resultado, sin embargo, es la abundancia de datos que no son de interés ni para los usuarios ni para la organización. Pero más importante aún, cuando los datos se duplican a través de los archivos no es probable que se hagan cambios en forma simultánea en todos los archivos. Las discrepancias que seguramente resultarán tornan virtualmente imposible saber qué detalles están más actualizados o que reporte contiene la información correcta.

En vez de permitir la redundancia no controlada, el diseño del sistema debe posibilitar compartir los datos a través de las aplicaciones. Se pueden satisfacer las necesidades comunes de datos si los mismos son accesibles a todas las aplicaciones que los requerirán.

2°.- **Proporcionar flexibilidad de acceso.** Frecuentemente los usuarios que solicitan información necesaria para la toma de decisiones suponen que pueden obtenerla porque saben que los datos requeridos se almacenan en una modalidad entendible para la computadora. Por desgracia, a menudo descubren que tales demandas no pueden ser satisfechas con rapidez debido a que:

- * Los datos se hallan en diferentes archivos.
- * Se tiene que escribir un programa para extraer los datos de los diversos archivos.
- * Tomará tiempo desarrollar el programa.
- * Se deben terminar primero otros trabajos.

Resulta irónico que los datos que sirven para las decisiones que tienen el mayor beneficio potencial frecuentemente no sean accesibles de modo inmediato. Estas necesidades exigen el acceso a cualquier dato, en cualquier clave sin considerar dónde se encuentran almacenados los datos, o cómo se debe tener acceso a ellos.

3°.- **Aportar relacionabilidad.** La *relacionabilidad* es la capacidad de definir las relaciones entre entidades o tipos de registros y de *recabar* datos basándose en esas relaciones. Es obvia la importancia de describir los atributos de una entidad cuando se definen registros. Un sistema bien diseñado debe tener la capacidad de ofrecer y brindar la relación de entidades para garantizar un buen servicio, así, se alcanzará una buena administración de la base de datos.

4°.- **Mantener la independencia de los datos.** Los usuarios no tienen por qué limitar su consideración de los datos por la modalidad en que el sistema de cómputo almacena los detalles. La *independencia de los datos* se refiere a la capacidad para tener separados los aspectos lógico y físico de los datos.

Aspecto lógico de los datos: Es el aspecto conceptual que el usuario tiene de los datos, incluyendo los nombres y el contenido de los elementos de los datos, así como la manera en que están organizados; se elige cierto aspecto debido a la forma en que los datos se utilizarán por parte de una persona o en una aplicación.

Aspecto físico de los datos: Es la manera en que los datos se organizan y almacenan en la computadora y en el almacenamiento secundario, incluyendo las estructuras de archivos que almacenan los datos y los métodos de acceso que recaban los detalles.

Es difícil de lograr una independencia completa. En cierto punto, las consideraciones físicas y lógicas deben reunirse de suerte que se puedan procesar los datos.

5°.- Asegurar el desarrollo futuro. El cambio es un concepto clave para la comprensión de la necesidad de alcanzar el objetivo de asegurar el desarrollo futuro de la base de datos. Es posible prever qué cambio ocurrirá en muchas áreas: (claves de registros, estructuras de almacenamiento físico, etc.)

La necesidad de cambios nunca se eliminará. Pero el objetivo es minimizar la dificultad para efectuar modificaciones cuando se vuelven necesarias. Algunos procedimientos para lograrlo son mejores que otros.

6°.- Preservar la integridad de los datos. La *integridad de los datos* se refiere a la *confiabilidad* de los mismos. La redundancia no controlada en los archivos y la necesidad de múltiples actualizaciones casi siempre conduce a problemas de integridad.

Los problemas de integridad también surgen por errores en los datos. Cuando se crean y mantienen archivos es menester efectuar revisiones para cerciorarse de que los datos son correctos. No debe permitirse que los datos que no cumplan con determinadas especificaciones o que contengan errores ingresen en la base de datos.

7°.- **Garantizar la seguridad de los datos.** La *seguridad* con respecto a la base de datos es el resultado de controlar el derecho de acceso a la base y la capacidad de las personas para recabar, agregar o eliminar registros. La seguridad es especialmente trascendente, y difícil de alcanzar en los entornos de grandes bases de datos donde los usuarios interactúan con el sistema a través de líneas de comunicación.

El objetivo de garantizar la seguridad de los datos es evitar todo acceso o empleo no autorizado de la base de datos ya sea en forma accidental o intencional.

6.4.1.- DISEÑO MEDIANTE UN MODELO DE DATOS RELACIONAL

Cuando se diseña una base de datos y se establece el esquema, es importante conocer el modelo de datos subyacente del sistema de administración de base de datos que será utilizado. El *modelo de datos* define cómo se muestran las relaciones entre las entidades de la base de datos.

Una *entidad* es algo que posee ciertos atributos o propiedades a las cuales se les puede asignar valores. Estos valores pueden ser numéricos o no. La forma en que los datos se organizan en la jerarquía: campos, registros y archivos refleja la relación entre atributos, entidades y conjunto de entidades. Así un *campo* es una unidad elemental de información que representa un atributo de una entidad; un *registro* es una colección de campos de una entidad y un *archivo* es una colección de registros de las entidades contenidas en un conjunto de entidades.

Cada sistema de administración de base de datos (SABD) se sirve de un modelo de datos específico. Tres modelos de datos son de uso común en los sistemas de bases de datos: el relacional, el jerárquico y el reticular o de red.

Tomando en cuenta que el Sistema de Cuentas por Cobrar (SCC) es un sistema que se encuentra estrechamente relacionado con el sistema de Control Escolar, se optó por elegir y utilizar diseño bases de datos relacionales.

Utilizando bases de datos relacionales se pretende que el Sistema de Cuentas por Cobrar haga uso de algunas de las bases de datos del Sistema de Control Escolar (SCE), en especial la base de datos de alumnos, ésto para eficientar el sistema y evitar la duplicidad de información en la unidad de disco duro.

El *modelo de datos relacional*, es el modelo mas ampliamente utilizado entre los sistemas de base de datos que en la actualidad adquieren las empresas. El modelo de base de datos en si está basado en una relación de los registros contenidos en un archivo de base de datos. No existe un orden o secuencia en el archivo, la relación es una estructura lógica.

Todos los modelos de datos se guardarán siguiendo una de las organizaciones de archivos que puede emplear un sistema de cómputo, una estructura de almacenamiento aleatoria, indexada o secuencial.

El modelo de datos relacional es en el presente el fundamento más generalizado para nuevos sistemas de administración de base de datos, en virtud de que:

- 1°.- Las bases de datos relacionales son conceptualmente muy simples y fáciles de entender.
- 2°.- Puesto que las relaciones no requieren ser predefinidas, la base de datos puede transformarse para satisfacer condiciones cambiantes.
- 3°.- Las relaciones quedan implícitas en los valores de los datos.

6.4.2 ESTRUCTURAS DE LAS BASES DE DATOS UTILIZADAS EN EL SISTEMA DE CUENTAS POR COBRAR

El sistema de cuentas por Cobrar cuenta con 11 bases de datos, seis de ellas son propias del sistema como son: (Edocuent.dbf, diario.dbf, pago.dbf, concepto.dbf, y mes.dbf) y las cinco restantes (Alumnos.dbf, impres.dbf, escapes.dbf, inicio.dbf y config.dbf) son bases de datos externas que utilizo como apoyo para cumplir con los requerimientos de ejecución. Cabe mencionar que la base de datos (DMes/Día/Año.dbf) se crea diariamente, en la cual se guardan los movimientos realizados en dicha fecha.

Estructura de EDOCUENT.DBF			
Nombre campo	Tipo	Anchura	Dec
MATRIALUM	Carácter	7	
BECA	Numérico	3	0
INSCRIPCIO	Numérico	7	2
CARGO1	Numérico	7	2
PAGADO1	Lógico	1	
SALDO1	Numérico	7	2
CARGO2	Numérico	7	2
PAGADO2	Lógico	1	
SALDO2	Numérico	7	2
CARGO3	Numérico	7	2
PAGADO3	Lógico	1	
SALDO3	Numérico	7	2
CARGO4	Numérico	7	2
PAGADO4	Lógico	1	
SALDO4	Numérico	7	2
CARGO5	Numérico	7	2
PAGADO5	Lógico	1	
SALDO5	Numérico	7	2
CARGO6	Numérico	7	2
PAGADO6	Lógico	1	
SALDO6	Numérico	7	2
SALDOANT	Numérico	7	2

Tabla 6.1

Estructura de ALUMNOS.DBF			
Nombre campo	Tipo	Anchura	Dec
INSCRITO	Lógico	1	
MATRIALUM	Carácter	7	
NOMALUM	Carácter	35	
SEMESALUM	Carácter	2	
SEXALUM	Carácter	1	
DOMALUM	Carácter	35	
COLALUM	Carácter	20	
CPALUM	Carácter	5	
CIUDALUM	Carácter	20	
TELALUM1	Carácter	15	
ESCPROALUM	Carácter	20	
NOMTUTALUM	Carácter	35	
TELALUM2	Carácter	15	
FEINSALUM	Fecha	8	
GRUPO	Carácter	3	
BAJA	Lógico	1	
MAT1	Carácter	5	
P11	Numérico	4	1
I11	Numérico	2	0
P12	Numérico	4	1
I12	Numérico	2	0
F13	Numérico	4	1
I13	Numérico	2	0
MAT2	Carácter	5	
.	.	.	.
.	.	.	.
MAT13	Carácter	5	
P131	Numérico	4	1
I131	Numérico	2	0
P132	Numérico	4	1
I132	Numérico	2	0
F133	Numérico	4	1
I133	Numérico	2	0

Tabla 6.2

Las bases de datos mostradas anteriormente (EDOCUENT.DBF y ALUMNOS.DBF), son aquellas bases de datos en las cuales se lleva a cabo la relación de los campos para poder efectuar las operaciones internas en el sistema. Dicha relación se establece a través del campo MATRIALUM el cual se encuentra contenido en ambas bases. Al momento en que se desea efectuar alguna de las diversas operaciones que ofrece el sistema, se pide la matrícula del alumno, al momento en que se teclea dicha matrícula, se establece la relación entre la base de datos EDOCUEN.DBF y la base de datos ALUMNOS.DBF de la escuela a la que pertenece el alumno y checa si se encuentra, en caso de que no lo encuentre aparece un mensaje de que el alumno no se encuentra, no está dado de alta en ese grupo o simplemente no está reinscrito.

La relación de los datos se efectúa en el archivo fuente PAGOS.PRG en la función Pago_colegiatura (.). Dicha relación se establece de la siguiente manera:

SET RELATION TO matrialum INTO edocuent

Estructura de DIARIO.DBF			
Nombre campo	Tipo	Anchura	Dec
TIPO	Carácter	1	
CLAVE	Carácter	4	
DESCRIP	Carácter	15	
FECHA	Fecha	8	
IMPORTE	Númerico	7	2

Tabla 6.3

Estructura de PAGO.DBF			
Nombre campo	Tipo	Anchura	Dec
PAGO	Carácter	25	

Tabla 6.4

Estructura de CONCEPTO.DBF			
Nombre campo	Tipo	Anchura	Dec
CONCEPTO	Carácter	55	

Tabla 6.5

Estructura de MES.DBF			
Nombre campo	Tipo	Anchura	Dec
TIPO	Carácter	1	
MES	Carácter	12	

Tabla 6.6

Estructura de INICIO.DBF			
Nombre campo	Tipo	Anchura	Dec
KARDEX	Lógico	1	
CFIYA	Lógico	1	
CUOTAS	Lógico	1	

Tabla 6.7

Estructura de DMES/DIA/AÑO.DBF			
Nombre campo	Tipo	Anchura	Dec
CLAVE	Carácter	5	
ESCUELA	Carácter	55	
FECHA	Fecha	8	
HORA	Carácter	5	
MATRIALUM	Carácter	7	
CONCEPTO	Carácter	50	
FORMAPAGO	Carácter	25	
MONTO	Númérico	7	2

Tabla 6.8

Estructura de IMPRES.DBF			
Nombre campo	Tipo	Anchura	Dec
NIMP	Númérico	3	0
DSIMP	Carácter	35	

Tabla 6.9

Estructura de ESCAPES.DBF			
Nombre campo	Tipo	Anchura	Dec
NIMP	Numérico	3	0
ESCAPE	Carácter	12	
VALOR	Carácter	40	

Tabla 6.10

Estructura de CONFIG.DBF			
Nombre campo	Tipo	Anchura	Dec
CLAVE	Carácter	5	
NIVEL	Carácter	1	
NIMP	Numérico	3	0
PUERTO	Carácter	8	

Tabla 6.11

6.4.3.- VALIDACIÓN DE LAS ENTRADAS

La validación de entradas es el conjunto de acciones tomadas durante el ingreso de los datos con la finalidad de detectar errores en los datos antes de que ingresen en la base de datos. La detección de los errores en este nivel evita costosas actualizaciones después que los datos han sido aceptados en el almacenamiento y facilita corregir errores rápidamente, sin tener que esperar un largo periodo de recorrido y regreso. Los errores disminuyen la calidad de los datos y pueden contribuir a fallas en el sistema y a una degradación de la base de datos tan grave que la misma ya no sea un recurso preciso y confiable para la empresa.

6.4.4.- SEGURIDAD Y CONFIDENCIALIDAD

Las *medidas de seguridad* están orientadas a preservar la información en los datos impidiendo cualquier intromisión que pudiera conducir a la destrucción de archivos 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) es un sistema en línea, con el propósito de destruir, modificar o tener acceso a los datos sin permiso. La confidencialidad o 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 a 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.

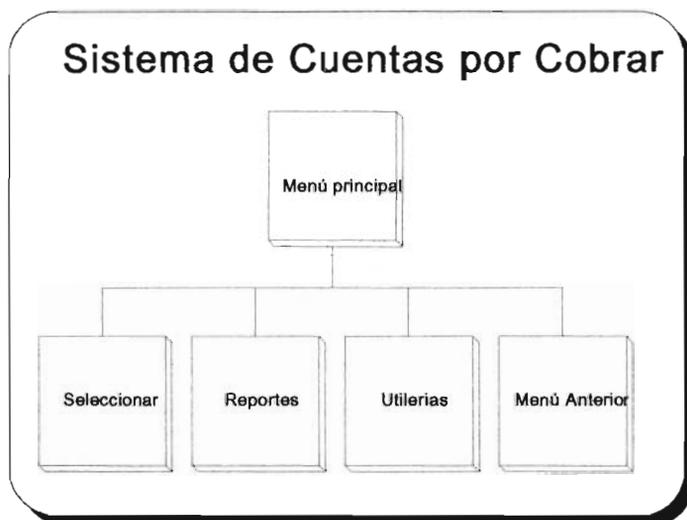
Una tercera fase de la reglamentación es la *vigilancia*. La conservación de registros de todas las personas que utilizan los datos y el examen de los registros de modo periódico es una forma relativamente sencilla de vigilar el acceso y tal vez detectar actividades indebidas.

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 encriptamiento.

7.- DESCRIPCIÓN DEL SISTEMA PROPUESTO

El sistema de Cuentas por Cobrar esta integrado por distintos módulos, los cuales a su vez cuentan con variadas opciones como se ilustra a continuación.

7.1.- MENÚ PRINCIPAL



Esquema 7.1

Dependiendo de la opción que el usuario elija o desee utilizar, se desplegarán distintas ventanas con variadas opciones según haya sido la elección del usuario.

A continuación se presenta como está integrado el sistema de Cuentas por Cobrar en lo que se refiere a su uso, proporcionando al mismo tiempo, una explicación de cada uno de los módulos así como de cada una de las opciones del sistema.

El módulo de **Seleccionar** ofrece al usuario la opción de elegir la escuela con la que desea trabajar para llevar a cabo cualquiera de las variadas opciones que ofrece el sistema.

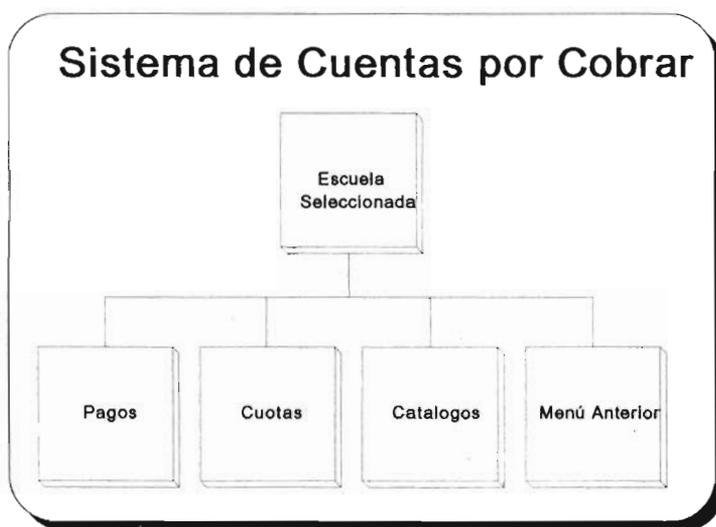
En el módulo de **Reportes** el sistema permite al usuario emitir la impresión de distintos tipos de reportes como son Antigüedad de saldos, Becarios, Corte de caja e Ingresos por escuela.

El módulo de **Utilerias** le permite al usuario llevar a cabo la reindexación de las principales bases de datos del sistema.

La última opción que se ofrece al usuario es la del módulo de **Menú Anterior** en la cual, el usuario puede regresar al Shell del sistema.

- ♦ **7.1.1.-** En el módulo **Seleccionar** se ofrece la opción para que el usuario pueda escoger la escuela con la que desea trabajar; se muestra una pantalla en la cual se visualizan las escuelas que están dadas de alta, su clave interna y el nivel al que pertenece dicha escuela.

Una vez que el usuario haya elegido la escuela con la que se valla a trabajar, independientemente de la escuela que se trate, ya sea de nivel preparatoria, licenciatura, diplomado, maestría o doctorado, el sistema ofrece cuatro módulos de trabajo con sus respectivas opciones, apareciendo otro menú adicional, el cuál se muestran a continuación:



Esquema 7.2

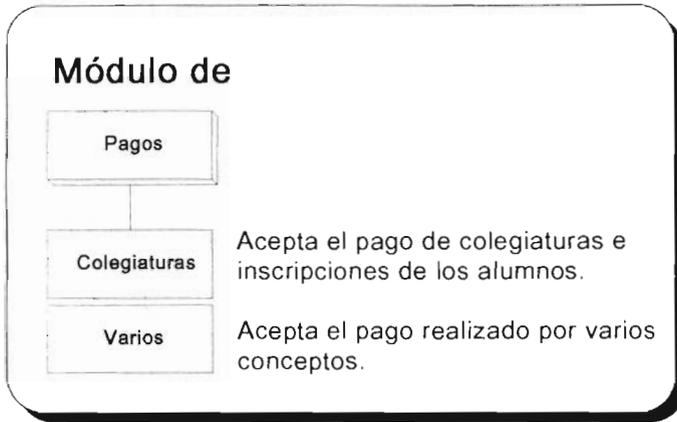
En el módulo de **Pagos** se puede llevar a cabo el pago de colegiaturas por escuela, así como el pago por conceptos varios.

En el módulo de **Cuotas** se permite cargar las cuotas (inscripciones y colegiaturas) de los alumnos en forma general, es decir, a todos los alumnos de una escuela determinada, o en forma individual (por alumno) de un grupo determinado y de la escuela que se haya elegido. También se ofrece la posibilidad de asignar el porcentaje de beca por alumno.

En el módulo de **Catálogos** se realizan las operaciones de asignación y manejo del catálogo del sistema como son las formas de pago (efectivo, cheque, tarjeta de crédito, etc.) y el concepto por el cual se obtuvo un ingreso (constancias de estudio, certificados, memorándums, examen de titulación, etc.). En ambas opciones se permite dar la alta, baja o modificación de un concepto o forma de pago.

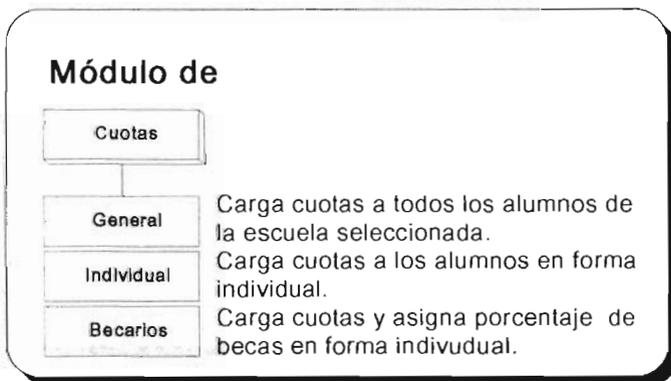
Por último se ofrece la opción de **Menú Anterior**, en la cual se tiene el regreso al menú principal del sistema.

- ♦ 7.1.1.1.- En caso de que la opción seleccionada sea el módulo de **Pagos**, aparecerán dos opciones a elegir "**Colegiaturas**" y "**Varios**", como se ilustra a continuación:



Esquema 7.3

- ♦ 7.1.1.2.- Si la opción elegida fue el módulo de **Cuotas**, aparecerá una ventana con tres opciones a elegir tales como *“General”*, *“Individual”* y *“Becarios”*, los cuales se presentan a continuación:



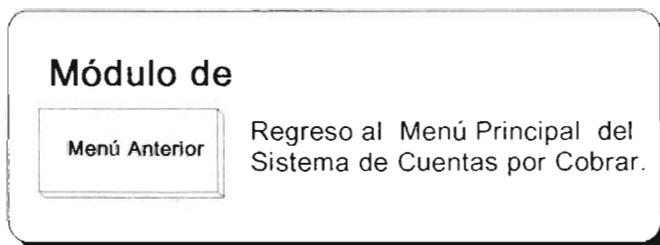
Esquema 7.4

- ♦ 7.1.1.3.- Si el usuario escoge el módulo de **Catálogos**, se visualizará otra ventana ofreciendo dos opciones más, las cuales son “*Conceptos de pago*” y “*Formas de pago*” que se presentan a continuación:



Esquema 7.5

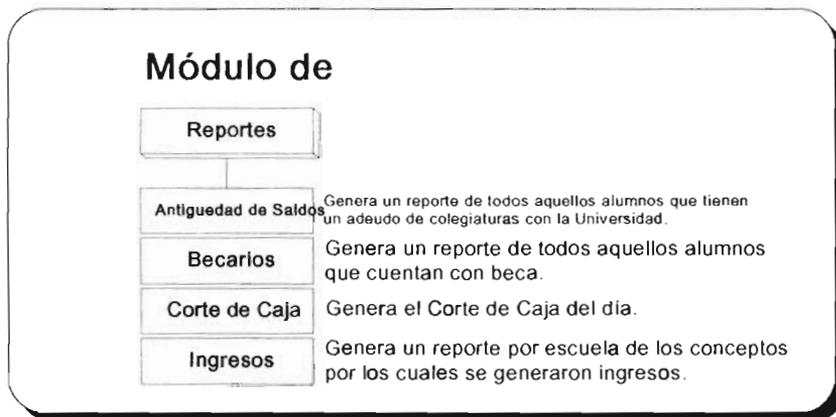
- ♦ 7.1.1.4.- En lo que se refiere a la opción de **Menú anterior** se ofrece la opción de retorno al menú principal del sistema de Cuentas por Cobrar como se muestra:



Esquema 7.6

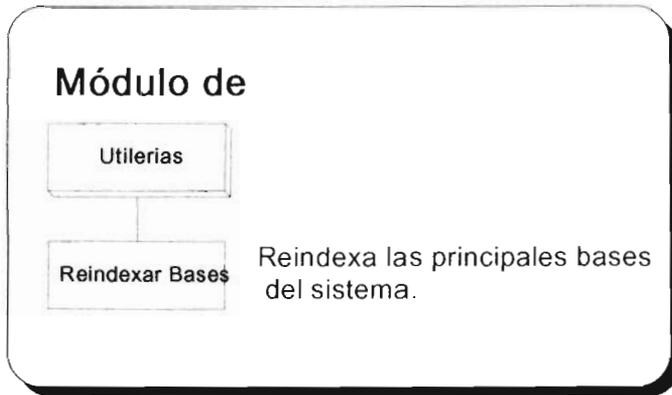
Si la opción seleccionada no fue la del módulo de Seleccionar del Menú Principal del sistema de Cuentas por Cobrar, se ofrecen como se mostró anteriormente otras tres opciones más que son Reportes, Utilerías y Menú Anterior.

- ♦ **7.1.2.-** Si el módulo elegido del Menú Principal fue el de **Reportes**, el sistema muestra una ventana que ofrece cuatro opciones o tipos de reportes, de los cuales, el usuario puede mandar a imprimir, esto dependiendo de cual haya sido su elección. Dichas opciones se muestran a continuación:



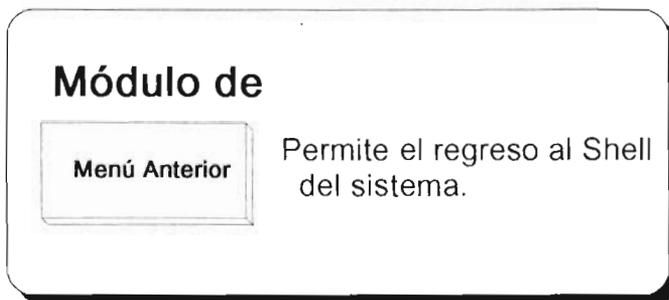
Esquema 7.7

- ♦ **7.1.3.-** En caso de que el usuario haya seleccionado el módulo de **Utilerías**, aparecerá una sola opción de **Reindexar Bases**, como se muestra a continuación:



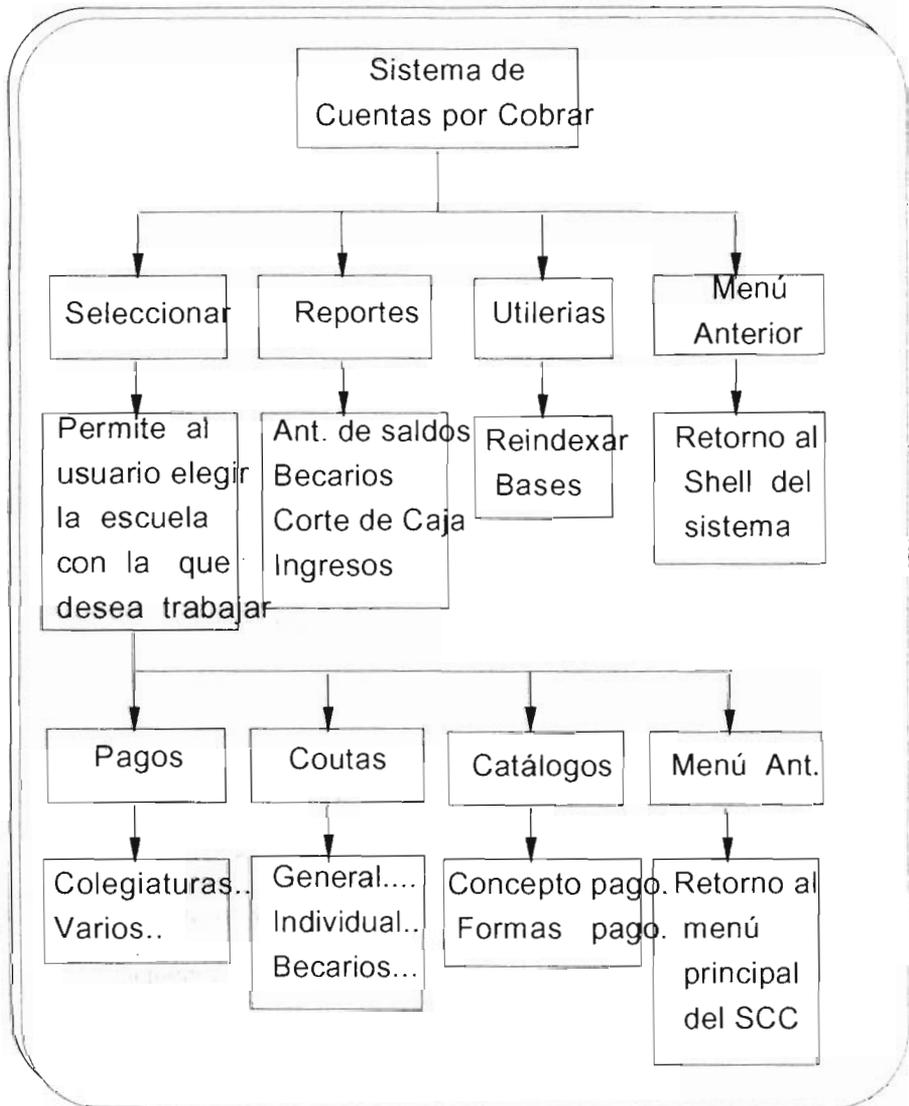
Esquema 7.8

- ♦ 7.1.4.- La última opción que se ofrece en el menú principal del sistema de Cuentas por Cobrar es el módulo de regreso al **Menú Anterior** como a continuación se presenta:



Esquema 7.9

7.2.- DIAGRAMA DEL SISTEMA PROPUESTO



Esquema 7.10

8.- CONCLUSIONES Y RECOMENDACIONES

A lo largo del presente trabajo de tesis se ha demostrado que el Sistema de Cuentas por Cobrar es una valiosa herramienta que contribuye en gran escala a la Universidad Vasco de Quiroga para lograr el fortalecimiento productivo de las actividades internas del área administrativa así como en el engrandecimiento y mejoramiento del desarrollo de la institución puesto que se han abarcado y cumplido los objetivos y las metas establecidas desde la presentación de la propuesta.

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

Con la implantación del Sistema de Cuentas por Cobrar también se logró que el procedimiento de pagos, el cobro de colegiaturas, inscripciones así como la emisión del corte de caja, ingresos por escuela y antigüedad de saldos se efectúe eficazmente ya que se realizan en tiempo real y en forma automática, reduciendo el tiempo invertido en cada una de las operaciones diarias en comparación con el diagnosticado cuando dichas operaciones se hacían a través de procedimientos manuales, anulando al mismo tiempo cualquier margen de error que se pudiera presentar en caja al realizar cualquiera de las diversas actividades cotidianas.

El Sistema de Cuentas por Cobrar se encuentra diseñado de tal forma que la interfaz con el usuario es amigable y transparente en su utilización, y puede ser utilizado por cualquier persona, no siendo necesario que los usuarios cuenten con conocimientos en computación puesto que sus menús y sus pantallas son de fácil entendimiento, además de contar en todo momento con ayuda en línea que le proporciona al usuario una breve explicación o descripción de cada una de las opciones de los menús; así mismo, se ofrece en cada una de las pantallas ayuda en línea de las teclas auxiliares o de función que el usuario puede utilizar para complementar su funcionamiento.

Tomando en cuenta el despunte que han marcado los sistemas informáticos en los últimos tiempos, en la actualidad para muchas empresas, instituciones y organizaciones el uso e implantación de sistemas de información constituyen el corazón en el desarrollo de sus actividades cotidianas, por lo tanto, se hace presente la necesidad de encaminar en forma adecuada el uso de los mismos para lograr obtener resultados óptimos y de calidad, siendo prescindible dar mantenimiento a aquellas aplicaciones de las cuales hacen uso los usuarios finales. En base a lo anterior, a continuación presento algunas recomendaciones que sería importante seguir para obtener una mayor eficiencia del Sistema de Cuentas por Cobrar:

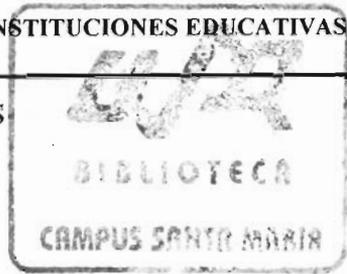
- ♦ Capacitar al personal que va a hacer uso del sistema para evitar un mal uso del mismo, y así, evitar errores que pudieran perjudicar los procedimientos de las áreas implicadas, en especial en lo que se refiere al módulo de pagos.

- ♦ Asignar a los usuarios claves de acceso por parte del administrador de sistemas para evitar la infiltración de información y de usuarios no autorizados. Con esto se logrará asignar privilegios de acceso a los usuarios autorizados a ciertos niveles y a determinadas actividades, deslindándoles en forma efectiva sus responsabilidades.
- ♦ Cuando se lleve a cabo el periodo de reinscripciones, solicitar al departamento de control escolar la inicialización del semestre en cada uno de los grupos de las diferentes escuelas en el Sistema de Control Escolar, así como la reinscripción de los alumnos en su nuevo grupo para evitar que el Sistema de Cuentas por Cobrar rechace el cobro de su reinscripción al emitir un mensaje de que el alumno no esta reinscrito o no esta dado de alta en ese grupo.
- ♦ Para el caso de alumnos de nuevo ingreso, solicitar a control escolar que los alumnos estén dados de alta en el Sistema de Control Escolar en la escuela que les corresponda y que previamente se hayan formado los grupos, en caso de no ser así, el sistema rechazará el cobro de su inscripción.
- ♦ Asignar previamente al periodo de inscripciones el cargo de cuotas (colegiaturas) en cada uno de los niveles educativos de la universidad, (preparatorias y licenciaturas).

- ♦ Asignar previamente al periodo de inscripciones el porcentaje de beca a aquellos alumnos que cuenten con ella, logrando agilizar el cobro de colegiaturas, evitando así un mal cobro y por ende una entrada de ingresos errónea en la base de datos.
- ♦ Para eficientar el control de acceso al Sistema de Cuentas por Cobrar se tomó del Sistema de Control Escolar la función correspondiente con el nombre Claves (), la cual se omite en el apéndice 9.B. - Código fuente por razones obvias.
- ♦ Tomando en cuenta de que el Sistema de Cuentas por Cobrar se encuentra estrechamente ligado con el Sistema de Control Escolar, en caso de que los usuarios trabajen en algún ambiente de redes compatibles como lo es NOVELL, al momento de llevar a cabo la ejecución del mismo el sistema detecta si el equipo de cómputo se encuentra trabajando bajo este ambiente o no.
- ♦ Cuando se trate de la aceptación de inscripciones, pago de módulos en caso de diplomados o de materias en caso de postgrados, solicitar a control escolar que los alumnos estén dados de alta en el Sistema de Control Escolar en el nivel que les corresponda y que previamente se hayan formado los grupos, en caso de no ser así, el sistema rechazará el cobro de su inscripción , del módulo o de las materias.

- En caso de que el usuario introduzca erróneamente un ingreso falso en el sistema, ya sea en la opción de pago de colegiaturas o de pagos varios en el módulo de pagos, localizar al diseñador del sistema o a alguna persona autorizada para corregir dicho error y evitar que el corte de caja, así como los reportes de ingresos por escuela proporcionen una cantidad falsa de ingresos al emitir el reporte de los mismos.
- Realizar un respaldo mensual para evitar la pérdida de información o el borrado accidental de las principales y más importantes bases del sistema.

- Sistema de Seguimiento del programa?*
- *Seguimiento del programa?*
 - elementos que deben revisarse
 - actualizarse.
 - *Ergonomía - aplicada en la implementación.*
 - *Nivel de capacitación técnica del personal actual*
 - *Relación concreta cuentas por cobrar con Control Escolar.*
 - Verificación de alumnos inscritos económicamente y académicamente.*
 - *Riesgo de manejo de efectivo*
 - Banco - Conectados



9.- APÉNDICES

El uso de las computadoras, sobre todo las computadoras personales, se ha extendido tan rápidamente que ya no es posible hacer caso omiso de su existencia. Actualmente las computadoras proporcionan información requerida para aumentar el poder intelectual de las organizaciones. Tales computadoras no operan en un vacío; su uso afecta a las personas y a las organizaciones en todo el mundo.

Los sistemas de información nos permiten emplear las computadoras para desarrollar y facilitar múltiples actividades de operación vital que contribuyen en el fortalecimiento y engrandecimiento de las organizaciones. Si dichas actividades se desarrollan a través de procesos manuales, aparte de consumir tiempo y personal, se evita que tales entidades prosperen y obtengan un impulso a nivel interno y externo.

Los sistemas de información o software, son las instrucciones que controlan el funcionamiento de la máquina, los cuales, se escriben para controlar el procesamiento de una tarea determinada y obtener de los mismos determinados resultados deseados.

- Forma de Pago - Efectivo único
 - Preservar los datos frables
 - Ventanilla única - ?
- Possibilidad

→ Los recargos están acumulados y como se amilan

9.A.- MANUAL DEL USUARIO

CONTENIDO

Propósito

Este manual tiene la intención de explicarle al usuario en forma fácil y entendible la manera correcta de utilizar el Sistema de Cuentas por Cobrar para que su uso sea transparente. No siendo necesario contar por parte del usuario con experiencia o instrucción en sistemas.

Organización

Este manual contiene once capítulos, los cuales se describen a continuación:

Capítulo 1 Características y requerimientos Pág. 85

Se proporcionan algunas de las principales características del sistema de Cuentas por Cobrar y requerimientos para su ejecución.

Capítulo 2 Instalación Pág. 87

Se establecen los pasos necesarios para llevar a cabo la instalación del sistema.

Capítulo 3 Sistema de Cuentas por Cobrar Pág. 90

En este capítulo se presenta un diagrama de como se encuentra diseñado y modulado el Sistema de Cuentas por Cobrar.

Capítulo 4 Introducción Pág. 91

En este capítulo se proporciona una breve explicación a partir del momento que se lleva a cabo la ejecución del sistema.

Capítulo 5 Menú principal Pág. 93

En esta sección se muestra y describe detalladamente cada una de las opciones del menú principal del sistema.

Capítulo 6 Módulo “Seleccionar” Pág. 95

Este capítulo describe el objetivo de la opción del módulo de *Seleccionar*, aquí el usuario puede elegir la escuela con la que desea trabajar para ejecutar cualquiera de las opciones que ofrece el sistema.

Capítulo 7 Módulo de “Pagos” Pág. 98

En este capítulo se presenta una descripción de las operaciones que se pueden realizar con la elección del módulo de *Pagos*, pudiendo llevar a cabo el pago de colegiaturas por escuela, así como la aceptación de pagos por conceptos varios.

Capítulo 8 Módulo de “Cuotas” Pág. 116

En el capítulo quinto, el módulo de *Cuotas* permite cargar las cuotas (inscripciones y colegiaturas) a los alumnos en forma general, es decir, a todos los alumnos de determinada escuela, o en forma individual (por alumno) de un grupo determinado y de la escuela que se haya elegido. Otra de las opciones que se ofrece en éste módulo es la asignación de los porcentajes de beca por alumno.

Capítulo 9 Módulo de “Catálogos” Pág. 124

El presente capítulo describe el módulo de *Catálogos* así como la forma de operación de las distintas opciones que se ofrecen en el mismo. El usuario puede registrar, dar de baja o modificar las formas de pago más comunes (efectivo, cheque, tarjeta de crédito, etc.) y los conceptos por los cuales se obtienen ingresos (constancias de estudio, certificados, memorándums, examen de titulación, etc.).

Capítulo 10 Módulo de “Menú Anterior” Pág. 137

Este capítulo proporciona la opción del módulo de *Menú Anterior* para que el usuario pueda regresar al menú principal del Sistema de Cuentas por Cobrar, éste en caso de que el usuario se encuentre en el menú secundario del sistema

Capítulo 11 Módulo de “Reportes” Pág. 139

El octavo capítulo describe la forma de como puede llevarse a cabo la emisión de distintos reportes como son (Antigüedad de saldos, Becarios, Corte de caja e Ingresos por escuela.).

Capítulo 12 Módulo de “Utilerías” Pág. 153

Este capítulo muestra como llevar a cabo la reindexación de las principales bases de datos del sistema a través del módulo de *Utilerías*.

Capítulo 13 Módulo de “Menú Anterior” Pág. 157

En este último módulo del menú principal se ofrece la opción de *Menú Anterior*, en la cual se permite el regreso al Shell del sistema.

Capítulo 14 Módulo de “Claves de acceso” Pág. 159

Este último capítulo va destinado principalmente al administrador del sistema, ya que se considera como la persona indicada y autorizada para realizar la asignación de claves y privilegios a nuevos usuarios, la eliminación de usuarios del sistema y modificación las claves de los mismos.

Capítulo 1

“Características y Requerimientos”

Esta versión del Sistema de Cuentas por Cobrar incluye algunas características que transformarán el equipo de cómputo en una herramienta potente para su uso interno dentro de las diversas áreas que integran las organizaciones. Estas características se describen brevemente en esta sección.

- El Sistema de Cuentas por Cobrar puede ser instalado, si así se requiere, en ambiente MS-DOS, no viéndose afectado en absoluto al momento de su ejecución.
- Para obtener un uso adecuado del equipo de cómputo, el Sistema de Cuentas por Cobrar puede ser instalado bajo ambiente Windows para trabajo en grupo, así, se podrá obtener un mejor aprovechamiento de los recursos, información y aplicaciones que ofrece este sistema operativo.
- El Sistema de Cuentas por Cobrar puede ser instalado en sistemas de cómputo que se encuentran operando bajo ambiente NOVELL y en ambiente monousuario.

La siguiente tabla muestra los requisitos mínimos que se requieren de hardware para la instalación del Sistema de Cuentas por Cobrar.

REQUISITOS	
<i>Componente</i>	<i>Descripción</i>
Computadora	Procesador 386SX o superior.
MS-DOS	Se recomienda la versión 6.0 (para comprobar su versión escriba ver a continuación del símbolo de MS-DOS.)
Memoria	3MB de RAM; se recomiendan 4MB para que no cause conflictos entre las aplicaciones en Windows.
Espacio en disco	10 MB de espacio libre en el disco; se recomiendan 15 MB.
Componentes	Una unidad de Floppy de 3.5". Un monitor VGA o de resolución superior.
Hardware de red	Para conectar el sistema de cómputo mediante alguno de los distintos tipos de red, se necesitarán cables, tarjetas y conectores.

Capítulo 2

“Instalación”

Dentro de las peculiaridades que ofrece el sistema de Cuentas por Cobrar , 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 para trabajo en grupos y NOVELL. A continuación se establecen los pasos necesarios para realizar la instalación del sistema según la opción de trabajo seleccionada.

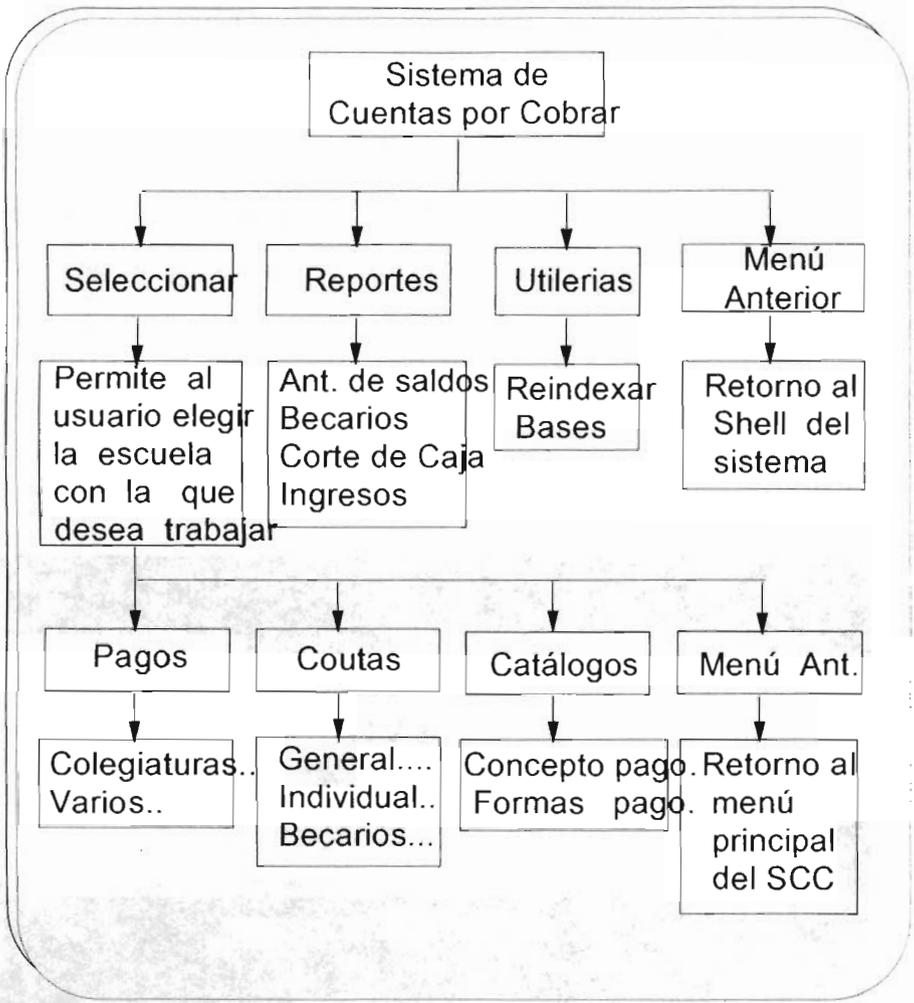
- ♦ Para instalar el sistema en ambiente MS-DOS
 1. Inicialice la computadora.
 2. Introduzca el disco de instalar en la unidad de disquetes.
 3. A continuación del simbolo del sistema, escriba la letra de la unidad que esté utilizando, seguida de dos puntos (:) y presione ENTER o INTRO como se muestra a continuación: C:\> A: <INTRO>
 4. Escriba **Instalar** y presione la tecla INTRO.

Nota: Si el sistema fue instalado para trabajar en ambiente monousuario, será necesario que el administrador de sistemas solicite a control escolar una copia del último respaldo realizado en el Sistema de Control Escolar y llevar a cabo la recuperación de la información a través del mismo para de esta forma poder iniciar la ejecución del Sistema de Cuentas por Cobrar.

- ◆ Para instalar el sistema en ambiente Windows para trabajo en grupos
 1. Inicialice la computadora.
 2. Ejecute Windows de la siguiente manera: `C:\> WIN` y oprima la tecla ENTER.
 3. Introduzca el disco de instalar en la unidad de disquetes.
 4. Ejecute la aplicación del Administrador de archivos. Una vez ejecutado, ubique el indicador del mouse en el icono de la unidad de disco donde haya introducido el disquete y haga clic para que lea la unidad.
 5. Leída la unidad, ubique el indicador del mouse en el archivo que lleva el nombre **Instawin.bat** y haga doble clic para que se inicie la instalación del sistema.
 6. Concluida la instalación, salga de la aplicación del administrador de archivos y salga de Windows. Posteriormente vuelva a ejecutar Windows como lo hizo en el paso 2 para que se configure el sistema.
 7. Para ejecutar el sistema, basta con ubicar el indicador del mouse en el icono y hacer doble clic.

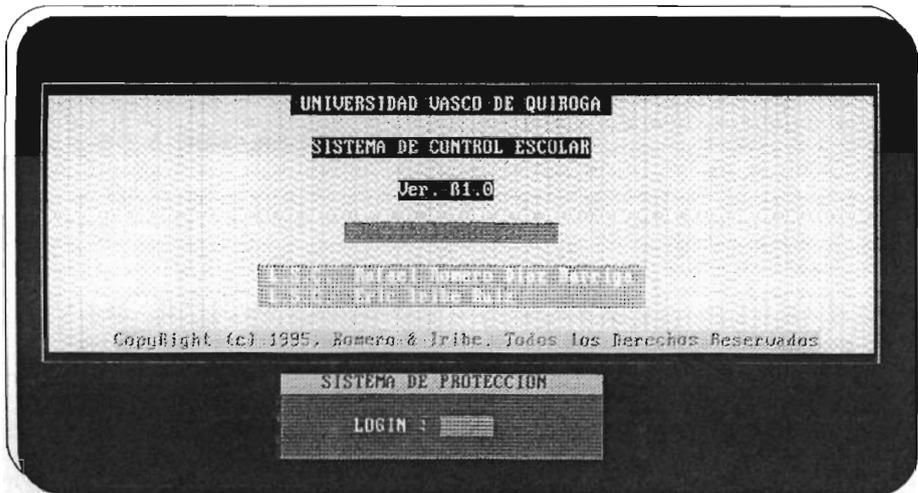
- ♦ Para instalar el sistema en ambiente NOVELL
 1. Inicialice la computadora.
 2. Introdúzcase al ambiente de red.
 3. Introduzca el disco de instalar en la unidad de disquetes.
 4. A continuación del símbolo del sistema, escriba la letra de la unidad que esté utilizando, seguida de dos puntos (:) y presione ENTER o INTRO como se muestra a continuación: H:\> A: <INTRO>
 5. Escriba **Instared** y presione la tecla INTRO.

Capítulo 3
"Sistema de Cuentas por Cobrar"



Capítulo 4
“Introducción”

Para poder iniciar la ejecución del Sistema de Cuentas por Cobrar, es necesario teclear a continuación del prompt del sistema lo siguiente: **C:\> SCE** y oprimir la tecla **INTRO** o **ENTER**, siempre y cuando se valla a ejecutar en ambiente MS-DOS. En caso de que se desee ejecutar en ambiente Windows para trabajo en grupos, sólo basta con ubicarse en el grupo del sistema, mostrar el indicador del mouse en el icono indicado y hacer doble clic en el botón del mouse. Una vez que se ejecuta el sistema, ya sea en Windows o en MS-DOS, como primera instancia se presenta una pantalla de presentación en la cual se le solicita al usuario teclee su clave o login de acceso tal como se muestra a continuación:



Cabe mencionar que el usuario tiene tres oportunidades para tener acceso al sistema, en caso de que se fracase en los tres intentos, el sistema emitirá el siguiente mensaje “INTRUSO EN EL SISTEMA. NO TIENE PERMITIDO EL ACCESO”. Así se ofrece un óptimo control de acceso tanto al sistema como a la información y se evita que personas ajenas se infiltren.

Nota: El Sistema de Cuentas por Cobrar ofrece en todo momento durante su utilización ayuda en línea, por lo cual, es importante que el usuario tome en cuenta los siguientes puntos:

- El cuadro de diálogo superior muestra al usuario la fecha de trabajo, el menú del sistema en el cual se encuentra trabajando y/o el nombre de la escuela donde esta realizando alguna operación.
- Cada una de las pantallas con que cuenta el sistema ofrece al usuario ayuda en línea en la parte inferior de la pantalla, ya sea mostrando una breve explicación de las opciones de los diferentes menús o haciendo referencia a las teclas auxiliares o de función con las que cuenta el usuario para complementar su trabajo.

Sirviendo este manual de igual manera a aquellos usuarios que cuentan con otros privilegios, omitiéndose sólo algunas pantallas al momento de ejecutar el sistema con su clave.

Una vez que el usuario, en este caso el administrador, haya realizado la ejecución del sistema visualizará una pantalla como la que se muestra el inicio de este capítulo, en la cual se ofrecen tres opciones a elegir. Control Escolar, Cuentas por Cobrar y Salir al DOS.

La primera opción que se ofrece en el menú principal del sistema, es la de acceso al Sistema de Control Escolar. Considerando que el fin de este manual es hacer referencia exclusivamente a la utilización del Sistema de Cuentas por Cobrar y tomando en cuenta que el Sistema de Control Escolar contará con su propio manual, se omitirá explicación alguna en esta opción.

Como segunda opción, se ofrece el acceso al Sistema de Cuentas por Cobrar.

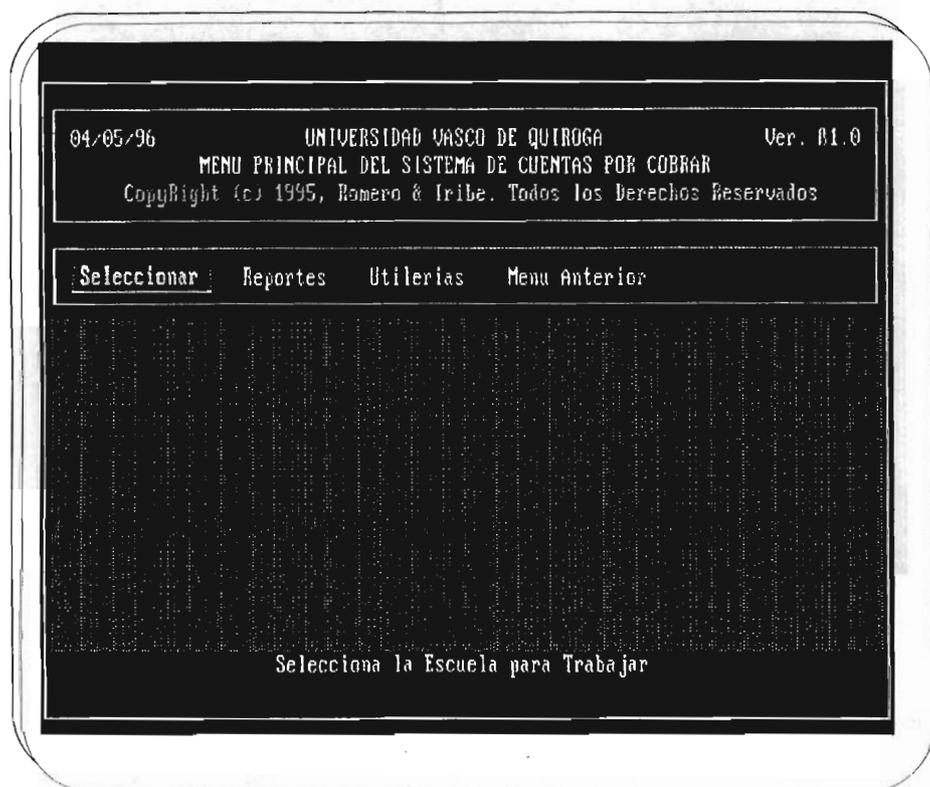
Y por último, se ofrece la opción de Salir al DOS.

Para tener acceso al Sistema de Cuentas por Cobrar o a cualquiera de las opciones que ofrece el sistema, sólo basta con seleccionar la opción deseada moviendo el cursor con las teclas de direccionamiento (**flechas**) y oprimir la tecla **ENTER** o **INTRO**. Al realizar dicha operación se mostrará la siguiente pantalla:

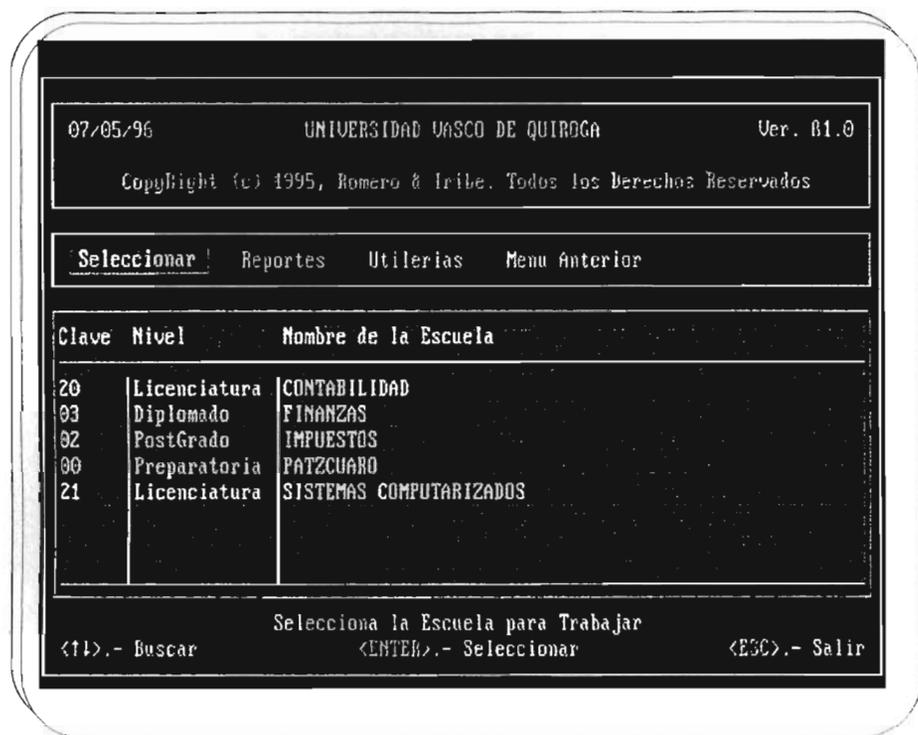
Capítulo 6

Módulo "Seleccionar"

Una vez que el usuario a inicializado el sistema de Cuentas por Cobrar la primera opción que se ofrece es el módulo "Seleccionar" como lo muestra esta pantalla:



El módulo **Seleccionar** ofrece la opción para que el usuario pueda escoger la escuela con la que desea trabajar, se muestra una pantalla en la cual se visualizan las escuelas que están dadas de alta, su clave interna y el nivel al que pertenece dicha escuela.

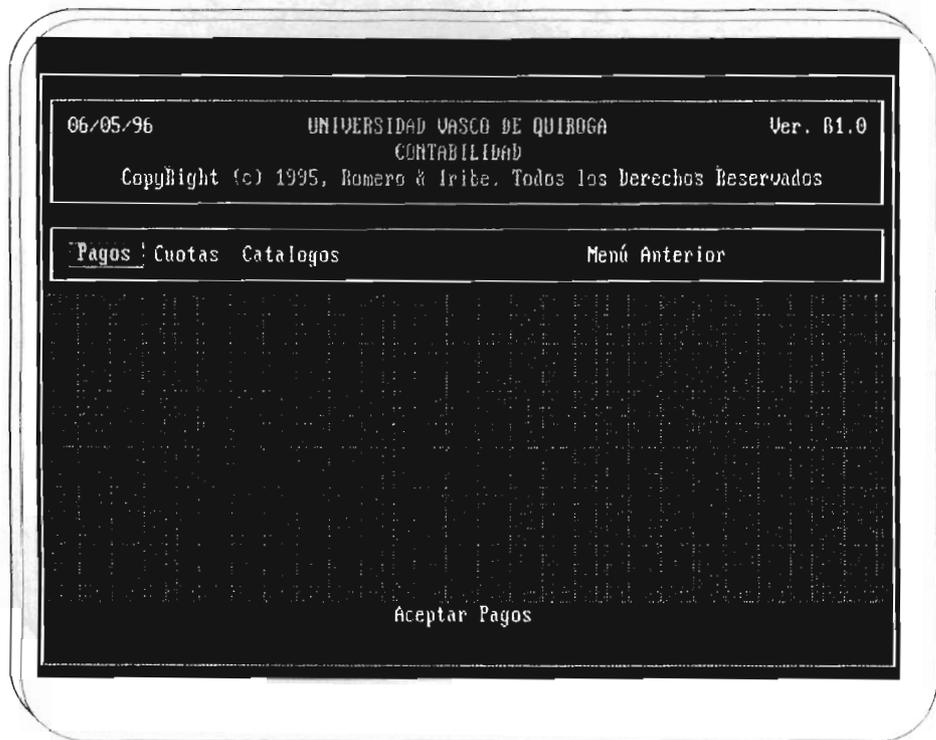


Para llevar a cabo la búsqueda y selección de una escuela, el usuario puede hacer uso de las teclas de direccionamiento (**flechas**) o las teclas **Page Up** y **Page Down**; una vez encontrada la escuela deseada, ubicar el cursor en dicha escuela y oprimir la tecla **ENTER**.

Ya elegida la escuela en la que se valla a realizar alguna operación, el sistema ofrece un menú secundario con cuatro módulos de trabajo (Pagos, Cuotas, Catálogos y Menú Anterior) los cuales se explican en los siguientes capítulos:

Capítulo 7

Módulo "Seleccionar"



Si el usuario desea realizar alguna operación en este módulo, es necesario ubicar el cursor en la opción **Pagos** como lo muestra la pantalla y oprimir la tecla **ENTER**.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Para que el usuario trabaje en la opción de colegiaturas es necesario que ubique el cursor en la opción indicada y oprima la tecla **ENTER**. Después de hecho esto visualizará la siguiente pantalla:

```
07/05/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. R1.0
                  CONTABILIDAD
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Pagos Cuotas Catalogos          Menú Anterior

Grupo      : 10-1          Matricula   :
Alumno     :
Mes a pagar :
Importe    :
Cantidad   :
Recargos   :
Total      :
Forma de pago :
Impresora predeterminada : Epson FX

Acepta el pago de Colegiaturas y/o Inscripción
<F1>.- Nombre  <F2>.- Impresora  <F3>.- Alinear Recibos  <ESC>.- Cancelar
```

El llenado de cada uno de los campos que se presentan en esta pantalla no debe implicar dificultad alguna pues como su nombre lo dice, solicita el tipo de información que tiene que ser vaciada. A continuación se da una breve explicación de cada uno de los campos.

Tanto en el campo “Grupo” como en el campo “Matrícula” el usuario deberá solicitar al alumno su grupo y número de matrícula para que se pueda llevar a cabo la operación.

En caso de que el alumno no cuente con su número de matrícula, para facilitar la operación, el usuario puede proporcionar el grupo y posteriormente oprimir la tecla de función **F1** y el cursor aparecerá en el campo “Alumno” ahí el usuario puede teclear el nombre del alumno empezando por sus apellidos para que se realice la búsqueda.

También es conveniente que el usuario solicite a control escolar una lista de cada uno de los grupos de las diferentes escuelas, en las cuales aparece el nombre del alumno, su número de matrícula y el grupo en el que esta inscrito, de tal forma que pueda hacer una rápida consulta para concluir con el movimiento.

Cubiertos estos dos primeros campos, automáticamente aparecerá el nombre del alumno y una pantalla del lado derecho, la cual le muestra y permite al usuario elegir el mes que se va a cubrir con el pago.

06/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. B1.0
CONTABILIDAD		
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Pagos Cuotas Catalogos		Menú Anterior
Grupo : 10-0	Matricula	Mes
Alumno : CALDERAS MARTINEZ MARIA DEL CARMEN		
Mes a pagar :		Inscripción
Importe :		Marzo
Cantidad :		Abril
Recargos :		Mayo
Total :		Junio
Forma de pago :		Julio
Impresora predeterminada : Epson FX		
Acepta el pago de Colegiaturas y/o Inscripción		
<F1>.- Buscar	<ENTER> - Seleccionar	<ESC>.- Salir

Para seleccionar el mes a pagar es necesario ubicar el cursor en la opción deseada y oprimir la tecla **ENTER**. Al momento de hacerlo, en forma automática se llena el campo "Mes a pagar" y aparece el importe que tiene que pagar el alumno. Conviene en este momento que el usuario pregunte al alumno si cuenta con beca, en caso de que así sea, tendrá que salir de este módulo y a través del módulo de "Cuotas" en la opción "becarios" (capítulo 8) asignarle el porcentaje de beca otorgado.

06/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. 01.0
CONTABILIDAD		
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Pagos Cuotas Catalogos		Menú Anterior
Grupo :	10-U	Matricul
Alumno :	CALDERAS MARTINEZ MARIA DEL CARMEN	
Mes a pagar :	MAYO	
Importe :	\$ 840.00	
Cantidad :	\$ 840.00	
Recargos :	\$ 0.00	
Total :	\$ 840.00	
Forma de pago :		
Impresora predeterminada :	Epson FX	

Forma de Pago
CHEQUE BANCOMER
CHEQUE DE CAJA
EFFECTIVO
TARJETA ATLANTICO
TARJETA BANAMEX
TARJETA BANCOMER

Acepta el pago de Colegiaturas y/o Inscripción

<F1>.- Buscar <ENTER>.- Seleccionar <ESC>.- Salir

La elección de la "forma de pago" se lleva a cabo moviendo el cursor con las teclas de direccionamiento, ubicándolo en la opción deseada y oprimiendo la tecla **ENTER**.

Si el alumno no cuenta con beca, el usuario tiene que preguntar a la persona que va a realizar el pago la cantidad que va a cubrir y dependiendo de lo que le digan, llenar el campo "cantidad" con números y oprimir la tecla **ENTER**.

Si no se le van a cobrar recargos al alumno, el usuario deberá teclear un cero en el campo "recargos", de lo contrario, no permitirá que continúe con la operación, pero si sí se le van a cobrar recargos basta con poner la cantidad y oprimir **ENTER**.

Concluida la operación anterior, automáticamente aparecerá la cantidad a cubrir por el alumno en el campo "total", es decir, la cantidad que se tiene que cobrar y aparecerá otra pantalla del lado derecho que le permitirá al usuario elegir la forma de pago.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Al momento de elegir la forma de pago se llenará dicho campo y por último aparecerá un mensaje en la esquina inferior derecha con la leyenda "Guardar movimiento?", en caso de que la respuesta sea "N" (No) se cancelará la operación y el cursor regresará a la campo matrícula, si la respuesta es "S" (Si) se guardará el movimiento y automáticamente se elaborará el recibo. Como a continuación se presenta:

06/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. 01.0	
CONTABILIDAD			
Copyright (c) 1995, Romero & Iribe Todos los Derechos Reservados			
Pagos	Cuotas	Catalogos	
		Menú Anterior	
Grupo	: 10-U	Matricula	: 9120001
Alumno	: CALDERAS MARTINEZ MARIA DEL CARMEN		
Mes a pagar	: MAYO		
Importe	: \$ 840.00		
Cantidad	: \$ 840.00		
Recargos	: \$ 0.00		
Total	: \$ 840.00		
Forma de pago	: EFECTIVO		
Impresora predeterminada	: Epson FX	Guardar Movimiento	: N
Acepta el pago de Colegiaturas y/o Inscripción			

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Para salir de cualquiera de las opciones que se ofrecen en los diferentes módulos del sistema, el usuario debe oprimir la tecla **ESC** y automáticamente el sistema retrocederá.

Otra de las funciones que se pueden llevar a cabo en la opción de colegiaturas del módulo de pagos en el Sistema de Cuentas por Cobrar es el cambio de impresora con la que se desea trabajar. Para realizar el operación de cambio de impresora, el usuario tiene que oprimir la tecla de función **F2** y se presentará la siguiente pantalla:

06/05/96		UNIVERSIDAD VASCO DE QUIROGA		Ver. B1.0								
		CONTABILIDAD										
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados												
Pagos Cuentas Catalogos			Menú Anterior									
Grupo	:	10-U	<table border="1"><tr><td>IMPRESORA</td></tr><tr><td>HP Laser Portrait</td></tr><tr><td>HP Laser+ Landscape</td></tr><tr><td>HP Laser+ Portrait</td></tr><tr><td>HP LaserJet</td></tr><tr><td>HP LaserJet 500 Plus</td></tr><tr><td>HP LaserJet Plus</td></tr></table>			IMPRESORA	HP Laser Portrait	HP Laser+ Landscape	HP Laser+ Portrait	HP LaserJet	HP LaserJet 500 Plus	HP LaserJet Plus
IMPRESORA												
HP Laser Portrait												
HP Laser+ Landscape												
HP Laser+ Portrait												
HP LaserJet												
HP LaserJet 500 Plus												
HP LaserJet Plus												
Alumno	:											
Mes a pagar	:											
Importe	:											
Cantidad	:											
Recargos	:											
Total	:											
Forma de pago	:											
Impresora predeterminada	:	Epson										
Acepta el pago de Colegiaturas y/o Inscripción												
<F1>.- Nombre <F2>.- Impresora <F3>.- Alinear Recibos <ESC>.- Cancelar												

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Si la respuesta es "N" (No) se imprimirá una pequeña línea de equis en el recibo. Para asegurarse de que el recibo esta correctamente alineado, la línea de las equis xxxxxx deberá estar completamente centrada en la letra "U" donde aparece el nombre de la escuela en el recibo (xxxUniversidad). En caso de que el recibo se encuentre correctamente alineado hay que oprimir la tecla "S" en respuesta de que si esta alineado, y desaparecerá ese mensaje.

Si la impresora no esta lista o simplemente se encuentra apagada, se visualizará una pantalla como la siguiente:

```
06/05/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
                  CONTABILIDAD
          Copyright (c) 1995, Romero & Iribe. Todos los derechos Reservados

Pagos  Cuotas  Catalogos          Menú Anterior

Grupo   : 10-0   La impresora no esta lista   :
Alumno  :
                REINTENTAR  CANCELAR
Mes a pagar :
Importe  :
Cantidad :
Recargos :
Total    :
Forma de pago :
Impresora predeterminada : Epson FX

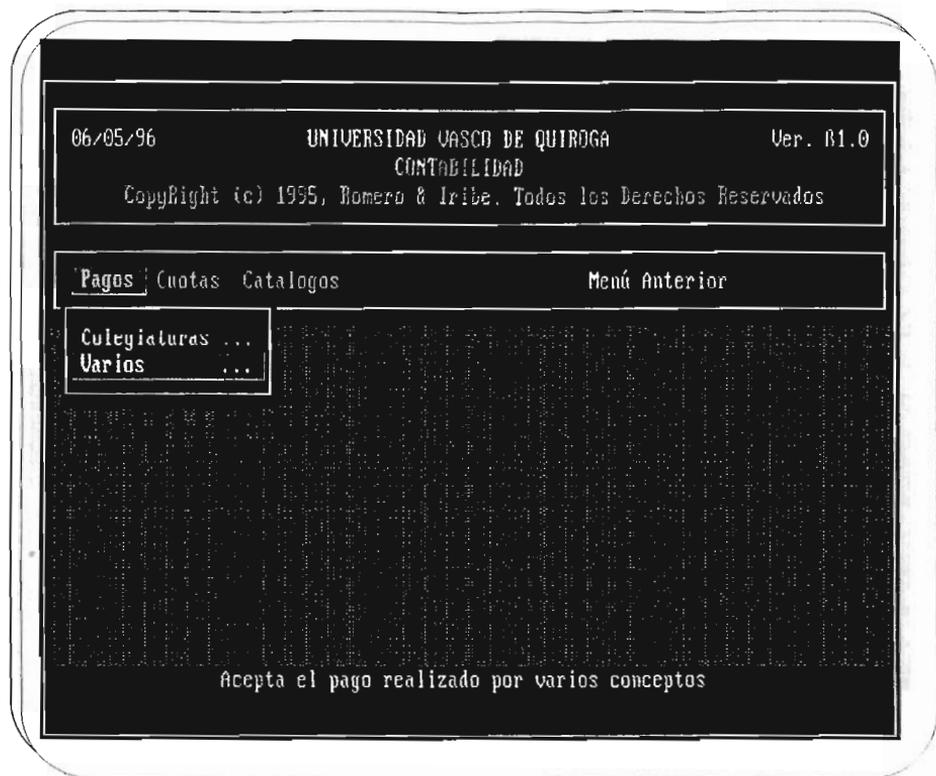
          Acepta el pago de Colegiaturas y/o Inscripción
<F1>.- Nombre  <F2>.- Impresora  <F3>.- Alinear Recibos  <ESC>.- Cancelar
```

Si se pretende dar como respuesta “**REINTENTAR**”, primero es necesario ya sea prender la impresora, alimentarla con los recibos o simplemente ponerla en línea. En caso de que la respuesta sea “**CANCELAR**” sólo basta con posicionar el cursos en dicha opción y oprimir la tecla **ENTER** y desaparecerá el mensaje.

Nota: En la siguientes opción del menú, de este módulo donde se solicite llenar el campo “Forma de pago”, la operación se realiza de la misma manera como se describió anteriormente; tomando en cuenta lo anterior, procederemos a simplificar un poco el llenado y la utilización de las diferentes pantallas que derivan del módulo “Pagos”.

En la siguiente opción “Varios”, que se explicará a continuación, se ofrece de igual manera la opción de cambio de impresora o la alineación de recibos. Tales operaciones se manejan del mismo modo como se explicó en las opción de “Colegiaturas” mencionada en este módulo, por lo que se omitirán dichas explicaciones en lo posterior.

La opción “Varios” del módulo de pagos está destinada a recibir pagos por conceptos varios, excepto colegiaturas e inscripciones. Para seleccionar esta opción, el usuario deberá ubicar el cursor como lo muestra la siguiente pantalla y oprimir **ENTER**.



La siguiente pantalla que visualizará el usuario es muy similar a la mostrada en “Colegiaturas” con la excepción de que aquí no se acepta el pago de colegiaturas y por ende no aparece el recuadro del lado derecho con los meses a cubrir, más bien, aparecerá otro recuadro de opciones, pero con los conceptos de pago.

06/05/96	UNIVERSIDAD VASCO DE QUIROGA CONTABILIDAD	Ver. 01.0
CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Pagos Cuotas Catalogos		Menú Anterior
Grupo	RECIBIR PAGO DE UNA PERSONA EXTERNA A LA UNIVERSIDAD	
Alumn	NO SI	
Conce		
Importe :		
Recargos :		
Total :		
Forma de pago :		
Impresora predeterminada :	Epson FX	
Acepta el pago realizado por varios conceptos		
<F2>.- Cambiar Impresora	<F3>.- Alinear Recibos	<ESC>.- Salir

Si la respuesta en "N" (No) aparecerá un mensaje diciendo "No esta dado de alta el grupo" y habrá que oprimir la tecla **ENTER** para que desaparezca. Si la respuesta es "S" (Si) el mensaje desaparecerá y el cursor se posicionará en el campo "Alumno" para que el usuario pueda teclear el nombre del interesado, y en los demás campos el procedimiento es el mismo.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Si el pago va a ser efectuado por un alumno hay que cubrir los campos como se realizó y explico en la opción de Colegiaturas.

Suponiendo que un alumno acude a caja a pagar una carta de pasante, el usuario deberá solicitarle el grupo en el que esta inscrito y su número de matrícula, seguidamente el sistema proporciona el nombre del alumno y ubica en la parte derecha de la pantalla un recuadro con los posibles conceptos para que el usuario elija el adecuado, como a continuación se muestra:

06/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. B1.0
CONTABILIDAD		
CopyRight (c) 1995, Romero & Irujo. Todos los Derechos Reservados		
Pagos Cuentas Catalogos		Menú Anterior
Grupo : 10-U	Matricula :	
Alumno : CALDERAS MARTINEZ MARIA DEL CARMEN		
Concepto :		Concepto
Importe :		CARTA DE PASANTE
Recargos :		CERTIFICADO PARCIAL
Total :		CONSTANCIA DE ESTUDIOS
Forma de pago :		EXAMEN ADICIONAL
Impresora predeterminada : Epson FX		EXAMEN DE ADMISION
		EXAMEN DE REGULARIZACI
Acepta el pago realizado por varios conceptos		
<F1>.- Buscar	<ENTER>.- Seleccionar	<ESC>.- Salir

SISTEMÁ DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

una vez ubicado el cursor en el concepto deberá oprimir **ENTER** aparecerá en el campo "Concepto" la opción elegida. El siguiente campo a llenar es el del "Importe", ahí el usuario vaciará el costo del concepto, posteriormente un cero en el campo "Recargos" ya que el pago de éstos no los genera , y enseguida aparecerá en el campo "Total" la cantidad total a pagar por el interesado. Como siguiente paso, se mostrará el recuadro que ofrece las diversas formas de pago para que sea elegida por el usuario:

06/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. 01.0
CONTABILIDAD		
CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Pagos Cuotas Catalogos		Menú Anterior
Grupo : IO-U	Matricul	Forma de Pago
Alumno : CALDERAS MARTINEZ MARIA DEL CARMEN		
Concepto : CARTA DE PASANTE		CHEQUE BANCOMER
Importe : \$ 150.00		CHEQUE DE CAJA
Recargos : \$ 0.00		EFFECTIVO
Total : \$ 150.00		TARJETA ATLANTICO
Forma de pago :		TARJETA BANAMEX
Impresora predeterminada : Epson FX		TARJETA BANCOMER
Acepta el pago realizado por varios conceptos		
<↑>.- Buscar	<ENTER>.- Seleccionar	<ESC>.- Salir

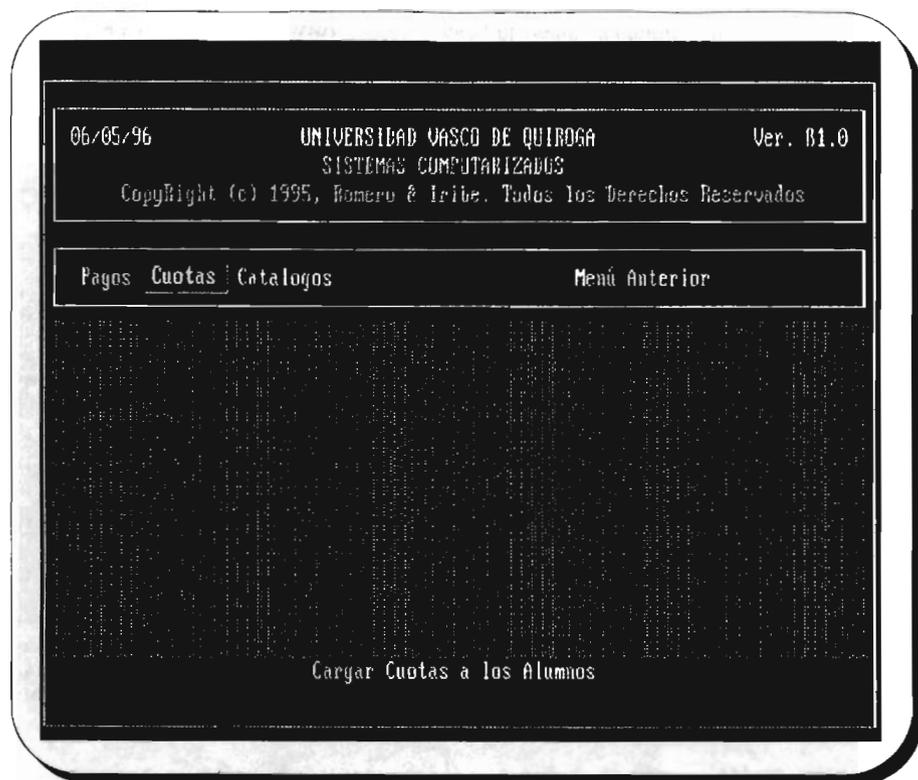
SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Como última acción, después de haber elegido la forma de pago, el sistema solicitará al usuario confirme si se guarda el movimiento. Al responder negativamente "N" el sistema posicionará el cursor en el campo "Matrícula", en caso contrario, y que la respuesta sea "S" se guardará el movimiento y se imprimirá el recibo.

06/05/96	UNIVERSIDAD VASCO DE QUIROGA CONTABILIDAD	Ver. 81.0	
CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados			
Pagos Cuotas Catalogos		Menú Anterior	
Grupo	: IO-U	Matricula	: 9120001
Alumno	: CALDERAS MARTINEZ MARIA DEL CARMEN		
Concepto	: CARTA DE PASANTE		
Importe	: \$	150.00	
Recargos	: \$	0.00	
Total	: \$	150.00	
Forma de pago	: CHEQUE DE CAJA		
Impresora predeterminada	: Epson FX	Guardar Movimiento	: N
Acepta el pago realizado por varios conceptos			

Capítulo 8

Módulo "Cuotas"



El módulo "Cuotas" permite al usuario asignar, como su nombre lo dice, las cuotas a los alumnos en tres formas.

Se puede hacer el cargo de las mismas a una escuela en forma "General", a un alumno en forma "Individual" y como última opción, efectuar la asignación de porcentajes a "Becarios".

En el módulo **Cuotas**, aparecerá una pantalla con tres opciones a elegir, tales como: "General", "Individual" y "Becarios", las cuales se presentan a continuación:



Para realizar la elección de ésta, así como de las siguientes opciones y módulos que brinda el sistema, oprima la tecla **ENTER**; por lo tanto, de ahora en adelante se omitirá este paso para que el usuario se valla familiarizando más con el sistema.

La opción “General” emite una pantalla en la cual el usuario puede realizar la asignación de cuotas en forma general a una escuela. Esta opción se debe llevar a cabo al inicializar el semestre en cada una de las escuelas que se encuentran registradas en el sistema, siendo necesario que control escolar a través del sistema propio haga tal operación.

La pantalla que se aprecia al entrar a esta opción, ofrece dos campos a llenar: “Inscripción” y “Colegiatura” en los cuales, tal como su nombre lo indica, el usuario debe poner el costo a cubrir por los alumnos en ambos conceptos durante el semestre.

06/05/96

UNIVERSIDAD VASCO DE QUIROGA

Ver. 01.0

SISTEMAS COMPUTARIZADOS

Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Pagos Cuotas Catalogos

Menú Anterior

Inscripción : \$ 700.00

Colegiatura : \$ 040.00

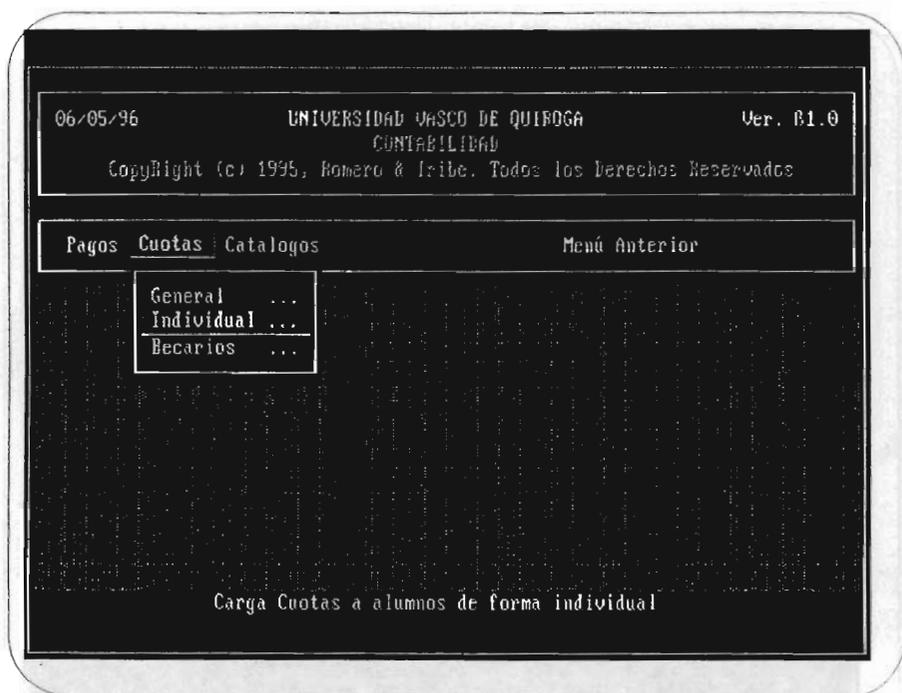
Esta correcta la información : N

Carga Cuotas a todos los alumnos de esta escuela

<ESC>.- Cancelar

Para concluir con esta operación, el sistema emite un mensaje en la parte inferior izquierda para que el usuario confirme si la operación realizada es correcta. El usuario debe oprimir la tecla "S" para guardar el movimiento y por consiguiente se carguen las cuotas en esa escuela. Si el usuario ha cometido un error, puede oprimir la tecla "N" y volver a realizar la operación.

En caso de que el usuario desee asignarle a algún alumno cuotas en forma individual debido a que cuenta con alguna autorización para que su pago de inscripción o de colegiatura sea menor, la segunda opción que ofrece el SCC en este módulo, le permite al usuario realizar este proceso sin dificultad alguna. Es necesario tomar en cuenta que si el alumno paga menos de la cantidad establecida por la universidad por concepto de beca, el usuario deberá realizar tal asignación en la opción "Becarios" y no en esta opción.



Los campos a cubrir en la siguiente pantalla son pocos y de fácil entendimiento para su llenado.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Primero se debe teclear el grupo en el que esta inscrito el alumno, posteriormente hay que proporcionarle al sistema la matrícula del mismo para que se lleve a cabo la búsqueda del alumno en la base de datos, una vez encontrado, aparecerá automáticamente el nombre del alumno y el usuario deberá vaciar la información pertinente en el campo "Inscripción" y "Colegiatura" como se hizo en la opción de asignación "General". El último paso es confirmarle al sistema si esta correcta la información o esta errónea, esto dependerá la respuesta por parte del usuario.

06/05/96

UNIVERSIDAD VASCO DE QUIROGA
CONTABILIDAD

Ver. B1.0

Copyright (c) 1995, Romero & Iribar. Todos los Derechos Reservados

Pagos Cuotas Catalogos

Menú Anterior

Grupo : 10-D

Matricula : 9120001

Alumno : CALDERAS MARTINEZ MARIA DEL CARMEN

Inscripción : \$ 700.00

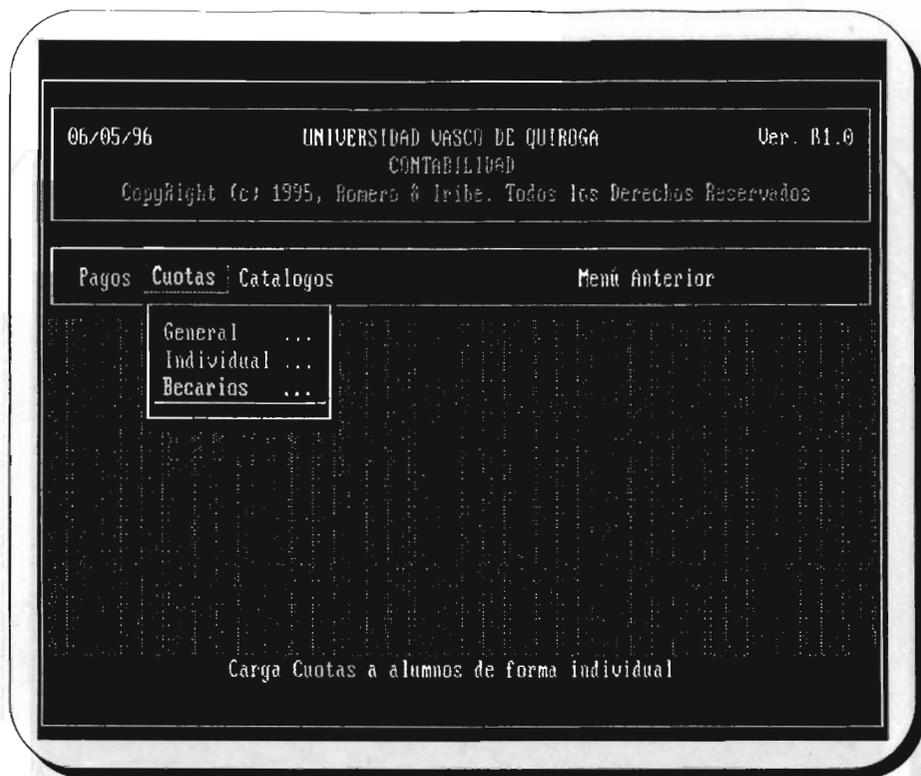
Colegiatura : \$ 700.00

Esta correcta la información : N

Carga Cuotas a alumnos de forma individual

<ESC>.- Cancelar

La tercera y última opción que se ofrece en el módulo pagos, es la asignación de porcentajes a aquellos alumnos que cuentan con beca.



La asignación de porcentajes de becas se debe realizar una vez que el departamento de control escolar ha inicializado el semestre y a llevado a cabo la formación de grupos, para facilitar esta operación se puede solicitar una lista de los alumnos y sus porcentajes de beca que les han otorgado.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Si esto no fuera posible, conforme vayan pasando los alumnos a cubrir sus colegiaturas el usuario puede preguntarles si cuentan con beca, de ser así, puede llevar a cabo esta operación antes de realizar cualquier cobro.

06/05/96	UNIVERSIDAD VASCO DE QUIROGA CONTABILIDAD	Ver. 01.0
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Pagos Cuotas Catalogos		Menú Anterior
Grupo	: 10-U	
Matricula	: 9120001	
Alumno	: CALDERAS MARTINEZ MARIA DEL CARMEN	
Porcentaje de Beca	: 50	
Esta correcta la información : N		
Carga Cuotas a alumnos de forma individual		<ESC>.- Cancelar

Esta operación termina indicándole al sistema que la operación esta correcta, oprimiendo la tecla "S" o "N" si hubiera algún error al momento en que el usuario introduce los datos.

Capítulo 9

Módulo "Catálogos"

06/05/96

UNIVERSIDAD VASCO DE QUIROGA
CONTABILIDAD

Ver. 01.0

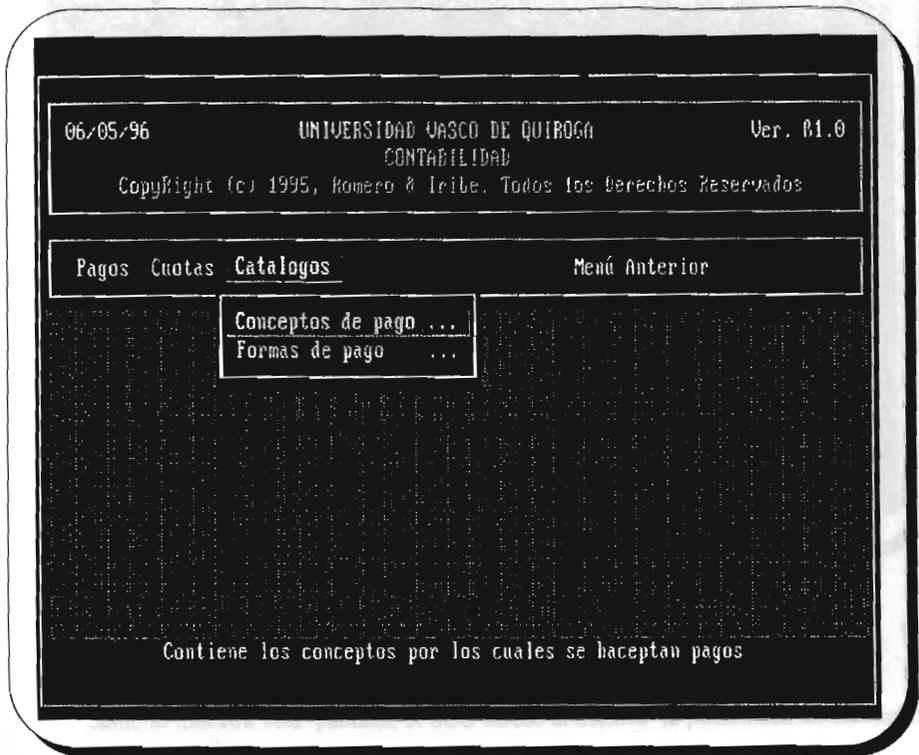
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Pagos Cuotas Catálogos

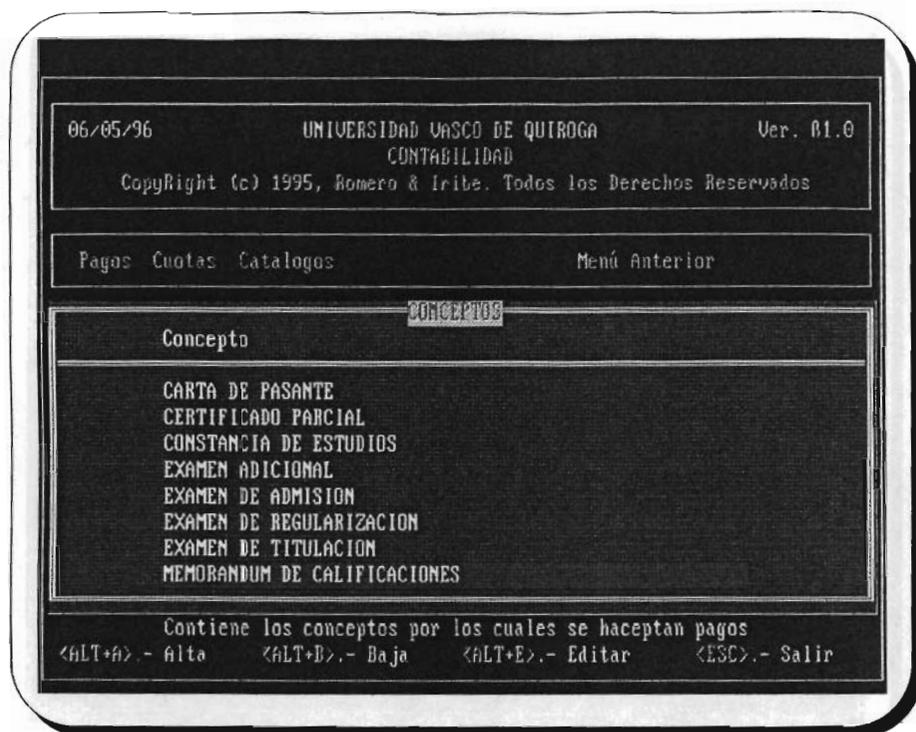
Menú Anterior

Manejo del catalogo de conceptos del sistema

Este módulo permite que el usuario lleve a cabo la adición de un nuevo concepto de pago y una nueva forma de pago, la eliminación y la edición de cualquiera de los ya existentes por medio de las dos opciones que ofrece el menú.



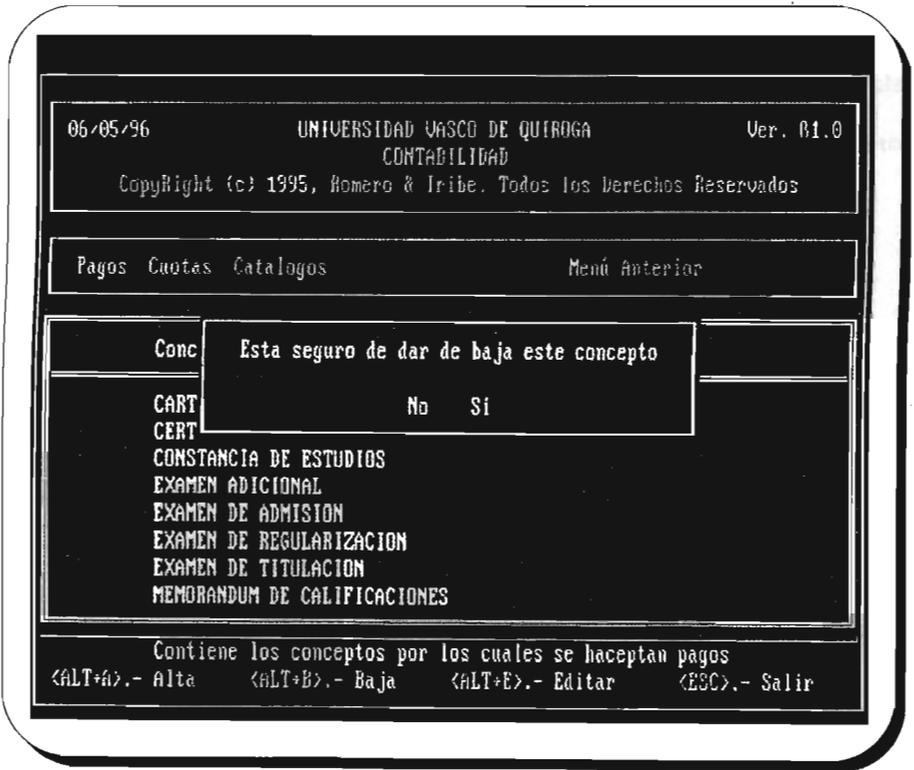
Al haber seleccionado la opción de “Conceptos” aparecerá una pantalla que muestra al usuario los conceptos que se encuentran registrados en el sistema, pudiendo observarlos auxiliándose de las teclas de direccionamiento (**flechas**) o de las teclas **Page Up** y **Page Down**.



Tal como lo muestra esta pantalla, el SCC ofrece al usuario la posibilidad de efectuar tres funciones auxiliares.

- ♦ La primera función le permite al usuario dar de alta un nuevo concepto de pago. Para poder realizar alguna, alta es necesario que el usuario oprima la tecla alterna **ALT** y sin dejar de presionarla oprima la tecla **A**, al hacerlo, observará lo siguiente:

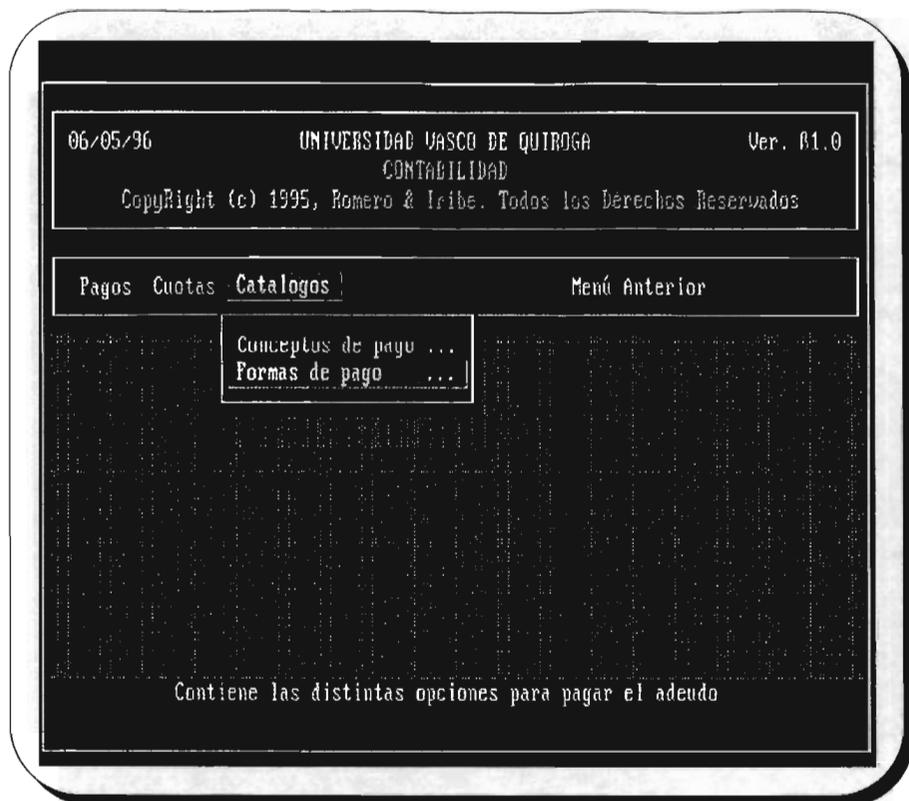
- La segunda función permite que el usuario lleve a cabo la eliminación de algún concepto. Para que se realice la operación el usuario deberá posicionarse en el concepto que desea eliminar. Dicha baja se efectuará cuando el usuario oprima la tecla alterna **ALT** y sin dejar de presionarla oprimir la tecla **B**. Como paso preventivo se agrego el desplegado de un mensaje que le pregunta al usuario si desea realizar la baja. Si se esta seguro de la operación basta con oprimir la tecla "S" o posicionar el cursor en la palabra "SI" auxiliándose de la tecla **flecha a la derecha** ya que por omisión el cursor se posiciona en la letra "NO" para evitar eliminar un concepto no deseado.



- ♦ La última función, permite la edición de alguno de los conceptos registrados ya que posiblemente al dar de alta un nuevo concepto tecleamos mal el nombre, así que, para corregirlo es necesario que se posicione el cursor en el concepto deseado y oprimir la tecla **ALT** y la tecla **E** como se ha hecho en las dos opciones anteriores.

Al hacerlo, aparecerá una pantalla en la cual se edita el concepto, estando ahí podemos realizar la corrección. Al terminar oprima la tecla **ENTER** dos veces o primero **ENTER** y después la tecla **G** para guardar la corrección.

En caso de que se haya equivocado, en vez de presionar la tecla **G** presione la tecla **C** para cancelar la operación. Si desea salir de esta pantalla y cancelar presione la tecla **ESC**.



Al seleccionar la opción “Formas de pago” podrá visualizar en la pantalla las formas de pago que se encuentran registrados en el sistema, con las cuales, los alumnos o las personas que realizan el pago pueden cubrir el adeudo, pudiendo observarlos auxiliándose de las teclas de direccionamiento (flechas) o de las teclas **Page Up** y **Page Down**.

06/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. 01.0
CONTABILIDAD		
CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Pagos Cuotas Catalogos		Menú Anterior
FORMA DE PAGO		
Forma de Pago		
CHEQUE BANCOMER CHEQUE DE CAJA EFECTIVO TARJETA ATLANTICO TARJETA BANAMEX TARJETA BANCOMER TARJETA BITAL TARJETA CONFIA		
Contiene las distintas opciones para pagar el adeudo <ALT+A>.- Alta <ALT+B>.- Baja <ALT+E>.- Editar <ESC>.- Salir		

La pantalla superior ofrece al usuario la posibilidad de efectuar tres funciones auxiliares como las descritas en la opción de "Conceptos de pago".

- La primera función le permite al usuario dar de alta una nueva forma de pago. Para poder realizar alguna alta es necesario que el usuario oprima la tecla alterna **ALT** y sin dejar de presionarla oprima la tecla **A**, al hacerlo, observará lo siguiente:

```
06/05/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. 81.0
                  CONTABILIDAD
                  Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Pagos  Cuotas  Catalogos          Menú Anterior

FORMA DE PAGO
Forma de Pago

CHEQUE BANCOMER
FORMA DE PAGO Agrega

Forma de Pago : TARJETA UVAQ

<G> Guarda o Cancela

TARJETA CONFIA

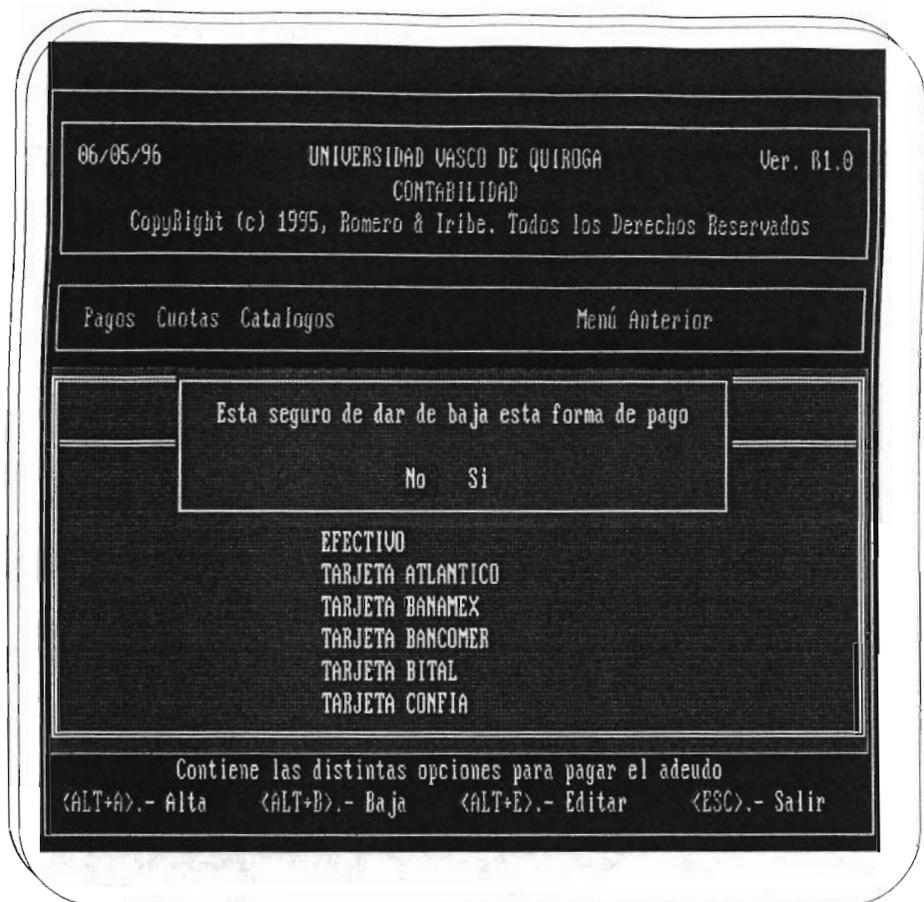
Contiene las distintas opciones para pagar el adeudo
<ALT+A>.- Alta  <ALT+B>.- Baja  <ALT+E>.- Editar  <ESC>.- Salir
```

Para dar de alta una nueva forma de pago, teclee el nombre que le corresponde, posteriormente oprima la tecla **ENTER** y finalmente oprima la tecla **G** o la tecla **ENTER** nuevamente para que se guarde la alta, apareciendo en el menú la opción agregada.

- ♦ La segunda función permite que el usuario lleve a cabo la eliminación de alguna forma de pago. Para que se realice la operación, el usuario deberá posicionar el cursor en la forma de pago que desea eliminar.

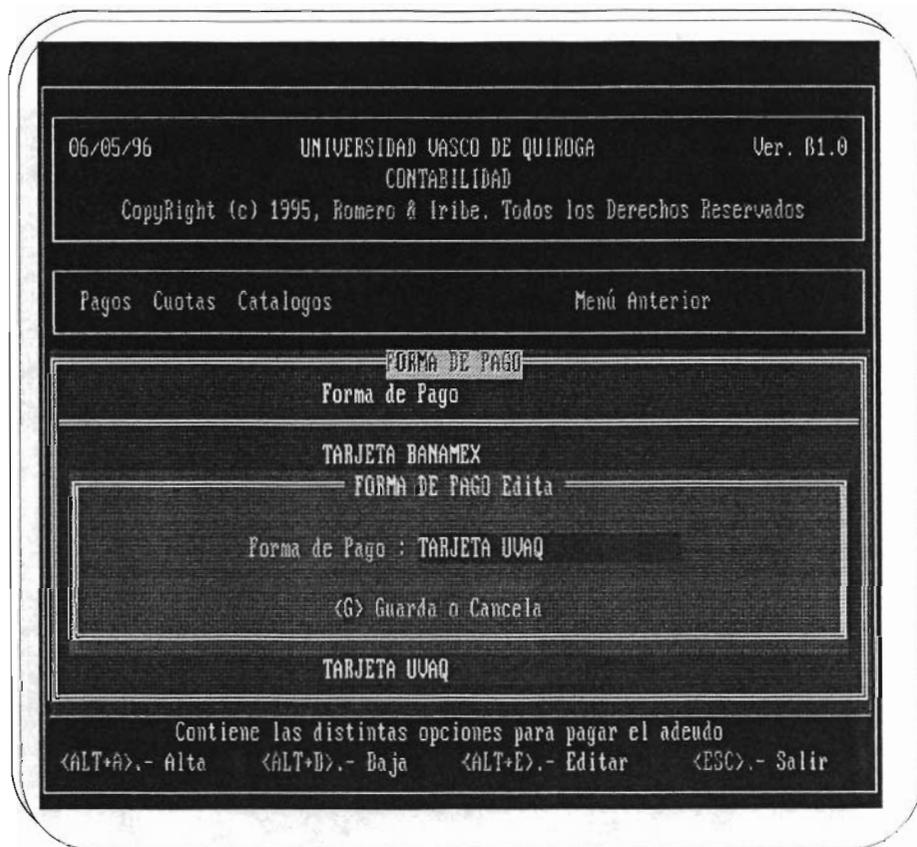
Dicha baja se efectuará cuando el usuario oprima la tecla alterna **ALT** y sin dejar de presionarla oprimir la tecla **B**.

Como paso preventivo se agrego el desplegado de un mensaje que le pregunta al usuario si desea realizar la baja. Si se esta seguro de la operación basta con oprimir la tecla **"S"** o posicionar el cursor en la palabra **"SI"** auxiliándose de la tecla **flecha a la derecha** ya que por omisión el cursor se posiciona en la letra **"NO"** para evitar eliminar una forma de pago no deseada.



- ♦ La última función, permite la edición de alguna de las formas de pago registradas ya que posiblemente al dar de alta una nueva forma tecleamos mal el nombre, así que, para corregirla es necesario que se posicione el cursor en la forma deseada y oprimir la tecla ALT y la tecla E como se ha hecho en las dos opciones anteriores

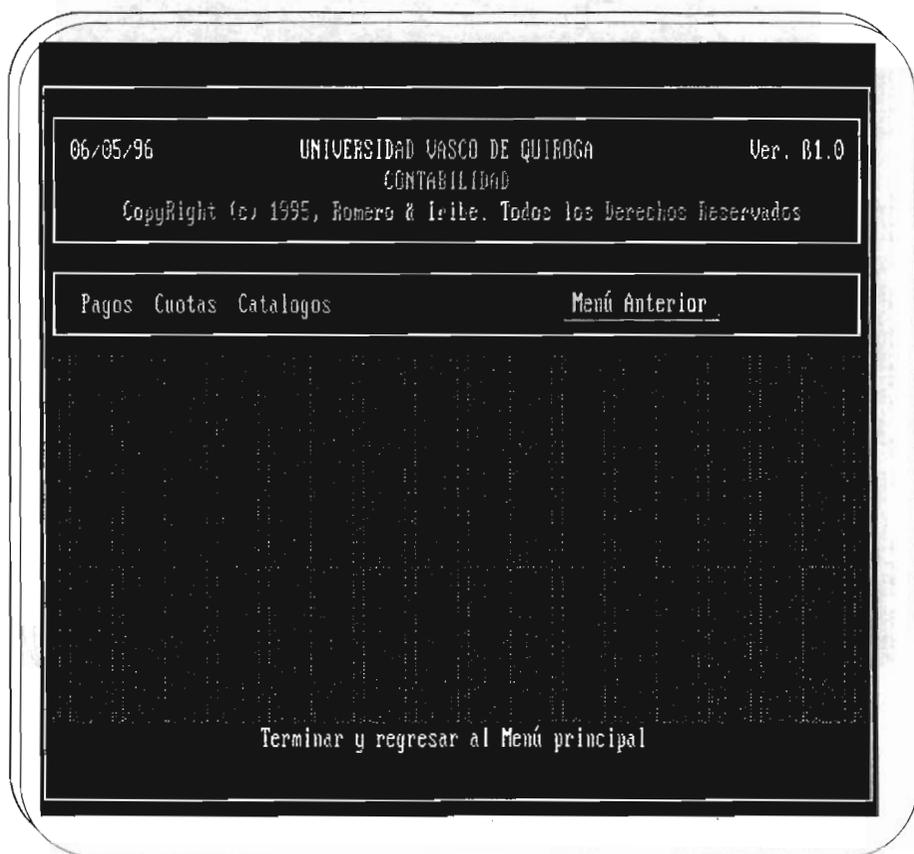
Al hacerlo, aparecerá una pantalla en la cual se edita la forma de pago, estando ahí podemos realizar la corrección. Al terminar oprima la tecla **ENTER** dos veces o primero **ENTER** y después la tecla **G** para guardar la corrección. En caso de que se haya equivocado, en ves de presionar la tecla **G** presione la tecla **C** para cancelar la operación. Si desea salir de esta pantalla y cancelar presione la tecla **ESC**.



Capítulo 10

Módulo "Menú Anterior"

Menú anterior ofrece la opción de retorno al menú principal del Sistema de Cuentas por Cobrar, requiriendo de dos pasos: El primero es seleccionar con el cursor la opción que se presenta en el menú como se muestra a continuación y oprimir la tecla **ENTER**:



Capítulo 11

Módulo "Reportes"

06/05/96

UNIVERSIDAD VASCO DE QUIROGA

Ver. 01.0

MENU PRINCIPAL DEL SISTEMA DE CUENTAS POR COBRAR

Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Seleccionar

Reportes

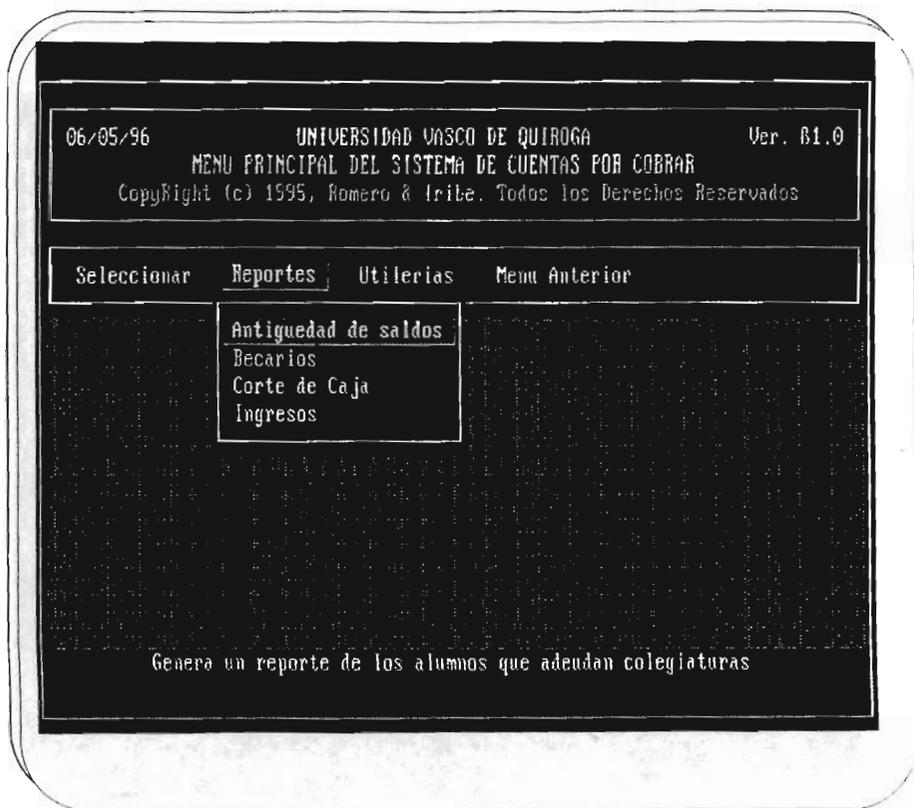
Utilerias

Menu Anterior

Genera los reportes principales del sistema

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Si el módulo elegido del Menú Principal del Sistema de Cuentas por Cobrar fue el de **Reportes**, el sistema muestra una ventana que ofrece cuatro opciones o tipos de reportes que el usuario puede mandar a imprimir, ésto dependiendo de cual haya sido su elección. Dichas opciones se muestran a continuación:



SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Dentro del departamento de contraloría, se hace prescindible la necesidad de contar con un reporte que proporcione los nombres de los alumnos y los meses que por alguna razón vienen arrastrando un saldo moroso con la institución. Por tal motivo se creó la opción de "Antigüedad de Saldos" para que la universidad cuente de manera fácil y rápida con un reporte que le indique tal índice. Para elegir cualquier opción, en este caso "Antigüedad de saldos" oprima la tecla **ENTER** y visualizará la siguiente pantalla:

```
06/05/96          UNIVERSIDAD VASCO DE QUIROGA          Ver. B1.0
MENU PRINCIPAL DEL SISTEMA DE CUENTAS POR COBRAR
Copyright (c) 1995, Romero & Iribe. Todos los derechos Reservados

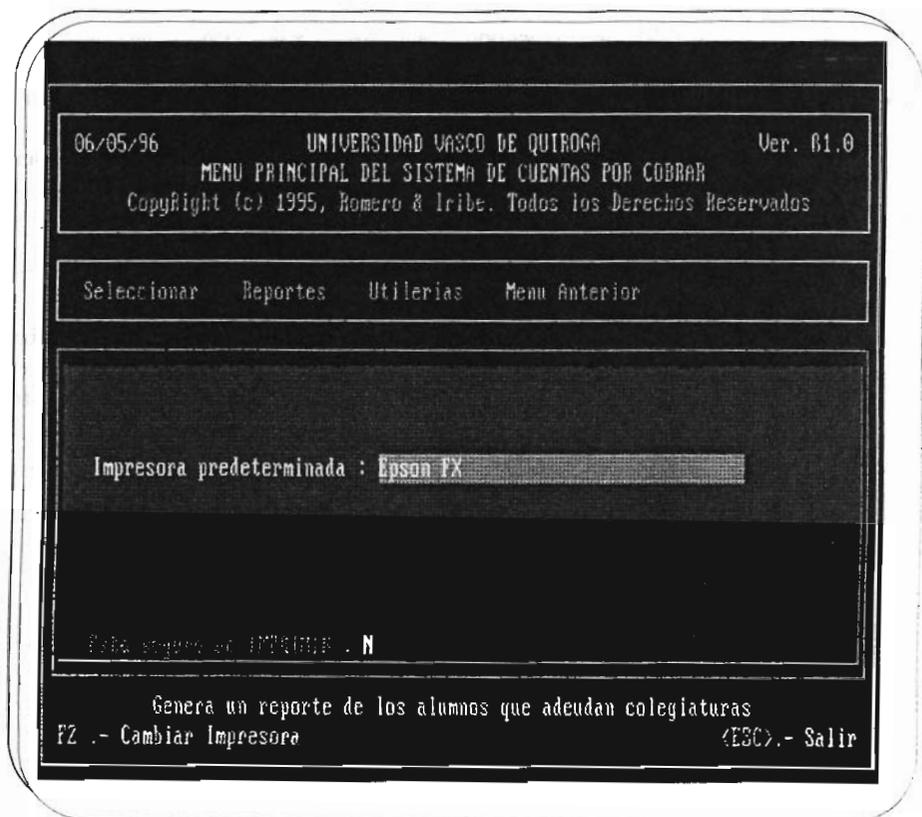
Seleccionar  Reportes  Utilerias  Menu Anterior

Clave Nivel      Nombre de la Escuela
-----
20  Licenciatura  CONTABILIDAD
33  Diplomado     FINANZAS
32  PostGrado     IMPUESTOS
30  Preparatoria  PATZCUARO
21  Licenciatura  SISTEMAS COMPUTARIZADOS

Genera un reporte de los alumnos que adeudan colegiaturas
<F1>.- Buscar          <ENTER>.- Seleccionar          <ESC>.- Salir
```

Para emitir la impresión del reporte de antigüedad de saldos, primero el usuario debe posicionar el cursor en el nombre de la escuela de la cual desea obtener el reporte. Para buscar la escuela se puede auxiliar de las teclas especificadas en la línea de ayuda del sistema y posteriormente oprimiendo la tecla indicada para seleccionarla.

Realizada la operación anterior, se observará la siguiente pantalla:



La impresión del reporte se realizará al momento que el usuario le confirme al sistema que sí esta seguro de imprimir, al hacerlo se imprimirá el reporte correspondiente.

Si por alguna razón se desea cambiar la impresora predeterminada por el sistema, oprima la tecla **F2** y se desplegará un recuadro del lado derecho que le permitirá al usuario elegir la impresora que va a utilizar.

Para cancelar la operación el usuario puede oprimir la tecla **"N"** para negar la impresión o simplemente oprimir la tecla **ESC** y se cancelará automáticamente, regresando al usuario al menú de opciones del módulo de "Reportes".

El segundo tipo de reporte que podemos emitir a través del sistema es el de "Becarios". Dependiendo de la escuela que haya seleccionado el usuario, el sistema proporciona un reporte con los nombres de los alumnos que cuentan con beca y los porcentajes que tienen asignados.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Para seleccionar esta opción, ubique el cursor en "Becarios" como se muestra a continuación oprima la tecla ENTER:



SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Seleccionada la opción en el menú del módulo, el sistema le solicita al usuario seleccione la escuela de la cual desea obtener el reporte, presentando una pantalla como la siguiente:

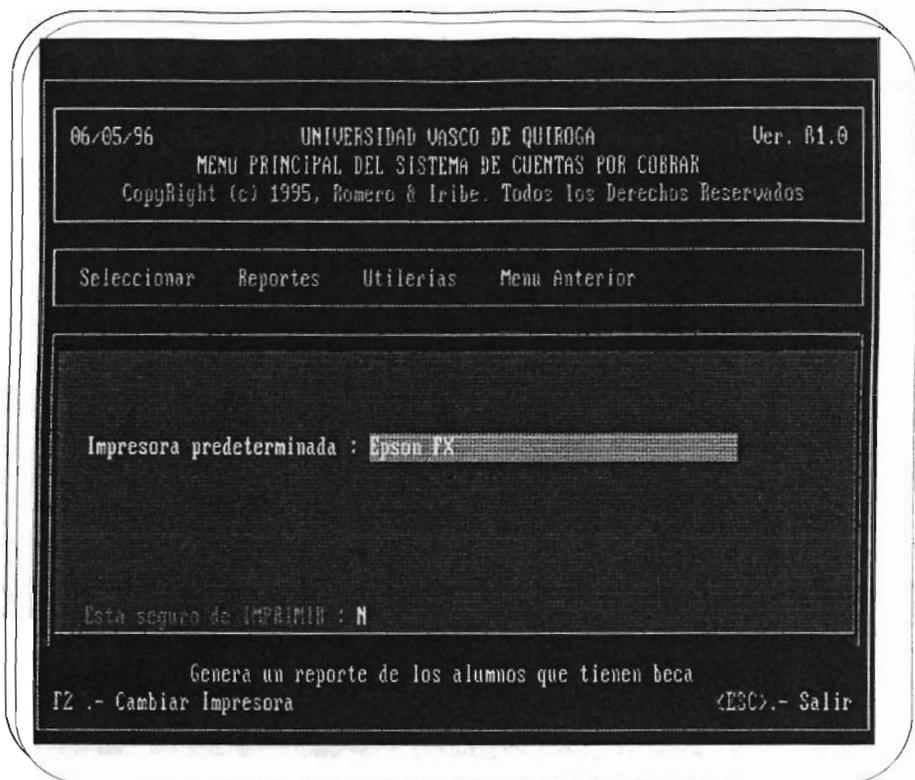
06/05/96 UNIVERSIDAD VASCO DE QUIROGA Ver. B1.0
MENU PRINCIPAL DEL SISTEMA DE CUENTAS POR COBRAR
Copyright (c) 1995, Romero & Irujo. Todos los Derechos Reservados

Seleccionar Reportes Utilerias Menu Anterior

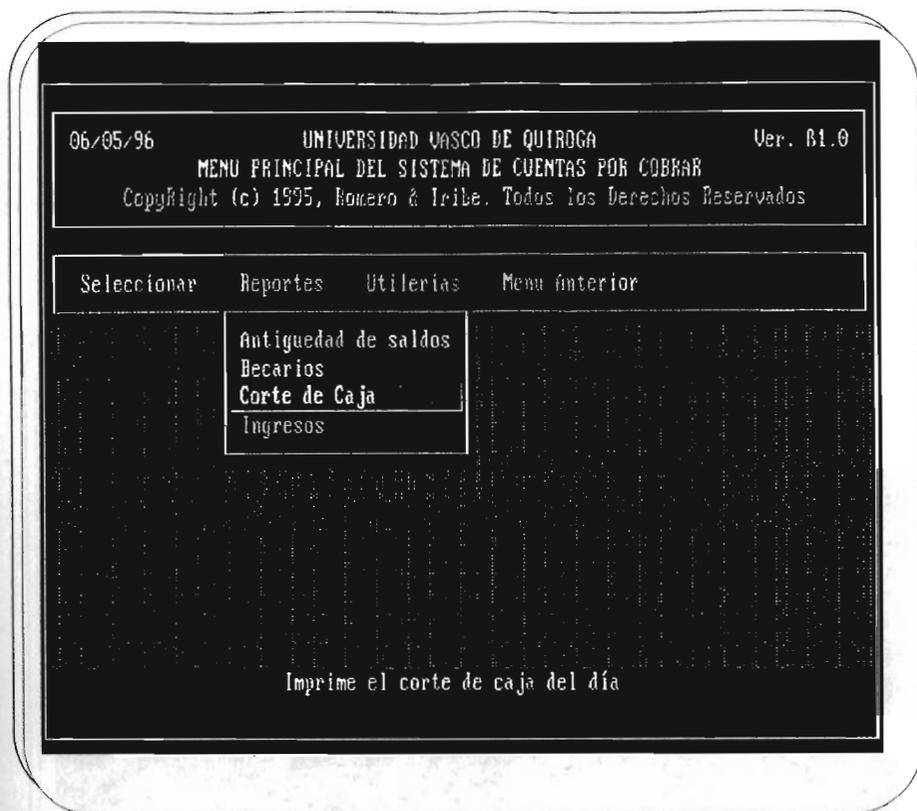
Clave	Nivel	Nombre de la Escuela
20	Licenciatura	CONTABILIDAD
03	Diplomado	FINANZAS
02	PostGrado	IMPUESTOS
00	Preparatoria	PATZCUARO
21	Licenciatura	SISTEMAS COMPUTARIZADOS

Genera un reporte de los alumnos que tienen beca
<F1>.- Buscar <ENTER>.- Seleccionar <ESC>.- Salir

La búsqueda de la escuela se realiza moviendo el cursor con las teclas **flecha arriba** y **flecha abajo** o con las teclas **Page Up** y **Page Down**, ya ubicada, presione **ENTER**. Posteriormente observará la siguiente pantalla.



Al igual que en la opción de "Antigüedad de Saldos", para confirmar la impresión del reporte oprima la tecla "S" y "N" o ESC para cancelar el proceso y salir. Si la impresora con la que se esta trabajando es diferente a la que muestra el sistema, la tecla de función F2 le permite seleccionar la apropiada.

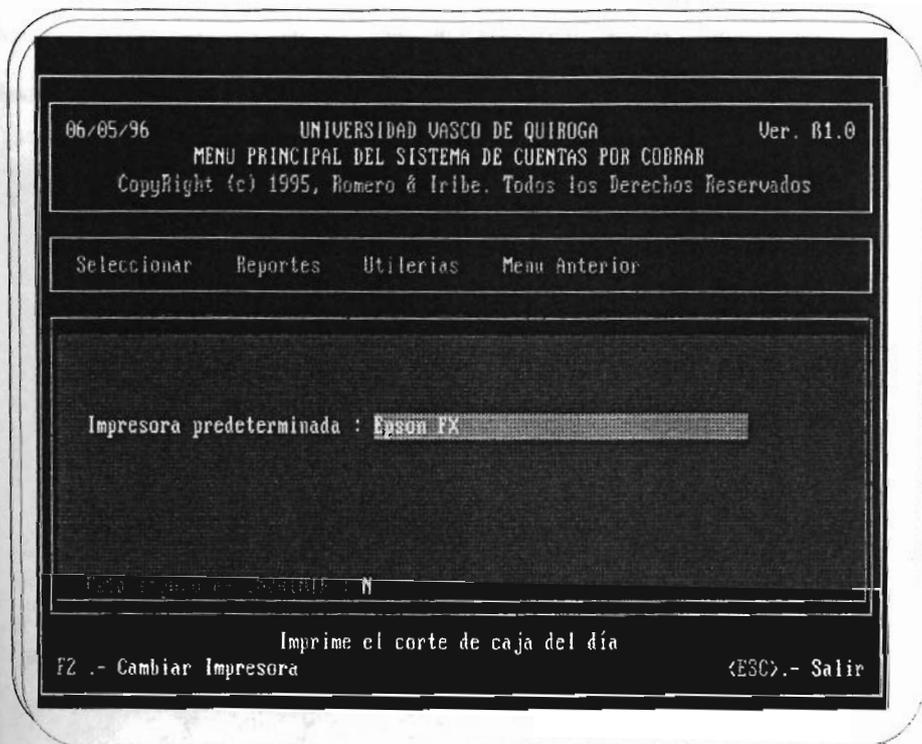


A través de la emisión del reporte de “Corte de Caja”, el usuario puede obtener palpablemente el resultado derivado del manejo diario del sistema de Cuentas por Cobrar. Este reporte debe ser impreso en forma diaria y al final del día, una vez que el usuario ha concluido con sus actividades de aceptación de pagos y va a checar las entradas, realizando en forma paralela la impresión del reporte de “Ingresos”.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

La operación para seleccionar "Corte de Caja" es la misma que se ha descrito en las opciones anteriormente expuestas:

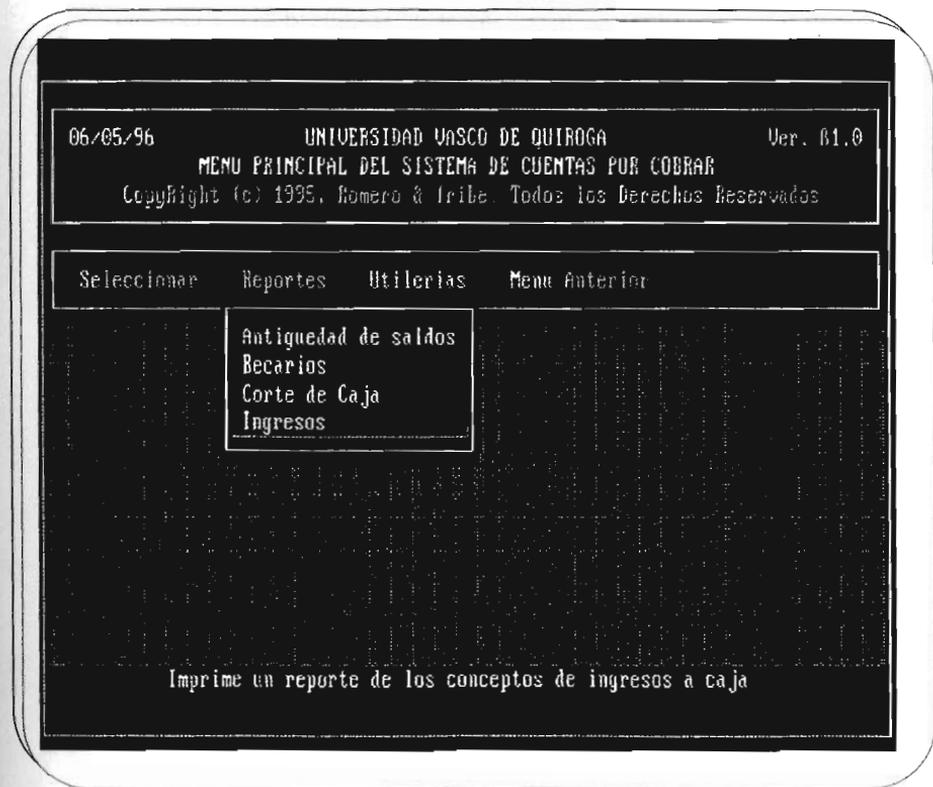
Como el corte de caja se hace en forma general, tomando en cuenta todos los ingresos que se obtuvieron en cada una de las escuelas la pantalla que visualizará el usuario es la siguiente:



SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

El corte de caja presenta al usuario una vez que este ha confirmado la impresión del reporte, los conceptos por los cuales se registraron ingresos en caja, las formas de pago con que fueron cubiertos y el total de los ingresos del día.

Anteriormente se señaló que se recomienda que el corte de caja se debe emitir en forma paralela con los reportes de ingresos para facilitar algunas actividades que desarrolla el usuario diariamente y por tres razones más, las cuales se mencionarán después de mostrar la opción de "Ingresos" en la siguiente pantalla.



La elección se efectúa ubicando el cursor como se ilustra arriba y presionando **ENTER**.

RECOMENDACIÓN: Es necesario que el usuario realice anotaciones del nombre de las escuelas en las cuales se registran ingresos, para que al final del día tenga conocimiento de cuales escuelas debe emitir el reporte de ingresos.

- ♦ La primera razón por la cual se aconseja emitir este tipo de reportes en forma posterior a la emisión del corte de caja, es para que el usuario pueda sumar los totales de cada uno de los reportes de ingresos de cada escuela y pueda comprobar que el total corresponda con el total del corte de caja.
- ♦ Como segunda razón, el usuario poder conocer las formas de pago por las cuales se registraron los ingresos y la cantidad total de cada una de ellas.
- ♦ Por último, este tipo de reportes le permite conocer al usuario los conceptos por los cuales se registraron ingresos y sus cantidades.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

La siguiente visualización corresponde a la pantalla que le permite al usuario seleccionar la escuela de la cual desea emitir el reporte:

06/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. 61.0
MENU PRINCIPAL DEL SISTEMA DE CUENTAS POR COBRAR		
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Seleccionar Reportes Utilerias Menu Anterior		
Clave	Nivel	Nombre de la Escuela
20	Licenciatura	CONTABILIDAD
33	Diplomado	FINANZAS
32	PostGrado	IMPUESTOS
30	Preparatoria	PATZCUARO
21	Licenciatura	SISTEMAS COMPUTARIZADOS
Imprime un reporte de los conceptos de ingresos a caja		
<F1>.- Buscar	<ENTER>.- Seleccionar	<ESC>.- Salir

Una vez seleccionada esta opción, el sistema le muestra al usuario una última pantalla en la cual le proporciona el tipo de impresora que tiene configurada y le solicita la confirmación para llevar a cabo la elaboración de dicho reporte.

06/05/96

UNIVERSIDAD VASCO DE QUIROGA

Ver. R1.0

MENU PRINCIPAL DEL SISTEMA DE CUENTAS POR COBRAR

Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Seleccionar Reportes Utilerias Menu Anterior

Impresora predeterminada : Epson FX

Esta seguro de IMPRIMIR : N

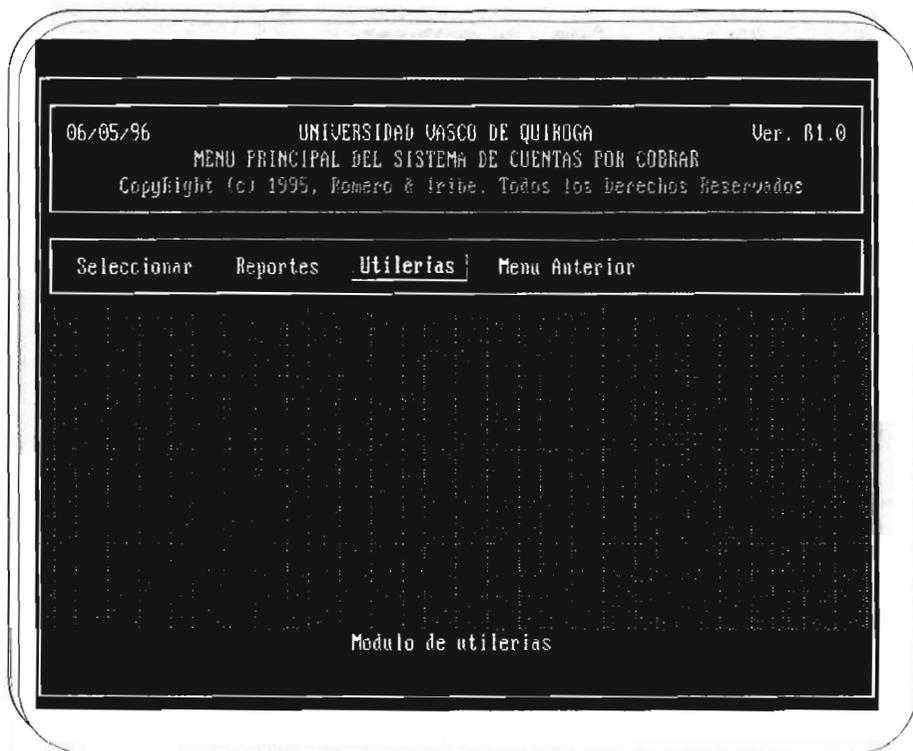
Imprime un reporte de los conceptos de ingresos a caja

F2 .- Cambiar Impresora

<ESC>.- Salir

Capítulo 12

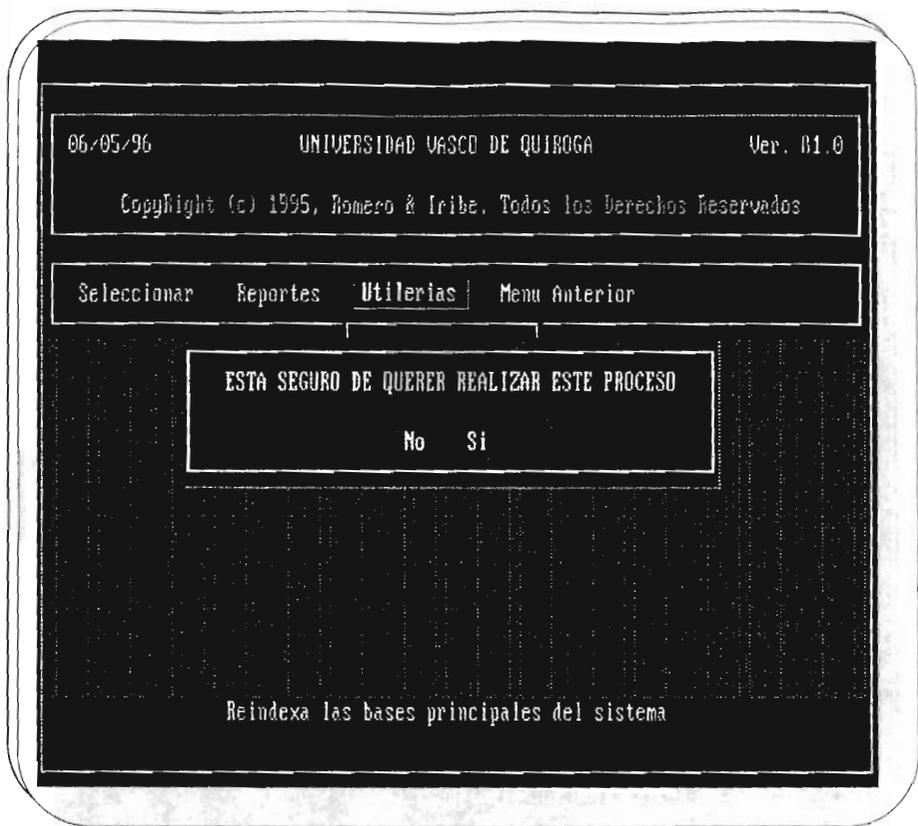
Módulo "Utilerias"



Utilerias es un módulo que ofrece una sola opción, la de "Reindexar Bases", como se muestra a continuación:



Si el usuario solicita se efectúe el proceso correspondiente a la opción, el sistema requiere le sea confirmada la acción al desplegar un mensaje de advertencia.



Al momento de confirmar la acción, el sistema efectúa la reindexación de las principales bases de datos, proporcionando en forma continua el nombre de las bases en las cuales se esta realizando dicho proceso, como se muestra a continuación:

06/05/96

UNIVERSIDAD VASCO DE QUIROGA

Ver. B1.0

CopyRight (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Seleccionar

Reportes

Utilerias

Menu Anterior

ESPERE UN MOMENTO POR FAVOR
SE ESTA REALIZANDO EL PROCESO

Indexando Archivo : ESCAPES.DBF

[29%]

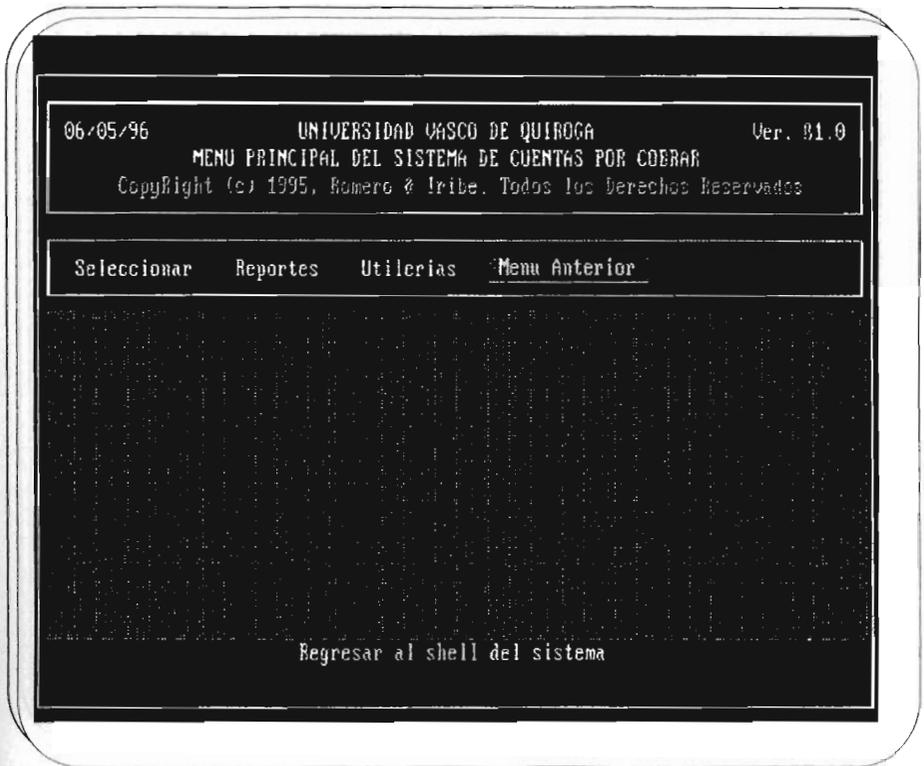


Reindexa las bases principales del sistema

Capítulo 13

Módulo "Menú Anterior"

La última opción que se ofrece en el menú principal del Sistema de Cuentas por Cobrar es el módulo de regreso al **Menú Anterior** o salida al Shell del sistema.



SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Para efectuar esta operación posicione el cursor en la opción que se presenta en el menú principal como se muestra en esta pantalla y presione la tecla **ENTER**.

Como paso seguido, posicione el cursor en la palabra "**SP**" del mensaje emitido por el sistema y oprima la tecla **ENTER** o simplemente oprima la tecla "**S**" para salir al Shell del sistema.



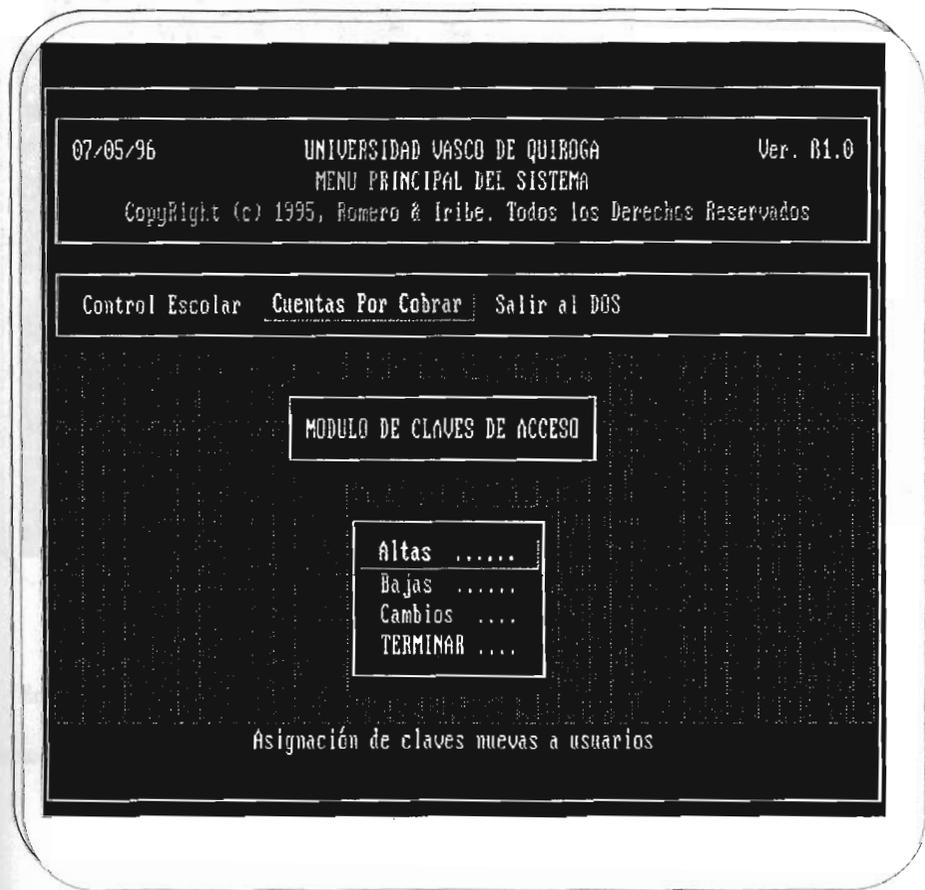
Capítulo 14
Módulo “Claves de Acceso”

Este capítulo esta destinado para ser leído por el administrador del sistema o por el encargado de los diferentes departamentos que integran el área administrativa debido a los siguientes puntos:

- ♦ La persona encargada del Sistema de Cuentas por Cobrar es una persona con un gran sentido de responsabilidad y con un alto significado de ética profesional pues la autoridad que se le confiere requiere de mucho profesionalismo.
- ♦ Por motivos de seguridad del sistema, de la información que contiene y principalmente por seguridad de la propia universidad; ya que la alteración ilícita de las bases de datos contenidas en el sistema provocada por personas ajenas a la institución ocasionaría trastornos en los diferentes procesos internos desarrollados en el áreas administrativa.
- ♦ La información que a continuación se describe detalla las operaciones que tiene que realizar el administrador del sistema para dar de alta un nuevo usuario y otorgarle sus privilegios dentro del mismo, dar de baja un usuario o permitir se efectúen modificaciones en su clave y/o privilegio.

Nota: Cabe mencionar que sólo la persona que esté como administrador del sistema o alguna otra persona asignada bajo responsabilidad del encargado del área administrativa, estará autorizada para hacerlo y contará con la autoridad para poder efectuar esta operación.

Para que el administrador pueda acceder a este módulo, tiene que oprimir la tecla de función F9, una vez presionada visualizará la siguiente pantalla:



El usuario puede seleccionar cualquiera de las opciones del menú navegando en el con ayuda de las teclas de direccionamiento **flecha arriba** y **flecha abajo**. Por omisión el sistema ofrece como primera opción "Altas". Al presionar **ENTER** apreciará la siguiente pantalla:

The screenshot shows a terminal window with a black background and white text. At the top, it displays the date '07/05/96', the university name 'UNIVERSIDAD VASCO DE QUIROGA', and the version 'Ver. 01.0'. Below this is the title 'MENU PRINCIPAL DEL SISTEMA' and a copyright notice: 'Copyright (c) 1995, Romero & Iribe. Todos los derechos Reservados'. A menu bar contains three options: 'Control Escolar', 'Cuentas Por Cobrar' (which is highlighted with a dashed border), and 'Salir al DOS'. Below the menu bar, there are four input fields: 'Clave Nueva', 'Compara Clave', 'Nivel', and 'Impresora'. To the right of the 'Impresora' field is a 'Puerto' field. At the bottom of the screen, it says 'Asignación de claves nuevas a usuarios' and '<ESC>.- Salir'.

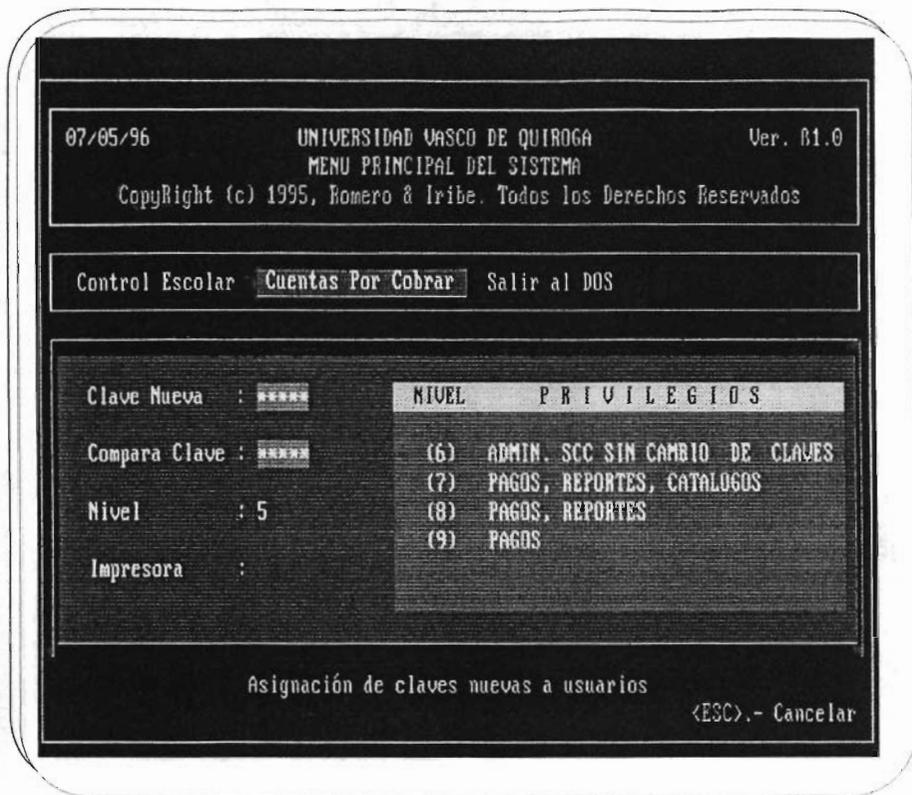
En el primer campo a llenar es necesario que el usuario teclee la clave que va a utilizar para tener acceso al sistema.

Como medida de seguridad del propio usuario, al momento de ir tecleando su clave aparecen asteriscos así, se evita que cualquier persona conozca la clave que ha tecleado

En el segundo campo el nuevo usuario deberá teclear nuevamente la clave para compararla con la anterior y confirmar que es la correcta; si la clave no es la misma aparece en pantalla un mensaje que dice “las clave no son iguales” y lo retorna al primer campo, pero si no hay error, en el lado derecho de la pantalla aparece un recuadro que le muestra al administrador del sistema los diferentes niveles de privilegios que ofrece el sistema, para observar los demás niveles el administrador puede oprimir la tecla de función **F4** y visualizará otro recuadro que le proporciona más privilegios.

El sistema por omisión otorga el nivel 5 para asignar un nivel basta con que el administrador oprima la tecla con el número de nivel que le va a asignar a ese usuario.

A continuación se muestra una pantalla que lo describe:

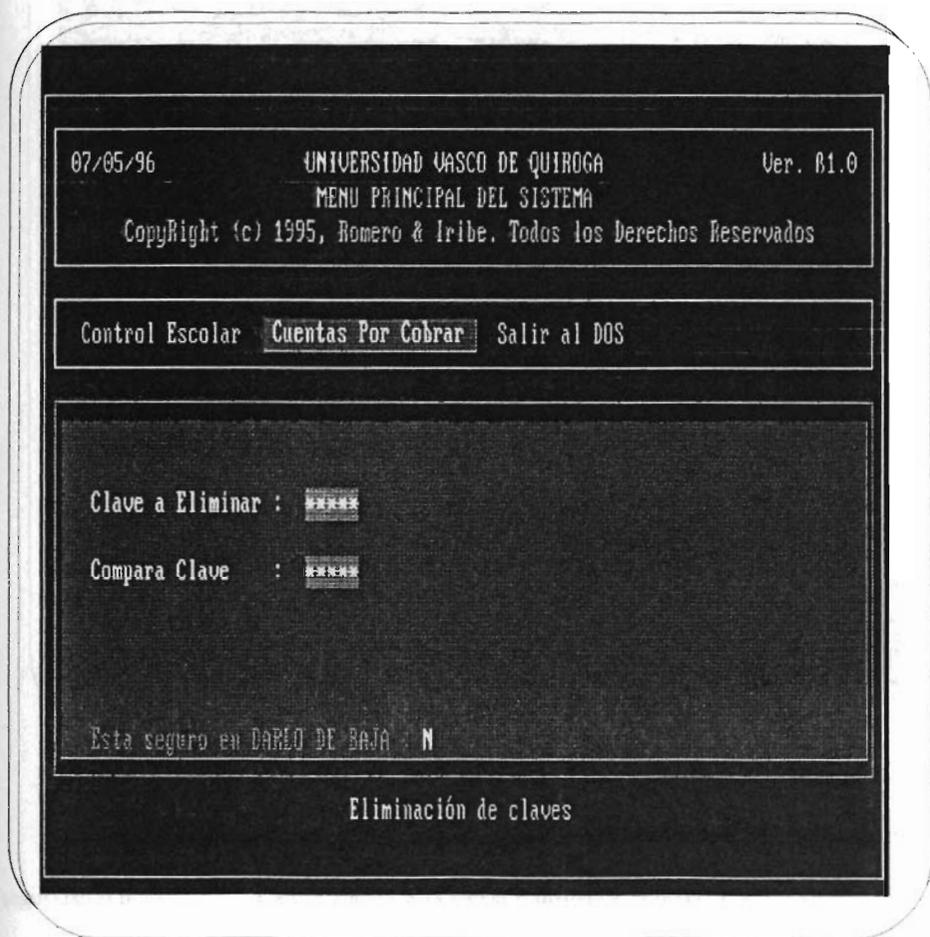


Al momento que se le otorga el nivel de privilegio aparece otro recuadro que le permite al administrador elegir el nombre de la impresora que se le va a destinar a ese usuario para trabajar.



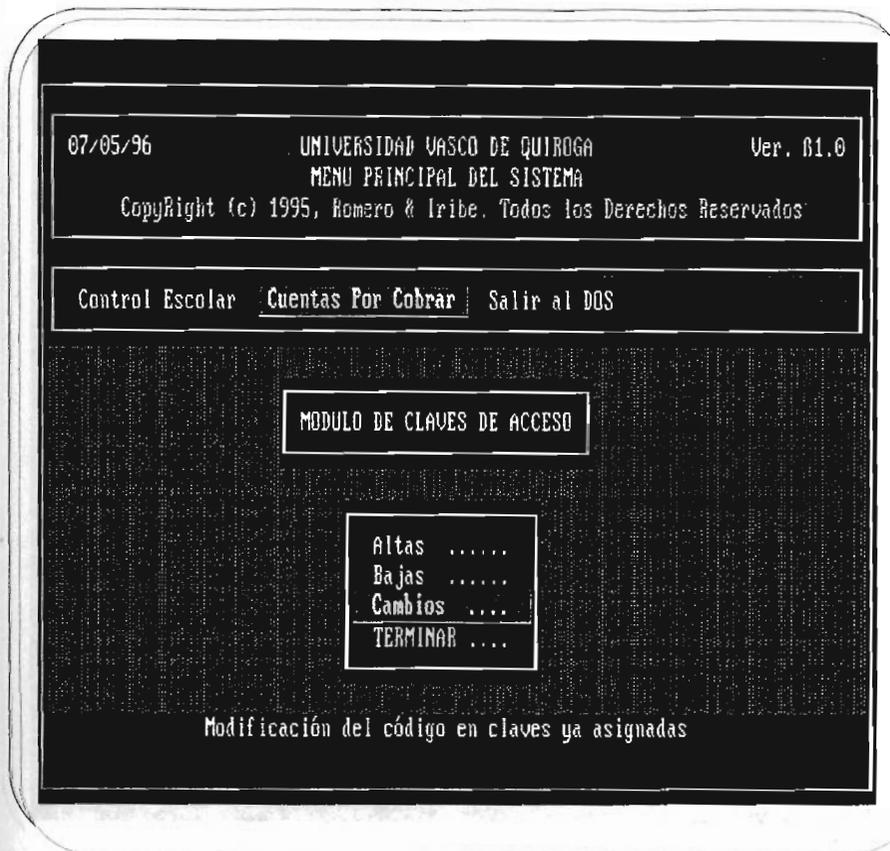
Ya seleccionada la impresora, el cursor se posiciona en el campo "Puerto" que por omisión, presenta LPT1, ahí el administrador deberá oprimir la tecla **ENTER**, por último el sistema pregunta al administrador si son correctos los datos, si así fuera, basta con oprimir la tecla "S" o la tecla "N" o **ESC** para cancelar la operación emitiéndose un mensaje que confirma al administrador del sistema que el usuario no ha sido dado de alta

Presionada la tecla **ENTER** se muestra una pantalla que solicita solo dos campos: la clave a eliminar y la clave a comparar. Si las claves son iguales el sistema le pide confirmación al administrador para realizar la baja, al responder "S" el sistema informa que la clave a sido dada de baja. Si las claves no son concuerdan se informa al administrador que las claves no son iguales y el cursor se posiciona en el campo "Clave a eliminar" para que se vuelva a realizar la operación.



SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

La operación se puede cancelar oprimiendo la tecla "N" o la tecla **ESC** confirmando al administrador que el registro no ha sido dado de baja. Para salir de la opción "Bajas" oprima la tecla de escape **ESC** las veces que sea necesario.



Algunas veces se hace presente la necesidad de realizar el cambio de claves por cuestión de seguridad o por motivos de autorización para elevar o disminuir el nivel de privilegio.

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Si lo que desea es un cambio o modificación, elija la opción con el nombre correspondiente siguiendo los pasos ya conocidos.

07/05/96	UNIVERSIDAD VASCO DE QUIROGA	Ver. B1.0
MENU PRINCIPAL DEL SISTEMA		
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados		
Control Escolar	<u>Cuentas Por Cobrar</u>	Salir al DOS
Clave a Modificar :	Comparar Clave :	
Teclee nueva clave :		
Nivel :		
Impresora :		Puerto :
Modificación del código en claves ya asignadas		
<ESC>.- Salir		

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

Lo primero que se solicita es que el usuario teclee la clave a modificar, posteriormente se tiene que volver a teclear la clave para que el sistema confirme si es la misma, en caso de que así sea el usuario deberá teclear la nueva clave de acceso.

Aquí el usuario no podrá volver a usar la misma clave ya que si lo intenta, se desplegará un mensaje diciéndole al usuario que esa clave ya existe y que introduzca una nueva.

Los campos restantes “Nivel”, “Impresora” y “Puerto” se llenan siguiendo el mismo procedimiento que se siguió en la opción de “Altas”.

El proceso se finaliza respondiéndole afirmativamente “S” al sistema, que si están correctos los datos y como respuesta obtendremos que la clave del usuario ha sido modificada.

07/05/96 UNIVERSIDAD VASCO DE QUIROGA Ver. 01.0
MENU PRINCIPAL DEL SISTEMA
Copyright (c) 1995, Romero & Iribe. Todos los Derechos Reservados

Control Escolar **Cuentas Por Cobrar** Salir al DOS

Clave a Modificar : **** Comparar Clave : ****

Teclee nueva clave : ****

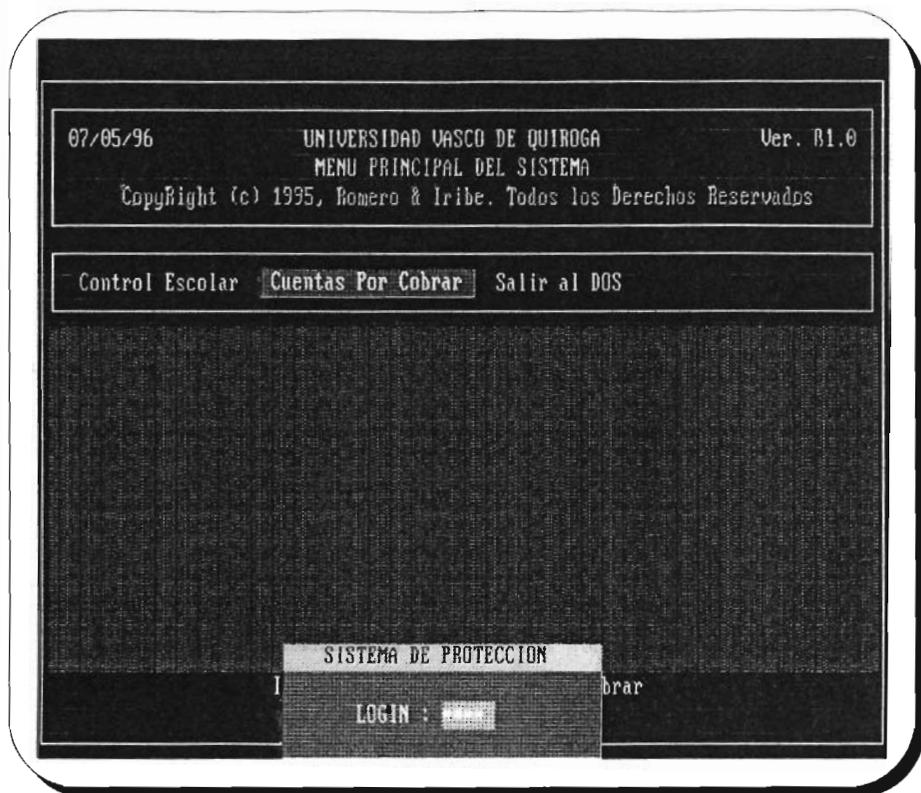
Nivel : 0

Impresora : Epson FX Puerto : LPT1

Son correctos los datos (S/N) : S

Modificación del código en claves ya asignadas
(ESC).- Cancelar

La última opción que proporciona este módulo al administrador del sistema es la opción de "Terminar", la cual se recomienda haga uso de ella siempre que vaya a dejar su estación de trabajo por alguna razón, así evitará que alguna persona ajena se infiltre y ocasione desperfectos en el sistema o altere información.



RECOMENDACIÓN FINAL: El sistema permite claves de cinco dígitos. Especifique al usuario que trate de evitar que la clave sea muy obvia, por ejemplo evitar poner las iniciales de sus nombres, las fechas de nacimiento, etc. El sistema permite que la clave este compuesta tanto por números como por letras.

9.B.- CÓDIGO FUENTE

El presente apéndice tiene como finalidad proporcionar todas las funciones creadas en código fuente para obtener como resultado final el archivo ejecutable del Sistema de Cuentas por Cobrar.

Tales funciones son presentadas conforme están contenidas en los distintos archivos como lo muestra el siguiente índice:

	Página
♦ Archivo SCE.PRG	174
♦ Archivo SCC.PRG	179
♦ Archivo SCE2.PRG	187
♦ Archivo PAGOS.PRG	205
♦ Archivo CUOTAS.PRG	218
♦ Archivo CATALOGO.PRG	231
♦ Archivo CONCEN.PRG	234
♦ Archivo MESPAGO.PRG	243
♦ Archivo UTILERIA.PRG	246
♦ Archivo REPO.PRG	253
♦ Archivo SCE3.PRG	271

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
/* Programa      : SISTEMA DE CUENTAS POR COBRAR UVAQ VER. 1.0
Elaborado por   : L.S.C. ERIC IRIBE RUIZ y L.S.C. RAFAEL ROMERO DIAZ BARRIGA
Funciones       : Sistema de Cuentas por Cobrar y Sistema de Control Escolar
Lenguaje        : CLIPPER 5.2
Enlazador       : RTLINK 3.14B
```

*/

SCE.PRG

```
#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 contendra la opcion deseada en el menú
LOCAL cColores // variable que contendra los colores del entorno DOS
LOCAL aOpcion := {"No","Si"} // opciones confirmación de salida
LOCAL cColor, color, lineamenu, nSalir, colores
PUBLIC nTiempo, nTEspera, 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 ultima línea y centrados
SET DELETE ON // Evita que los registros borrados se visualicen o se consulten
SET WRAP ON // Los menus pueden rotar
SET SCOREBOARD OFF // Desactiva los mensajes de sistema en la primera linea
SET ESCAPE OFF // Evita que el programa pueda ser abortado con Alt-C
SET TALK OFF
SET ECHO OFF // Se desactiva el echo 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 //
* .....//
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
colores := SETCOLOR (C_PANTALLA)
CLS // Se limpia la pantalla
color := SETCOLOR ()
CLS // Se limpia la pantalla
dispbox (7,1,20,maxcol()-1."+-+!+-+!.")
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. B1.0"
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 !scisnet() // Se verifica si se encuentra trabajando en ambiente de red
    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 está misma función termina la ejecución del //
*      sistema //
claves() //
* //
* //
* //
*.....//

TiempoDeEspera (5) // Tiempo de espera para el protector de pantalla
/* nPriv := 0
cPuerto := "LPT1"
NmImp := 102*/
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

/* 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,"+-+!+-+!,C_REYENO2")
DISPBOX (7,1,9,MAXCOL()-1,"+-+!+-+!,C_SOMBRA")
```

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

```
@ 3,3 SAY DATE() PICTURE "@e"
@ 3,68 SAY "Ver. β1.0"
cColor := SETCOLOR ("GR+N")
@ 3,24 SAY " UNIVERSIDAD VASCO DE QUITROGA "
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 válida en el menú. 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_scc ()
    SETCOLOR (colores)
    CLS
    QUIT
ENDIF
```

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

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
DO WHILE .T.
// SET KEY K_F12 TO depurar()
// SET KEY K_F11 TO dos_shell ()
SET KEY K_F9 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 menú principal
    @ 8 ,2 SAY SPACE(76) // Opciones del menú 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)
        //SET KEY K_F9 TO
        TONE (800)
        nSalir := ALERT ("Terminar y salir al DOS", aOpcion)
```

```
IF nSalir == 2
  SETCOLOR (colores)
  CLS
  Mend()    && remove mouse support
  QUIT
ENDIF
//SET KEY K_F9 TO claves
ENDCASE
ENDIF
ENDDO
RETURN NIL
```

SCC.PRG

/* Función principal del programa, se configuran la pantalla, el ambiente. Se dibuja la pantalla y se despliega el menú principal del sistema */

```

FUNCTION main_scc ()
  LOCAL nOp := 1 // variable que contendra la opcion deseada en el menú
  LOCAL cColores // variable que contendra los colores del entorno DOS
  LOCAL aOpcion := {"No","Si"} // opciones confirmación de salida
  LOCAL cColor, color, lineamenu, nSalir, colores, cColor2
// PUBLIC nTiempo, nTEspera, nPriv := NIL, cPucrto, login := SPACE(5), cPassword
  PUBLIC nmesec := SPACE(55), nmimp

SET EXCLUSIVE OFF
SET DECIMALS TO 1
SET MESSAGE TO MAXROW()-2 CENTER // Los mensajes de menú van en la ultima line y centrados
SET DELETE ON // Evita que los registros borrados se visualicen o se consulten
SET WRAP ON // Los menus pueden rotar
SET SCOREBOARD OFF // Desactiva los mensajes de sistema en la primera linea
SET ESCAPE OFF // Evita que el programa pueda se abortado con Alt-C
SET TALK OFF
SET ECHO OFF // Se desactiva el echo de la pantalla
SET STATUS OFF
SET BELL OFF // Se desactiva la campana de captura
SET CURSOR OFF // Se desactiva el cursor parpadeante

```

colores := SETCOLOR (C_PANTALLA)

// Abre y/o crea la base de diario.

```

*****
ok_bases := base_del_dia () //
IF !ok_bases //
  TONE (750) //
  ALERT ("NO SE PUEDE ABRIR LA BASE DEL DIA") //
  RETURN //
ENDIF //
*****

```

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

```

DISPBEGIN ()
  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,"+--+--+!,C_REYENO2")
  DISPBOX (7,1,9,MAXCOL()-1,"+--+--+!,C_SOMBRA")

```

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

```
@ 3,3 SAY DATE() PICTURE "@e"
@ 3,68 SAY "Ver. B1.0"
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 opcion valida en el menú. Para salir del ciclo se presiona la tecla ESC o la opcion "Salir al Dos", tras contestar si a la pregunta de abandonar se regresa al DOS y el sistema termina su ejecución */

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

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

```
DISPEND()
```

```
DO WHILE T.
```

```
SET KEY K_F9 TO claves ()
```

```
DISPBEGIN ()
```

```
cColor2 := SETCOLOR()
```

```
nCentra := (78-LEN ("MENU PRINCIPAL DEL SISTEMA DE CUENTAS POR COBRAR"))/2
```

```
cColor := SETCOLOR ("BG+N")
```

```
@ 4, nCentra SAY "MENU PRINCIPAL DEL SISTEMA DE CUENTAS POR COBRAR"
```

```
SETCOLOR (cColor2)
```

```
@ 22,1 SAY SPACE(78) // Opciones del menú principal
```

```
@ 8,2 SAY SPACE(76) // Opciones del menú principal
```

```
@ 8,3 PROMPT " Seleccionar " MESSAGE "Selecciona la Escuela para Trabajar"
```

```
@ 8,18 PROMPT " Reportes " MESSAGE "Genera los reportes principales del sistema"
```

```
@ 8,30 PROMPT " Utilerias " MESSAGE "Modulo de utilerias"
```

```
@ 8,43 PROMPT " Menu Anterior " MESSAGE "Regresar al shell del sistema"
```

```
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 ("Regresar al Menú Anterior", aOpcion)
```

```
IF nSalir == 2
```

```
RESTSCREEN (4,1,4,80,lineamenu) // Borra la linea que indica de que menu se trata
```

```
SETCOLOR (colores)
```

```
diario->(DBCLOSEAREA()) // Cierra base de diario
```

```
RETURN
```

```
ENDIF
```

```
ENDIF
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
IF nOp != 0 // Desistió

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
  // SET KEY K_F9 TO
  escuela := selecciona(.F.)
  IF escuela > 0
    nomesc := esc_selec := escuela->nomesc
    DBCLOSEAREA() // Se cierran las bases aqui para no perder el numero de registro seleccionado
    SET KEY K_F12 TO // Se apaga la tecla F12
    //SET KEY K_F9 TO claves
    menu_sist (esc_selec,escuela)
  ELSE
    //SET KEY K_F9 TO claves
    DBCLOSEAREA()
  ENDIF
CASE nOp == 2 // Menu de reportes
  menu_sec_sist (4)
CASE nOp == 3 // Se llama la rutina del menú de utilerias
  RESTSCREEN (4,1,4,80,lineamenu) // Borra la linea que indica de que menu se trata
  IF nPriv == 0 .OR. nPriv == 6
    menu_sec_sist (5)
  ELSE
    TONE (750)
    ALERT("NO TIENE PRIVILEGIOS PARA REALIZAR ESTA OPERACION")
  ENDIF
CASE nOp == 4 // ¿ Salir al DOS ?
  //SET KEY K_F9 TO
  TONE (800)
  nSalir := ALERT ("Regresar al Menú Anterior", aOpcion)
  IF nSalir == 2
    RESTSCREEN (4,1,4,80,lineamenu) // Borra la linea que indica de que menu se trata
    SETCOLOR (colores)
    diario->(DBCLOSEAREA()) // Cierra base de diario
    RETURN
  ENDIF
ENDCASE
ENDIF
ENDDO
RETURN

// Función que despliega el menu de trabajo general de las escuelas

STATIC 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
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

// Se cambia al directorio que contine 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")

diario->(DBCLOSEAREA())

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 menú principal

@ 8,2 SAY SPACE(76)

// Borra las opciones del menú anterior

@ 8,3 PROMPT " Pagos "

MESSAGE "Aceptar Pagos"

@ 8,10 PROMPT " Cuotas "

MESSAGE "Cargar Cuotas a los Alumnos"

@ 8,18 PROMPT " Catalogos "

MESSAGE "Manejo del catalogo de conceptos del sistema"

// @ 8,39 PROMPT " Utulerias "

MESSAGE "Utulerias del sistema"

@ 8,50 PROMPT " Menú Anterior "

MESSAGE "Terminar y regresar al Menú 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 Menú Anterior", aOpcion)

IF nSalir == 2

@ 4,2 SAY SPACE(76)

SET PATH TO

RETURN

ENDIF

ENDIF

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
IF nOpcion != 0 // Desistió
DO CASE
CASE nopcion == 1 // Pagos
  menu_sec_sist (1)
CASE nOpcion == 2 // Cuotas
  menu_sec_sist (2)
CASE nopcion == 3 // Catalogo
  menu_sec_sist (3)
* CASE nopcion == 4 // Utilerias
  menu_sec_sist (5)
* CASE nopcion == 4 // Menu anterior
  TONE (700)
  nSalir := ALERT ("Regresar al Menú Anterior", aOpcion)
  IF nSalir == 2
    @ 4,2 SAY SPACE(76)
    SET PATH TO
    RETURN
  ENDF
ENDCASE
ENDIF
ENDDO
RETURN NIL
```

STATIC PROCEDURE menu_sec_sist (nCol)

LOCAL nOpcion := 1, pantalla, nY1, nY2, nX1, nX2, cColor3 // variables locales

cColor3 := SETCOLOR ()

```
IF nCol == 1
  nX1 := 9
  nY1 := 3
  nX2 := 13
  nY2 := 22
ELSEIF nCol == 2
  nX1 := 9
  nY1 := 11
  nX2 := 14
  nY2 := 28
ELSEIF nCol == 3
  nX1 := 9
  nY1 := 18
  nX2 := 13
  nY2 := 42
ELSEIF nCol == 4
  nX1 := 9
  nY1 := 18
  nX2 := 15
  nY2 := 41
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
ELSEIF nCol == 5
    nX1 := 9
    nY1 := 30
    nX2 := 12
    nY2 := 48
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, "+-+|+-+| ")
    @ 22,1 SAY SPACE(78) // Borrar Mensajes del menú anterior
    IF nCol == 1
        @ 10 ,nY1 PROMPT " Colegiaturas ... " MESSAGE "Acepta el pago de Colegiaturas y/o Inscripción"
        @ 11 ,nY1 PROMPT " Varios ... " MESSAGE "Acepta el pago realizado por varios conceptos"
    ELSEIF nCol == 2
        @ 10 ,nY1 PROMPT " General ... " MESSAGE "Carga Cuotas a todos los alumnos de esta escuela"
        @ 11 ,nY1 PROMPT " Individual ... " MESSAGE "Carga Cuotas a alumnos de forma individual"
        @ 12 ,nY1 PROMPT " Becarios ... " MESSAGE "Carga Cuotas a alumnos de forma individual"
    ELSEIF nCol == 3
        @ 10 ,nY1 PROMPT " Conceptos de pago ... " MESSAGE "Contiene los conceptos por los cuales se hacen pagos"
        @ 11 ,nY1 PROMPT " Formas de pago ... " MESSAGE "Contiene las distintas opciones para pagar el adeudo"
    ELSEIF nCol == 4
        @ 10 ,nY1 PROMPT " Antigüedad de saldos " MESSAGE "Genera un reporte de los alumnos que adeudan colegiaturas"
        @ 11 ,nY1 PROMPT " Becarios " MESSAGE "Genera un reporte de los alumnos que tienen beca"
        @ 12 ,nY1 PROMPT " Corte de Caja " MESSAGE "Imprime el corte de caja del día"
        @ 13 ,nY1 PROMPT " Ingresos " MESSAGE "Imprime un reporte de los conceptos de ingresos a caja"
    ELSEIF nCol == 5
        @ 10 ,nY1 PROMPT " Reindexar Bases " MESSAGE "Reindexa las bases principales del sistema"
    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)
    SETCOLOR (cColor3)
    RETURN
ENDIF
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
IF nOpcion != 0
DO CASE
CASE nopcion == 1
IF nCol == 1 // Pago de colegiaturas y/o inscripción
pago_colegiatura ()
ELSEIF nCol ==2 // Cargo de cuotas generales
cuota_general ()
ELSEIF nCol ==3
conceptos (.T.) // Catalogo de conceptos
ELSEIF nCol == 5
reindexa (2)
ENDIF
CASE nOpcion == 2
IF nCol == 1
pago_varios ()
ELSEIF nCol ==2 // Cargo de cuotas individuales
cuota_individual ()
ELSEIF nCol ==3 // Catalogo de formas de pago
conceptos (.F.)
ELSEIF nCol ==4
repo_becarios () // Reporte de Becarios
ENDIF
CASE nopcion == 3
IF nCol == 1
ELSEIF nCol ==2 // Captura de becarios
becarios ()
ELSEIF nCol ==3
ELSEIF nCol == 4 // Corte de Caja
corte_caja ()
ENDIF
CASE nopcion == 4
IF nCol == 4 // Reporte de Ingresos
ingresos ()
ENDIF
ENDCASE
ENDIF
ENDDO
RETURN NIL
```

```
FUNCTION menu_apaga (tipo)
LOCAL cColor := SETCOLOR ("W/N")
DISPBEGIN ()
IF tipo
@ 8,3 SAY " Pagos "
@ 8,10 SAY " Cuotas "
@ 8,18 SAY " Catalogos "
* @ 8,29 SAY " Reportes "
* @ 8,39 SAY " Utilerias "
@ 8,50 SAY " Mcnú Anterior "
```

ELSE

@ 8,3 SAY " Seleccionar "

@ 8,18 SAY " Reportes "

@ 8,30 SAY " Utilerias "

@ 8,43 SAY " Menu Anterior "

ENDIF

SETCOLOR (cColor)

DISPEND ()

RETURN NIL

STATIC FUNCTION base_del_dia ()

LOCAL aBase := {}, cNomArch,cNomArch2,cNomArch3, IEdo

cNomArch := "D"+ ALLTRIM(STR(MONTH(DATE()))) + SUBSTR(ALLTRIM(STR(YEAR(DATE()))),3,2)

cNomArch2 := "I"+ ALLTRIM(STR(MONTH(DATE()))) + SUBSTR(ALLTRIM(STR(YEAR(DATE()))),3,2)

cNomArch3 := "C"+ ALLTRIM(STR(MONTH(DATE()))) + SUBSTR(ALLTRIM(STR(YEAR(DATE()))),3,2)

IF !FILE (cNomArch+".DBF")

AADD(aBase,{"Clave", "C", 5,0})

AADD(aBase,{"Escuela", "C",55,0})

AADD(aBase,{"Fecha", "D", 8,0})

AADD(aBase,{"Hora", "C", 5,0})

AADD(aBase,{"Matrerialum", "C", 7,0})

AADD(aBase,{"Concepto", "C",50,0})

AADD(aBase,{"formapago", "C",25,0})

AADD(aBase,{"Monto", "N", 7,2})

DBCREATE(cNomArch,aBase)

OPEN FILE &cNomArch TIME 30 EXCLUSIVE ALIAS diario

INDEX ON hora TO &cNomArch

INDEX ON escuela+concepto TO &cNomArch2

INDEX ON concepto TO &cNomArch3

diario->(DBCLOSEAREA())

ENDIF

OPEN FILE &cNomArch TIME 30 SHARED INDEX &cNomArch,&cNomArch2,&cNomArch3 ALIAS diario

IF NETERR ()

TONE (800)

ALERT ("ARCHIVO OCUPADO, ESPERE UN MOMENTO Y VUELVA A INTENTAR")

IEdo := .F.

ELSE

IEdo := .T.

ENDIF

RETURN IEdo

SCE2.PRG

// Apaga las opciones de los distintos menus

PROCEDURE limpiamenu (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 " Menú 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,limpiamenu(1),limpiamenu (3)) // Se apagan las opciones del menú

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 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)
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

```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
RESTSCREEN (10,1,MAXROW()-3,MAXCOL()-1,pantalla)
SETCOLOR (cColor)
RETURN NIL
```

```
PROCEDURE pantallacampos (tipo_campos)
```

```
LOCAL cColorLocal
```

```
DISPBEGIN ()
```

```
IF tipo_campos < 7
```

```
DISPBOX (10,1,maxrow()-3,maxcol()-1,"+-+!+-+!_",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 "N° 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 : "
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, {"FeInsAlum", "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,{"rfcmae","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,{"feingmae","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,{"matrrialum","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,{"CALS","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,{"matrrialum","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("edocuent",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 // Mensajes al pie de pantalla
  cLocalColor := SETCOLOR (C_PIE1)
  @ 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 selección"
    @ 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
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

// 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
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
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
```

*..... 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
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
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 habre 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
ENDIF
ENDDO
alumnos->(DBCLOSEAREA()) // Cierra la base de alumnos no indexada y la habre 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)
```

PAGOS.PRG

```

#include "Tipos.Ch"
#include "colores.Ch"
#include "caos.ch"

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

FUNCTION pago_colegiatura ()
  LOCAL nKey, ok_bases, cColor2 := SETCOLOR (),cAlum := SPACE(7), lOk
  LOCAL pantalla, cGrupo, nPago, cConcepto := SPACE (12), nRecargo, nOk
  LOCAL cCampo1 , cCampo2, nImporta, cColor, nBeca, nPagoAnterior, flag := .F.
  LOCAL cFormaPago, cCampo3
  PUBLIC nNumMes := 0
  PRIVATE impresora,nPrnActive, cPuerto, aPuertos, nprn := m->Nmlmp

/* Vars. que indican la posición donde se desplegara el nombre de la impresora si está es cambiada */
PUBLIC x := 20, y := 32

IF !FILE("edocuent.dbf")
  TONE(750)
  ALERT ("AUN NO SE HAN CARGADO LAS CUOTAS A LOS ALUMNOS")
  RETURN
ENDIF

ok_bases := abre_bases (19) // Base de impresoras
IF lok_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 lok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
  impres->(DBCLOSEAREA())
  RETURN
ENDIF

IF !FILE("grupo1.ntx")
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
  impres->(DBCLOSEAREA())
  escapes->(DBCLOSEAREA())
  RETURN
ENDIF

```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

ok_bases := abre_bases (10) // Abre la base de grupos

```
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
  impres->(DBCLOSEAREA())
  escapes->(DBCLOSEAREA())
  RETURN
ENDIF
```

ok_bases := abre_bases (29) // Base de estados de cuenta

```
IF !ok_bases
  TONE (750)
  grupos->(DBCLOSEAREA())
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ESTADOS DE CUENTA")
  impres->(DBCLOSEAREA())
  escapes->(DBCLOSEAREA())
  grupos->(DBCLOSEAREA())
  RETURN
ENDIF
```

IF !FILE("alumno1.ntx")

```
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
  impres->(DBCLOSEAREA())
  escapes->(DBCLOSEAREA())
  grupos->(DBCLOSEAREA())
  RETURN
ENDIF
```

ok_bases := abre_bases (7) // Base de alumnos

```
IF !ok_bases
  grupos->(DBCLOSEAREA())
  edocuent->(DBCLOSEAREA())
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  impres->(DBCLOSEAREA())
  escapes->(DBCLOSEAREA())
  grupos->(DBCLOSEAREA())
  RETURN
ENDIF
```

pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura
menu_apaga (.T.)

@ 9, 2 SAY REPLICATE ("-",45)

SET RELATION TO matrialum INTO edocuent

impresora := CsPrnName2(nPrn) // Inicializa la impresora

impres->(DBCLOSEAREA())

SET KEY K_F3 TO alineat

DO WHILE .T.

pie (17)

DISPBOX (10,1,MAXROW()-3,maxcol()-1,"+-+|+-+|_",C_REYENO) // Se dibuja la ventana de captura
 pantallacampo (22)

```
DO WHILE T
SET KEY K_F2 TO cambia_impresora ()
@ x,y SAY impresora COLOR C_CONFIG4
cGrupo := SPACE(3)
SET CURSOR ON
@ 11, 19 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)
  grupos->(DBCLOSEAREA())
  edocuent->(DBCLOSEAREA())
  alumnos->(DBCLOSEAREA())
  escapes->(DBCLOSEAREA())
  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
```

```

DO WHILE .T.
  cAlum := SPACE(7)
  flag := .F.
  DISPBEGIN ()
  cColor := SETCOLOR (C_BORRADO)
  @ 11, 59 SAY REPLICATE ("_",10)
  @ 12, 19 SAY REPLICATE ("_",50)
  @ 14, 19 SAY REPLICATE ("_",50)
  @ 15, 19 SAY REPLICATE ("_",50)
  @ 16, 19 SAY REPLICATE ("_",15)
  @ 17, 19 SAY REPLICATE ("_",15)
  @ 18, 19 SAY REPLICATE ("_",15)
  @ 19, 21 SAY REPLICATE ("_",50)
  @ 20, 50 SAY REPLICATE ("_",28)
  SETCOLOR (cColor)
  pic(19)
DISPEND()
SET CURSOR ON
SET KEY K_F1 TO cAlum := busca_nombre()

@ 11, 59 GET cAlum PICTURE "@!" VALID (!EMPTY(cAlum))
READ

SET KEY K_F1 TO
SET CURSOR OFF

IF LASTKEY () == K_ESC
  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)
      @ 12, 19 GET cNomAlum PICTURE "@!" // Despliega el nombre del alumno
      CLEAR GETS
    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)
    @ 11, 59 SAY REPLICATE ("_",10)
    SETCOLOR (cColor)
    @ 11, 59 SAY "No existe"
    INKEY (1)
    @ 11, 59 SAY " " "
    LOOP
  ENDIF
ENDIF

```

```

ENDIF
ENDIF
nPago := 0.0
nRecargo := -1
SET CURSOR ON
REPEAT
  nOk := 1
  cConcepto := catalogos(1)
  IF cConcepto = NIL
    TONE (750)
    nOk := ALERT ("Cancelar la operacion", {"NO", "SI"})
    IF nOk = 2
      EXIT
    ENDIF
  ENDIF
ENDIF
UNTIL cConcepto <> NIL
IF nOk = 2
  LOOP
ENDIF
pie(1)
IIF (nNumMes <= 7, nNumMes --, nNumMes -= 8)
nBeca := edocuent->beca / 100
IF nNumMes > 0
  cCampo1 := "SALDO"+ALLTRIM(STR(nNumMes))
  cCampo2 := "PAGADO"+ALLTRIM(STR(nNumMes))

  IF edocuent->&cCampo1. # 0 .AND. nBeca == 0
    nImporte := edocuent->(&cCampo1.)
    flag := .T.
  ELSEIF edocuent->&cCampo1. >= 0 .AND. !edocuent->&cCampo2.
    cCampo3 := "CARGO"+ALLTRIM(STR(nNumMes))

    *..... Se checa si se realizo ya un pago parcial
    IF edocuent->&cCampo3. # edocuent->&cCampo1.
      nImporte := edocuent->(&cCampo1.)
    ELSE
      *..... si no se han realizado pagos parciales...
      IF nBeca # 0
        nImporte := ((edocuent->(&cCampo1.))* (1-nBeca))
      ELSE
        nImporte := edocuent->(&cCampo1.)
      ENDIF
    ENDIF
  ELSE
    nImporte := 0.0
  ENDIF
ELSE
  nImporte := edocuent->(inscripcio)
ENDIF

```

```

IF nBeca == 1
    TONE (800)
    ALERT ("ESTE ALUMNO TIENE BECA DEL 100%, ESTA EXCENTO DE PAGO")
    LOOP
ENDIF
IF nImporte <= 0.0
    TONE (800)
    ALERT ("YA ESTA PAGADO TOTALMENTE EL MES SELECCIONADO")
    LOOP
ENDIF
DISPBEGIN ()
    IF edocuent->beca > 0 .AND. nNumMes > 0
        @ 14, 42 SAY "Porcentaje de Beca : "
        @ 14, 63 GET edocuent->beca
    ENDIF
    cColor := SETCOLOR (C_IMPORTE)
    @ 15, 19 SAY nImporte PICTURE "$ #,###.##"
    SETCOLOR (cColor)
    @ 14, 19 GET cConcepto PICTURE "@!"
    CLEAR GETS
DISPEND ()
@ 16, 19 GET nPago PICTURE "$ #,###.##" VALID !EMPTY(nPago)
@ 17, 19 GET nRecargo PICTURE "$ #,###.##" VALID nRecargo >= 0.0
READ
IF LASTKEY () == K_ESC
    LOOP
ENDIF
cColor := SETCOLOR ("GR+R")
nPagoReal := nPago+nRecargo
@ 18, 19 SAY nPagoReal PICTURE "$ #,###.##"
SETCOLOR (cColor)
REPEAT
    nOk := 1
    cFormaPago := catalogos(2)
    IF cFormaPago == NIL
        TONE (750)
        nOk := ALERT ("Cancelar la operacion",{ "NO", "SI" })
        IF nOk == 2
            EXIT
        ENDIF
    ENDIF
UNTIL cFormaPago <> NIL
IF nOk == 2
    LOOP
ENDIF
@ 19, 21 GET cFormaPago PICTURE "@!"
CLEAR GETS
IOk := "N"
@ 20, 52 SAY "Guardar Movimiento : "
@ 20, 73 GET IOk PICTURE "!A" VALID (IOk$"SN")
READ

```

```

IF !ok == "S"
  IF edocuent->(lockr (30))
    IF nNumMes > 0
      cCampo1 := "SALDO"+ALLTRIM(STR(nNumMes))
      cCampo2 := "CARGO"+ALLTRIM(STR(nNumMes))
      IF flag
        REPLACE edocuent->&cCampo1. WITH edocuent->&cCampo1. - nPago
      ELSE
        REPLACE edocuent->&cCampo1. WITH nImporte - nPago
      ENDF
      IF nImporte - nPago <= 0
        cCampo2 := "PAGADO"+ALLTRIM(STR(nNumMes))
        REPLACE edocuent->&cCampo2. WITH .T.
      ENDF
    ELSE
      REPLACE edocuent->inscripcion WITH edocuent->inscripcion - nPago
    ENDF
    edocuent->(DBCOMMIT())
    diario->(DBAPPEND())
    REPLACE diario->clave      WITH login      ;;
      diario->escuela         WITH nomesc      ;;
      diario->fecha           WITH DATE()      ;;
      diario->hora            WITH TIME()      ;;
      diario->matrerialum     WITH cAlum      ;;
      diario->formapago       WITH cFormaPago,;
      diario->monto           WITH nPagoReal
    IF nNumMes > 0
      cConcepto := "Colegiatura mes de " + cConcepto
      REPLACE diario->concepto WITH cConcepto
    ELSE
      REPLACE diario->concepto WITH cConcepto
    ENDF
    diario ->(DBCOMMIT())
    SET KEY K_F2 TO; SET KEY K_F3 TO
    ok_bases := abre_bases (19) // Base de impresoras
    IF !ok_bases
      TONE (750)
      ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
      LOOP
    ENDF
    recibo (cNomAlum,cAlum,nPago,nRecargo,cConcepto) // Imprime el recibo
    impres->(DBCLOSEAREA())
  ELSE
    ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
  ENDF
ENDIF
ENDDO
ENDDO
RETURN NIL

```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
FUNCTION pago_varios ()
LOCAL nKey, ok_bases, cColor2 := SETCOLOR (),cAlum := SPACE(7), lOk
LOCAL pantalla, cGrupo, nPago, cConcepto := SPACE (50), nRecargo, nOk
LOCAL cFormaPago, nPagoReal, lNoAlumno := .F., cNomAlum := SPACE(35)
LOCAL flag := .F.
PRIVATE impresora,nPrnActive, cPuerto, aPuertos, nprn := m->Nmlmp

/* Vars. que indican la posición donde se desplegara el nombre de la impresora si está 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")
    impres->(DBCLOSEAREA())
    RETURN
ENDIF

IF !FILE("grupo1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
    impres->(DBCLOSEAREA())
    escapes->(DBCLOSEAREA())
    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")
    impres->(DBCLOSEAREA())
    escapes->(DBCLOSEAREA())
    RETURN
ENDIF

IF !FILE("alumno1.ntx")
    TONE (700)
    ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
    impres->(DBCLOSEAREA())
    escapes->(DBCLOSEAREA())
    grupos->(DBCLOSEAREA())
    RETURN
ENDIF
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  grupos->(DBCLOSEAREA())
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  impres->(DBCLOSEAREA())
  escapes->(DBCLOSEAREA())
  grupos->(DBCLOSEAREA())
  RETURN
ENDIF

pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

menu_apaga (.T.)
@ 9, 2 SAY REPLICATE ("-",45)
impresora := CsPrnName2(nPrn) // Inicializa la impresora
impres->(DBCLOSEAREA())
SET KEY K_F3 TO aline()
DO WHILE .T.
  pie (17)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1,"+-+!+-+!_",C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (23)
  DO WHILE .T.
    SET KEY K_F2 TO cambia_impresora ()
    @ x,y SAY impresora COLOR C_CONFIG4
    cGrupo := SPACE(3)
    SET CURSOR ON
    @ 11, 19 GET cGrupo PICTURE "@R@! X9-A"
    READ
    SET CURSOR OFF

  IF LASTKEY () = K_ESC
    RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)
    SETCOLOR (cColor2)
    @ MAXROW()-1,1 SAY SPACE (78)
    grupos->(DBCLOSEAREA())
    alumnos->(DBCLOSEAREA())
    escapes->(DBCLOSEAREA())
    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
    ENDIF
  ENDIF
ENDWHILE
```

```

ELSE
  TONE (750)
  IF ALERT ("RECIBIR PAGO DE UNA PERSONA EXTERNA A LA UNIVERSIDAD",{ "NO","SI"}) == 2
    flag := .T.
    SET CURSOR ON
    @ 12, 19 GET cNomAlum PICTURE "@!" // Despliega el nombre del alumno
    READ

    SET CURSOR OFF
    IF LASTKEY () == K_ESC
      cColor := SETCOLOR (C_BORRADO)
      @ 12, 19 SAY REPLICATE ("_",50)
      SETCOLOR (cColor)
      LOOP
    ELSE
      !NoAlumno := .T.
      EXIT
    ENDIF
  ENDIF
  TONE (800)
  ALERT ("NO ESTA DADO DE ALTA EL GRUPO")
ENDIF
ENDIF
ENDDO

DO WHILE .T.
  // Este bloque sirve para evitar el borrar el nombre de una persona externa a la universidad que desee
  // hacer un pago
  IF !flag
    cAlum := SPACE(7)
    cNomAlum := SPACE(35)
  ELSE
    flag := .F.
  ENDIF

  DISPBEGIN ()
  cColor := SETCOLOR (C_BORRADO)
  @ 11, 59 SAY REPLICATE ("_",10)
  IF !NoAlumno
    @ 12, 19 SAY REPLICATE ("_",50)
  ENDIF
  @ 14, 19 SAY REPLICATE ("_",55)
  @ 15, 19 SAY REPLICATE ("_",15)
  @ 16, 19 SAY REPLICATE ("_",15)
  @ 17, 19 SAY REPLICATE ("_",15)
  @ 18, 19 SAY REPLICATE ("_",15)
  @ 19, 21 SAY REPLICATE ("_",50)
  @ 20, 50 SAY REPLICATE ("_",28)
  SETCOLOR (cColor)
  pic(18)
DISPEND()

```

```

IF !NoAlumno
SET CURSOR ON
@ 11, 59 GET cAlum PICTURE "@"!
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 .AND. alumnos->inscrito .AND. EMPTY(alumnos->baja)
cNomAlum := alumnos->(NomAlum)
@ 12, 19 GET cNomAlum PICTURE "@"! // Despliega el nombre del alumno
CLEAR GETS
ELSE
TONE (750)
ALERT ("ESTE ALUMNO NO ESTA DADO DE ALTA EN ESTE GRUPO O NO HA SIDO REINSCRITO")
LOOP
ENDIF
ELSE
IF alumnos->(FOUND())
TONE (750)
IF ALERT ("RECIBIR PAGO DE UNA PERSONA EXTERNA A LA UNIVERSIDAD",{"NO","SI"}) == 1
TONE (350)
cColor := SETCOLOR (C_BORRADO)
@ 11, 59 SAY REPLICATE ("_",55)
SETCOLOR (cColor)
@ 11, 59 SAY "No existe"
INKEY (1)
@ 11, 59 SAY " "
LOOP
ELSE
SET CURSOR ON
@ 12, 19 GET cNomAlum PICTURE "@"! // Despliega el nombre del alumno
READ
SET CURSOR OFF
IF LASTKEY () == K_ESC
!NoAlumno := .F.
LOOP
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
!NoAlumno := .F.
nPago := 0.0
nRecargo := -1
    
```

```

SET CURSOR ON
REPEAT
  nOk := 1
  cConcepto := catalogos(3)
  IF cConcepto = NIL
    TONE (750)
    nOk := ALERT ("Cancelar la operacion",{ "NO","SI" })
  IF nOk = 2
    EXIT
  ENDIF
ENDIF
UNTIL cConcepto <> NIL
IF nOk = 2 .AND. !EMPTY(cGrupo)
  LOOP
ELSEIF nOk = 2 .AND. EMPTY(cGrupo)
  EXIT
ENDIF
pic(1)
@ 14, 19 GET cConcepto PICTURE "@!"
CLEAR GETS
@ 16, 19 GET nPago PICTURE "$ #,###.##" VALID !EMPTY(nPago)
@ 17, 19 GET nRecargo PICTURE "$ #,###.##" VALID nRecargo >= 0.0
READ
IF LASTKEY () = K_ESC .AND. !EMPTY(cGrupo)
  LOOP
ELSEIF LASTKEY () = K_ESC .AND. EMPTY(cGrupo)
  EXIT
ENDIF
cColor := SETCOLOR ("GR+/R")
nPagoReal := nPago+nRecargo
@ 18, 19 SAY nPagoReal PICTURE "$ #,###.##"
SETCOLOR (cColor)
REPEAT
  nOk := 1
  cFormaPago := catalogos(2)
  IF cFormaPago = NIL
    TONE (750)
    nOk := ALERT ("Cancelar la operacion",{ "NO","SI" })
  IF nOk = 2
    EXIT
  ENDIF
ENDIF
UNTIL cFormaPago <> NIL
IF nOk = 2
  LOOP
ENDIF

```

```

@ 19, 21 GET cFormaPago PICTURE "@!"
CLEAR GETS
IOk := "N"
@ 20, 52 SAY "Guardar Movimiento : "
@ 20, 73 GET IOk PICTURE "!A" VALID (IOk$"SN")
READ
IF IOk == "S"
    diario->(DBAPPEND())
    REPLACE diario->clave      WITH login      ;;
        diario->escuela        WITH nomesc     ;;
        diario->fecha          WITH DATE()         ;;
        diario->hora           WITH TIME()         ;;
        diario->matrIALum      WITH cAlum          ;;
        diario->concepto       WITH cConcepto     ;;
        diario->formapago      WITH cFormaPago    ;;
        diario->monto          WITH nPagoReal
    diario->(DBCOMMIT())
    SET KEY K_F2 TO; SET KEY K_F3 TO
    ok_bases := abre_bases (19) // Base de impresoras
    IF !ok_bases
        TONE (750)
        ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
    LOOP
ENDIF
recibo (cNomAlum,cAlum,nPago,nRecargo,cConcepto) // Imprime el recibo
impres->(DBCLOSEAREA())
ENDIF
ENDDO
ENDDO
RETURN NIL

```

CUOTAS.PRG

```
#include "Tipos.Ch"
#include "InKey.Ch"
#include "colores.Ch"
```

```
FUNCTION cuota_individual ()
```

```
LOCAL nKey, ok_bases, cColor2 := SETCOLOR (),cAlum := SPACE(7)
```

```
LOCAL pantalla, cGrupo, nInscrip, nColeg, cOk, archivo
```

```
IF !FILE("edocuent.dbf")
```

```
  aBase := {}
```

```
  AADD(aBase,{"matrrialum","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(CPATH+"edocuent",aBase)
```

```
ok_bases := abre_bases (30)
```

```
// Abre la base estados de cuenta
```

```
IF !ok_bases
```

```
  TONE (750)
```

```
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ESTADOS DE CUENTA")
```

```
  RETURN
```

```
ENDIF
```

```
archivo := CPATH + "\edocuent.ntx"
```

```
INDEX ON matrrialum TO &archivo
```

```
edocuent->(DBCLOSEAREA())
```

```
ENDIF
```

```
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)
  grupos->(DBCLOSEAREA())
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  RETURN
ENDIF
```

```
ok_bases := abre_bases (29) // Base de estados de cuenta
IF !ok_bases
  TONE (750)
  grupos->(DBCLOSEAREA())
  alumnos->(DBCLOSEAREA())
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ESTADOS DE CUENTA")
  RETURN
ENDIF
IF !FILE("edocuent.ntx")
  INDEX ON matrialum TO edocuent
ENDIF
```

```
pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura
menu_apaga (.T.)
@ 9, 5 SAY REPLICATE ("-",45)
```

DO WHILE .T.

pic (2)

DISPBOX (10,1,MAXROW()-3,maxcol()-1,"+-+!+-+!_",C_REYENO) // Se dibuja la ventana de captura
pantallacampos (20)

DO WHILE .T.

cGrupo := SPACE(3)

SET CURSOR ON

@ 11, 19 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)

grupos->(DBCLOSEAREA())

alumnos->(DBCLOSEAREA())

edocuent->(DBCLOSEAREA())

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

DO WHILE .T.

cAlum := SPACE(7)

cColor := SETCOLOR (C_BORRADO)

@ 12, 19 SAY REPLICATE ("_",10)

@ 13, 19 SAY REPLICATE ("_",50)

@ 15, 19 SAY REPLICATE ("_",15)

@ 16, 19 SAY REPLICATE ("_",15)

@ 20, 5 SAY REPLICATE ("_",55)

SETCOLOR (cColor)

SET CURSOR ON

@ 12, 19 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 .AND. alumnos->inscrito .AND. EMPTY(alumnos->baja)
            cNomAlum := alumnos->(NomAlum)
            @ 13, 19 GET cNomAlum PICTURE "@!" // Despliega el nombre del alumno
            CLEAR GETS
        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, 19 SAY REPLICATE ("_",55)
        SETCOLOR (cColor)
        @ 12, 19 SAY "No existe"
        INKEY (1)
        @ 12, 19 SAY " "
        LOOP
    ENDIF
ENDIF
nInscrip := 0.0
nColeg := 0.0
SET CURSOR ON
@ 15, 19 GET nInscrip PICTURE "$ #,###.##" VALID (!EMPTY(nInscrip))
@ 16, 19 GET nColeg PICTURE "$ #,###.##" VALID (!EMPTY(nColeg))
READ
IF LASTKEY () == K_ESC
    LOOP
ENDIF
pie (1)
cOk := "N"
@ 20, 5 SAY "Esta correcta la información ."
@ 20, 36 GET cOk PICTURE "!A" VALID (cOk$"SN")
READ
IF cOk == "S"
    edocuent->(DBSEEK(cAlum))
    IF edocuent->(FOUND())
        IF lockr (30)
            almacena (nInscrip,nColeg) //Almacena la informacion en disco
        ELSE
            ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
        ENDIF
    ENDIF
ENDIF

```

```

ELSE
  DBAPPEND()
  REPLACE edocuent->matrrialum WITH cAlum
  almacena (nInscrip,nColeg)
ENDIF
DBCOMMIT ()
ENDIF
ENDDO
RETURN NIL

```

```

FUNCTION cuota_general ()
LOCAL nKey, ok_bases, cColor2 := SETCOLOR (), archivo
LOCAL pantalla, nInscrip, nColeg, cOk,cuadro

```

```

IF !FILE("edocuent.dbf")
  aBase := {}
  AADD(aBase,{"matrrialum","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(CPATH+"edocuent",aBase)

```

```

ok_bases := abre_bases (30) // Abre la base estados de cuenta
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ESTADOS DE CUENTA")
  RETURN
ENDIF
archivo := CPATH + "edocuent.ntx"
INDEX ON matrrialum TO &archivo
edocuent->(DBCLOSEAREA())
ENDIF

```

```
ok_bases := abre_bases (22) // Base de inicio en exclusiva
IF lok_bases
  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 (cColor2)
  RETURN
ENDIF

IF inicio->cuotas
  inicio->(DBCLOSEAREA())
  TONE (750)
  TONE (1000)
  TONE (1200)
  ALERT ("ERROR: No se pueden cargar cuotas, ya que no se ha inicializado el semestre en el Sistema de Control Escolar")
  SETCOLOR (cColor2)
  RETURN
ENDIF

IF !FILE("grupo1.ntx")
  inicio->(DBCLOSEAREA())
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA GRUPOS PARA ESTA ESCUELA")
  SETCOLOR (cColor2)
  RETURN
ENDIF

ok_bases := abre_bases (10) // Abre la base de grupos
IF lok_bases
  inicio->(DBCLOSEAREA())
  TONE (700)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE GRUPOS")
  SETCOLOR (cColor2)
  RETURN
ENDIF

IF !FILE("alumno1.ntx")
  grupos->(DBCLOSEAREA())
  inicio->(DBCLOSEAREA())
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
  SETCOLOR (cColor2)
  RETURN
ENDIF
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

ok_bases := abre_bases (7) // Base de alumnos

```
IF !ok_bases
  grupos->(DBCLOSEAREA())
  inicio->(DBCLOSEAREA())
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  SETCOLOR (cColor2)
  RETURN
ENDIF
```

ok_bases := abre_bases (29) // Base de estados de cuenta

```
IF !ok_bases
  TONE (750)
  grupos->(DBCLOSEAREA())
  alumnos->(DBCLOSEAREA())
  inicio->(DBCLOSEAREA())
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ESTADOS DE CUENTA")
  SETCOLOR (cColor2)
  RETURN
ENDIF
```

pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

menu_apaga (.T.)

@ 9, 5 SAY REPLICATE ("-",45)

DISPBOX (10,1,MAXROW()-3,maxcol()-1,"+ +|+ +|_",C_REYENO) // Se dibuja la ventana de captura

pantallacampos (21)

DO WHILE .T.

pic (2)

nInscrip := 0.0

nColeg := 0.0

cColor := SETCOLOR (C_BORRADO)

@ 20, 5 SAY REPLICATE ("_",40)

SETCOLOR (cColor)

SET CURSOR ON

@ 14, 19 GET nInscrip PICTURE "\$ #,###.###" VALID (!EMPTY(nInscrip))

@ 16, 19 GET nColeg PICTURE "\$ #,###.###" VALID (!EMPTY(nColeg))

READ

IF LASTKEY () == K_ESC

RESTSCREEN (7,1,MAXROW()-3,MAXCOL()-1,pantalla)

SETCOLOR (cColor2)

@ MAXROW()-1,1 SAY SPACE (78)

grupos->(DBCLOSEAREA())

alumnos->(DBCLOSEAREA())

edocuent->(DBCLOSEAREA())

inicio->(DBCLOSEAREA())

RETURN -1

ENDIF

```
pic (1)
cOk := "N"
@ 20, 5 SAY "Esta correcta la información ."
@ 20, 36 GET cOk PICTURE "!A" VALID (cOk$"SN")
READ
SET CURSOR OFF
```

```
IF cOk = "S"
// Cuadro de dialogo
*****
alumnos->(DBGOTOP()) //
cuadro := SAVESCREEN (9,18,16,63) //
DISPBOX (9,18,15,62, "+-+!+-+! ", C_REYENO3) //
color := SETCOLOR ("R+/N") //
@ 11,21 SAY " ESPERE UN MOMENTO POR FAVOR" //
SETCOLOR (color) //
color := SETCOLOR ("GR+*/N") //
@ 12,26 SAY "SE ESTAN CARGANDO LAS CUOTAS" //
SETCOLOR (color) //
*****
```

```
DO WHILE alumnos->(EOF())
edocuent->(DBSEEK(alumnos->matrialum))
IF edocuent->(FOUND())
IF lockr (30)
almacena (nInscrip,nColeg) //Almacena la informacion en disco
ELSE
ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
EXIT
ENDIF
ELSE
DBAPPEND()
REPLACE edocuent->matrialum WITH alumnos->matrialum
almacena (nInscrip,nColeg)
ENDIF
DBCMMIT ()
alumnos->(DBSKIP())
ENDDO
EXIT
ENDIF
ENDDO
RESTSCREEN (9, 18, 16, 63, cuadro)
DISPBEGIN ()
RESTSCREEN (7, 1, MAXROW()-3, MAXCOL()-1, pantalla)
SETCOLOR (cColor2)
@ MAXROW()-1, 1 SAY SPACE (78)
inicio->cuotas := .T. // Indica que ya se cargaron las cuotas
grupos->(DBCLOSEAREA())
alumnos->(DBCLOSEAREA())
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
edocuent->(DBCLOSEAREA())
inicio->(DBCLOSEAREA())
DISPEND ()
TONE (750)
TONE (1200)
ALERT ("LA OPERACION SE REALIZO EXITOSAMENTE. SE HAN CARGADO LAS NUEVAS CUOTAS A LOS ALUMNOS")
RETURN NIL
```

```
FUNCTION almacena (nInscrip,nColeg)
LOCAL i,cCampo,nSaldoAnt := 0
FOR i := 1 TO 6
  cCampo := "SALDO"+ALLTRIM(STR(i))
  nSaldoAnt += edocuent->(&cCampo.)
NEXT
REPLACE edocuent->inscripcio WITH nInscrip ;;
  edocuent->cargo1 WITH nColeg ;;
  edocuent->cargo2 WITH nColeg ;;
  edocuent->cargo3 WITH nColeg ;;
  edocuent->cargo4 WITH nColeg ;;
  edocuent->cargo5 WITH nColeg ;;
  edocuent->cargo6 WITH nColeg ;;
  edocuent->pagado1 WITH .F. ;;
  edocuent->pagado2 WITH .F. ;;
  edocuent->pagado3 WITH .F. ;;
  edocuent->pagado4 WITH .F. ;;
  edocuent->pagado5 WITH .F. ;;
  edocuent->pagado6 WITH .F. ;;
  edocuent->saldo1 WITH nColeg ;;
  edocuent->saldo2 WITH nColeg ;;
  edocuent->saldo3 WITH nColeg ;;
  edocuent->saldo4 WITH nColeg ;;
  edocuent->saldo5 WITH nColeg ;;
  edocuent->saldo6 WITH nColeg ;;
  edocuent->saldoant WITH nSaldoAnt
/* edocuent->saldo1 WITH 0.0 ;;
  edocuent->saldo2 WITH 0.0 ;;
  edocuent->saldo3 WITH 0.0 ;;
  edocuent->saldo4 WITH 0.0 ;;
  edocuent->saldo5 WITH 0.0 ;;
  edocuent->saldo6 WITH 0.0 ;;
*/
RETURN
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

FUNCTION becarios ()

LOCAL nKey, ok_bases, cColor2 := SETCOLOR (),cAlum := SPACE(7)

LOCAL pantalla, cGrupo, nBecas := 0, cOk

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)

grupos->(DBCLOSEAREA())

ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")

RETURN

ENDIF

ok_bases := abre_bases (29) // Base de estados de cuenta

IF !ok_bases

TONE (750)

grupos->(DBCLOSEAREA())

alumnos->(DBCLOSEAREA())

ALERT ("NO SE PUEDE ABRIR LA BASE DE ESTADOS DE CUENTA")

RETURN

ENDIF

pantalla := SAVESCREEN (7,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura

menu_apaga (.T.)

@ 9, 5 SAY REPLICATE ("-.45)

DO WHILE .T.

pie (2)

DISPBOX (10,1,MAXROW()-3,maxcol()-1,"+-+!+-+!_",C_REYENO) // Se dibuja la ventana de captura

pantallacampos (21.5)

```

DO WHILE .T.
  cGrupo := SPACE(3)
  SET CURSOR ON
  @ 11, 19 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)
    grupos->(DBCLOSEAREA())
    alumnos->(DBCLOSEAREA())
    edocuent->(DBCLOSEAREA())
    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

```

```

DO WHILE .T.
  cAlum := SPACE(7)
  cColor := SETCOLOR (C_BORRADO)
  @ 12, 19 SAY REPLICATE ("_",10)
  @ 13, 19 SAY REPLICATE ("_",50)
  @ 15, 26 SAY REPLICATE ("_",15)
  @ 16, 19 SAY REPLICATE ("_",15)
  @ 20, 5 SAY REPLICATE ("_",55)
  SETCOLOR (cColor)
  SET CURSOR ON
  @ 12, 19 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 .AND. alumnos->inscrito .AND. EMPTY(alumnos->baja)
      cNomAlum := alumnos->(NomAlum)
      @ 13, 19 GET cNomAlum PICTURE "@!" // Despliega el nombre del alumno
      CLEAR GETS
    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, 19 SAY REPLICATE ("_",55)
    SETCOLOR (cColor)
    @ 12, 19 SAY "No existe"
    INKEY (1)
    @ 12, 19 SAY "      "
    LOOP
  ENDIF
ENDIF
nInscrip := 0.0
nColeg := 0.0
SET CURSOR ON
@ 15, 26 GET nBecas PICTURE "999" VALID !EMPTY(nBecas)
READ
IF LASTKEY () = K_ESC
  LOOP
ENDIF
pie (1)
cOk := "N"
@ 20, 5 SAY "Esta correcta la información ."
@ 20, 36 GET cOk PICTURE "!A" VALID (cOk$"SN")
READ
IF cOk = "S"
  edocuent->(DBSEEK(cAlum))
  IF edocuent->(FOUND())
    IF lockr (30)
      REPLACE edocuent->beca WITH nBecas
    ELSE
      ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
    ENDIF
  ENDIF

```

```
ELSE
  DBAPPEND()
  REPLACE edocuent->beca WITH nBecas
ENDIF
DBCOMMIT ()
ENDIF
ENDDO
ENDDO
RETURN NIL
```

CATALOGO.PRG

```

#include "Diploma.Ch"
#include "Tipos.Ch"
#include "InKey.Ch"
#include "SetCurs.Ch"           // Del Clipper para control del Cursor

#define wTop  aWin[1]
#define wLeft  aWin[2]
#define wBottom aWin[3]
#define wRight aWin[4]
#define wFrame aWin[5]
#define wColor aWin[6]
#define wTitle aWin[7]
#define wBack  aWin[8]
#define wOpen  aWin[9]
#define wLEN   10

FUNCTION catalogos (opcion)
* ..... Vars. Locales
FIELD mes
LOCAL cColores := SETCOLOR (),ok_bases,cConcepto,cColores2
PUBLIC aWin1,aWin2
* ..... CONFIGURACION
SetCursor(0)

* ..... BASES DE DATOS
IF opcion == 1
  ok_bases := abre_bases(27)
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE MES")
    RETURN
  ENDIF
ELSEIF opcion == 2
  ok_bases := abre_bases(28)
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE FORMAS DE PAGO")
    RETURN
  ENDIF
ELSEIF opcion == 3
  ok_bases := abre_bases(26)
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE CATALOGO DE CONCEPTOS")
    RETURN
  ENDIF
ENDIF
REINDEX

```

```

* ..... Proceso general
cConcepto := main_catalogo(opcion) // Proceso

* ..... Fin de trabajo
IF opcion = 1
  nNumMes := mes->(RECNO())
  mes->(DBCLOSEAREA())
ELSEIF opcion = 2
  pago->(DBCLOSEAREA())
ELSEIF opcion = 3
  concepto->(DBCLOSEAREA())
ENDIF
DISPBEGIN ()
  WinClose(aWin1)
  SETCOLOR ("W/N")
  @ MAXROW()-1,1 SAY SPACE (78)
  SETCOLOR (cColores)
DISPEND ()
RETURN cConcepto

FUNCTION main_catalogo(opcion)
  Local oBrowse, nSelect := Select(), nKey := 0, lRefresh := Cierto
  LOCAL nReg,cReg

  * ..... Filtra
  IF opcion = 1
    IF MONTH(DATE()) >= 2 .AND. MONTH(DATE()) <= 8
      SET FILTER TO mes->tipo = "N"
    ELSE
      SET FILTER TO mes->tipo = "P"
    ENDIF
    mes->(DBGOTOP())
  ENDIF

  * ..... Construye
  oBrowse := ConstruyeBrowse(opcion)
  * ..... Control
  pie (4)

  Do While nKey != K_ESC
    Do While lRefresh .And. !oBrowse:stabilize()
      EndDo
    nKey := reempinkey (0)
    lRefresh := Cierto
  Do Case
    Case nKey == K_ENTER
      IF opcion == 1
        RETURN mes->mes
      ELSEIF opcion == 2
        RETURN pago->pago
      ELSE

```

```

    RETURN concepto->concepto
ENDIF
Case nKey == K_LEFT
  oBrowse:Left()
Case nKey == K_RIGHT
  oBrowse:Right()
Case nKey == K_UP
  oBrowse:Up()
Case nKey == K_DOWN
  oBrowse:Down()
Case nKey == K_PGDN
  oBrowse:PageDown()
Case nKey == K_PGUP
  oBrowse:PageUp()
Case nKey == K_ESC
  RETURN NIL
OtherWise
  TONE (1000)
  TONE (1200)
EndCase
EndDo

SELECT ( nSelect )
RETURN NIL

/* -----
ConstruyeBrowse() -> oBrowse
. Constructor del Objeto Browse.
----- */

STATIC FUNCTION ConstruyeBrowse(opcion)
Local oTb
aWin1 := DefWin(11, 53, 20, 78, FRAME2, 2)
oTb := TBrowseDB(12, 56, 19, 77)
oTb:HeadSep := "-|-|"
oTb:ColSep := "|_|"
oTb:ColorSpec := Kolor(2)
IF opcion == 1
  oTb:AddColumn( TbColumnNew(" Mes", {||mes->mes}))
ELSEIF opcion == 2
  oTb:AddColumn( TbColumnNew(" Forma de Pago", {||pago->pago}))
ELSE
  oTb:AddColumn( TbColumnNew(" Concepto", {||concepto->concepto}))
ENDIF
DispBegin()
  WinOpen( aWin1 )
  oTb:ForceStable()
DispEnd()
RETURN oTb

```

CONCEN.PRG

```

#include "Diploma.Ch"
#include "Tipos.Ch"
#include "InKey.Ch"
#include "SetCurs.Ch"           // Del Clipper para control del Cursor

#define wTop      aWin[1]
#define wLeft     aWin[2]
#define wBottom   aWin[3]
#define wRight    aWin[4]
#define wFrame    aWin[5]
#define wColor    aWin[6]
#define wTitle    aWin[7]
#define wBack     aWin[8]
#define wOpen     aWin[9]
#define wLEN      10

FUNCTION conceptos (tipo)
* ..... Vars. Locales
LOCAL cColores := SETCOLOR (), ok_bases, pantalla
PUBLIC aWin1,aWin2, cTipo

* ..... BASES DE DATOS
IF tipo
  ok_bases := abre_bases(26)
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE CONCEPTOS")
    RETURN
  ENDIF
ELSE
  ok_bases := abre_bases(28)
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE FORMAS DE PAGO")
    RETURN
  ENDIF
ENDIF
REINDEX
pantalla := SAVESCREEN (0,0,MAXROW(),MAXCOL())
@ 9, 2 SAY REPLICATE ("-",70)
SET CURSOR OFF
menu_apaga(.T.)
    
```

```
* ..... Proceso general
main_conceptos(tipo) // Proceso
```

```
* ..... Fin de trabajo
IF tipo
concepto->(DBCLOSEAREA())
ELSE
pago->(DBCLOSEAREA())
ENDIF
DISPBEGIN ()
WinClose(aWin1)
@ MAXROW()-1,1 SAY SPACE (78)
RESTSCREEN (0,0,MAXROW(),MAXCOL(),pantalla)
DISPEND ()
RETURN NIL
```

```
FUNCTION main_conceptos(tipo)
Local oBrowse, nSelect := Select(), nKey := 0, lRefresh := Cierto
LOCAL nReg,cReg,cMensaje
```

```
* ..... Construye
oBrowse := ConstruyeBrowse(tipo)
```

```
* ..... Inicio de la Base
IF tipo
concepto->(DBGOTOP())
ELSE
pago->(DBGOTOP())
ENDIF
```

```
* ..... Control
pie (16)
Do While nKey != K_ESC
Do While lRefresh .And. !oBrowse:stabilize()
EndDo
nKey := reempinkey (0)
lRefresh := Cierto
Do Case
Case nKey == K_LEFT
oBrowse:Left()
Case nKey == K_RIGHT
oBrowse:Right()
Case nKey == K_UP
oBrowse:Up()
Case nKey == K_DOWN
oBrowse:Down()
Case nKey == K_PGDN
oBrowse:PageDown()
Case nKey == K_PGUP
oBrowse:PageUp()
```

```

Case nKey == K_ALT_A .OR. nKey == K_INS
  nReg := RECNO()
  If IRefresh := Edita(Cierto,tipo)
    oBrowse:RefreshAll()
  EndIf
  DBGOTO(nReg)
Case nKey == K_ALT_E .Or. nKey == K_SPACE
  nReg := RECNO()
  If IRefresh := Edita(Falso, tipo)
    oBrowse:RefreshCurrent()
  EndIf
  DBGOTO(nReg)
Case nKey == K_ALT_B .OR. nKey == K_DEL
  TONE (750)
  IF tipo
    cMensaje := "Esta seguro de dar de baja este concepto"
  ELSE
    cMensaje := "Esta seguro de dar de baja esta forma de pago"
  ENDIF
  IF ALERT (cMensaje,{"No","Si"}) == 2
    IF lockr (30)
      nReg := RECNO()
      cReg := &(IndexKey(IndexOrd()))
      DELETE
      DBUNLOCK()
      DbSeek (cReg, Cierto)
      oBrowse:RefreshAll()
    ELSE
      ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
    ENDIF
  ENDIF
Case nKey == K_ESC
  EXIT
  OtherWise
    TONE (1000)
    TONE (1200)
EndCase
EndDo
SELECT ( nSelect )
RETURN NIL

```

```

/* -----
ConstruyeBrowse() -> oBrowse
. Constructor del Objeto Browse.
----- */

STATIC FUNCTION ConstruyeBrowse(tipo)
Local oTb
aWin1 := DefWin(10, 1, 21, 78, FRAME2, 2, IIF(tipo,"CONCEPTOS","FORMA DE PAGO"))
oTb := TBrowseDB(11, 2, 20, 77)
oTb.HeadSep := "-|--"
oTb.ColSep := "| |"
oTb.ColorSpec := Kolor(2)
IF tipo
oTb.AddColumn( TbColumnNew("Concepto", {|| concepto->concepto}))
ELSE
oTb.AddColumn( TbColumnNew("Forma de Pago", {|| pago->pago}))
ENDIF
DispBegin()
WinOpen( aWin1 )
oTb.ForceStable()
DispEnd()
RETURN oTb

/* -----
Edita(lNew) -> NIL
. Función para agregar o editar un concepto.
----- */

STATIC FUNCTION Edita(lNuevo,tipo)
LOCAL cConcepto, cFormaPago,aWin, cColor, nCursor, GetList := {}, cOp := "G"

* ..... INICIA
cColor := SetColor( Kolor(3) )
If lNuevo
pic(16)
IF tipo
cConcepto := SPACE(55)
ELSE
cFormaPago := SPACE(25)
ENDIF
Else
IF tipo
cConcepto := concepto->concepto
ELSE
cFormaPago := pago->pago
ENDIF
EndIf

nCursor := SetCursor(1)

```

```

* ..... VENTANA
aWin := DefWin( 14, 3, 19, 76, FRAME2, 3, IIF(tipo, " CONCEPTOS " + IIF(INuevo, "Agrega ", "Edita "),
FORMA DE PAGO " + IIF(INuevo, "Agrega ", "Edita "))
DispBegin()
  WinOpen(aWin)
  IF tipo
    @ 1, 2 IN aWin SAY 'Concepto .'
  ELSE
    @ 1, 16 IN aWin SAY 'Forma de Pago .'
  ENDIF
  @ 3, 24 IN aWin SAY "<> Guarda o Cancela"
DispEnd()
SETCOLOR (cColor)
cColor := SETCOLOR(kolor(5))

IF tipo
  @ 1, 14 IN aWin GET cConcepto PICTURE '@!'
ELSE
  @ 1, 32 IN aWin GET cFormaPago PICTURE '@!'
ENDIF
@ 3, 25 IN aWin GET cOp PICTURE '!' WHEN !EMPTY(cConcepto) VALID cOp $ "CG"
READ
Do Case
  Case LastKey() == K_ESC .Or. cOp == "C"
    cOp := "C"
  Case INuevo .AND. tipo .AND. DBSEEK(cConcepto)
    TONE (750)
    ALERT ("Ya Existe El Concepto")
    cOp := "C"
  Case INuevo .AND. !tipo .AND. DBSEEK(cFormaPago)
    TONE (750)
    ALERT ("Ya Existe esta Forma de Pago")
    cOp := "C"
  CASE INuevo
    APPEND BLANK
    IF tipo
      REPLACE concepto->concepto WITH cConcepto
    ELSE
      REPLACE pago->pago WITH cFormaPago
    ENDIF
  OtherWise
    IF lockr (30)
      IF tipo
        REPLACE concepto->concepto WITH cConcepto
      ELSE
        REPLACE pago->pago WITH cFormaPago
      ENDIF
      DBUNLOCK()
    ELSE
      ALERT ("EL REGISTRO SE ENCUENTRA OCUPADO, NO SE PUEDE REALIZAR LA OPERACION EN ESTE MOMENTO")
    ENDIF

```

```

EndCase
DBCOMMIT ()
WinClose(aWin) // Cierra ventana
SetColor(cColor)
SetCursor(nCursor)
*/
RETURN cOp != "C"

/* -----
Kolor(nJuego, nColor) -> cColor
. Regresa un juego <nJuego> de colores o un color del juego si se
  pasa <nColor>.
----- */

FUNCTION Kolor(nJuego, nColor)
STATIC aColor := { {"W/B", "W+/R", "W+/B", "B/W"}, ;
                  {"W+/B", "W+/R", "W+/B", "B/W+"}, ;
                  {"W/Rb", "W+/Rb", "W+/Rb", "Gr+/Rb"}, ;
                  {"N/W", "W+/R", "B/W", "W/B"}, ;
                  {"W/B", "GR+/R", "W/B", "RB/W"} }

If IsNil(nColor)
    RETURN aColor[nJuego, 1] + "," + aColor[nJuego, 2] + "," + aColor[nJuego, 3]
EndIf
RETURN aColor[nJuego, nColor]

/* -----
DefWin(nT, nL, nB, nR, cFrame, nJuego, cTitle) -> aWin
. Define una ventana.
----- */

FUNCTION DefWin(nT, nL, nB, nR, cFrame, nJuego, cTitle)
Local nPos, aWin := Array(wLEN), i
wTop := nT
wLeft := nL
wBottom := nB
wRight := nR
wFrame := IIF( IsNil(cFrame), FRAME1, cFrame )
wColor := IIF( IsNil(nJuego), 1, nJuego )
wTitle := IIF( IsNil(cTitle), "", cTitle )
wBack := NIL
wOpen := Falso
RETURN aWin

```

```

/* -----
WinOpen(aWin, cFrame, cTitle) -> aWin
. Despliega en Forma de menú PopUp.
----- */

FUNCTION WinOpen(aWin, cFrame, cTitle)
Local nCursor := SetCursor(SC_NONE), IClear := Falso, cBack, nRow, nCol
nRow := Row()
nCol := Col()
wTitle = IIF( IsNil(cTitle), wTitle, cTitle )
wFrame = IIF( IsNil(cFrame), wFrame, cFrame )
DispBegin()
  If !wOpen
    If Empty(wBack)
      wBack := SaveScreen(wTop, wLeft, wBottom, wRight)
      IClear := Cierto
    Else
      cBack := wBack
      wBack := SaveScreen(wTop, wLeft, wBottom, wRight)
      RestScreen(wTop, wLeft, wBottom, wRight, cBack)
    EndIf
  EndIf
  DispBox(wTop, wLeft, wBottom, wRight, wFrame, Kolor(wColor, 3) )
  If IClear
    DispBox(wTop + 1, wLeft + 1, wBottom - 1, wRight - 1, ;
      Space(9), Kolor(wColor, 1) )
    nRow := wTop + 1
    nCol := wLeft + 1
  EndIf
  If !Empty(wTitle)
    @ wTop, wLeft + Int((wRight - wLeft - Len(wTitle)) / 2) ;
    SAY PadR(wTitle, Min(Len(wTitle), wRight - wLeft)) ;
    COLOR Kolor(wColor, 4)
  EndIf
DispEnd()
SetCursor(nCursor)
wOpen := Cierto
SetPos(nRow, nCol)
RETURN aWin

```

```
/* -----  
WinClose(aWin) -> aWin  
  . Despliega en Forma de menú PopUp.  
----- */
```

```
FUNCTION WinClose(aWin)  
Local cBack  
Do Case  
  Case wOpen .And. !Empty(wBack)  
    cBack := SaveScreen(wTop, wLeft, wBottom, wRight)  
    RestScreen(wTop, wLeft, wBottom, wRight, wBack)  
    wBack := cBack  
  Case wOpen .And. Empty(wBack)  
    wBack := SaveScreen(wTop, wLeft, wBottom, wRight)  
EndCase  
wOpen := Falso  
RETURN aWin
```

```
/* -----  
WinRow(aWin, nRow) -> nWinRow  
  . Regresa el renglón relativo a la ventana.  
----- */
```

```
FUNCTION WinRow(aWin, nRow)  
RETURN nRow + wTop + 1
```

```
/* -----  
WinCol(aWin, nCol) -> nWinCol  
  . Regresa la columna relativa a la ventana.  
----- */
```

```
FUNCTION WinCol(aWin, nCol)  
RETURN nCol + wLeft + 1
```

```
/* -----  
WinRows(aWin) -> nRows  
  . Regresa el número de renglones útiles de <aWin>.  
----- */
```

```
FUNCTION WinRows(aWin)  
RETURN wBottom - wTop - 1
```

```
/* -----  
WinCols(aWin) -> nCols  
  . Regresa el número de renglones útiles de <aWin>.  
----- */
```

```
FUNCTION WinCols(aWin)  
RETURN wRight - wLeft - 1
```

```
FUNCTION DoNothing  
RETURN NIL
```

```
FUNCTION Mes(ndMes) // <- Transforma ndMes -> cMes  
Static aMeses := { "Ene", "Feb", "Mar", "Abr", "May", "Jun", ;  
                  "Jul", "Ago", "Sep", "Oct", "Nov", "Dic" }  
If IsDate(ndMes)  
    ndMes := Month(ndMes)  
EndIf  
RETURN IIF( Between(ndMes, 1, 12), aMeses[ndMes], Space(3) )
```

MESPAGO.PRG

```

#include "Diploma.Ch"
#include "Tipos.Ch"
#include "InKey.Ch"
#include "SetCurs.Ch"           // Del Clipper para control del Cursor

#define wTop      aWin[1]
#define wLeft     aWin[2]
#define wBottom   aWin[3]
#define wRight    aWin[4]
#define wFrame    aWin[5]
#define wColor    aWin[6]
#define wTitle    aWin[7]
#define wBack     aWin[8]
#define wOpen     aWin[9]
#define wLEN      10

FUNCTION mes_a_pagar ()
* ..... Vars. Locales
FIELD mes
LOCAL cColores := SETCOLOR (),ok_bases,cConcepto,cColores2
PUBLIC aWin1,aWin2

* ..... CONFIGURACION
SetCursor(0)

* ..... BASES DE DATOS

ok_bases := abre_bases(27)
IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE MES")
    RETURN
ENDIF
REINDEX

* ..... Proceso general
cConcepto := main_mes_a_pagar() // Proceso

* ..... Fin de trabajo
SET FILTER TO
mes->(DBCLOSEAREA())
DISPBEGIN ()
WinClose(aWin1)
SETCOLOR ("W/N")
@ MAXROW()-1,1 SAY SPACE (78)
SETCOLOR (cColores)
DISPEND ()
RETURN cConcepto
    
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
FUNCTION main_mes_a_pagar()
  Local oBrowse, nSelect := Select(), nKey := 0, lRefresh := Cierto
  LOCAL nReg,cReg

  * ..... Filtra
  IF MONTH(DATE()) >= 2 .AND. MONTH(DATE()) <= 8
    SET FILTER TO tipo = "N"
  ELSE
    SET FILTER TO tipo = "P"
  ENDIF
  mes->(DBGOTOP())

  * ..... Construye
  oBrowse := ConstruyeBrowse()

  * ..... Control
  pie (4)
  Do While nKey != K_ESC
    Do While lRefresh .And. !oBrowse:stabilize()
      EndDo
    nKey := reempinkey (0)
    lRefresh := Cierto
    Do Case
      Case nKey == K_ENTER
        RETURN mes->mes
      Case nKey == K_LEFT
        oBrowse:Left()
      Case nKey == K_RIGHT
        oBrowse:Right()
      Case nKey == K_UP
        oBrowse:Up()
      Case nKey == K_DOWN
        oBrowse:Down()
      Case nKey == K_PGDN
        oBrowse:PageDown()
      Case nKey == K_PGUP
        oBrowse:PageUp()
      Case nKey == K_ESC
        RETURN NIL
      Otherwise
        TONE (1000)
        TONE (1200)
    EndCase
  EndDo
  SELECT ( nSelect )
  RETURN NIL
```

```
/*-----  
  ConstruyeBrowse() -> oBrowse  
    . Constructor del Objeto Browse.  
-----*/
```

STATIC FUNCTION ConstruyeBrowse()

```
Local oTb  
aWin1 := DefWin(11, 55, 20, 78, FRAME2, 2)  
oTb := TBrowseDB(12, 56, 19, 77)  
oTb:HeadSep := "-|-"  
oTb:ColSep := "|"  
oTb:ColorSpec := Kolor(2)  
oTb:AddColumn( TbColumnNew(" Mcs", {||mes->mes}))  
DispBegin()  
  WinOpen( aWin1 )  
  oTb:ForceStable()  
DispEnd()  
RETURN oTb
```

UTILERIA.PRG

```
#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 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 haber 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,"+-+!+-+! ",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 HABRJR 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","_")
  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 HABRJR 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","_")
  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 HABRJR 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","_")

```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
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","_")
  GaugeDisplay(nHandle)
  INDEX ON clavmae 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","_")
  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"
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
nHandle := GaugeNew (19, 20, 21, 60, "13", "14", "_")
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 )
```

```
// Assign default values
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
```

```
// Resolve parameters
IF cBackColor <> NIL
  aHandle[ B_BACKCOLOR ] := cBackColor
ENDIF

IF cBarColor <> NIL
  aHandle[ B_BARCOLOR ] := cBarColor
ENDIF

IF cBarCharacter <> NIL
  aHandle[ B_BARCHAR ] := cBarCharacter
ENDIF

// OK, the defaults are set, now let's make sure it will fit on the
// screen correctly
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)

// Draw bar to show current percent

*GaugeUpdate(aHandle. aHandle[B_PERCENT])

RETURN(aHandle)

* Actualiza el porcentaje

FUNCTION GaugeUpdate

LOCAL nCenter := ROUND((aHandle[B_RIGHT] - aHandle[B_LEFT]) / 2.0) + 1

LOCAL cOldColor := SETCOLOR(aHandle[B_BARCOLOR])

LOCAL nBarRatio := (aHandle[B_RIGHT] - (aHandle[B_LEFT] + 1))

LOCAL 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)

 @a nRow + aHandle[B_TOP], aHandle[B_LEFT] + 1 SAY :

 REPLICATE(aHandle[B_BARCHAR], nCols)

NEXT

SETCOLOR(cOldColor)

* RETURN(aHandle)

RETURN .T.

REPO.PRG

```
#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"
```

```
FUNCTION centro (cTexto)
RETURN (80-LEN(ALLTRIM(cTexto)))/2
```

```
/* Función que cambia el numero pasado como parametro a su correspondiente en letra. unicamente
dentro del rango de 0-9999 */
```

```
FUNCTION cambia (nNumero)
LOCAL cNumero := SPACE(7), cC := {}, nC := {}, aux, tmp, cNum := {}
STATIC aMiles := {"UN MIL","DOS MIL","TRES MIL","CUATRO MIL";
                 "CINCO MIL","SEIS MIL","SIETE MIL","OCHO MIL";
                 "NUEVE MIL"}
STATIC aCientos := {"CIENTO","DOS CIENTOS","TRES CIENTOS","CUATRO CIENTOS";
                   "QUINIENTOS","SEIS CIENTOS","SIETE CIENTOS","OCHO CIENTOS";
                   "NOVECIENTOS"}
STATIC aDecenas := {"UN","DOS","TRES","CUATRO","CINCO","SEIS","SIETE";
                   "OCHO","NUEVE","DIEZ","ONCE","DOCE","TRECE","CATORCE";
                   "QUINCE","DIECISEIS","DIECISIETE","DIECIOCHO","DIECINUEVE";
                   "VEINTE","VEINTIUNO","VEINTIDOS","VEINTITRES","VEINTICUATRO";
                   "VEINTICINCO","VEINTISEIS","VEINTISIETE","VEINTIOCHO";
                   "VEINTINUEVE","TREINTA","TREINTA Y UNO","TREINTA Y DOS";
                   "TREINTA Y TRES","TREINTA Y CUATRO","TREINTA Y CINCO","TREINTA Y SEIS";
                   "TREINTA Y SIETE","TREINTA Y OCHO","TREINTA Y NUEVE","CUARENTA";
                   "CUARENTA Y UNO","CUARENTA Y DOS","CUARENTA Y TRES","CUARENTA Y CUATRO";
                   "CUARENTA Y CINCO","CUARENTA Y SEIS","CUARENTA Y SIETE","CUARENTA Y OCHO";
                   "CUARENTA Y NUEVE","CINCUENTA","CINCUENTA Y UNO","CINCUENTA Y DOS";
                   "CINCUENTA Y TRES","CINCUENTA Y CUATRO","CINCUENTA Y CINCO","CINCUENTA Y SEIS";
                   "CINCUENTA Y SIETE","CINCUENTA Y OCHO","CINCUENTA Y NUEVE","SESENTA";
                   "SESENTA Y UNO","SESENTA Y DOS","SESENTA Y TRES","SESENTA Y CUATRO";
                   "SESENTA Y CINCO","SESENTA Y SEIS","SESENTA Y SIETE","SESENTA Y OCHO";
                   "SESENTA Y NUEVE","SETENTA","SETENTA Y UNO","SETENTA Y DOS","SETENTA Y TRES";
                   "SETENTA Y CUATRO","SETENTA Y CINCO","SETENTA Y SEIS","SETENTA Y SIETE";
                   "SETENTA Y OCHO","SETENTA Y NUEVE","OCHENTA","OCHENTA Y UNO","OCHENTA Y DOS";
                   "OCHENTA Y TRES","OCHENTA Y CUATRO","OCHENTA Y CINCO","OCHENTA Y SEIS";
                   "OCHENTA Y SIETE","OCHENTA Y OCHO","OCHENTA Y NUEVE","NOVENTA","NOVENTA Y UNO";
                   "NOVENTA Y DOS","NOVENTA Y TRES","NOVENTA Y CUATRO","NOVENTA Y CINCO";
                   "NOVENTA Y SEIS","NOVENTA Y SIETE","NOVENTA Y OCHO","NOVENTA Y NUEVE"}
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
FOR i = 1 to LEN (alltrim(str(nNumero)))
  temp := SUBSTR(alltrim(str(nNumero)),i,1)
  if temp == "."
    exit
  endif
  AADD(cNum,temp)
NEXT
ASIZE (cC,5)
ASIZE (nC,5)

*.....MILES
IF LEN(cNum) == 4
  nC[1] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),1,1))
  IF nC[1] # 0
    cC [1] := aMiles[nC[1]]
  ELSE
    cC [1] := SPACE(1)
  ENDIF
ELSE
  cC[1] := SPACE(1)
ENDIF

*.....Cientos
IF LEN(cNum) >= 3
  IF LEN(cNum) == 3
    nC[2] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),1,1))
    nC[3] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),2,2))
  ELSE
    nC[2] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),2,1))
    IF nC[2] # 0
      cC [2] := aCientos[nC[2]]
    ELSE
      cC {2} := SPACE(1)
    ENDIF
    nC[3] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),3,2))
  ENDIF
  IF nC[3] == 0 .AND. nC[2] == 1
    cC[2] := "CIE"
  ELSE
    IF nC[2] # 0
      cC [2] := aCientos[nC[2]]
    ELSE
      cC [2] := SPACE(1)
    ENDIF
  ENDIF
ELSE
  cC[2] := SPACE(1)
ENDIF
```

```

*.....Decenas
IF LEN(cNum) >= 2 .OR. LEN(cNum) == 1
  IF LEN(cNum) = 4
    nC [3] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),3,2))
    IF nC[3] # 0
      cC [3] := aDecenas[nC[3]]
    ELSE
      cC [3] := SPACE(1)
    ENDIF
  ELSEIF LEN(cNum) = 3
    nC [3] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),2,2))
    IF nC[3] # 0
      cC [3] := aDecenas[nC[3]]
    ELSE
      cC [3] := SPACE(1)
    ENDIF
  ELSE //IF LEN(cNum) = 2 .OR. LEN(cNum) == 1
    nC [3] := VAL(SUBSTR(ALLTRIM(STR(nNumero)),1,2))
    IF nC[3] # 0
      cC [3] := aDecenas[nC[3]]
    ELSE
      cC [3] := SPACE(1)
    ENDIF
  ENDIF
ELSE
  cC[2] := SPACE(1)
ENDIF

IF temp == "."
  IF LEN(cNum) = 4
    cC[4] := SUBSTR(ALLTRIM(STR(nNumero)),6,1)
    IF LEN(ALLTRIM(STR(nNumero))) == 7
      cC[5] := SUBSTR(ALLTRIM(STR(nNumero)),7,1)
    ELSE
      cC[5] := SPACE(1)
    ENDIF
  ELSEIF LEN(cNum) = 3
    cC[4] := SUBSTR(ALLTRIM(STR(nNumero)),5,1)
    IF LEN(ALLTRIM(STR(nNumero))) == 6
      cC[5] := SUBSTR(ALLTRIM(STR(nNumero)),6,1)
    ELSE
      cC[5] := SPACE(1)
    ENDIF
  ELSEIF LEN(cNum) = 2
    cC[4] := SUBSTR(ALLTRIM(STR(nNumero)),4,1)
    IF LEN(ALLTRIM(STR(nNumero))) == 5
      cC[5] := SUBSTR(ALLTRIM(STR(nNumero)),5,1)
    ELSE
      cC[5] := SPACE(1)
    ENDIF
  .

```

```

ELSEIF LEN(cNum) == 1
  cC[4] := SUBSTR(ALLTRIM(STR(nNumero)),3,1)
  IF LEN(ALLTRIM(STR(nNumero))) == 4
    cC[5] := SUBSTR(ALLTRIM(STR(nNumero)),4,1)
  ELSE
    cC[5] := SPACE(1)
  ENDIF
ENDIF
ENDIF
ELSE
  cC[4] := cC[5] := "0"
ENDIF

RETURN ALLTRIM(ALLTRIM(cC[1]) + "-" + ALLTRIM(cC[2]) + "-" + ALLTRIM(cC[3]) + " PESOS " + cC[4] + cC[5] + " 100 M.N.")

/* 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 */
STATIC FUNCTION sele_esc(scc)
  LOCAL nKey, ok_bases, cEscuela

  ok_bases := abre_bases (1) // Base de escuelas
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ESCUELAS")
    RETURN
  ENDIF

  pantalla := SAVESCREEN (10,1,MAXROW()-3,MAXCOL()-1) // Se resguarda el area de pantalla de captura
  pic (4)
  IIF (scc == NIL, scc := .T., scc := scc)
  IIF (scc.limpiamenus(1).limpiamenus (3)) // Se apagan las opciones del menú
  DISPBOX (10,1,maxrow()-3,maxcol()-1,"+-+|+-+|_".C_REYENO) // Se dibuja la ventana de captura
  @ 9,14 SAY REPLICATE ("-",15)
  cColor:= SETCOLOR (C_CAP)

  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

```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

/* 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)

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)

cEscuela := escuela->nomesc

escuela->(DBCLOSEAREA())

RETURN cEscuela

ELSE

escuela->(DBCLOSEAREA())

RETURN NIL

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 NIL

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

FUNCTION scc_encabezado (nHoja, tipo)

LOCAL cTexto1 := "UNIVERSIDAD VASCO DE QUIROGA".x. nPr1 := m->nimp

IF tipo == 1

cTexto2 := "Corte de Caja"

x := 14

ELSEIF tipo == 2

cTexto2 := "Reporte de Ingresos"

x := 11

ELSEIF tipo == 3

cTexto2 := "Reporte de Becarios"

x := 11

ENDIF

// Se imprime el encabezado de la boleta

@ PROW(), PCOL() SAY CsSendPrn3 ("GRUESA",nPr1)

@ 1,6 SAY cTexto1

@ 4,x SAY cTexto2

@ PROW(), PCOL() SAY CsSendPrn3 ("-GRUESA",nPr1)

@ 6,2 SAY "HOJA N° "+ALLTRIM(STR(nHoja))

@ 6,65 SAY "HORA : " + TIME()

IF tipo == 2

@ 7,2 SAY "ESCUELA : " + cEscuela

ENDIF

@ 8,2 SAY "TAXIMAROA # 62 COL. FELIX IRETA"

@ 8,61 SAY ALLTRIM(STR(DAY(DATE()))+" DE "+UPPER(CMONTH(DATE)))+" DE "+ALLTRIM(STR(YEAR(DATE))))

@ 9,2 SAY REPLICATE("-",78)

IF tipo == 1

@ 10,2 SAY " FORMA DE PAGO IMPORTE"

ELSEIF tipo == 2

@ 10,2 SAY " CONCEPTO IMPORTE"

ELSEIF tipo == 3

@ 10,2 SAY " ALUMNO % DE BECA"

ENDIF

@ 11,2 SAY REPLICATE("-",78)

RETURN

FUNCTION corte_caja ()

LOCAL nPrnActive, cPuerto, aPuertos, nprn := m->Nnimp

LOCAL pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1)

LOCAL cColor2, aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F.

LOCAL cuadro.ok := "N", nSumaTotal := 0.0, nHoja := 1

LOCAL aFormaPago := {}, aSumas := {}, cFormaPago, suma := 0.0, nR, i

PRIVATE impresora

/* Vars. que indican la posición donde se desplegará el nombre de la impresora si está es cambiada */

PUBLIC x := 14, 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")
  impres->(DBCLOSEAREA())
  RETURN
ENDIF

impresora := CsPrnName2(nPrn)         // Inicializa la impresora
impres->(DBCLOSEAREA())
menu_apaga (.F.)
@ 9, 15 SAY REPLICATE ("-",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
  pantallacampos (24)
  SET KEY K_F2 TO cambia_impresora ()
  @ 14,32 SAY impresora COLOR C_CONFIG4
  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

  IF ok == "N" .OR. LASTKEY () == K_ESC
    EXIT
  ENDIF
  SET KEY K_F2 TO
  ok := "N"
ok_bases := abre_bases (19) // Base de impresoras

IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
  EXIT
ENDIF

diario->(DBSETFILTER( {|| diario->(fecha) == DATE()}. "diario->(fecha) == DATE()" ))
diario->(DBGOTOP())
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")
    @ 12,22 SAY "SE ESTA IMPRIMIENDO EL CORTE DE CAJA"
    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)
    diario->(DBSETORDER(2))
    diario->(DBGOTOP())
    nSumaTotal := 0.0
    DO WHILE diario->(!EOF())
      INKEY (.5)
      IF LASTKEY () == K._ESC
        EXIT
      ENDIF
      cFormaPago := diario->formapago
      AADD (aFormaPago,cFormaPago)
      suma := 0
      DO WHILE cFormaPago == diario->formapago
        suma += diario->monto
        diario->(DBSKIP())
      ENDDO
      AADD (aSumas,suma)
      nSumaTotal += suma
    ENDDO
    IF LASTKEY () == K._ESC
      flag := .T.
      EXIT
    ENDIF

    diario->(DBGOTOP())
    diario->(DBSETORDER(1))
    SET (_SET_PRINTFILE. ALLTRIM(cPuerto))
    SET (_SET_DEVICE. "PRINTER")
    scc_encabezado (nHoja,1)
    nR := PROW() + 1
    FOR i = 1 TO LEN (aFormaPago)
      INKEY (.5)
      IF LASTKEY () == K._ESC
        EXIT
      ENDIF
  
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
@ nR.2 SAY aFormaPago [i] PICTURE "@!"
@ nR.60 SAY aSumas [i] PICTURE "$ #.###.##"
nR++
IF nR >= 50
  nHoja++
  scc_encabezado (nHoja.1)
  nR := PROW() + 2
ENDIF
NEXT
IF LASTKEY () == K_ESC
  flag := .T.
  EXIT
ENDIF
nR := PROW() + 1
@ nR.2 SAY REPLICATE (" ".78)
nR++
@ nR.50 SAY "Total ."
@ nR.60 SAY nSumaTotal PICTURE "$ #.###.##"
EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
REST$CREEN (9.18.16.63. cuadro)
flag := .T.
EXIT
ELSE
  TONE (350)
  nOp := ALERT ("La impresora no esta lista".aOpcion)
  IF nOp == 2
    flag := .T.
    EXIT
  ENDIF
ENDIF
ENDDO
IF flag
  flag := .F.
  impres->(DBCLOSEAREA())
  EXIT
ENDIF
ENDDO

diario->(DBCLEARFILTER())
*SET FILTER.TO
escapes->(DBCLOSEAREA())
REST$CREEN (1.1,MAXROW(),MAXCOL()-1,pantalla)
SETCOLOR (cColor2)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN -1
RETURN
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

/* Funcion que imprime los conceptos y el monto acumulado por concepto de la escuela que se seleccione */

FUNCTION ingresos ()

```
LOCAL nPrnActive, cPuerto, aPuertos, nprn := n->NmImp
LOCAL pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1)
LOCAL cColor2, aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F.
LOCAL cuadro.ok := "N", nSumaTotal := 0.0, nHoja := 1
LOCAL aConcepto := {}, aSumas := {}, cConcep. suma := 0.0, nR, i
PRIVATE impresora.cEscuela
```

/* Vars. que indican la posición donde se desplegara el nombre de la impresora si está es cambiada */

```
PUBLIC x := 14, 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")
```

```
    impres->(DBCLOSEAREA())
```

```
ENDIF
```

```
impresora := CsPrnName2(nPrn)
```

```
// Inicializa la impresora
```

```
impres->(DBCLOSEAREA())
```

```
menu_apaga (.F.)
```

```
@ 9,15 SAY REPLICATE ("-.".60)
```

```
cEscuela := sele_escf(.F.)
```

```
IF cEscuela == NIL
```

```
    escapes->(DBCLOSEAREA())
```

```
    RETURN
```

```
ENDIF
```

```
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 (24)
```

```
    SET KEY K_F2 TO cambia_impresora ()
```

```
@ 14,32 SAY impresora COLOR C_CONFIG+
```

```
    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
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
IF ok == "N" .OR. LASTKEY () == K_ESC
  EXIT
ENDIF
SET KEY K_F2 TO
ok := "N"

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
  EXIT
ENDIF

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")
    @ 12,19 SAY "SE ESTA IMPRIMIENDO EL REPORTE DE INGRESOS"
    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)
    diario->(DBSETORDER(3))
    diario->(DBGOTOP())
    nSumaTotal := 0.0
    DO WHILE diario->(EOF())
      INKEY (.5)
      IF LASTKEY () == K_ESC
        EXIT
      ENDIF
      cConcep := diario->concepto
      IF UPPER(SUBSTR(cConcep,1,11)) == "COLEGIATURA"
        cConcep := "COLEGIATURA"
      ENDIF
      AADD (aConcepto.cConcep)
      suma := 0
      DO WHILE cConcep == diario->concepto .OR.:
        cConcep == UPPER(SUBSTR(diario->concepto,1,11))
        IF diario->escuela == cEscuela .AND. diario->fecha == DATE()
          suma += diario->monto
        ENDIF
        diario->(DBSKIP())
      ENDDO
    ENDWHILE
  ENDWHILE
```

```

AADD (aSumas.suma)
nSumaTotal += suma
ENDDO
IF LASTKEY () == K_ESC
    flag := .T.
    EXIT
ENDIF
diario->(DBGOTOP())
diario->(DBSETORDER(1))
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "PRINTER")
scc_encabezado (nHoja.2)
nR := PROW() + 1

FOR i = 1 TO LEN (aConcepto)
    INKEY (.5)
    IF LASTKEY () == K_ESC
        EXIT
    ENDIF
    @ nR,2 SAY aConcepto [i] PICTURE "@!"
    @ nR,60 SAY aSumas [i] PICTURE "$ #.### ##"
    nR++
    IF nR >= 50
        nHoja++
        scc_encabezado (nHoja.2)
        nR := PROW() + 2
    ENDIF
NEXT
IF LASTKEY () == K_ESC
    flag := .T.
    EXIT
ENDIF
nR := PROW() + 1
@ nR,2 SAY REPLICATE (" ",78)
nR++
@ nR,50 SAY "Total :"
@ nR,60 SAY nSumaTotal PICTURE "$ #.### ##"
EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
RESTSCREEN (9.18.16.63, cuadro)
flag := .T.
EXIT
ELSE
    TONE (350)
    nOp := ALERT ("La impresora no esta lista",aOpcion)
    IF nOp == 2
        flag := .T.
        EXIT
    ENDIF
ENDIF
ENDIF

```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
ENDDO
IF flag
    impres->(DBCLOSEAREA())
    flag := .F.
    EXIT
ENDIF
ENDDO
escapes->(DBCLOSEAREA())
RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
SETCOLOR (cColor2)
@. MAXROW()-1.1 SAY SPACE (78)
RETURN -1
RETURN

FUNCTION repo_becarios ()
    LOCAL nPrnActive, cPuerto, aPuertos, nprn := m->NmImp
    LOCAL pantalla := SAVESCREEN (1,1,MAXROW(),MAXCOL()-1).cColor1 := SETCOLOR()
    LOCAL cColor2, aOpcion := {"REINTENTAR","CANCELAR"}. flag := .F.
    LOCAL cuadro,ok := "N", nHoja := 1,cEscuela. nR. i,cPath
    PRIVATE impresora

/* Vars. que indican la posición donde se desplegara el nombre de la impresora si está es cambiada */
PUBLIC x := 14, y := 32

menu_apaga (.F.)
@ 9, 15 SAY REPLICATE ("-",60)
cEscuela := selecciona(.F.)

IF cEscuela # -1
    escuela->(DBGOTO(cEscuela))
    cPath := escuela->direct
    cClave := escuela->clave
    cTipo := escuela->tipo
    cTipoSep := escuela->tiposep
    SET PATH TO &cPath
    escuela->(DBCLOSEAREA())
ELSE
    RETURN
ENDIF

ok_bases := abre_bases (29) // Base de estados de cuenta
IF !ok_bases
    TONE (750)
    grupos->(DBCLOSEAREA())
    ALERT ("NO SE PUEDE ABRIR LA BASE DE ESTADOS DE CUENTA")
    RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
    SETCOLOR (cColor1)
    RETURN
ENDIF
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
IF !FILE("alumno1.ntx")
  TONE (700)
  ALERT ("NO SE HAN DADO DE ALTA ALUMNOS EN ESTA ESCUELA")
  RESTSCREEN (1,1,MAXROW(),MAXCOL()-1.pantalla)
  SETCOLOR (cColor1)
  RETURN
ENDIF

ok_bases := abre_bases (7) // Base de alumnos
IF !ok_bases
  edocuent->(DBCLOSEAREA())
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE ALUMNOS")
  RESTSCREEN (1,1,MAXROW(),MAXCOL()-1.pantalla)
  SETCOLOR (cColor1)
  RETURN
ENDIF
SET RELATION TO matrialum INTO edocuent

ok_bases := abre_bases (19) // Base de impresoras
IF !ok_bases
  edocuent->(DBCLOSEAREA())
  alumnos->(DBCLOSEAREA())
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
  RESTSCREEN (1,1,MAXROW(),MAXCOL()-1.pantalla)
  SETCOLOR (cColor1)
  RETURN
ENDIF

ok_bases := abre_bases (20) // Base de codigos de escapes
IF !ok_bases
  edocuent->(DBCLOSEAREA())
  alumnos->(DBCLOSEAREA())
  impres->(DBCLOSEAREA())
  TONE (750)
  ALERT ("NO SE PUEDE ABRIR LA BASE DE CODIGOS DE ESCAPE")
  RESTSCREEN (1,1,MAXROW(),MAXCOL()-1.pantalla)
  SETCOLOR (cColor1)
  RETURN
ENDIF

impresora := CsPrnName2(nPrn) // Inicializa la impresora
impres->(DBCLOSEAREA())

cColor2 := SETCOLOR(C_CAP)
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
DO WHILE .T.
  pie (13)
  DISPBOX (10,1,MAXROW()-3,maxcol()-1,"+-+!+-+! ".C_REYENO) // Se dibuja la ventana de captura
  pantallacampos (24)

  SET KEY K_F2 TO cambia_impresora ()
  @ 14,32 SAY impresora COLOR C_CONFIG4

  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

  IF ok == "N" .OR. LASTKEY () == K_ESC
    EXIT
  ENDIF
  SET KEY K_F2 TO
  ok := "N"

  ok_bases := abre_bases (19) // Base de impresoras
  IF !ok_bases
    TONE (750)
    ALERT ("NO SE PUEDE ABRIR LA BASE DE IMPRESORAS")
    EXIT
  ENDIF

  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")
      @ 12,19 SAY "SE ESTA IMPRIMIENDO EL REPORTE DE BECARIOS"
      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")
      scc_encabezado (nHoja.3)
      nR := PROW() + 1
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
DO WHILE alumnos->(EOF())
  INKEY (.5)
  IF LASTKEY () == K_ESC
    EXIT
  ENDIF
  IF edocuent->beca > 0 .AND. alumnos->inscrito .AND. alumnos->(EMPTY(baja))
    @ nR,2 SAY alumnos->nomalum PICTURE "@,!"
    @ nR,60 SAY edocuent->beca PICTURE "99"
    nR++
  ENDIF
  IF nR >= 50
    nHoja++
    scc_encabezado (nHoja,3)
    nR := PROW() + 2
  ENDIF
  alumnos->(DBSKIP())
ENDDO
IF LASTKEY () == K_ESC
  flag := .T.
  EXIT
ENDIF
nR := PROW() + 1
@ nR,2 SAY REPLICATE (".",78)
EJECT
SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
SET (_SET_DEVICE, "SCREEN")
RESTSCREEN (9,18,16,63, cuadro)
flag := .T.
EXIT
ELSE
  TONE (350)
  nOp := ALERT ("La impresora no esta lista".aOpcion)
  IF nOp == 2
    flag := .T.
    EXIT
  ENDIF
ENDIF
ENDDO
IF flag
  impres->(DBCLOSEAREA())
  flag := .F.
  EXIT
ENDIF
ENDDO
escapes->(DBCLOSEAREA())
alumnos->(DBCLOSEAREA())
edocuent->(DBCLOSEAREA())
RESTSCREEN (1,1,MAXROW(),MAXCOL()-1,pantalla)
SETCOLOR (cColor2)
@ MAXROW()-1,1 SAY SPACE (78)
RETURN -1
RETURN
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
FUNCTION recibo (cNombre.cMatricula.nImporte.nRecargo.cConcepto)
LOCAL cColor2. aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F.
LOCAL cuadro.ok := "N", nR, i

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")
    @ 12.27 SAY "SE ESTA IMPRIMIENDO EL RECIBO"
    SETCOLOR (color)
    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "PRINTER")
    SETPRC(1,1)
    nR := PROW()
    @ PROW(). PCOL() SAY CsSendPrn3 ("INICIO", nPrn)
    @ nR. 65 SAY ALLTRIM(STR(DAY(DATE))) " " ALLTRIM(STR(MONTH(DATE))) " " ALLTRIM(STR(YEAR(DATE)))
    @ nR+3. 23 SAY NomEsc
    @ nR+5. 12 SAY cNombre
    @ nR+5. 63 SAY cMatricula
    @ nR+7. 12 SAY nImporte+nRecargo PICTURE "#.###.##"
    @ PROW(). PCOL() SAY CsSendPrn3 ("COMPRIMIDA", nPrn)
    @ nR+7. 52 SAY cambia(nImporte+nRecargo)
    @ PROW(). PCOL() SAY CsSendPrn3 ("-COMPRIMIDA", nPrn)
    @ nR+13.5 SAY ALLTRIM(UPPER(cConcepto))+ " : $ "+ALLTRIM(STR(nImporte))
    IF nRecargo # 0
      @ nR+14.5 SAY "RECARGOS : $ " + ALLTRIM(STR(nRecargo))
    ENDIF
    @ nR+32.50 SAY SPACE(1)
    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
RETURN
```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
FUNCTION alinea ()
LOCAL cColor2, aOpcion := {"REINTENTAR","CANCELAR"}, flag := .F.
LOCAL cuadro.ok := "N", nR, i

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 ESTA IMPRIMIENDO EL RECIBO"
    SETCOLOR (color)
    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "PRINTER")
    SETPRC(1.1)
    @ PROW(). PCOL() SAY CsSendPrn3 ("INICIO", nPrn)
    nR := PROW()
    DO WHILE .T.
      @ nR,1 SAY "xxxxxxxxxxxxxxxxxxxx"
      IF ALERT ("ESTA CORRECTAMENTE ALINEADO EL RECIBO",{ "NO","SI" }) == 2
        *@ 7,1 SAY SPACE(1)
        EXIT
      ENDIF
    ENDDO
    SET (_SET_PRINTFILE, ALLTRIM(cPuerto))
    SET (_SET_DEVICE, "SCREEN")
    RESTSCREEN (9.18,16.63, cuadro)
    SET CURSOR ON
    EXIT
  ELSE
    TONE (350)
    nOp := ALERT ("La impresora no esta lista",aOpcion)
    IF nOp == 2
      EXIT
    ENDIF
  ENDIF
ENDDO
RETURN
```

SCE3.PRG

```

#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 )
/* lArea -> Indicará si se desea abrir en un área nueva .T. -> NEW
Por defecto en nueva area
lModo -> .T. = compatido. .F. = exclusivo
*/
LOCAL lControl, cPantal, nVpos, cuadro, color
LOCAL lValor := FALSE // Fichero COMMON.CH

// Comprobación de parámetros. Si no se pasa el nombre de fichero devuelve .F.

IF ValType( cFichero ) != "C"
RETURN lValor
END IF

// Si no se pasa el segundo parámetro 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 área de trabajo asume que se realizará en la primera que se encuentre
// libre, y si se omite el tiempo lo reintentará 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 IValor
END IF

// Para diferenciar si es tiempo indefinido
IControl := IIf( nTiempo == 0 , TRUE , FALSE )

cuadro := SAVESCREEN (9,18,16,63) // Se guarda el area de pantalla donde se
    // desplegaria en caso necesario el menseaje
    // de acceso

// ( Punto 1 )
WHILE nTiempo > 0 .OR. IControl

    DbUseArea( lArea, cFichero, cAlias, lModo )

    IF !NetErr() // Si no hay ningún tipo de error
        IValor := .T.
        EXIT
    ELSE
        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 INTENTANDO ACCEDER A LA RED"
        SETCOLOR (color)
    END IF

    INKEY ( 1 )

    nTiempo--
END WHILE

RESTSCREEN (9,18,16,63.cuadro)

RETURN IValor

// Bloquea y/o intenta bloquear el registro de trabajo
FUNCTION lockr(segun, 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 := segun
  ciclo =(segundos=0)
  DO WHILE (ciclo .OR. segundos>0)
    DISPBOX (9.18.15.62."+++'+++' ".C_REYENO?) // 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
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.
  ENDIF
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

```

SISTEMA DE CUENTAS POR COBRAR PARA INSTITUCIONES EDUCATIVAS

```
// Si se omite el parámetro 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 éxito

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 reintentará el bloqueo
    DO WHILE segundos > 0 .OR. IControl
      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( 1 )
      segundos-- // Decrementamos el parámetro de tiempo pasado
      // Si el bloqueo no tiene éxito se reintentará 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 (550)
    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
```

10.- BIBLIOGRAFÍA

- Baena Paz, Guillermina. *Manual para Elaborar Trabajos de Investigación Documental*, México, 8ª, Ed. Editores Mexicanos Unidos S.A., 1991, 124 pp.
- Fraguas, Alfonso. Marín, Francisco. Quirós, Antonio. Torres, Antonio. *Clipper en Redes de Área Local*, Madrid España, Ed. Ra-Ma, 1994, 658 pp.
- García Badell, José Javier. *Clipper 5.2 a su Alcance*. España, 2ª Ed. Mc Graw Hill, 1994, 844 pp.
- Gillenson, Mark L. *Introducción a las Bases de Datos*. México, Ed. Mc Graw Hill, 1988, 391 pp.
- Hernández Montiel, M. Atziry. *Sistema de Control de Recursos para Obras Públicas del Estado de Michoacán*, Tesis Profesional, Morelia, 1996, 154 pp.
- Jiménez Hernández, Ricardo. *Administración de Centros de Cómputo*, México, 3ª Ed Trillas S.A de C.V., 1991, 267 pp.
- Korth, Henry F. Silberschatz, Abraham. *Fundamentos de Bases de Datos*, 1ª Ed. Mc Graw Hill, México, 1988, 525 pp.
- Lipschutz, Seymour. *Estructura de Datos*, México, Ed. Mc Graw Hill, 1987, 390 pp.
- Marín Quirós, Francisco. Quirós Casado, Antonio. Torres Lozano, Antonio. *Clipper5 Referencia Rápida*, México, Ed. Ra-Ma, 1991, .
- Pressman, Roger S. *Ingeniería del Software*. Un Enfoque Práctico, México, 3ª Ed. Mc Graw Hill, 824 pp.
- Reyes Vargas, Wendy. *Automatización del Sistema de Inscripción de las Sociedades Comerciales en Registro Público de la Propiedad y del Comercio del Estado*, Tesis Profesional, Morelia, 1995, 145 pp.
- Ramalho, José A. *Clipper 5.2 Avanzado*, Avaraca Madrid, 2ª Ed. Mc Graw Hill, 1994, 391 pp.
- Sanders, Donald H. *Informática Presente y futuro*, México, 3ª Ed. Mc Graw Hill, 1990, 887 pp.

Senn, James A. *Análisis y Diseño de Sistemas de Información*, México, 2ª Ed. Mc Graw Hill, 1992, 942 pp.

Senn, James A. *Sistemas de Información para la Administración*, México, Ed. Grupo Editorial Iberoamérica, 1991, 727 pp.

Spence, Rick. *Clipper 5.2 Guía Avanzada para el Programador*, Wilmington, Delaware. E.U.A., Ed. Ra-Ma, 1994, 782 pp.

Straley, Stephen J. *Clipper 5.2*, México, Ed. Limusa S.A. de C.V., 1993, 832 pp.