

Universidad Autónoma del
Estado de Hidalgo
Instituto de Ciencias Básicas e
Ingeniería

“DISEÑO DEL SISTEMA DE CONTROL
DE INGRESOS DE LA EMPRESA
D@tcom Escuela de Computación”

TESIS

Que para obtener el Título de
Licenciado en Computación

PRESENTA:

Juan Miguel Gamboa Miranda

Asesor:

Lic. En Comp. Luís Islas Hernández

Pachuca de Soto, Hgo., Noviembre de 2005

ÍNDICE

| | |
|--|------------|
| Introducción | I |
| Objetivos | II |
| Justificación | III |
| Capítulo 1. Marco Teórico | 1 |
| 1.1.- Introducción | 2 |
| 1.2.- Lenguajes de Programación | 2 |
| 1.3.- Bases de Datos | 3 |
| 1.3.1.- Concepto y ventajas | 4 |
| 1.3.2.- Efecto sinérgico | 4 |
| 1.3.3.- Diferentes estructuras de bases de datos | 5 |
| 1.3.4.- Clasificación de las bases de datos por su uso y Aplicación | 7 |
| 1.3.5.- Componentes de un sistema de bases de datos | 8 |
| 1.4.- Análisis de sistemas | 10 |
| 1.4.1.- Planificación de un proyecto de sistemas | 10 |
| 1.4.1.1.- Proyecto de sistema o de software | 10 |
| 1.4.1.2.- Objetivos de la planificación del proyecto | 11 |
| 1.4.1.3.- Actividades asociadas al proyecto de software | 11 |
| 1.4.1.4.- Recursos | 12 |
| 1.4.1.5.- Estimación del proyecto de software | 14 |
| 1.4.1.6.- Diferentes modelos de estimación | 15 |
| 1.5.- Metodología | 16 |
| 1.5.1.- Concepto de metodología | 16 |
| 1.5.2.- Metodologías para el análisis de sistemas | 17 |
| 1.5.2.1.- Introducción | 17 |
| 1.5.2.2.- Metodología en cascada | 17 |
| 1.5.2.3.- Metodología en espiral | 20 |
| 1.5.2.4.- Etapas del método de prototipos | 22 |
| 1.5.2.5.- Metodología orientada a objetos | 24 |
| Capítulo 2. Situación actual | 26 |
| 2.1.- Descripción de la organización | 27 |
| 2.1.1.- Historia | 27 |
| 2.2.- Áreas de la organización | 27 |
| 2.2.1.- Listado de áreas | 27 |
| 2.2.2.- Descripción de actividades por área | 28 |
| 2.2.3.- Organigrama | 29 |
| 2.3.- Descripción del problema | 30 |
| 2.3.1.- Administración problemática por rezago en la información | 30 |
| Capítulo 3. Fases de la automatización | 31 |
| 3.1.- Análisis | 32 |
| 3.1.1.- Entrevista | 32 |

| | |
|--|------------|
| 3.1.2.- Requerimientos de la empresa de acuerdo a los datos obtenidos en la entrevista | 33 |
| 3.1.3.- Necesidades detectadas en la institución educativa | 33 |
| 3.1.4.- Alternativas para la solución | 34 |
| 3.1.5.- Criterios a evaluar | 35 |
| 3.1.6.- Estudio de factibilidad | 35 |
| 3.1.6.1.- Razones por las que se deba realizar el sistema | 35 |
| 3.1.6.2.- Razones por las que no se deba implantar el sistema | 36 |
| 3.1.6.3.- Propuesta sugerida | 36 |
| 3.1.6.4.- Planeación del proceso de desarrollo | 37 |
| 3.1.6.5.- Determinación de las herramientas a utilizar | 37 |
| 3.1.6.6.- Programa preliminar para el desarrollo | 38 |
| 3.1.6.7.- Estimados preliminares de costo para el desarrollo del sistema | 39 |
| 3.1.6.8.- Estimación costo/beneficio | 39 |
| 3.2.- Diseño | 40 |
| 3.2.1.- Diagramas de flujo | 40 |
| 3.2.1.1.- Diagrama de contexto | 40 |
| 3.2.1.2.- Diagrama de flujo de nivel 0 | 42 |
| 3.2.1.3.- Diagramas de flujo de nivel 1 | 43 |
| 3.2.1.4.- Diagrama entidad/relación | 46 |
| 3.2.2.- Diseño lógico del software | 47 |
| 3.3.- Implementación | 83 |
| 3.4.- Pruebas finales | 84 |
| Conclusiones | 86 |
| Glosario | 88 |
| Apéndice | 96 |
| Bibliografía | 182 |

INTRODUCCIÓN

Este documento tiene como finalidad recolectar y determinar los datos que serán necesarios para poder detectar y solucionar por medio de la sistematización un problema.

Para que dicha solución pueda llevarse a cabo será necesario obtener datos para comenzar a conocer el estado actual de la empresa reconociendo los principales problemas para encontrar una alternativa más viable en cuanto al manejo de la información se refiere.

Los datos necesarios para tal efecto se describen más adelante pero entre ellos podemos mencionar, el giro principal de la empresa o negocio, una entrevista realizada para poder determinar tanto la actividad como las necesidades clave de la empresa, contendrá también las necesidades requeridas en ese momento, los problemas que enfrenta en la actualidad, y un estudio de factibilidad para saber si es necesario implantar el sistema o no.

El análisis preliminar es el punto de partida para iniciar cualquier análisis de un proyecto especificado, de este documento dependen todos los datos realizados más adelante como son el ANÁLISIS, DISEÑO, DESCRIPCIÓN DE PROCEDIMIENTOS, GENERACIÓN DE PRUEBAS, EL CONTROL DE LA CALIDAD, LA IMPLANTACIÓN Y POR ÚLTIMO LA INSTALACIÓN.

Por esta razón el presente documento contiene información importante referente a la empresa y debe ser clara, seria y precisa.

OBJETIVOS

GENERAL

Solucionar problemas empresariales utilizando los conocimientos necesarios de desarrollo de software y la tecnología para el buen empleo de la información.

Específicos

Lograr un ahorro en tiempo y dinero en el manejo de la información dentro de la institución.

Llevar un adecuado control de datos creando un ambiente de certeza en todo momento.

Aumentar la velocidad de acceso a la información evitando tiempos prolongados de espera.

Generar los reportes necesarios para la agilización de tareas administrativas.

Proporcionar una alternativa de crecimiento del proyecto incrementando soluciones no incluidas en el proyecto.

Justificación

Hoy en día las necesidades de información de las empresas crecen a ritmos muy acelerados, esto trae como consecuencia que se tenga que hacer uso de los adelantos tecnológicos para tener un mejor control de la misma.

Las necesidades referidas son tan variadas como la existencia de las empresas mismas.

Este documento presenta una alternativa de solución para la empresa llamada **D@tcom** Escuela de Computación, la cual presenta problemas para el registro de la captación de recursos económicos haciendo que el manejo de la información se vea entorpecida y afectada en los tiempos destinados para la generación de reportes.

Por tal motivo, aquí daremos un enfoque de solución al problema, tratando siempre de que la información sea lo más fiel posible para que su manejo permita generar los documentos necesarios en tiempo y forma, teniendo como consecuencia un ahorro de tiempo y esfuerzo, además de mayor confiabilidad en la información agilizando muchas tareas y toma de decisiones que en otros casos tendrían un retraso por un lapso de hasta 1 mes.

También conviene decir que el uso de las tecnologías incurre en la mayoría de las ocasiones en un mejor manejo de la información y/o en un ahorro de tiempo, no olvidando la fiabilidad de la misma.

Aprovechando la tecnología dispuesta para el control expuesto, se plantea crear un sistema para llevar a cabo la solución de dicho problema.

Capítulo I

Marco Teórico

En éste capítulo encontraremos una descripción de los temas relevantes concernientes a las bases de datos, al análisis de sistemas así como diferentes metodologías.

Estos temas nos ayudarán a crear una imagen mental de lo que se pretende lograr en las secciones siguientes del desarrollo de éste trabajo, es decir, nos servirá de guía para lograr los objetivos a perseguir.

1.1.- Introducción

Antes que nada hablemos un poco del lenguaje de programación que vamos a utilizar a lo largo del proyecto, así como de las bases de datos y del manejador de las bases de datos.

Nuestro proyecto va a utilizar Visual Basic (en su versión 6.0), Microsoft Access® (en su versión 2003) y ODBC.

Integrando éstos tres elementos podremos desarrollar nuestro sistema por completo. Se decide hacer uso de éstos tres elementos debido a que Visual Basic por ejemplo, es un lenguaje de programación bastante flexible para el programador y aunque existen lenguajes mucho más robustos y dinámicos se selecciona por la razón de que no se va a manejar una gran cantidad de información. Por la misma razón se seleccionó Microsoft Access®, que en la base de datos que se va a generar solamente se almacenarán (en números redondos) 300 registros, lo cual es fácil de controlar mediante éste programa. Por otra parte, también se decide hacer uso del manejador ODBC (Administrador de Origen de Datos) porque es compatible con Visual Basic y Microsoft Access® y también es un manejador de bases de datos muy seguro siempre y cuando se integre de manera correcta en el diseño del sistema.

1.2.- Lenguajes de Programación

Antes de empezar un poco de historia, el lenguaje Visual Basic, ha sufrido diferentes transformaciones y cambios desde que apareció la primera versión **Visual Basic 1.0** en el año 1991.

Según ha ido avanzando a lo largo de ésta década, Visual Basic ha ido incorporando características que le han permitido adecuarse a los tiempos hasta llegar al actual Visual Basic 6.0.

Visual Basic 6.0 es un de los programas más populares en el desarrollo de aplicaciones tanto para programadores expertos como para principiantes, dado que se pueden realizar grandes aplicaciones en poco tiempo y su aprendizaje es muy sencillo, esto es debido a que es un producto con una interfaz gráfica de usuario de muy fácil manejo e intuitiva además de poder incorporar de una manera sencilla controles visuales, además de poder integrar imágenes y multimedia.

Este lenguaje está basado en el lenguaje Basic y en la programación orientada a objetos por lo que es un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos.

La palabra “Visual” hace referencia al método que se utiliza para crear la interfaz gráfica de usuario de nuestra aplicación. En lugar de escribir numerosas líneas de código para implementar una interfaz, muchas veces con n aspecto poco vistoso, con Visual Basic únicamente tendremos que hacer click con el ratón arrastrar y soltar los objetos prefabricados al lugar deseado dentro de un formulario.

La palabra “Basic” hace referencia, como hemos citado antes, al lenguaje Basic (Beginners AllPurpose Symbolic Instruction Code, que en castellano significa: Código de Instrucciones Simbólicas de uso General para Principiantes, en una traducción un tanto libre). Visual Basic ha evolucionado desde este lenguaje y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows, ésto es debido a que el fabricante es el mismo, Microsoft.

Es importante hacer notar el hecho de que se pueda reutilizar código de aplicaciones escritas en ASP, Visual C++, Visual J++, Delphi, o cualquier otro lenguaje de programación compatible con la tecnología COM de Microsoft, hace muy fácil trasladar aplicaciones ya existentes en una empresa. Es más, las herramientas de Office Microsoft Excel y Microsoft Access se pueden programar (macros) mediante este lenguaje de programación.

Además, Visual Basic le permite crear aplicaciones para uso personal, para un grupo de trabajo, para una empresa, aplicaciones distribuidas a través de Internet, aplicaciones de Bases de Datos y otras muchas que usted pueda imaginar.

La ventaja principal de este lenguaje de programación es su sencillez para programar aplicaciones de cierta complejidad para Windows y sus desventajas son la necesidad de archivos adicionales además del ejecutable y cierta lentitud en comparación con otros lenguajes aunque hoy en día éste último factor es cada vez menos determinante debido a la gran potencia de los ordenadores. Es obligado decir sin embargo, que sigue siendo BASIC. No se pueden comparar sus prestaciones con otros lenguajes cuando deseamos llegar al fondo de la máquina y controlar uno a uno sus registros. No es ese el fin perseguido con VB y si es necesario llegar a esas precisiones será necesario utilizar otro lenguaje que permita bajar el nivel de programación. (Visual-C). o realizar librerías (DLLs) que lo hagan. En la mayor parte de las aplicaciones, las herramientas aportadas por VB son más que suficiente para lograr un programa fácil de realizar y de altas prestaciones¹.

1.3.- Bases de datos

Las bases de datos surgen desde mediados de los años sesenta la historia de las bases de datos, en 1970 Codd propuso el modelo relacional, este modelo es el

¹ Francisco Javier Ceballos, 2000, Enciclopedia de Microsoft Visual Basic 6, Ed. Mc Graw-Hill

que ha marcado la línea de investigación por muchos años, ahora se encuentran los modelos orientados a objetos.

1.3.1.- Concepto y ventajas

Base de Datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo².

Ventajas del uso de la base de datos en la organización

1. Independencia de datos y tratamiento.
Cambio en datos no implica cambio en programas y viceversa (Menor coste de mantenimiento).
2. Coherencia de resultados.
Reduce redundancia :
 - Acciones lógicamente únicas.
 - Se evita inconsistencia.
3. Mejora en la disponibilidad de datos
 - No hay dueño de datos (No igual a ser públicos).
 - Ni aplicaciones ni usuarios.
 - Guardamos descripción (Idea de catálogos).
4. Cumplimiento de ciertas normas.
Restricciones de seguridad:
 - Accesos (Usuarios a datos).
 - Operaciones (Operaciones sobre datos).
5. Otras ventajas:
 - Más eficiente gestión de almacenamiento.

1.3.2.- Efecto sinérgico

Nivel Físico.

Es el nivel real de los datos almacenados, es decir, cómo se almacenan los datos, ya sea en registros, o como sea. Este nivel es usado por muy pocas personas que deben estar cualificadas para ello. Este nivel lleva asociada una representación de los datos, que es lo que denominamos Esquema Físico.

Nivel Conceptual.

Es el correspondiente a una visión de la base de datos desde el punto de vista del mundo real. Es decir tratamos con la entidad u objeto representado, sin importarnos como está representado o almacenado. Este nivel lleva asociado el Esquema Conceptual.

² Página dedicada a las bases de datos – <http://www.dbinternet.com.ar/metodo.htm>

Nivel Visión.

Son partes del esquema conceptual. El nivel conceptual presenta toda la base de datos, mientras que los usuarios por lo general sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas. Un ejemplo sería el caso del empleado que no tiene por qué tener acceso al sueldo de sus compañeros o de sus superiores. El esquema asociado a éste nivel es el Esquema de Visión.

Los 3 niveles vistos, componen lo que conocemos como arquitectura de base de datos a 3 niveles.

A menudo el nivel físico no es facilitado por muchos DBMS (Sistemas Manejadores de Bases de Datos), esto es, no permiten al usuario elegir como se almacenan sus datos y vienen con una forma estándar de almacenamiento y manipulación de los datos.

La arquitectura a 3 niveles se puede representar³ como sigue:

- Subesquema de Visión, Subesquema de Visión, Subesquema de Visión, ...
- Esquema Conceptual
- Esquema Físico

1.3.3.- Diferentes Estructuras De Bases De Datos

Modelos de Datos

Para representar el mundo real a través de esquemas conceptuales se han creado una serie de modelos:

1. Mundo Real Esquemas Conceptuales
2. Modelización

Existen multitud de estos modelos que se conocen como Modelos de Datos, algunos de estos modelos son:

- Modelo Relacional de Datos
- Modelo de Red
- Modelo Jerárquico

Modelo Relacional de Datos.

Representa al mundo real mediante tablas relacionadas entre sí por columnas comunes. Ejemplo:

| Num_empleado | Nombre | Sección |
|---------------------|---------------|----------------|
| 33 | Pepe | 25 |
| 34 | Juan | 25 |

³ Apuntes, cursos, entre otros – <http://www.lafacu.com>

| Num_sección | Nombre |
|-------------|---------|
| 25 | Textil |
| 26 | Pintura |

Modelo de Red.

Representamos al mundo real como registros lógicos que representan a una entidad y que se relacionan entre sí por medio de flechas. Ejemplo:

33 Pepe 25 Textil

34 Juan

Modelo Jerárquico.

Tiene forma de árbol invertido. Un padre puede tener varios hijos pero cada hijo sólo puede tener un padre. Ejemplo:

Empresa

25 Textil 26 Pintura 27 Construcción

33 Pepe 34 Juan 36 Perico

Al llegar a este punto adoptaremos una convención con respecto a las restricciones que se pueden dar en un modelo, en cuanto a la capacidad de relacionarse que pueda tener cada entidad de ese modelo con el resto de entidades.

Como se ha dicho el modelo jerárquico sólo admite relaciones 1 : 1 ó 1 : N.

En caso de que tuviésemos la necesidad de otro tipo de asociaciones y queramos usar el esquema jerárquico, recurriríamos a una duplicación de la información en el esquema, pero sólo a nivel esquemático.

Ejemplo:

Tenemos dos entidades (cliente y cuenta), queremos que un cliente pueda poseer varias cuentas, y que una cuenta pueda tener como titulares a varios clientes. Usando el modelo jerárquico tendríamos que recurrir a una duplicación de los datos en el esquema⁴.

Cliente Cuenta

Cuenta Cliente

Como ya se ha dicho esta duplicación sólo ocurriría a nivel esquemático, pero a nivel físico existiría un único registro por cliente, y por cuenta que relacionaríamos mediante varios punteros con todo lo necesario.

En el modelo de red no existen restricciones, si queremos representar que un cliente puede tener varias cuentas, cada una de las cuáles sólo puede tener un titular, y cada cuenta ésta en una sola sucursal, que por supuesto puede ser compartida por varias cuentas, éste sería el esquema:

⁴ Página especializada en bases de datos – <http://www.ur.mx/ur/faciya/carreras/cursos/sis>

Cliente Cuenta Sucursal

Con el modelo relacional podríamos tener ambas entidades definidas de la siguiente forma:

Cliente = (Nº Cliente: Acceso Principal; Nombre, Dirección, Nº Cuenta: Acceso Ajeno)

Cuenta = (Nº Cuenta: Acceso Principal; Saldo)

Se podría hablar de otro modelo que está un nivel más arriba y que se denomina Modelo Entidad-Relación (E/R). Este modelo sólo realiza el diseño, no realiza la implementación, por lo tanto, una vez hecho el diseño se puede llevar al modelo relacional, de red o jerárquico. Define dos entidades y una forma de relación entre ellas.

Ejemplo:

Nombre Dcción Num_Cli Num_cuenta Saldo

CLIENTE Cliente - CUENTA

Cuenta

Entidad Relación Entidad

1.3.4.- Clasificación de las bases de datos por su uso y aplicación

TIPOS DE MODELOS DE DATOS

Existen fundamentalmente tres alternativas disponibles para diseñar las bases de datos: el modelo jerárquico, el modelo de red y el modelo relacional. **MODELO JERARQUICO.** Puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.

MODELO DE RED. Este modelo permite la representación de muchos a muchos, de tal forma que cualquier registro dentro de la base de datos puede tener varias ocurrencias superiores a él. El modelo de red evita redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector.

MODELO RELACIONAL. Este modelo se está empleando con más frecuencia en la práctica, debido a la ventajas que ofrece sobre los dos modelos anteriores, entre ellas, el rápido entendimiento por parte de usuarios que no tienen conocimientos profundos sobre Sistemas de Bases de Datos.

BASES DE DATOS DISTRIBUIDAS. Las bases de datos distribuidas se están utilizando cada vez más en la misma medida en que se usan las arquitecturas de cliente-servidor y groupware. Los principales problemas que se generan por el uso de la tecnología de bases de datos distribuidas son en lo referente a duplicidad de datos y a su integridad al momento de realizar actualizaciones a los mismos. Además, el control de la información puede constituir una desventaja, debido a que se encuentra diseminada en diferentes localidades geográficas⁵.

⁵ Página especializada en bases de datos – <http://www.ur.mx/ur/faciya/carreras/cursos/sis/>

TENDENCIAS FUTURAS. En el futuro la mayoría de las organizaciones cambiarán la forma convencional de manejo de la información a la arquitectura de base de datos a las ventajas derivadas de su uso. El uso de las bases de datos distribuidas se incrementará de manera considerable en la medida en que la tecnología de comunicación de datos brinde más facilidades para ello. El uso de bases de datos facilitará y soportará en gran medida a los Sistemas de Información para la Toma de Decisiones.

1.3.5.- Componentes de un Sistema de Base de Datos

Recursos que componen un SBD

Un SBD estará formado por:

- Personas
- Máquinas
- Programas
- Datos

Descripción de las funciones de cada componente de un sistema de base de datos.

Los datos.

Es lo que se conoce como base de datos propiamente dicha. Para manejar estos datos utilizamos una serie de programas.

Los Programas.

Son los encargados de manejar los datos, son conocidos como DBMS (Data Base Management System) o también SGBD (Sistema Gestor de Base de Datos). Los DBMS tienen dos funciones principales que son:

- La definición de las estructuras para almacenar los datos.
- La manipulación de los datos.

Descripción de los sistemas de procesamiento de archivos

La información necesita ser guardada y manipulada para que sea útil y este es uno de los objetivos principales que han tenido las computadoras dentro de la empresa.

A principios de su incorporación al ámbito empresarial, las computadoras trabajaban con lo que se conoce como "Sistema de procesamiento de Archivos" en los que se guardaban los datos para ser procesados por programas escritos especialmente para cada clase de archivos; esto degeneraba en un sistema monolítico y de difícil mantenimiento.

Sistema de procesamiento de archivos.

Tiene una serie de inconvenientes que son reducidos en los SGBD:

- Dificultad de Acceso a ciertos datos o información.
- Si no existen programas para acceder o calcular cierta información, no puede accederse a ella. Ej.: Calcular totales, o registros con cierta condición...

Aislamiento de datos.

Los datos pueden estar en varios archivos con distintos formatos, que complican la creación de programas nuevos.

- Falta de Integridad:

Es complicado mantener ciertas condiciones en la información. Ejemplo: Que el saldo sea superior a cierta cantidad, que un empleado no esté adscrito a un número de departamento que no exista...

- Problemas de Atomicidad en las operaciones:

A veces es esencial que para la consistencia de la BD se efectúen varias operaciones como si fueran una única operación, evitando que se produzcan fallos en medio de dicha operación. Ejemplo: En una transferencia bancaria se da el caso de quitar dinero de una cuenta y añadirlo a la otra.

Problemas en el acceso concurrente.

Si varios usuarios acceden a la vez a un dato pueden producirse errores.

Ejemplo: Si se saca dinero de una misma cuenta desde dos sitios distintos.

Problemas de seguridad.

Dificultad para controlar que ciertos usuarios no accedan a ciertos datos.

Evolución histórica.

Viendo la necesidad de mejorar este estándar se desarrollaron los "Sistemas gestores de Bases de Datos Relacionales" (SGBDR) cuyas características hacen al sistema mucho más eficiente que los sistemas de manejo de archivos.

Algunas de las características son que existe solo una copia de los datos para que todos los programas trabajen con ella, esto es lo que se denomina obtención de redundancia mínima y de esta manera se podrá eliminar la inconsistencia de los datos.

La capacidad de interactuar en un ambiente cliente/servidor donde los clientes o usuarios (ya sea de una intranet o desde Internet) pudiendo trabajar con un conjunto único de datos alojados en un servidor y donde varios clientes podrían estar trabajando al mismo tiempo.

Estas son solo algunas de las características que tiene el modelo de base de datos relacional y en Linux existen diversos motores de base de datos que permiten trabajar ya sea con base de datos existentes o creando nuevas con todas las capacidades de trabajo en red de este sistema operativo.

Numerosas empresas se han volcado al desarrollo de SGBDR's como Oracle e Informix pero la más conocida por su carácter gratuito para el uso personal y la inclusión en numerosas distribuciones es el PostgreSQL.

En un principio adoptado para el sistema UNIX, fue portado a Linux y modificado para utilizar el lenguaje de consultas SQL, donde se llamo Posgres95.

Mejorado en el 96 tomo su nuevo PostgreSQL y adoptó las nuevas características del SQL - 92.

Actualmente posee ciertas limitaciones técnicas como el no permitir consultas paralelas, es decir, que no pueden dos usuarios consultar una misma tabla al mismo tiempo, lo que impide su adopción en un ambiente de trabajo de grandes dimensiones. Pero el uso del lenguaje de consulta SQL lo hace especial para ambientes de trabajo que cuente con múltiples plataformas y también así para su utilización para ser consultado directamente desde internet por medio de una página Web.

PostgreSQL requiere de cierto conocimiento del lenguaje SQL por lo que es recomendable el estudio de bibliografía que trate este tema antes de trabajar en el diseño de una base de datos con él.

No estaría bien terminar sin nombrar otros sistemas de base de datos como el Sybase SQL Anywhere Studio de la empresa Sybase cuya versión beta se encuentra disponible en el web site de la compañía www.sybase.com/products/anywhere/linuxbeta.

Otro conocido en el campo de las bases de datos es Oracle y el portar Oracle 8 a Linux fue un gran avance que da solidez a este sistema operativo para su adopción en las grandes y medianas empresas.

Descripción de sistemas de procesamiento en línea y por lotes

Un sistema de procesamiento por lotes (batch) ejecuta trabajos (jobs).

En un sistema de procesamiento por lotes, organizar la memoria en particiones fijas es simple y efectivo: cada trabajo se carga en la memoria cuando le toque su turno, y se queda en memoria hasta que termine. Si hay suficientes particiones para mantener a la CPU siempre ocupada, no habría razones para usar nada diferente.

Procesamiento en línea: A diferencia del procesamiento por lotes o batch, el procesamiento en línea implica que los programas se ejecuten de tal forma que los datos se actualicen de inmediato en los archivos de la computadora. A este tipo de procesamiento se le conoce también como tiempo real.

Las aplicaciones de tiempo real son indispensables en aquellos casos en que los datos contenidos en los archivos se modifican varias veces en el transcurso de un día y se consultan en forma casi inmediata con las modificaciones que se efectuaron. Un ejemplo de lo anterior es un sistema de reservaciones en alguna línea aérea⁶.

1.4.- Análisis de sistemas

1.4.1.- Planificación de un proyecto de sistemas.

1.4.1.1.- Proyecto de Sistema o Software

Es el proceso de gestión para la creación de un sistema o software, la cual encierra un conjunto de actividades, una de las cuales es la estimación, estimar es

⁶ Piattini Mario, Adoración de Miguel, Marcos Esperanza

echar un vistazo al futuro y aceptamos resignados cierto grado de incertidumbre. Aunque la estimación, es más un arte que una Ciencia, es una actividad importante que no debe llevarse a cabo de forma descuidada. Existen técnicas útiles para la estimación de costes de tiempo. Y dado que la estimación es la base de todas las demás actividades de planificación del proyecto y sirve como guía para una buena ingeniería, sistemas y software.

Al estimar tomamos en cuenta no solo del procedimiento técnico a utilizar en el proyecto, sino que se toman en cuenta los recursos, costos y planificación. El tamaño del proyecto es otro factor importante que puede afectar la precisión de las estimaciones. A medida que el tamaño aumenta, crece rápidamente la interdependencia entre varios elementos del software.

La disponibilidad de información histórica es otro elemento que determina el riesgo de la estimación.

1.4.1.2.- Objetivos de la Planificación del Proyecto.

El objetivo de la planificación del proyecto de software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

El Objetivo de la planificación se logra mediante un proceso de descubrimiento de la información que lleve a estimaciones razonables.

1.4.1.3.- Actividades asociadas al proyecto de software.

Ámbito del Software.

Es la primera actividad de llevada a cabo durante la planificación del proyecto de software.

En esta etapa se deben evaluar la función y el rendimiento que se asignaron al software durante la ingeniería del sistema de computadora para establecer un ámbito de proyecto que no sea ambiguo, e incomprensible para directivos y técnicos

Describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad, se evalúan las funciones del ámbito y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación. Las restricciones de rendimiento abarcan los requisitos de tiempo de respuesta y procesamiento, identifican los límites del software originados por el hardware externo, por la memoria disponible y por otros sistemas existentes.

El Ambito se define como un pre-requisito para la estimación y existen algunos elementos que se debe tomar en cuenta como es:

- La Obtención de la Información necesaria para el software. Para esto el analista y el cliente se reúnen sobre las expectativas del proyecto y se ponen de acuerdo en los puntos de interés para su desarrollo.

1.4.1.4.- Recursos

La segunda tarea de la planificación del desarrollo de software es la estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de software, esto simula a una pirámide donde las Herramientas (hardware y software), son la base que proporciona la infraestructura de soporte al esfuerzo de desarrollo, en segundo nivel de la pirámide se encuentran los componentes reutilizables. Y, en la parte mas alta de la pirámide se encuentra el recurso primario, las personas (el recurso humano).

Cada recurso queda especificado mediante cuatro características:

- Descripción del Recurso.
- Informes de disponibilidad.
- Fecha cronológica en la que se requiere el recurso.
- Tiempo durante el que será aplicado el recurso.

Recursos Humanos.

La Cantidad de personas requeridas para el desarrollo de un proyecto de software solo puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo, personas mes o personas años), y seleccionar la posición dentro de la organización y la especialidad que desempeñará cada profesional.

Recursos o componentes de software reutilizables.

Cualquier estudio sobre recursos de software estaría incompleto sin estudiar la reutilización, esto es la creación y la reutilización de bloques de construcción de software.

Tales bloques se deben establecer en catálogos para una consulta más fácil, estandarizarse para una fácil aplicación y validarse para la también fácil integración.

El Autor Bennatan sugiere cuatro categorías de recursos de software que se deberían tener en cuenta a medida que se avanza con la planificación:

- Componentes ya desarrollados.
- Componentes ya experimentados.
- Componentes con experiencia parcial.
- Componentes nuevos.

Recursos de entorno.

El entorno es donde se apoya el proyecto de Software, llamado a menudo entorno de ingeniería de software, incorpora hardware y software.

El hardware proporciona una plataforma con las herramientas (software) requeridas para producir los productos que son el resultado de la buena práctica de la ingeniería del software, un planificador de proyectos debe determinar la ventana temporal requerida para el hardware y el software, y verificar que estos recursos estén disponibles. Muchas veces el desarrollo de las pruebas de validación de un proyecto de software para la composición automatizada puede necesitar un compositor de fotografías en algún punto durante el desarrollo. Cada elemento de hardware debe ser especificado por el planificador del proyecto de software⁷.

⁷ Victor Gerez y Manuel Grijalva, El enfoque de sistemas, Limusa, 1980, México

1.4.1.5.- Estimación del proyecto de software.

En el principio el costo del Software constituía un pequeño porcentaje del costo total de los sistemas basados en Computadoras. Hoy en día el Software es el elemento más caro de la mayoría de los sistemas informáticos.

Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y perdidas, la estimación del costo y del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables: humanas, técnicas, de entorno, políticas, que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo.

Para realizar estimaciones seguras de costos y esfuerzos tienen varias opciones posibles:

- Deje la estimación para más adelante (obviamente podemos realizar una estimación al cien por cien fiable después de haber terminado el proyecto.
- Base las estimaciones en proyectos similares ya terminados.
- Utilice técnicas de descomposición relativamente sencillas para generar las estimaciones de costos y esfuerzo del proyecto.
- Desarrolle un modelo empírico para el cálculo de costos y esfuerzos del software.

Desdichadamente la primera opción, aunque atractiva no es práctica.

La Segunda opción puede funcionar razonablemente bien si el proyecto actual es bastante similar a los esfuerzos pasados y si otras influencias del proyecto son similares. Las opciones restantes son métodos viables para la estimación del proyecto de software. Desde el punto de vista ideal, se deben aplicar conjuntamente las técnicas indicadas usando cada una de ellas como comprobación de las otras.

Antes de hacer una estimación, el planificador del proyecto debe comprender el ámbito del software a construir y generar una estimación de su tamaño.

Estimación basada en el proceso.

La técnica más común para estimar un proyecto es basar la estimación en el proceso que se va a utilizar, es decir, el proceso se descompone en un conjunto

relativamente pequeño de actividades o tareas, y en el esfuerzo requerido para llevar a cabo la estimación de cada tarea.

Al igual que las técnicas basadas en problemas, la estimación basada en el proceso comienza en una delineación de las funciones del software obtenidas a partir del ámbito del proyecto. Se mezclan las funciones del problema y las actividades del proceso. Como último paso se calculan los costos y el esfuerzo de cada función y la actividad del proceso de software.

1.4.1.6.- Diferentes modelos de estimación.

Existen diferentes modelos de estimación como son:

Los Modelos Empíricos:

Donde los datos que soportan la mayoría de los modelos de estimación obtienen una muestra limitada de proyectos. Por esta razón, el modelo de estimación no es adecuado para todas las clases de software y en todos los entornos de desarrollo. Por lo tanto los resultados obtenidos de dichos modelos se deben utilizar con prudencia.

El Modelo COCOMO.

Barry Boehm, en su libro clásico sobre economía de la Ingeniería del Software, introduce una jerarquía de modelos de estimación de software con el nombre de COCOMO, por su nombre en inglés (COConstructive, COSt, MOdel) modelo constructivo de costos. La jerarquía de modelos de Boehm esta constituida por los siguientes:

- **Modelo I.** El Modelo COCOMO básico calcula el esfuerzo y el costo del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas.
- **Modelo II.** El Modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de conductores de costos que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- **Modelo III.** El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costos en cada caso (análisis, diseño, etc.) del proceso de ingeniería de software⁸.

⁸ Roger S. Presuman, Ingeniería del software, Ed. Mc Graw-Hill, 4ª. Edición

Herramientas Automáticas De Estimación.

Las herramientas automáticas de estimación permiten al planificador estimar costos y esfuerzos, así como llevar a cabo análisis del tipo, que pasa si, con importantes variables del proyecto, tales como la fecha de entrega o la selección del personal. Aunque existen muchas herramientas automáticas de estimación, todas exhiben las mismas características generales y todas requieren de una o más clases de datos.

A partir de estos datos, el modelo implementado por la herramienta automática de estimación proporciona estimaciones del esfuerzo requerido para llevar a cabo el proyecto, los costos, la carga de personal, la duración, y en algunos casos la planificación temporal de desarrollo y riesgos asociados.

En resumen el planificador del proyecto de software tiene que estimar tres cosas antes de que comience el proyecto: cuánto durará, cuánto esfuerzo requerirá y cuanta gente estará implicada. Además el planificador debe predecir los recursos de hardware y software que va a requerir y el riesgo implicado.

Para obtener estimaciones exactas para un proyecto, generalmente se utilizan al menos dos de las tres técnicas referidas anteriormente. Mediante la comparación y la conciliación de las estimaciones obtenidas con las diferentes técnicas, el planificador puede obtener una estimación más exacta. La estimación del proyecto de software nunca será una ciencia exacta, pero la combinación de buenos datos históricos y técnicas puede mejorar la precisión de la estimación.

1.5.- Metodología

1.5.1.- Concepto de metodología

Metodología. Conjunto de métodos empleados para el desarrollo de sistemas automatizados.

Una metodología completa es algo más que una notación, un proceso, y herramientas. Además de una notación, de un proceso, y de herramientas, éstas metodologías completas proporcionan:

- Guías para estimar costos,
- Manejo del proyecto en las tareas y entregas,
- Medidas y métricas,
- Formas definidas y dirección en las entregas de la construcción,
- Políticas y procedimientos para garantizar la calidad del software,
- Descripciones de los roles y programas de entrenamiento detallados,
- Ejemplos totalmente trabajados,
- Ejercicios de entrenamiento,

- Técnicas para adaptar el método, y técnicas definidas

1.5.2.- Metodologías para el análisis de sistemas

1.5.2.1.- Introducción.

Para el desarrollo de cualquier producto de software se realizan una serie de tareas entre la idea inicial y el producto final. Ese desarrollo sigue una determinada metodología o modelo de desarrollo. Un modelo de desarrollo establece el orden en el que se harán las tareas en el proyecto y nos provee de requisitos de entrada y salida para cada una de las actividades.

El desarrollo de este proyecto se ha basado en un modelo de desarrollo en espiral basado en prototipos. Se ha optado por esta metodología ya que se adapta considerablemente al desarrollo que sigue una aplicación del sistema como la que se pretende desarrollar, en la cuál vamos basándonos en prototipos que se desarrollan apoyándose en el anterior de forma incremental y en la que además se pretende probar una arquitectura o tecnología.

La gran ventaja de este modelo de desarrollo que se considera de vital importancia es la de tener puntos de control en cada iteración. Además el modelo es muy flexible a requisitos cambiantes, lo que es muy propio que se de en productos software de las características que se pretenden desarrollar.

Por lo tanto, en este capítulo se pretende exponer el desarrollo de este proyecto según esta metodología.

1.5.2.2.- Metodología en cascada (ó clásico)

Propuesto por Royce en 1970. Este modelo admite iteraciones. Después de cada etapa se realiza una revisión para comprobar si se puede pasar a la siguiente. Trabaja en base a documentos (SRD, SDD...)

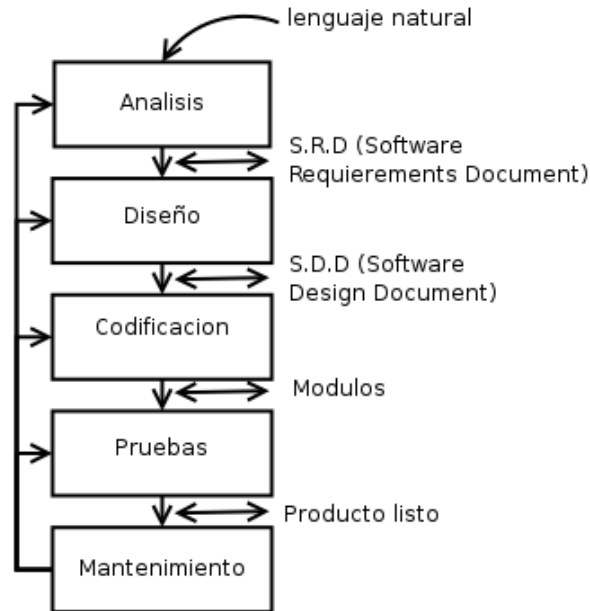


Figura 1.- Ciclo de vida de la metodología en cascada.

Ventajas

- Planificación sencilla
- Calidad del producto alta
- Permite trabajar con personal poco cualificado

Inconvenientes

- Necesidad de tener todos los requisitos al principio
- Si se han cometido errores en una fase es difícil volver atrás
- No se dispone del producto hasta el final
- Síndrome del 90%
- Más lento y más caro que los demás

Tipos de proyecto para los que es adecuado

- Los que se dispone de todas las especificaciones desde el principio
- Se desarrolla un tipo de producto que no es novedoso
- Proyectos complejos que se entienden bien desde el principio

Metodología en cascada con subproyectos

El sistema se divide en subsistemas independientes que se pueden llevar en paralelo.

La ventaja es que se puede tener más gente trabajando en paralelo, pero la desventaja es que pueden surgir interdependencias entre proyectos⁹.

Cada proyecto tiene sus propias fechas.

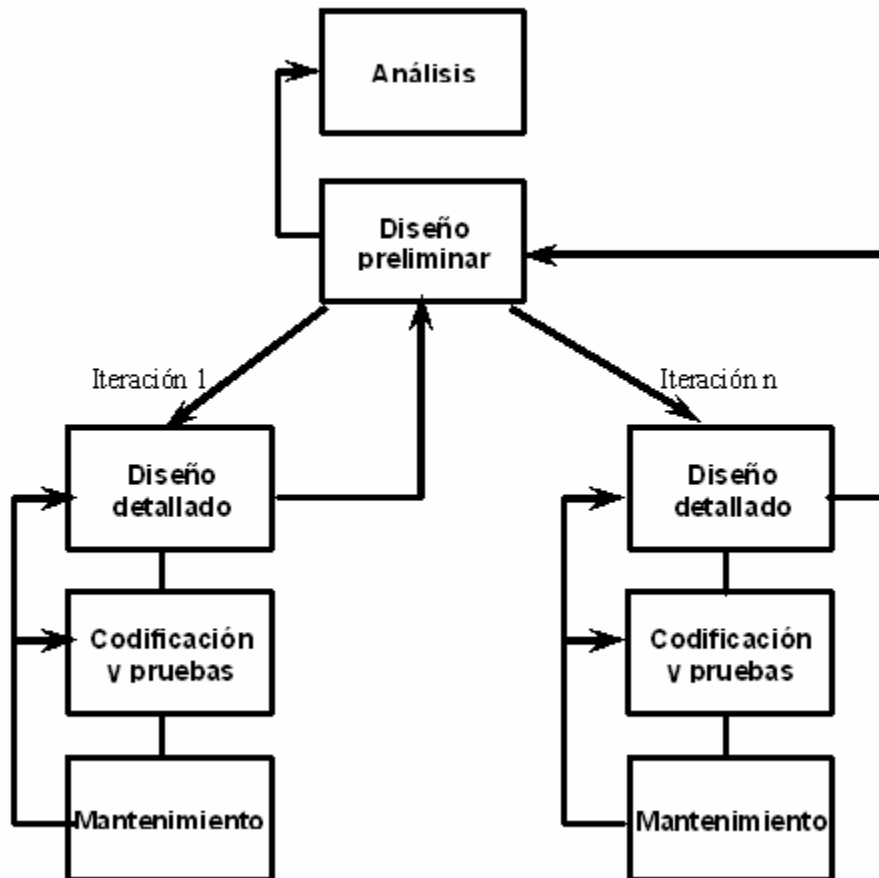


Figura 2.- Ciclo de vida en cascada con subproyectos.

⁹ James A. Senn, Análisis y diseño de sistemas de información, Ed. Mc Graw-Hill

Metodología en cascada incremental

Se realiza un ciclo de vida en cascada para cada pequeña funcionalidad.

Ventaja

- No es necesario disponer de todos los requisitos al principio

Desventaja

- Los errores de detección de requisitos aparecen tarde.

Metodología en cascada con reducción de riesgos

Para evitar los problemas en el entendimiento de los requisitos se puede hacer un desarrollo interactivo durante las fases de análisis y diseño global.

1. Preguntar al usuario
2. Hacer el diseño global que depende del punto 1
3. Hacer un prototipo de interfaz de usuario, hacer entrevistas con los usuarios enseñándoles el prototipo y volver con ello al punto 1 para identificar más requisitos o corregir mal entendidos.
4. A partir de aquí igual al ciclo de vida en cascada

1.5.2.3.- Metodología en espiral

Este modelo permite el desarrollo de software mediante etapas o niveles los cuales aumentan su grado de refinamiento cada vez más para así permitir un avance continuo y evolutivo (ver fig. 3). En este modelo se produce una interacción con el cliente durante todo el proceso de desarrollo posibilitando el mayor entendimiento de los requisitos.

En este modelo existen cuatro actividades principales:

Planificación: En esta etapa se realiza la determinación de requisitos, se proponen alternativas y se señalan las restricciones.

Análisis de riesgo: Evaluar alternativas mediante el uso de prototipos para determinar el mejor desempeño y la manera más adecuada para desarrollar una función.

Ingeniería: Se desarrolla el siguiente nivel considerando los riesgos, es aquí donde se aplican todas aquellas técnicas ingenieriles para asegurar la calidad.

Evaluación del cliente: El prototipo generado es evaluado por el cliente quien señala todas sus apreciaciones sobre éste.

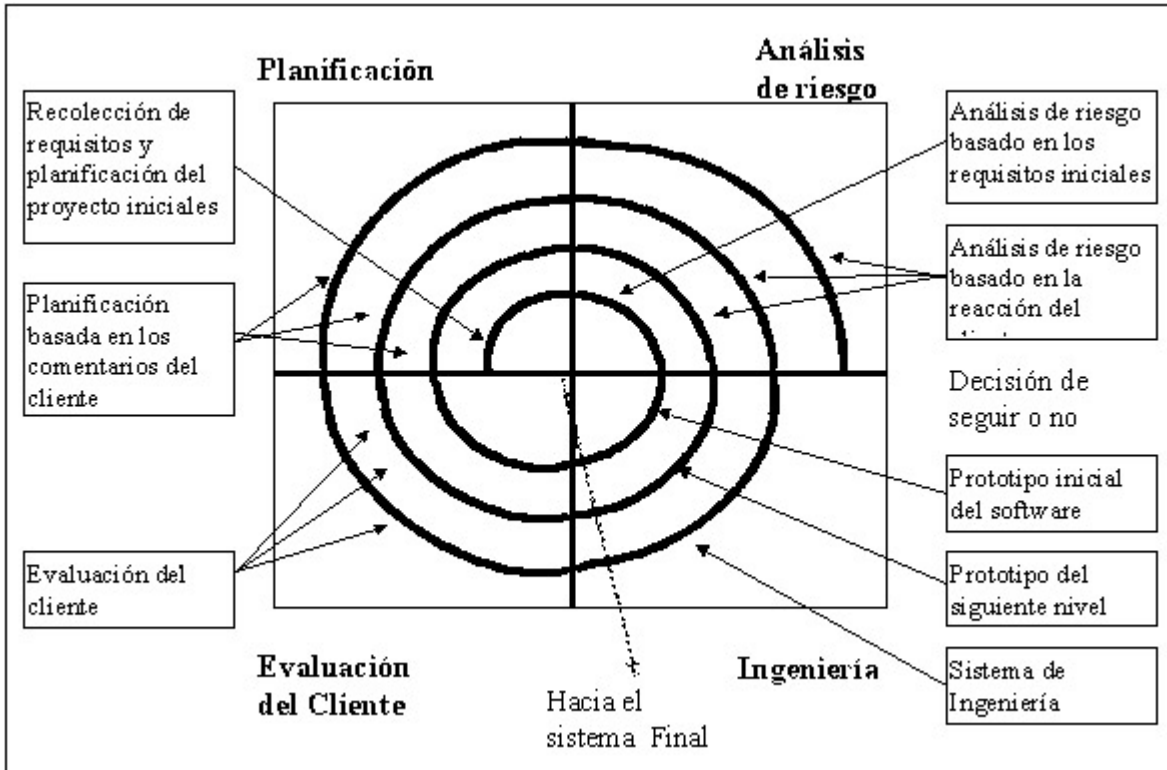


Figura 3: *Modelo en espiral.*

Este modelo es el enfoque más realista para el desarrollo de software y sistemas a gran escala. Emplea un enfoque evolutivo, permitiendo al desarrollador y al cliente entender y reaccionar a las alternativas de cada nivel. Por esto el modelo en espiral será escogido para enmarcar la metodología a desarrollar ya que estas características facilitan la aplicación de técnicas multimediales que requieren de muchas pruebas y evaluaciones paulatinas a medida que el software va tomando forma.

Por último y como el modelo de desarrollo en espiral marca, se lleva a cabo un plan de elaboración de la siguiente fase. Este plan se elabora en base a los resultados obtenidos en las pruebas anteriores. Si los resultados han sido satisfactorios, la siguiente fase se planteará como un incremento de la funcionalidad del sistema, en caso contrario se planteará como un incremento de la robustez y consistencia del mismo¹⁰.

¹⁰ Universidad Autónoma de Sonora, <http://www.uas.mx/cursoswebct/Progsist/material.htm>

1.5.2.4.-Etapas del Método de prototipos

1.- Identificación de requerimientos conocido.

La determinación de los requerimientos de una aplicación es tan importante para el método de desarrollo de prototipo como lo es para los métodos del ciclo clásico de desarrollo de sistemas o análisis estructurado (aunque las tácticas son diferentes). Por consiguiente, antes de crear el prototipo, los analistas y usuarios deben trabajar juntos para identificar los requerimientos conocidos que tiene que satisfacerse. Para hacerlo determinan los fines para lo que servirá el sistema y el alcance de sus capacidades.

2.- Desarrollo de un modelo de trabajo

Es útil comenzar el proceso de construcción del prototipo con el desarrollo de un plan general que permita a las personas conocer lo que se espera de ellas y del proceso de desarrollo. Es difícil, y en ocasiones imposibles, fijar una fecha tentativa de terminación. La experiencia con el sistema es la que determina eventualmente cuando el sistema esta terminado.

Para comenzar la primera iteración, usuarios y analistas identifican de manera conjunta los datos que son necesarios para el sistema y especifican la salida que debe producir la aplicación.

Las decisiones de diseño necesarias para desarrollar la salida del sistema cambian muy poco en relación con las tomadas en otros métodos de desarrollo. Sin embargo, con un prototipo, se espera que las especificaciones iniciales estén incompletas.

En el desarrollo de un prototipo se preparan los siguientes componentes:

- El lenguaje para el diálogo o conversación entre el usuario y el sistema
- Pantallas y formato para la entrada de datos
- Módulos esenciales de procesamiento
- Salida del sistema

Al construir el prototipo se deben seguir los estándares para datos que emplea la organización.

En esta etapa es más importante la rapidez con que se construye el prototipo que la eficiencia de operación. Es por esto que el analista no intenta optimizar la velocidad de operación del sistema (ver fig. 4).

3.- El prototipo y el usuario

Es responsabilidad del usuario trabajar con el prototipo y evaluar su característica y operación. La experiencia con el sistema bajo condiciones permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios así como la eliminación de características inadecuadas o innecesarias.

4.- Revisión del prototipo

Información sobre lo que les gusta y lo que les desagrada a los usuarios. La información obtenida tendrá influencia sobre las características de la siguiente versión de la aplicación.

Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo. El analista es el responsable de realizar las modificaciones.

5.- Repetición del proceso las veces que sea necesario.

El proceso finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando ya es evidente que no se obtendrá mayor beneficio.

6.- El abandono o dejarlo como está.

Cuando se verifica que no es posible desarrollar el sistema para satisfacer los objetivos deseados, ya sea por la tecnología existente o por el factor económico¹¹.

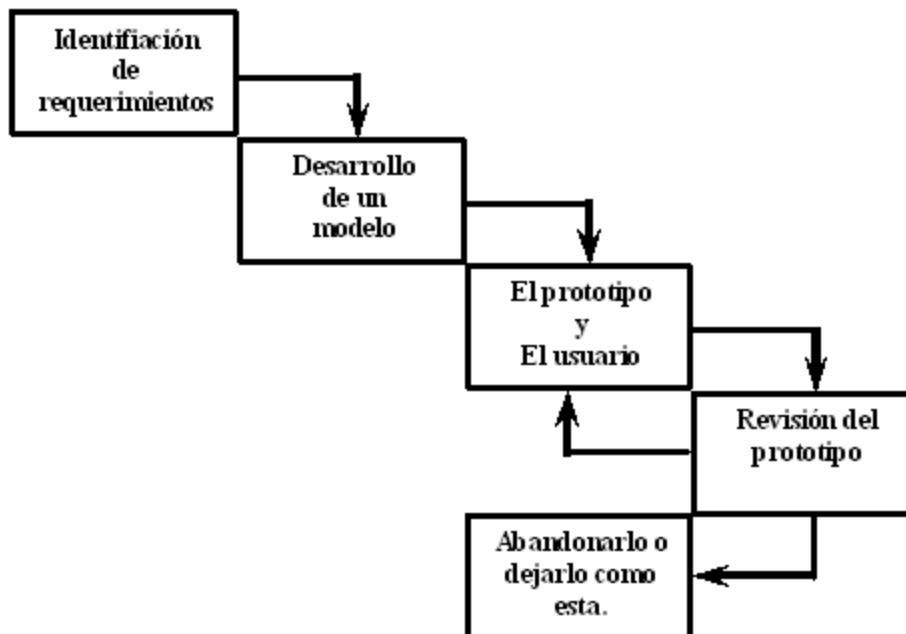


Figura 4.-

Metodología de diseño de prototipos

¹¹ Página dedicada a los sistemas de información, <http://elizabethpeguero.8m.com/eliza.htm>

1.5.2.5.- Metodología orientada a objetos

Aproximación posterior, se supone que sustituirá a la anterior debido a que:

- Se basa en componentes, es decir, el código es reutilizable.
- Fácil de mantener porque los cambios están localizados.

Objetos

Objeto: un objeto consta de una estructura de datos y una colección de métodos que manipulan esos datos. Los datos dentro de un objeto son atributos. Un objeto sólo puede ser manipulado a través de su interfaz. Interfaz es el conjunto de métodos visibles desde el exterior.

Conceptos sobre objetos

Encapsulamiento: Se ocultan los detalles de la implementación de las clases.

Clases: Describen abstracciones de datos y operaciones necesarias para su manipulación. Dentro de la clase se definen:

- *Atributos:* Algún tipo de dato contenido en una clase.
- *Métodos:* Algoritmos internos a la clase.

Polimorfismo: Comportamiento de un objeto diferente en función de los parámetros.

Herencia: Cuando varias clases comparten características comunes, estas se ponen en una clase antecesora.

Asociación: Relación entre clases que cooperan con alguna finalidad.

Durante la etapa de análisis se identifican los objetos y durante el diseño se definen las características de los objetos.

Ciclo de vida del software

El ciclo de vida consta de las siguientes etapas principales:

1. **Análisis:** Construye un modelo de los requisitos.
2. **Diseño:** A partir del análisis se deducen las estructuras de datos en las que descompone el sistema y la interfaz de usuario
3. **Codificación:** Se obtiene código ejecutable
4. **Pruebas:** Se comprueba que todo sea correcto.

5. **Mantenimiento:** Tras la entrega se asegura que siga funcionando y se adapta a los cambios.

Todas las fases tienen documentos de entrada y de salida.

Algunos autores dividen el diseño en dos partes:

- Diseño global o arquitectónico
- Diseño detallado



CAPÍTULO II

Situación Actual

Aquí encontraremos una descripción general de la organización, la cual nos arrojará los datos necesarios para determinar el sentido del desarrollo de todo el proyecto, lo que quiere decir que, todos los problemas aquí se visualizarán de primera instancia y se tendrán en consideración muchos detalles que son muy importantes para la solución del problema global.

2.1.- Descripción de la organización.

2.1.1.- Historia.

El Grupo Educativo **D@tcom** nace en el mes de noviembre del año 1991 en el centro de la ciudad de Apizaco, Tlaxcala, en un pequeño local comercial con apenas 5 computadoras, primero ofreciendo servicios de renta de equipo, internet, impresiones, etc.

Después de 1 año de trabajo surgió la idea de comenzar con la enseñanza de la computación de la misma forma como algunas otras escuelas tales como CCPM, CETEC entre otras.

Se comenzó a trabajar con un estimado de 50 alumnos que con un poco de trabajo fue causando un gran impacto en la ciudadanía tlaxcalteca, originando un crecimiento constante, para que tres años después se abriera la segunda escuela de computación, al año siguiente se abrieron dos escuelas más haciendo un total de cuatro escuelas de las cuales una se estableció en el estado de Hidalgo, el plantel Ciudad Sahagún del cual se está realizando el proyecto actual, al año siguiente se abrieron otros cuatro planteles, dos más en el estado de Hidalgo y dos en el estado de Puebla, para hacer un total de ocho escuelas.

Este grupo educativo tiene el sueño de llevar a cada uno de los rincones de este país el conocimiento de la computación, fomentando en cada uno de sus alumnos los valores que día a día nuestra sociedad está olvidando aplicar y enseñar a los jóvenes.

2.2.- Áreas de la organización

2.2.1.- Listado de áreas

- Dirección
- Recepción
- Docencia
- Mantenimiento técnico
- Limpieza
- Dirección de fomento educativo
- Módulo de información
- Asesoramiento de fomento educativo

2.2.2.- Descripción de actividades por área

Dirección

Es la que se encarga de la toma de decisiones de toda la institución, da autorizaciones así como la supervisión de todas las actividades entorno a la misma, planeando el futuro y crecimiento del plantel.

Recepción

Ésta es el área de interés, debido a que aquí se presentan los problemas, se encarga de la administración de los recursos, tales como:

1. Los ingresos
2. Los egresos
3. Los trámites administrativos para alumnos
4. Información general del estado de los alumnos
5. Control de materiales, entre otros

Docencia

Aquí se encuentra el personal de enseñanza, dispuesto a colaborar para el buen desempeño de las actividades de los alumnos apegándose a los reglamentos y currícula establecida para todos los planteles del Grupo Educativo.

Mantenimiento técnico

Dentro de ésta área se encuentra todo lo involucrado con la reparación y limpieza del equipo de cómputo para el funcionamiento óptimo del mismo, así como la instalación de los programas necesarios para el cumplimiento de los programas de estudios.

Limpieza

Se encarga de la pulcritud del plantel en donde podemos mostrar: las aulas de clase, oficinas en general, pasillos, cristales, salas de conferencias, etc.

Dirección de fomento educativo

Es el encargado de coordinar al departamento de asesoramiento de fomento educativo para completar el objetivo de realizar inscripciones de alumnos fuera del plantel.

Módulo de información

Aquí se brinda toda la información pertinente de los planes de estudios del plantel para cumplir con el objetivo de realizar inscripciones de personas con el interés de aprender.

Asesoramiento de fomento educativo

Éste departamento es el que realiza la tarea de visitar domicilios con la firme intención de realizar inscripciones proporcionando todos los informes necesarios tanto del plantel como de los planes de estudios disponibles.

2.2.3.- Organigrama

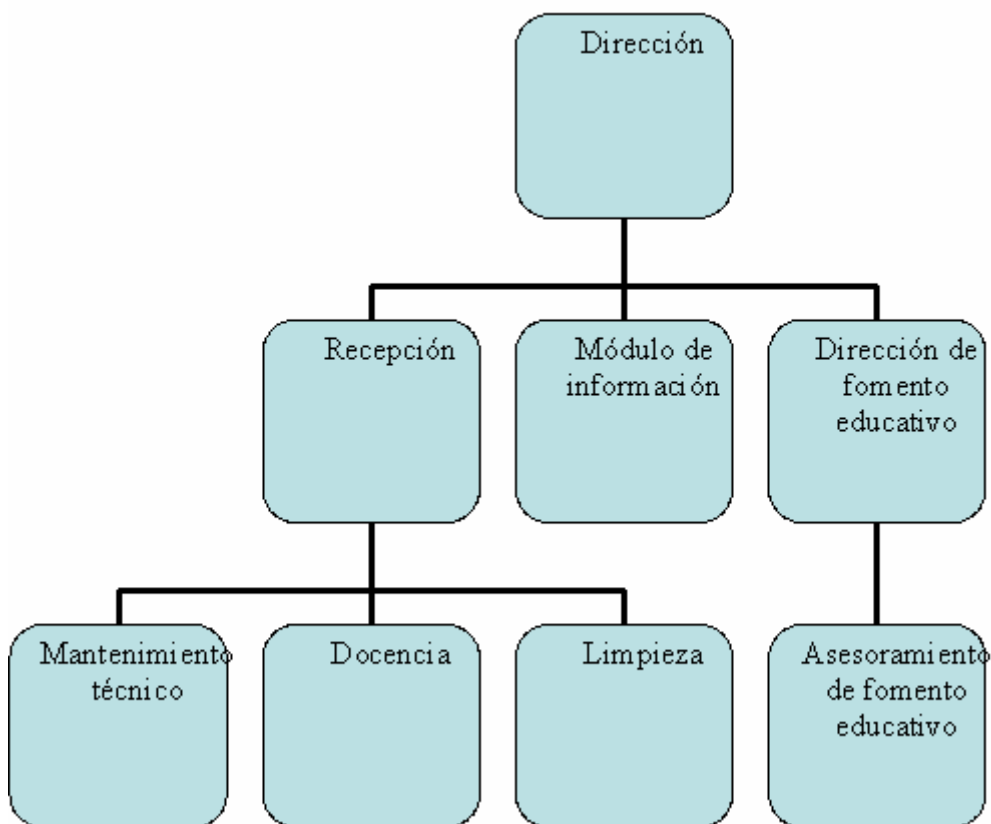


Figura 5.- Organigrama de la empresa

2.3.- Descripción del problema

2.3.1.- Administración problemática por rezago en la información

El plantel administrativamente lleva su control de ingresos como comúnmente se conoce “arrastrando el lápiz”, es decir, que la elaboración de recibos de cobro se realiza escribiendo a mano los datos necesarios para el recibo, y capturando el folio en listas con el formato propio del plantel, esto conlleva a que los datos necesarios para la elaboración de reportes del estado de pagos por alumno sea muy lento y con muchos errores, además de que al realizar el corte semanal del área de recepción sea del mismo modo muy lento y con muchos errores.

De aquí que podemos identificar todos y cada uno de los puntos que retrasan muchas de las actividades administrativas del plantel, por lo que en el presente documento se expone la alternativa de solución para que exista una administración mas dinámica y sin errores garantizando la actualización constante de los datos necesarios para muchas tareas administrativas y con la posibilidad de expandir el proyecto en un futuro contemplando los puntos expuestos anteriormente en la sección de requerimientos.

No olvidemos decir que, con el desarrollo del proyecto, lograremos ahorros significativos tanto en tiempo invertido en el personal como en gastos adicionales causados por errores administrativos.



Capítulo III

Fases de la Automatización

Este capítulo se dedica a todas y cada una de las fases que sigue el sistema para una buena implementación, desde el análisis pasando por el diseño e implementación, hasta las pruebas necesarias para eliminar la mayor cantidad de errores posible, teniendo en cuenta tanto el lenguaje de programación como las bases de datos necesarias.

Creado por Phantom en 3DS Max 4, 23.08.2003

3.1.- Análisis

De acuerdo a lo expuesto en el marco teórico, debemos especificar la metodología que utilizamos para éste fin que aunque se proporciona más adelante la mencionaremos, “Metodología en Espiral”.

Para llevar a cabo el análisis se debe tener una buena definición del problema (expuesto anteriormente), para que se pudiera llevar a cabo una mejor definición del problema presentamos aquí las preguntas realizadas al personal necesario y así obtener todos los detalles de la problemática que surge en torno al área administrativa.

3.1.1.- Entrevista

Nombre de la empresa o negocio.

D@tcom Escuela de Computación

Preguntas

- 1.- Cuál es el giro principal de la empresa?
- 2.- Qué tipo de servicios que brinda la Institución?
- 3.- Cuál es la cantidad aproximada de alumnado con la que cuenta la Institución?
- 4.- Cómo organiza la población estudiantil la institución?
- 5.- Cómo lleva a cavo el control del alumnado?
- 6.- Cuenta con un equipo de computo propio para el manejo del sistema?
- 7.- Cuál es el tiempo aproximado que tarda en capturar la información referente al alumnado?
- 8.- El llevar el control de sus actividades le genera gastos?
- 9.- Qué necesidad como dueño o encargado del puesto tiene para realizar mejor sus actividades?
- 10.- Le gustaría implantar un sistema que resuelva todos esos problemas?
- 11.- Qué tipo de problemas enfrenta al realizar sus actividades y descríbalos uno a uno?
- 12.- Como cliente, que le gustaría que hiciera el sistema?

3.1.2.- Requerimientos de la empresa de acuerdo a los datos obtenidos en la entrevista

El requerimiento principal de la institución es el de implantar un sistema que lleve el control de todos los ingresos como pago de colegiaturas, renta de equipo, impresión de documentos, venta de accesorios, venta de consumibles, venta de equipos de cómputo, servicios de reparación y mantenimiento, etc., así como los adeudos semanales generados por el concepto de colegiaturas.

El sistema a implantar deberá observar los siguientes puntos:

- El sistema deberá mostrar de forma detallada la información referente a los ingresos de la institución (antes mencionados).
- Este deberá generar los reportes semanales correspondientes a los conceptos de ingreso considerados.
- Deberá aplicar descuentos, recargos y cortes semanales de manera sencilla, eficiente y eficaz.
- Podrá ser actualizada la información solo por personas autorizadas.

También se pudieron observar algunos requerimientos adicionales que se contemplarán en una segunda parte de éste proyecto debido a la extensión del mismo, estos son:

- Un control de calificaciones por alumno.
- Un reporte para la generación de boletas de calificaciones en donde se incluya el desglose de inasistencias por módulo.
- Un reporte de vigencias de inscripciones semestre con semestre.
- Un reporte para elaborar constancias de estudios incluyendo la vigencia con respecto a la fecha de inscripción.

3.1.3.- Necesidades detectadas en la institución educativa (D@tcom)

De acuerdo a la entrevista se pueden observar los siguientes puntos:

Necesidades observadas.

- Realizar el cobro por los productos y/o servicios que ofrece la institución educativa (DATCOM).
- Que la información que se requiere en el momento pueda ser accedida con mayor rapidez.
- Saber en cualquier momento determinado el control detallado de pagos y adeudos semanales generados por los usuarios y alumnos que reciben el servicio.
- Que los datos se encuentren actualizados en todo momento ya que están en constante cambio.

En pocas palabras la necesidad que detectamos en el Grupo Educativo D@TCOM es el realizar de una manera más rápida y clara el control de ingresos de la misma adicionando dinamismo en las actividades administrativas del plantel.

3.1.4.- Alternativas para la solución

Identificación de alternativas.

La alternativa de solución es un sistema que permita llevar el control de los ingresos de la institución pero enfocándonos principalmente a los pagos de colegiatura, los adeudos y los descuentos realizados a la hora de pagar su colegiatura y otros ingresos generados por usuarios externos. Que suelen ser actividades muy laboriosas y difíciles de realizar y además si no se llevan a cabo correctamente pueden generar errores.

A continuación se propone la siguiente alternativa:

- El implantar un sistema para el control de ingresos que de solución a la problemática que actualmente enfrenta la institución educativa.

Comenzaremos con el análisis de las necesidades del plantel, la creación de una especificación del diseño para el sistema y la preparación de una propuesta indicando los los ingresos que tiene la institución a un determinado tiempo.

Una solución al problema anterior es la elaboración de un producto de software como ya se especifico anteriormente que nos permita el fácil uso de las actividades.

3.1.5.- Criterios a evaluar:

- Que el sistema cumpla con los requerimientos de la institución (corrección en los datos).
- Que la información sea clara y concisa (facilidad de uso).
- Que la información este protegida contra acceso y modificaciones no autorizadas (integridad).
- Que la información contenida tienda a ser actualizada, ésto sólo por personal autorizado.
- Uno de los puntos más importantes y que es el propósito de todo análisis es que mejore la situación actual de la institución.

Por lo tanto podemos decir que la alternativa sugerida hasta ahora ha cumplido con lo establecido.

3.1.6.- Estudio de factibilidad

De acuerdo a la funcionalidad del sistema y al estudio realizado para conocer los requerimientos del mismo podemos deducir lo siguiente:

3.1.6.1.- Razones por las que se deba realizar el sistema

- Ahorro de tiempo a la hora de capturar datos
- Datos precisos a la hora de consultar datos
- Información ordenada
- Cálculo de operaciones matemáticas óptimas
- Administración mejor de los recursos económicos.
- Rápido acceso a los datos(tiempo)
- Evitar la perdida de la información
- Evitar papelería excesiva (espacio)
- Rápida actualización de datos
- Privacidad de los datos

3.1.6.2.- Razones por las que no se deba implantar el sistema

- **Los costos**, puede ser más barato seguir llevando acabo las funciones y almacenando la información del sistema en forma manual, porque surgen gastos como:
 - * Consumo de electricidad
 - * Costo de equipo
 - * Costo de mantenimiento
 - * Costos de instalación
 - * Sobre todo el costo del producto (sistema)

Por lo que nos podemos dar cuenta hay mas razones positivas para implantar el sistema. Si nos damos cuenta los requerimientos de la institución son con el fin de brindar un servicio y llevar el control de actividades realizadas para obtener un ahorro tiempo y lograr almacenar datos precisos, sin olvidar que los ahorros reales se verán materializados a largo plazo.

Por lo tanto todo apunta a que el sistema sea realizado con éxito y llevar acabo su implantación.

3.1.6.3.- Propuesta sugerida

La propuesta sugerida para la elaboración del sistema es la siguiente:

- la persona encargada de la administración y cobranza disminuirá su tiempo al realizar las actividades a un 50% del tiempo que actualmente utiliza para realizar su trabajo.
- El sistema tendrá la capacidad de realizar cortes semanales deslindando responsabilidades al encargado de administración y cobranza.
- Tendrá la capacidad de aceptar o rechazar los pagos no autorizados.
- Brindará al usuario los reportes necesarios con la información requerida en el momento.
- La información estará ordenada de acuerdo al orden de importancia esto con el fin de poder identificar cada uno de los procesos con mayor rapidez y también evitar que el usuario lo registre en hojas que a su vez es una actividad muy tediosa y tardada.
- Permitirá que la información pueda ser actualizada cada vez que el usuario lo desee (solo personal autorizado). Esto ocasiona que los datos almacenados sean validos y precisos.
- El sistema estará protegido ante todo tipo de intrusos y factores que puedan modificar o provocar la pérdida de información.

Como pudimos darnos cuenta, las necesidades de la empresa en este caso una institución educativa son suficientes para poder implantar un sistema para el mismo fin.

3.1.6.4.- Planeación del proceso de desarrollo

El modelo de ciclo de vida del software a utilizar es:

“El Método en espiral“

El cual consiste en:

- Requerimientos del sistema
- Requerimientos del software
- Análisis
- Diseño del programa
- Codificación
- Pruebas
- Operaciones

3.1.6.5.- Determinación de las herramientas a utilizar

- DIAGRAMAS DE FLUJO DE DATOS
- DICCIONARIO DE DATOS
- ESPECIFICACION DE PROCESOS
- DIAGRAMA ENTIDAD/RELACION
- DIAGRAMAS DE TRANSICION DE ESTADOS
- DIAGRAMA DE ESTRUCTURAS

3.1.6.6.- Programa preliminar para el desarrollo

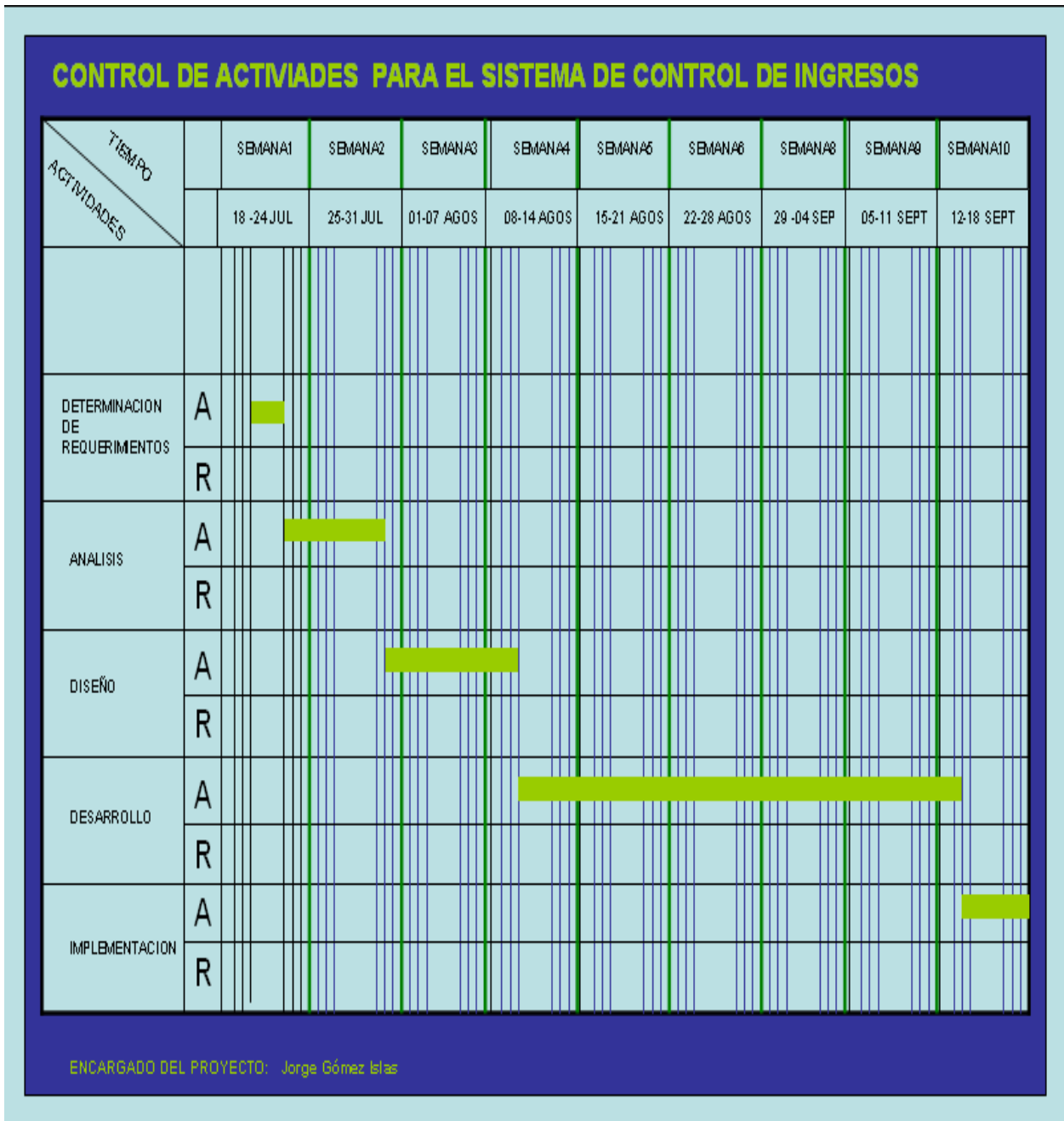


Figura 6.- Diagrama de Gantt (programa preliminar)

3.1.6.7.- Estimados preliminares de costo para el desarrollo del sistema

| TAREAS | TIEMPO APROXIMADO DE REALIZACION | COSTO | TOTAL |
|--------------------------|----------------------------------|------------|-------------|
| REQUISITOS DEL SISTEMA | 1 DIAS | \$ 500.00 | \$1.000.00 |
| ANALISIS | 1 SEMANAS | \$3.000.00 | \$9.000.00 |
| DISEÑO | 1.5 SEMANAS | \$3.000.00 | \$9.000.00 |
| GENERACION DE CODIGO | 2 SEMANAS | \$4.000.00 | \$4.000. |
| PRUEBAS DE SOFTWARE | 2 SEMANAS | \$4.000.00 | \$6.000.00 |
| INSTALACION DEL SOFTWARE | 4 HORAS | ----- | ----- |
| SOPORTE TECNICO | ----- | ----- | ----- |
| PERSONAL DE OPERACION | ----- | ----- | ----- |
| SUBTOTALES | | | |
| | | TOTAL \$ | \$29.000.00 |

3.1.6.8.- Estimación de costo/beneficio

La estimación de costos y beneficios para la institución la podemos ver desde el punto de vista siguiente:

BENEFICIOS TANGIBLES:

Con el uso del sistema de control ingresos podrá ahorrar en cuestión de papelería, copias de cada uno de los formatos utilizados y también el reducir su personal laboral por ejemplo: Si el puesto es ocupado por 2 personas esto puede reducirse a solo una, es decir, se reduciría el personal ya que el sistema llevará a cabo actividades que realiza el personal que labora en el área de administración y cobranza, debido a que se necesitará una sola persona para controlar el sistema.

BENEFICIOS INTANGIBLES:

Dentro de este tipo de beneficios podemos decir que el sistema va a realizar todas esas tareas monótonas como el pago de colegiaturas, el cobro de adeudos y descuentos para aquellos estudiantes que así lo requieran. Las actividades se realizan manualmente porque como ya explicamos anteriormente no cuentan con una herramienta de software y por tanto todas esas tareas no significarán aburrimiento para el personal en cuanto al control de los ingresos económicos de la institución.

COSTOS TANGIBLES:

Dentro de este tipo de costos podemos decir que, con la implantación del sistema de control de alumnos se podrían sustituir algunos sueldos pero también va a generar algunos gastos como: el mantenimiento del equipo, el soporte técnico, compra de material para impresora como tinta, el uso de energía eléctrica, y si no se adquiere impresora por el momento el gasto en impresiones.

3.2.-Diseño

3.2.1.- Diagramas de flujo

3.2.1.1.- Diagrama de contexto

En ésta sección debemos apoyarnos en muy diversas herramientas las cuales ya fueron mencionadas anteriormente, una de ellas es la de los diagramas de flujo que en todo momento nos mostrarán nuestro sistema de manera completamente gráfica y así tener otra perspectiva para solucionar problemas de forma eficaz lo que nos llevará a crear un sistema eficiente veamos la siguiente figura (figura 7).

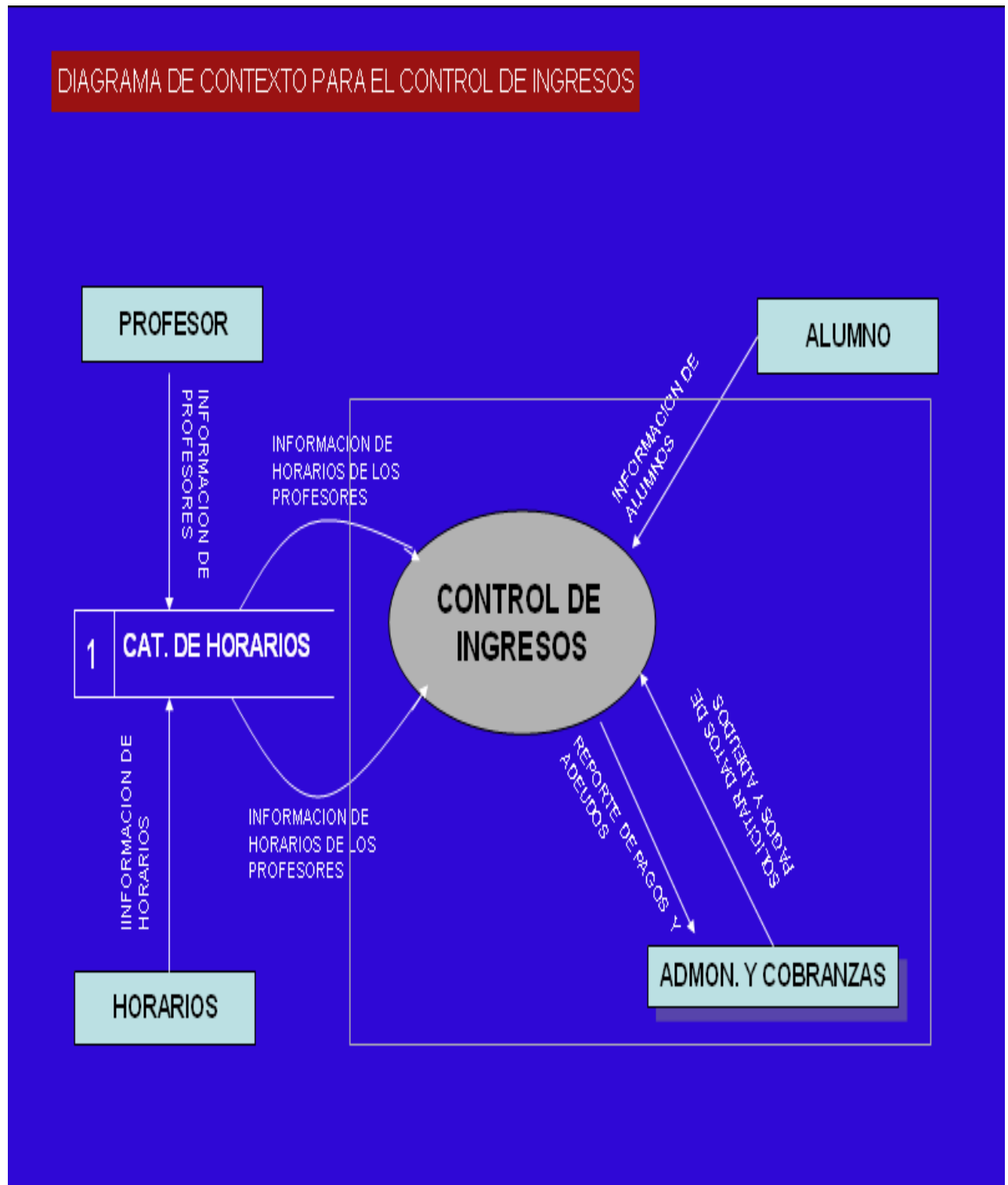


Figura 7.- Diagrama de contexto

3.2.1.2.- Diagrama de flujo de nivel 0

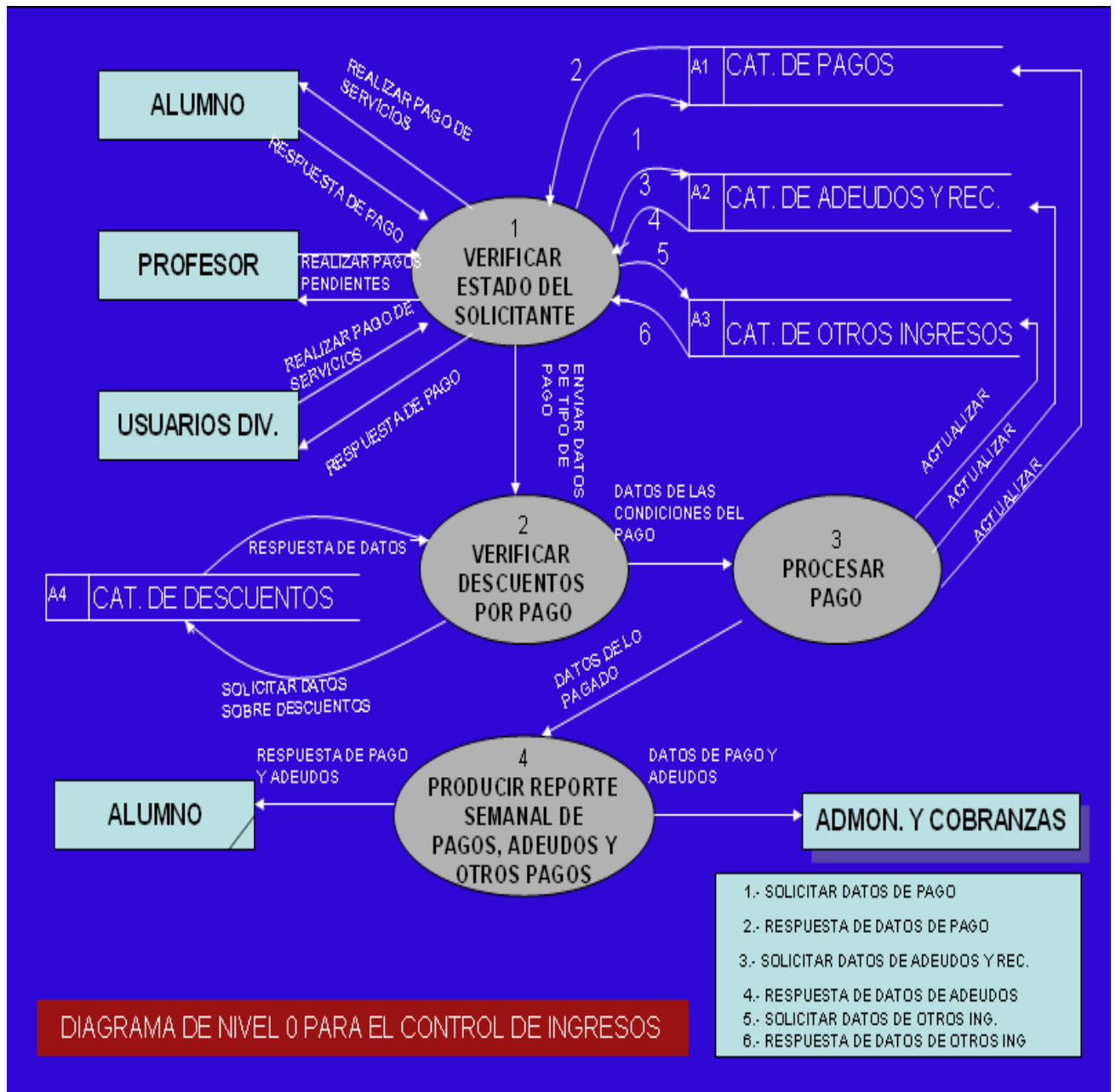


Figura 8.- Diagrama de nivel 0

3.2.1.3.- Diagramas de flujo de nivel 1

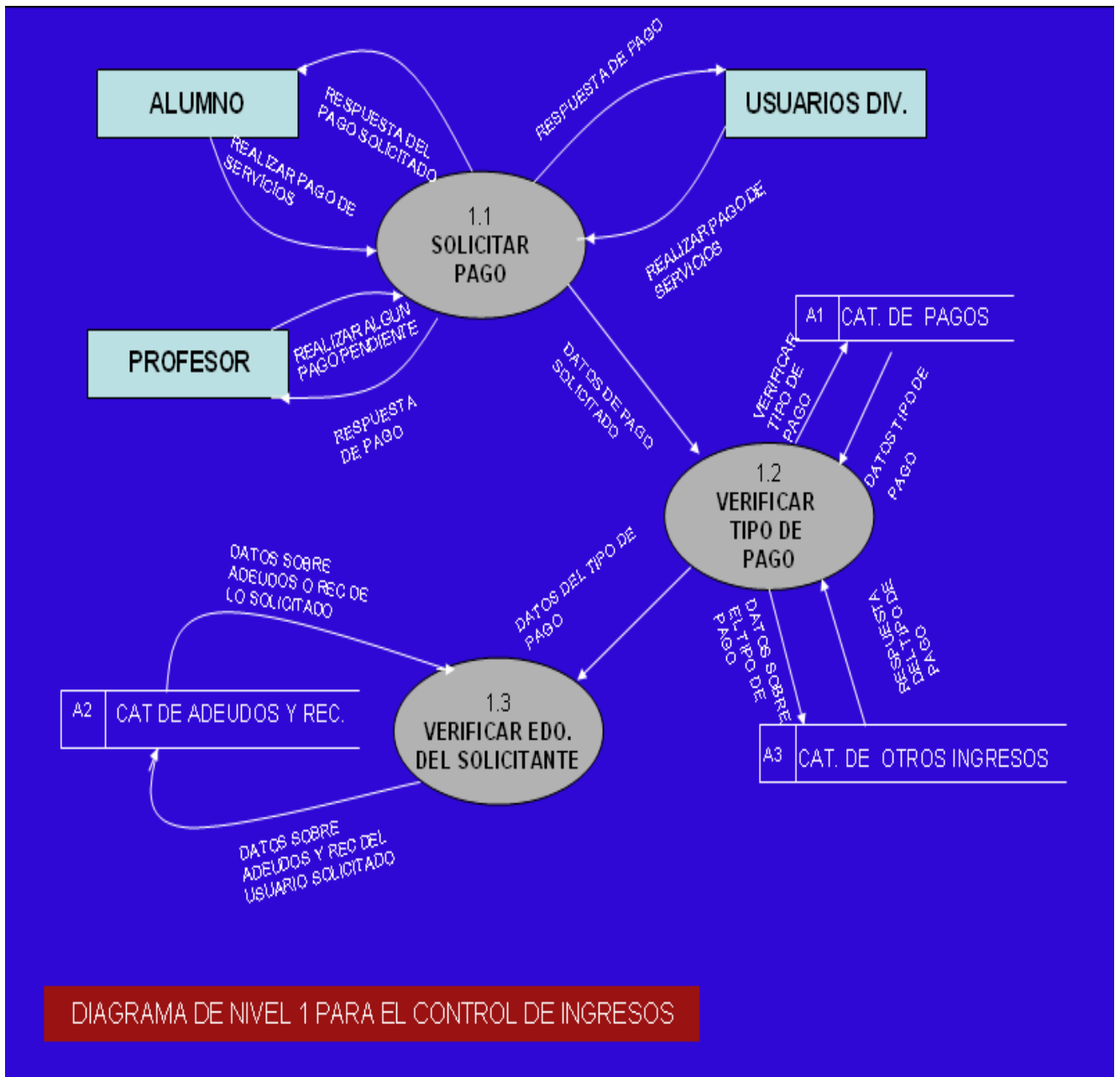


Figura 9.- Diagrama de nivel 1

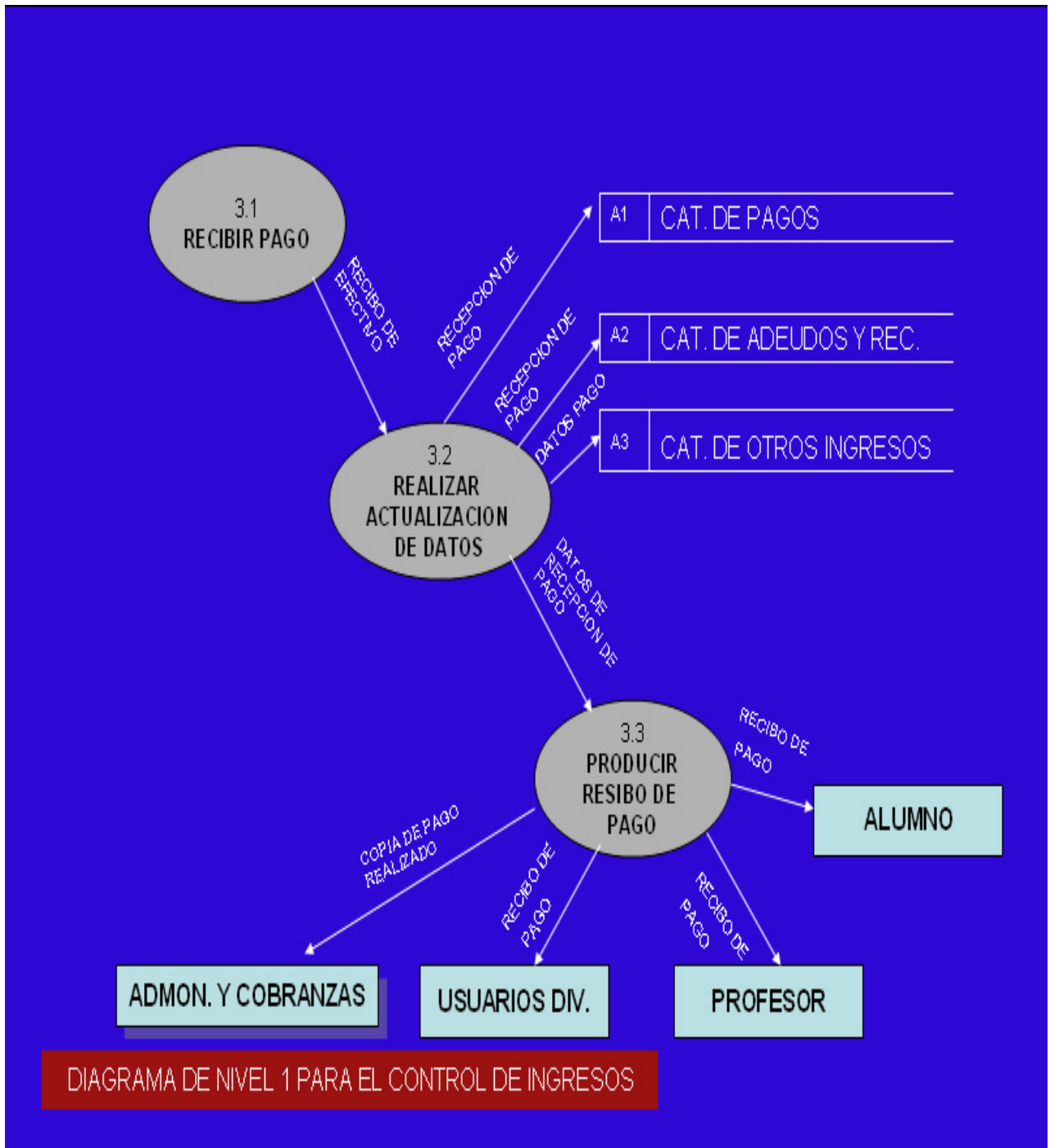


Figura 10.- Diagrama de nivel 1 (continuación)

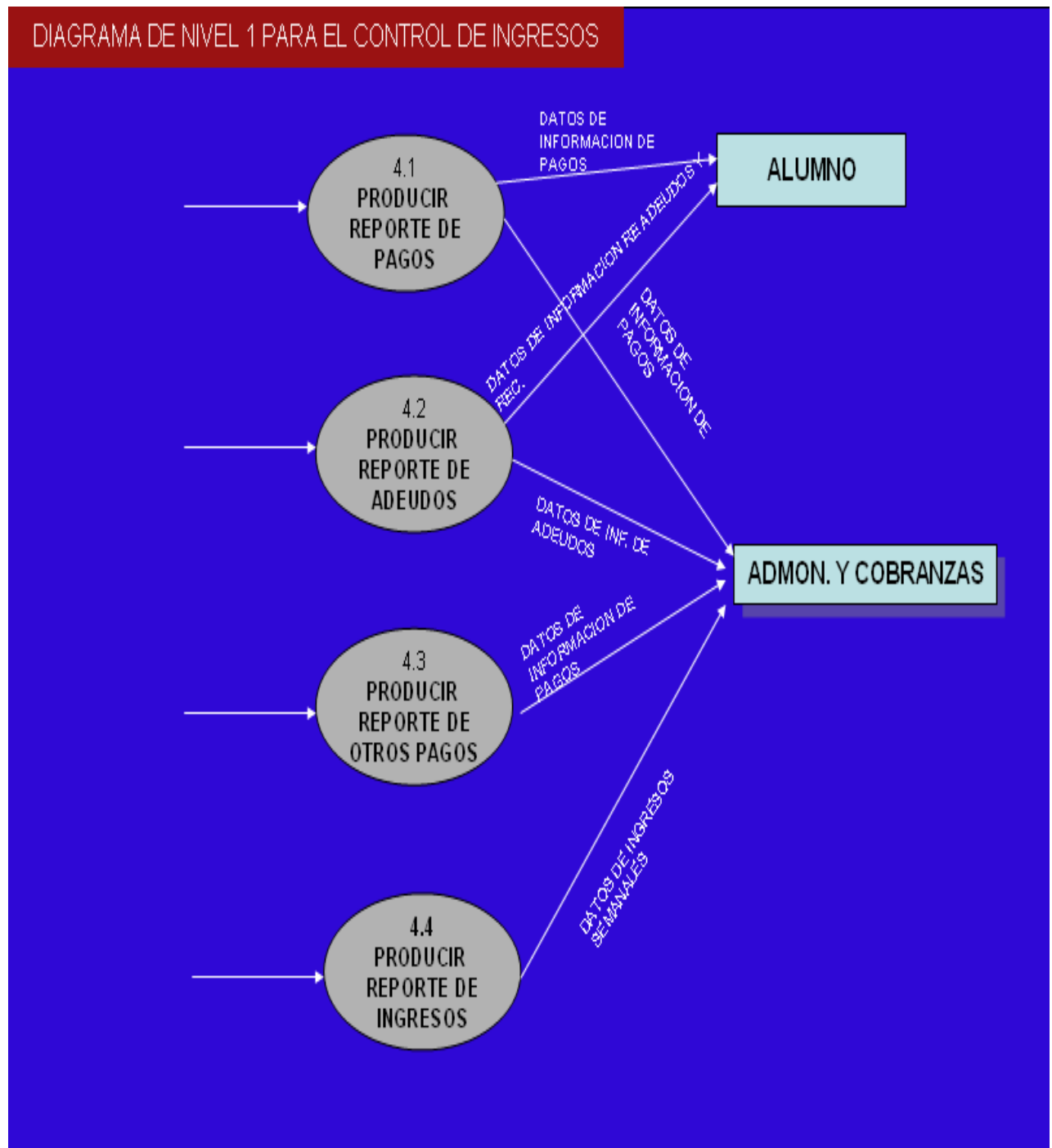


Figura 11.- Diagrama de nivel 1 (continuación)

3.2.1.4.- Diagrama entidad/relación



Figura 12.- Diagrama entidad/relación

3.2.2.- Diseño lógico del software

COMPONENTES DEL SISTEMA

El Sistema de Control de Pagos (SICEP) se encuentra integrado por los siguientes componentes:

- ✓ **Datos de Servicios**
- ✓ **Datos de Descuentos**
- ✓ **Pagos de Servicios a usuarios tanto internos como externos**
- ✓ **Datos de Cortes de Caja**
- ✓ **Datos de Adeudos**

DESCRIPCION DE CADA UNO DE LOS MÓDULOS DEL SISTEMA

Una vez que se ha realizado exitosamente la instalación del sistema, proseguimos a describir a nivel detallado la operatividad de cada uno de los módulos que integran el Sistema de Control de Pagos (SICEP):

En el momento que usted ejecute el sistema aparecerá una pantalla de inicio que le pedirá la contraseña que lo identifica como un usuario permitido la cual aparece a continuación:



Figura 13.- Formulario de seguridad.

En caso de que la clave de acceso sea incorrecta (no sea un usuario permitido) aparecerá un mensaje de error indicando que la contraseña es incorrecta.

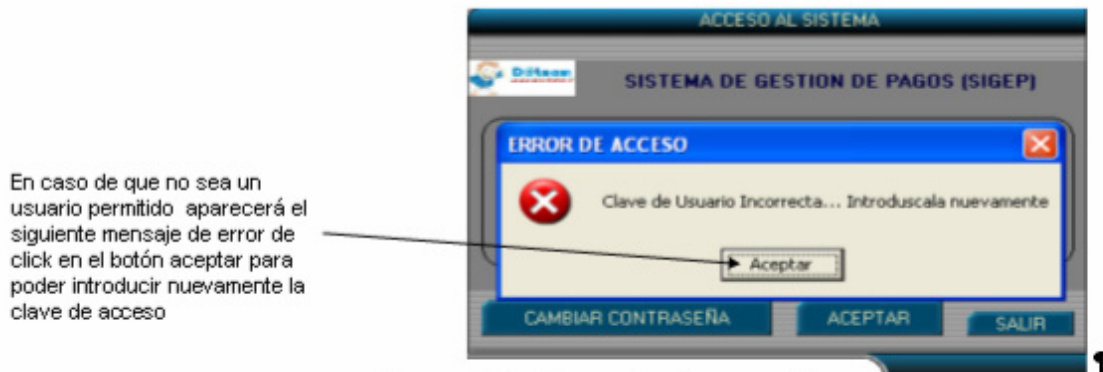


Figura-14.-Mensaje de error.

Una vez que aparece un mensaje de error, después de esa oportunidad tiene 2 intentos más para proporcionar la clave de acceso de lo contrario aparecerá un mensaje que indicara que llego al límite de intentos y las oportunidades se habrán agotado y el sistema se cerrará como se muestra a continuación:

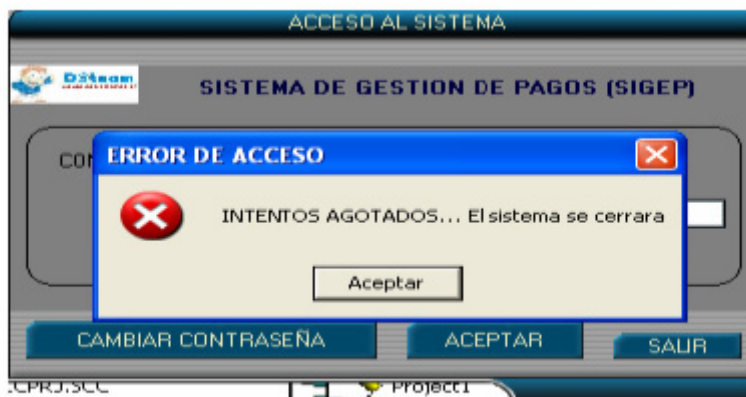


Figura-15.-Mensaje de intentos agotados.

De lo contrario una vez introducida la clave correcta al sistema (es un usuario permitido) podrá acceder al sistema sin ningún problema.

En caso de que sea un usuario nuevo destinado a operar el sistema, existe una opción que permite dar de alta a un nuevo usuario esto presionando el botón "CAMBIAR CONTRASEÑA" en el modulo de ACCESO AL SISTEMA.

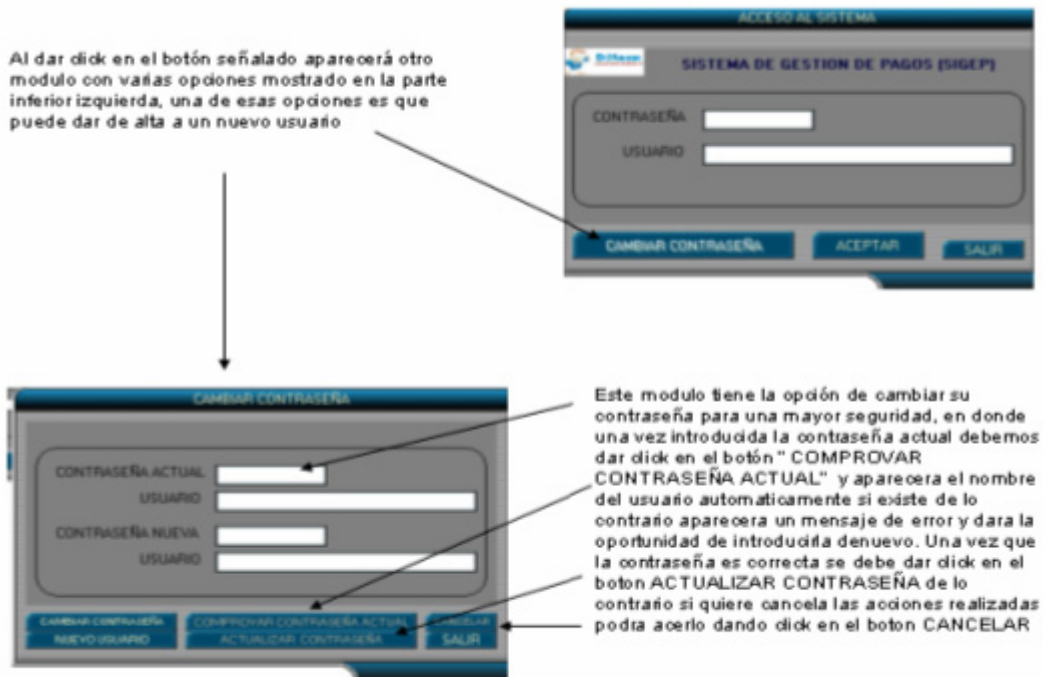


Figura 16.- Formulario de contraseña.

Si es un usuario nuevo de clic en el botón NUEVO USUARIO ubicado en el modulo CAMBIAR CONTRASEÑA:

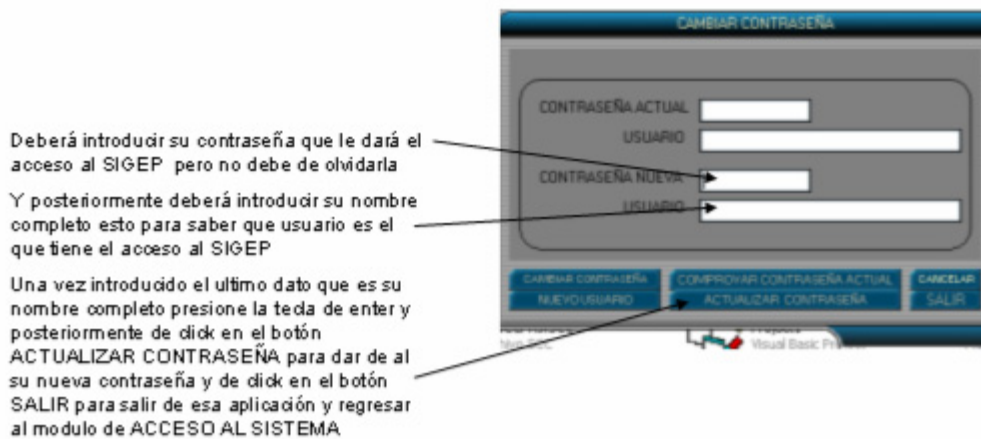


Figura 17.- Formulario de alta de usuarios.

Si usted es un usuario permitido entonces puede acceder al sistema:

Usted acaba de entrar al SISTEMA DE CONTROL DE PAGOS (SICPE) esperando que éste le sea útil y fácil de manejar con un ambiente agradable.

El SICEP fue realizado para poder llevar a cabo todas las actividades relacionadas con los ingresos de la institución educativa D@TCOM, lo cual resulta difícil y monótono llevarlo a cabo manualmente, contempla las siguientes actividades:

- ✓ Pagos parciales de colegiatura
- ✓ Descuentos por concepto de pago (interno como externo)
- ✓ Recargos a pagos por concepto de colegiatura
- ✓ Permite realizar corte caja diario a diferentes usuarios
- ✓ Permite cancelar pagos realizados
- ✓ Muestra el total de ingresos semanales
- ✓ Muestra de forma detallada los adeudos por concepto de colegiatura
- ✓ Muestra de forma detallada el total de ingresos por concepto de colegiatura
- ✓ Permite imprimir ticket de corte de caja
- ✓ Muestra el numero de semanas de adeudo por alumno

Una vez que usted ingresa al SICEP aparecerá el menú principal del sistema el cual se muestra a continuación:

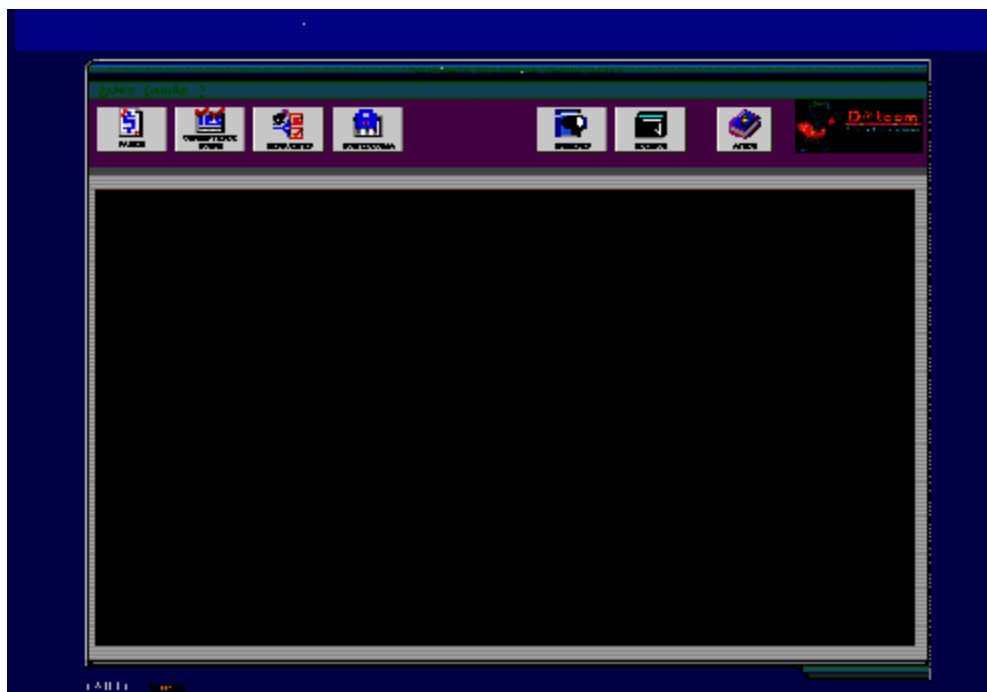


Figura 18.- Formulario Principal

En el cual puede acceder a las aplicaciones desde el menú principal o la barra de iconos.

Para acceder a las aplicaciones desde el menú principal deberá seleccionar ya sea, el menú ARCHIVO el cual contiene:

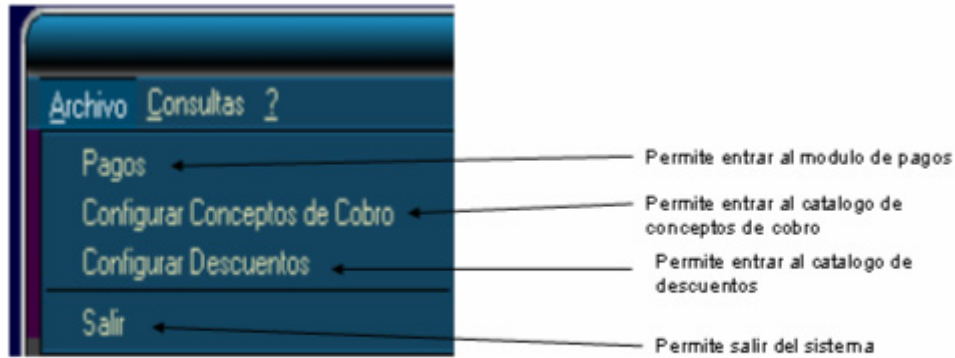


Figura 19.- Menú Principal.

Para acceder a las aplicaciones desde el menú CONSULTAS deberá seleccionar el MENU CONSULTAS el cual contiene:

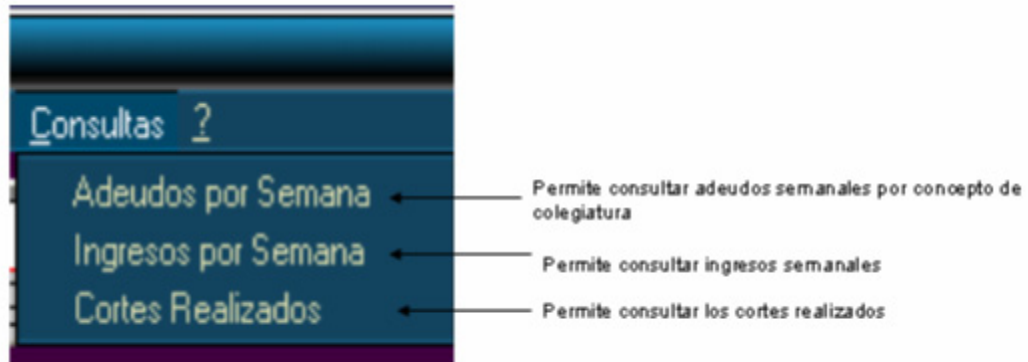


Figura 20.- Menú Consultas

Para acceder a las aplicaciones desde el menú AYUDA deberá seleccionar el MENU AYUDA el cual contiene:

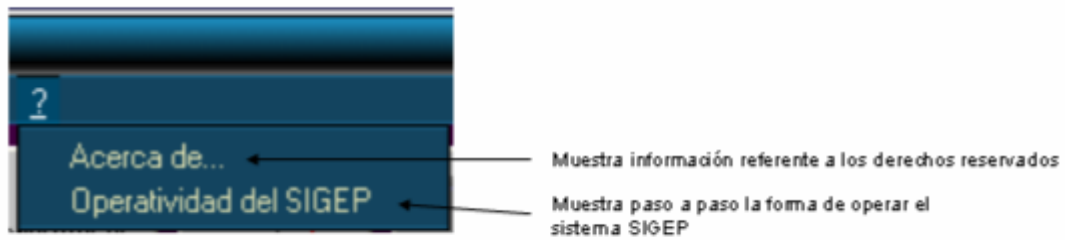


Figura 21.- Menú ayuda.

Otra manera de acceder a las aplicaciones del SISTEMA DE CONTROL DE PAGOS (SICEP) es por medio de la barra de iconos localizada debajo de la barra de menús y los cuales se describen a continuación:

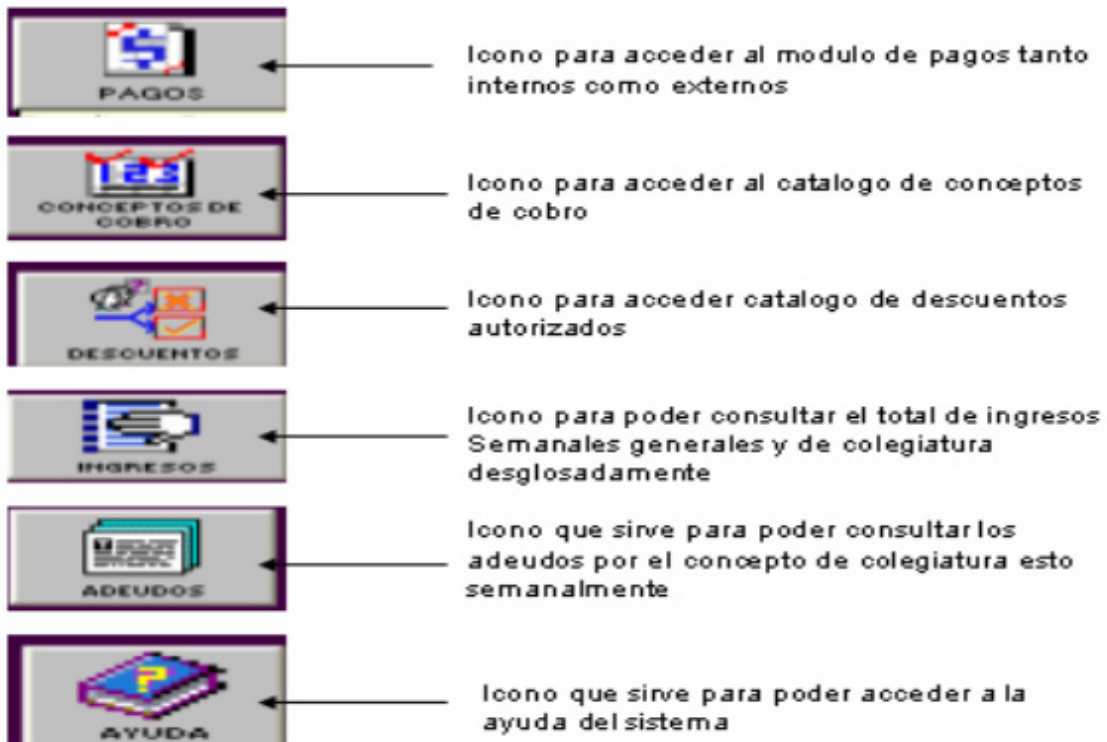


Figura 22.- Íconos.

Una vez que ya se explicó la forma de poder acceder a cada uno de los módulos que contiene el sistema, comenzaremos por explicar cada uno de los módulos detalladamente.

COMO REALIZAR EL PROCESO DE PAGO

Para poder realizar el proceso de pagos en el SICEP de una manera sencilla y rápida, en dos o tres click's, depende el tipo de pago (ya sea interno o externo) ya que el sistema fue estandarizado para poder realizar ambos pagos.

A continuación siga los siguientes pasos para poder realizar un pago.

Pasos para cuando se trata de un pago Interno:

1.- Ingrese al módulo de PAGOS, lo puede hacer desde el menú ARCHIVO y PAGOS para acceder al módulo correspondiente o si lo prefiere desde el botón de PAGOS contenido en barra de iconos.

2.- Una vez que ingresó al modulo de pagos aparecerá la siguiente ventana solicitándole nuevamente su ID (identificación para saber que es un usuario permitido).

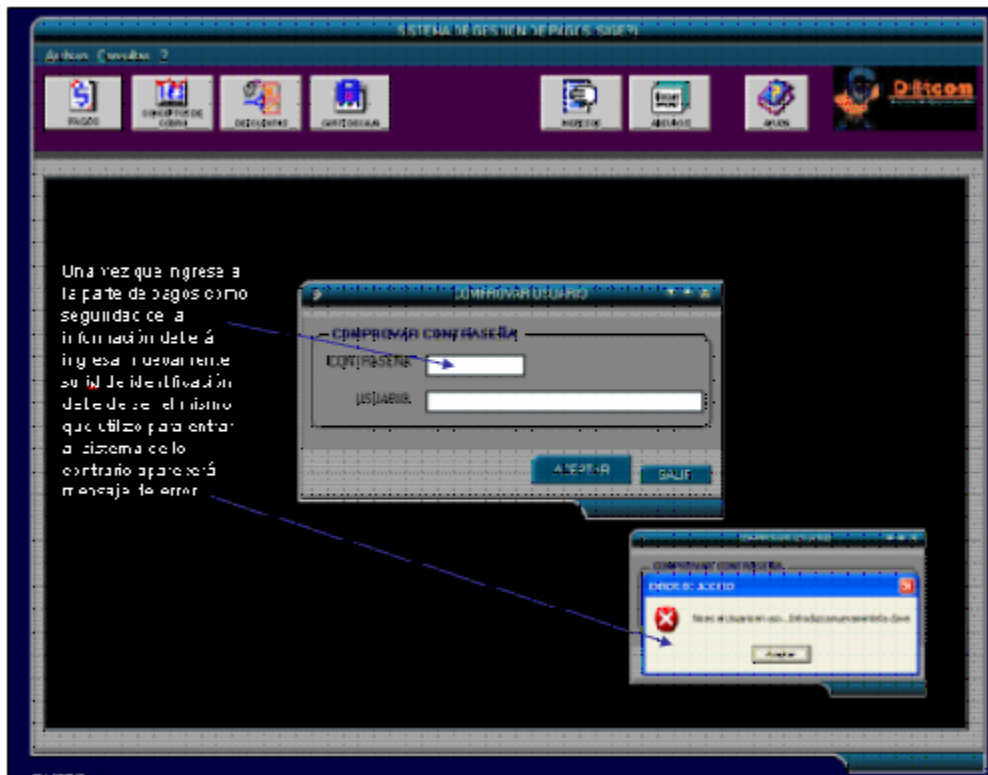


Figura 23.- Formulario de solicitud de ID.

3.- Si la contraseña es correcta entonces, el sistema le permitirá entrar al módulo de pagos que se muestra continuación:

The screenshot shows a software interface for a payment system. At the top, there is a menu bar with the following options: NUEVO PAGO, ACTUALIZAR PAGO, MODIFICAR PAGO, BORRAR PAGO, CANCELAR NUEVO PAGO, CANCELAR PAGO REALIZADO, and SALIR DE ESTA OPCION. Below the menu, the interface displays several fields and controls:

- Fecha:** 23/03/2005
- FOLIO:** (empty field)
- Tip de Pago:** with radio buttons for 'Efectivo' and 'Cheque'.
- MATRICULA:** (input field) with a button labeled 'BUSCAR MATRICULA DEL ALUMNO'.
- NOMBRE DEL CLIENTE:** (input field)
- CONFIRMA:** (input field)
- Nº DE SEMANAS A PAGAR:** (input field)

On the right side of the form, there are three vertical menus:

- Top menu: - CANCELAR PAGO REALIZADO -
- Middle menu: - Efectivo - Cheque -
- Bottom menu: - CONFIRMAR PAGO -

Figura 24.- Formulario de Pagos.

4.- A continuación se muestra el procedimiento de cómo realizar un pago (INTERNO).

The screenshot shows a web-based payment form. At the top, there is a title 'PAGOS' and a navigation menu with icons and labels: 'NUEVO PAGO', 'ACTUALIZAR PAGO', 'MODIFICAR PAGO', 'BORRAR PAGO', 'CANCELAR NUEVO PAGO', 'CANCELAR PAGO REALIZADO', and 'SALIR DE ESTA Opcion'. The form itself is divided into several sections. On the left, there are radio buttons for 'Interno' and 'Externo'. To the right, the date is set to '26/05/2005' and the 'FOLIO' is '1'. Below this, there is a 'MATRICULA' field with the value '1' and a button 'BUSCAR MATRICULA DEL ALUMNO'. The 'NOMBRE DEL CLIENTE' field is currently empty. Below that is a 'CONCEPTO' dropdown menu and a 'No. DE SEMANAS A PAGAR' dropdown menu. On the far right, there are summary fields: 'No. de Semanas que Adude' (empty), 'Descripcion' (empty), 'Total Adude' (displaying '\$'), and 'Adude Anual Adude' (displaying '\$ 0'). Five orange arrows originate from the form and point to a list of five numbered instructions at the bottom of the page.

1.- damos clic en el botón NUEVO PAGO para realizar un nuevo pago

2.- damos clic en INTERNO

3.- Introducimos el numero la matricula del alumno en caso de que no la recuerde el alumno de clic en el botón de la derecha para buscarla

4.- una vez introducida la clave correcta del alumno automáticamente e aparecerá el nombre del alumno

5.- en esta parte indicara el numero de pago que también es automático

Figura 25.- Formulario de Acción de Pagos (parte 1).

6.-en esta parte aparecerá el numero de semanas que debe

7.- en esta parte aparecerán las semanas especificas que debe

8.- en esta parte aparecerá el total en efectivo de las semanas que debe el alumno seleccionado

9.- en esta parte aparecerá el adeudo anual del alumno esto solo en caso de que se este empezando un nuevo año y se deba del atrasado

NOTA: Todos estos puntos relacionados en cuanto a adeudos se refieren solo al concepto de COLEGIATURA

Figura 26.- Formulario de Acción de Pagos (parte 2).

5.- Ya proporcionamos la clave del alumno ahora vamos a realizar un pago relacionado con el mismo.

Aquí tenemos 2 formas de realizar un pago de una persona que pertenece a la institución:

- a) cuando el alumno va a realizar el pago de colegiatura
- b) cuando el alumno va a realizar el pago por cualquier otro concepto diferente al concepto de colegiatura.

Vamos a ver el primer caso cuando el alumno va a pagar colegiatura:

The screenshot shows a software interface for processing payments. At the top, a toolbar contains several action buttons: 'NUEVO PAGO', 'ACTUALIZAR PAGO', 'MODIFICAR PAGO', 'BORRAR PAGO', 'CANCELAR NUEVO PAGO', 'CANCELAR PAGO REALIZADO', and 'SALIR DE ESTA OPCION'. The main form area is titled 'PAGOS'. It features a 'Tipo de Pago' section with radio buttons for 'Interno' and 'Externo'. Below this, there is a 'MATRICULA' field with the value '3' and a 'BUSCAR MATRICULA DEL ALUMNO' button. The 'NOMBRE DEL CLIENTE' field displays 'Victor Telles Peralta'. A 'CONCEPTO' dropdown menu is open, showing a list of options: 'Colegiatura', 'Credencial', 'Impresiones', 'Renta de Equipo de Computo', 'Examen Estemporaneo', 'Constancia de Calificaciones', 'Examen Extraordinario', and 'Venta de Equipo de Computo'. The 'No. DE SEMANAS A PAGAR' dropdown menu is also open. On the right side, a summary box displays: 'FECHA' 30/10/2005, 'FOLIO' 41, 'No. de Semanas que Adeuda' 1, 'Descripcion' 43, 'Total Adeudo' \$ 115, and 'Adeudo Anual Atrasado' \$ 0. Two blue arrows point from the text below to the 'CONCEPTO' dropdown and the 'No. DE SEMANAS A PAGAR' dropdown.

10.- Seleccionamos del catalogo de CONCEPTOS DE COBRO la opción de COLEGIATURA

11.- Al momento de Seleccionar la opción de COLEGIATURA se tienen que seleccionar el numero de semanas a pagar

Figura 27.- Formulario de Pago de Colegiatura.

Podemos observar que cuando se trata de colegiatura de inmediato el puntero se posiciona en el numero de semanas que se van a cubrir para el alumno seleccionado.

Y posteriormente se visualizará otra ventana que a continuación se muestra:

12- una vez seleccionando el número de semanas a pagar aparecerá la siguiente ventana

13- la ventana muestra la descripción del No de SEM. Que se están cubriendo

14- muestra el importe con letra

15- y muestra el total a pagar con número detalladamente con descuentos y recargos en cero porque no se han aplicado mostrando el total a pagar

Figura 28.- Ventana del Estado General de Pagos del Alumno.

Hasta este momento se ha realizado el pago de colegiatura de manera directa sin aplicar NI DESCUENTOS NI RECARGOS, a continuación se muestra esta parte:

The screenshot shows a software interface for payments. At the top, there's a title bar 'PAGOS' and a menu with options: 'NUEVO PAGO', 'ACTUALIZAR PAGO', 'MODIFICAR', and 'SALIR DE ESTA OPCION'. Below the menu, there are fields for 'Tipo de Pago' (Interno/Externo), 'MATRICULA' (4), 'NOMBRE DEL CLIENTE', and 'CONCEPTO' (Colegiatura). A date '31/10/2005' and other financial data are visible on the right. A dialog box titled 'APLICAR DESCUENTOS / RECARGOS' is open, showing 'DESCUENTO' and 'RECARGO' options. Below the dialog, there's a table 'SEMANAS A PAGAR' and a summary section with 'SUBTOTAL \$ 240', 'DESCUENTO \$ 0', 'RECARGOS \$ 0', and 'TOTAL A PAGAR \$ 240'. A button 'ACERTAR' is also present.

| Semana | Fecha de Pago | Total de Pago |
|--------|---------------|---------------|
| 41 | 31/10/2005 | 80 |
| 42 | 31/10/2005 | 80 |
| 43 | 31/10/2005 | 80 |

Subtotal: \$ 240
Descuento: \$ 0
Recargos: \$ 0
Total a pagar: \$ 240

Una vez realizado el pago si queremos aplicarle algún descuento o recargo dar clic en el icono señalado

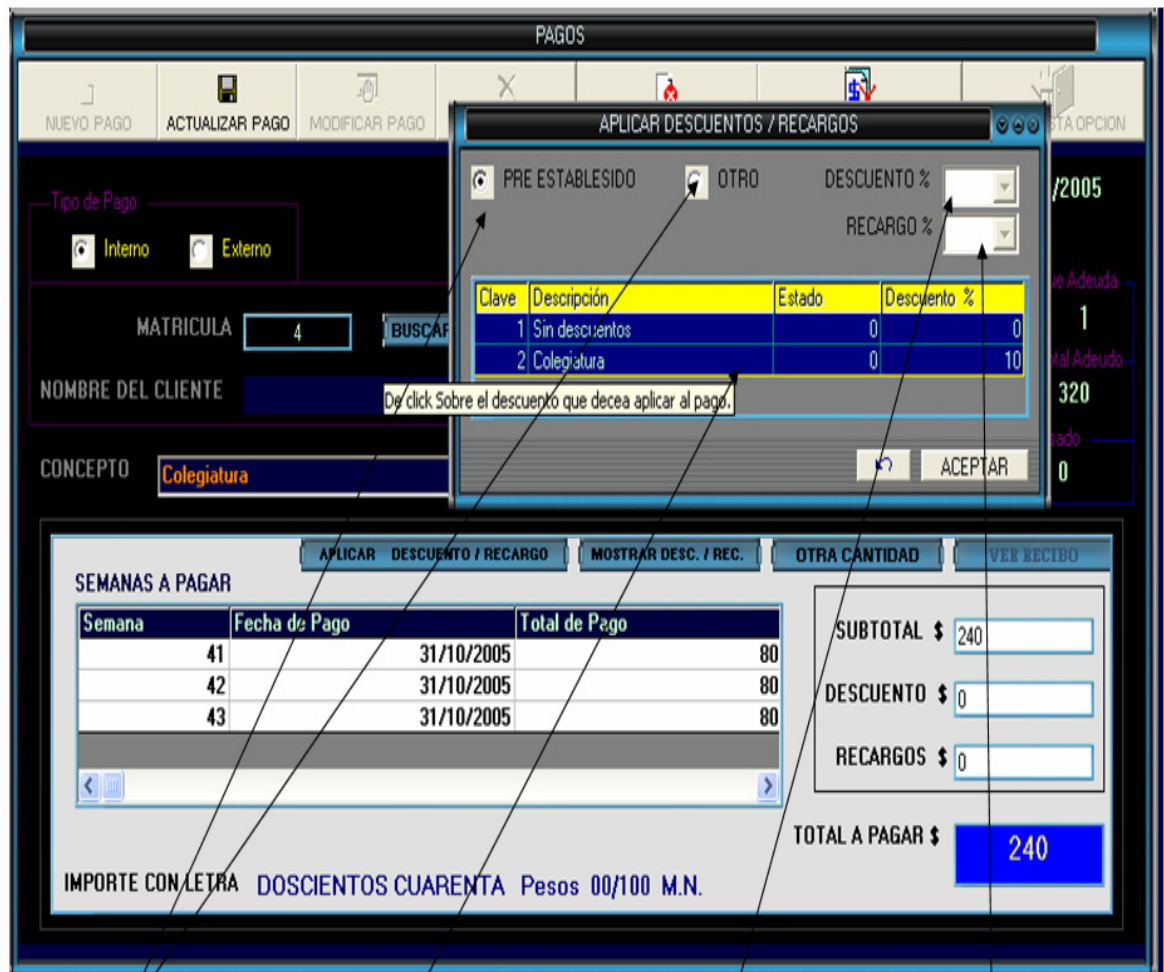
Aparecerá la siguiente ventana en donde se puede visualizar si quiere aplicar ya sea un descuento o un recargo al pago pero no los dos juntos

Este botón sirve para deshacer la operación realizada

Y este para aceptar la operación realizada

Figura 29.- Formulario de Descuentos.

Ya activamos la pantalla para realizar el descuento o recargo al pago realizado ahora realizaremos la actividad.



Si elegimos aplicar descuento aparecen dos opciones PREESTABLECIDO y OTRO

Si se elige PREESTABLECIDO aparecerá el desglose de conceptos de descuento para que de clic sobre el deseado

Si decide aplicar un descuento que no existe en establecido entonces seleccione OTRO y podrá elegir el deseado en esta parte

Si decide aplicar un recargo solo podrá hacerlo desde esta parte

NOTA: si se equivoco en la operación o quiere cambiar de DESCUENTO A RECARGO o viceversa presione el botón ATRAS

Figura 30.- Aplicación de Descuentos

Ya vimos cómo aplicar los descuentos y recargos pero aún no se aplican al pago, para poder aplicarlos al pago siga los pasos siguientes:

The screenshot shows a software interface for applying payments. At the top, there are navigation buttons: NUEVO PAGO, ACTUALIZAR PAGO, MODIFICAR PAGO, BORRAR PAGO, CANCELAR NUEVO PAGO, CANCELAR PAGO REALIZADO, and SALIR DE ESTA OPCION. The main area contains fields for 'Tipo de Pago' (Interno/Externo), 'FECHA' (31/10/2005), 'FOLIO' (41), 'MATRICULA' (4), 'NOMBRE DEL CLIENTE' (Leonardo Guevara Ostría), 'CONCEPTO' (Colegiatura), and 'No. DE SEMANAS A PAGAR' (3). A table titled 'SEMANAS A PAGAR' lists weeks 41, 42, and 43 with their respective dates and a total of 80 for each. To the right, a summary shows 'SUBTOTAL \$ 240', 'DESCUENTO \$ 24', 'RECARGOS \$ 0', and 'TOTAL A PAGAR \$ 216'. A button 'MOSTRAR DESC. / REC.' is highlighted with an arrow. Below the interface, three arrows point to the 'MOSTRAR DESC. / REC.' button, the 'TOTAL A PAGAR' field, and the 'DESCUENTO' field, with corresponding explanatory text.

| Semana | Fecha de Pago | Total de Pago |
|--------|---------------|---------------|
| 41 | 31/10/2005 | 80 |
| 42 | 31/10/2005 | 80 |
| 43 | 31/10/2005 | 80 |

Summary values:
SUBTOTAL \$ 240
DESCUENTO \$ 24
RECARGOS \$ 0
TOTAL A PAGAR \$ 216

Una vez que aplico el descuento o recargo y le dio ACEPTAR de clic en el botón MOSTRAR para aplicarlo al pago

Una vez que dio clic en MOSTRAR se vera alterada el descuento o recargo según fue el caso

Pr lo tanto también se vera alterado el total tanto en letra como en numero

Figura 31.- Aplicación de Descuentos o Recargos.

Hasta ahora hemos visto cómo realizar un pago interno pero por el concepto de COLEGIATURA ahora vamos a ver el caso de:

Un pago interno por un concepto diferente al de COLEGIATURA.

Para este caso, es muy similar a lo anterior, excepto que la pantalla de TOTAL cambia como se muestra a continuación:

Como podemos darnos cuenta es un usuario interno

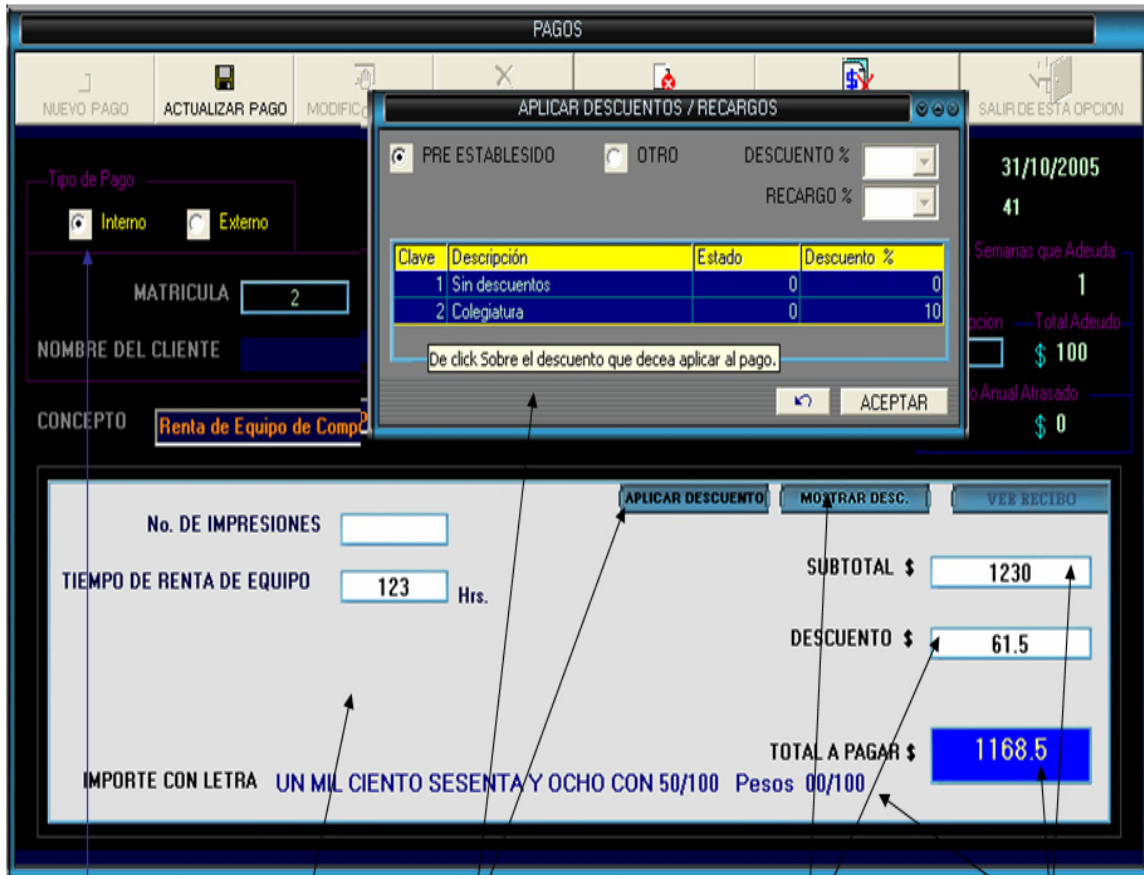
El concepto es diferente al de colegiatura en el momento que es seleccionado se activa la siguiente pantalla con los totales automáticamente

En el caso que el concepto se trate de IMPRESIONES O RENTA DE EQUIPO el sistema permitirá introducir la cantidad o el tiempo según sea el caso en esta parte presionando la tecla enter al introducir los datos

Dependiendo el concepto aparecerán los totales automáticamente y también con letra

Figura 32.- Pagos por Concepto Diferente al de Colegiaturas.

Pero también se tiene la opción de aplicar DESCUENTOS pero no REGARGOS (los recargos solo funcionan con el concepto de COLEGIATURA) a continuación se describe esta opción:



Como podemos darnos cuenta es un usuario interno con un concepto diferente al de COLEGIATURA

Tenemos la pantalla de totales

Ahora le aplicaremos descuento al pago presionamos el botón APLICAR DESCUENTO recuerde que los recargos no se pueden aplicar en esta opción y aparecerá la pantalla de descuentos

En donde una vez aplicado el descuento y haber presionado el botón ACEPTAR debemos dar clic en el botón MOSTRAR Desc para aplicar el Desc. Al pago

Una vez que se presiono el botón MOSTRAR Desc los total se alteraran con nuevo resultado

Figura 33.- Cobro de Recargos.

Toca el turno de los pagos realizados por usuarios externos (pagos externos).

Es muy sencillo a un más que los pagos INTERNOS, sólo siga los pasos descritos a continuación:

1.- Abra el modulo de PAGOS, hasta este momento se considera que usted ya tiene el conocimiento de ingresar al módulo ya que fue explicado anteriormente, usted ingresará a la siguiente ventana:

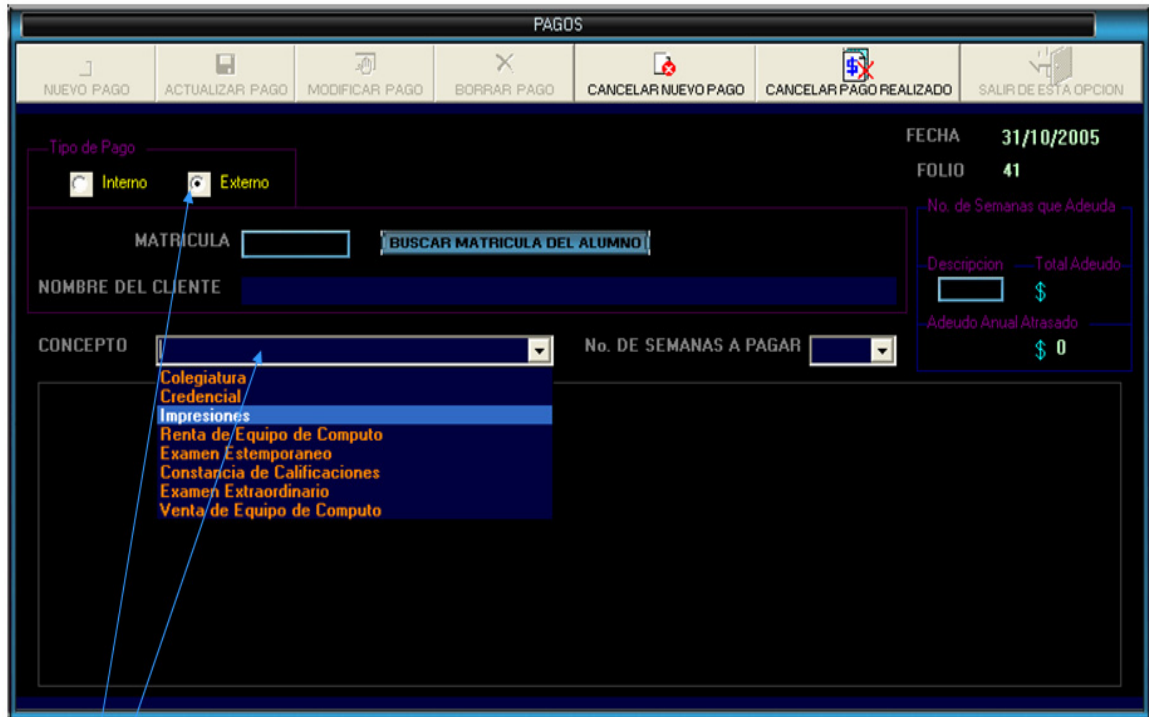
De clic sobre el botón NUEVO PAGO para realizar un nuevo pago

Presione el botón CANCELAR para cancelar el pago en caso de que ya no quiera realizarlo

NOTA: solo se puede cancelar un pago si no se a grabado

Figura 34.- Formulario de Pagos.

Una vez que usted se encuentra en la ventana de pagos siga los éstos pasos (solo cuando se trata de un usuario EXTERNO, es decir, que no es alumno de la institución educativa):



Una vez que da de alta un nuevo pago y este es externo automáticamente el puntero se posiciona en el CONCEPTO DE COBRO ignorando la matricula del alumno ya que es externo

Figura 35.- Pasos para Pagos Externos (paso 1)

El procedimiento es más fácil, ya que sólo debe seleccionar el concepto de COBRO, una vez que fue seleccionado, aparecerá la pantalla que se mostrará en la siguiente figura (ver figura 36), en donde el total ya es automático excepto para cuando son IMPRESIONES O RENTA DE EQUIPO DE CÓMPUTO porque hay que establecer la cantidad o el tiempo de renta según sea el caso.

En los pagos externos también puede realizar descuentos, recuérdese que los RECARGOS no existen en los pagos EXTERNOS y se realizan exactamente igual que en los pagos INTERNOS.

The screenshot displays a payment system interface with the following elements:

- Menu:** NUEVO PAGO, ACTUALIZAR PAGO, MODIFICAR PAGO, BORRAR PAGO, CANCELAR NUEVO PAGO, CANCELAR PAGO REALIZADO, SALIR DE ESTA OPCION.
- Form Fields:**
 - Tipo de Pago: Interno (selected), Externo.
 - MATERICULA: [Empty field] with button BUSCAR MATRICULA DEL ALUMNO.
 - NOMBRE DEL CLIENTE: [Empty field]
 - CONCEPTO: Venta de Equipo de Computo (selected).
 - No. DE SEMANAS A PAGAR: [Empty field]
- Summary Box:**
 - Buttons: APLICAR DESCUENTO, MOSTRAR DESC., VER RECIBO.
 - Inputs: No. DE IMPRESIONES, TIEMPO DE RENTA DE EQUIPO [Empty] Hrs.
 - Values: SUBTOTAL \$ 6000.00, DESCUENTO \$ 0, TOTAL A PAGAR \$ 6000.
 - Note: IMPORTE CON LETRA SEIS MIL Pesos 00/100 M.N.
- Right Panel:** FECHA 31/10/2005, FOLIO 41, No. de Semanas que Adeuda, Descripción, Total Adeudo, Adeudo Anual Atrasado \$ 0.

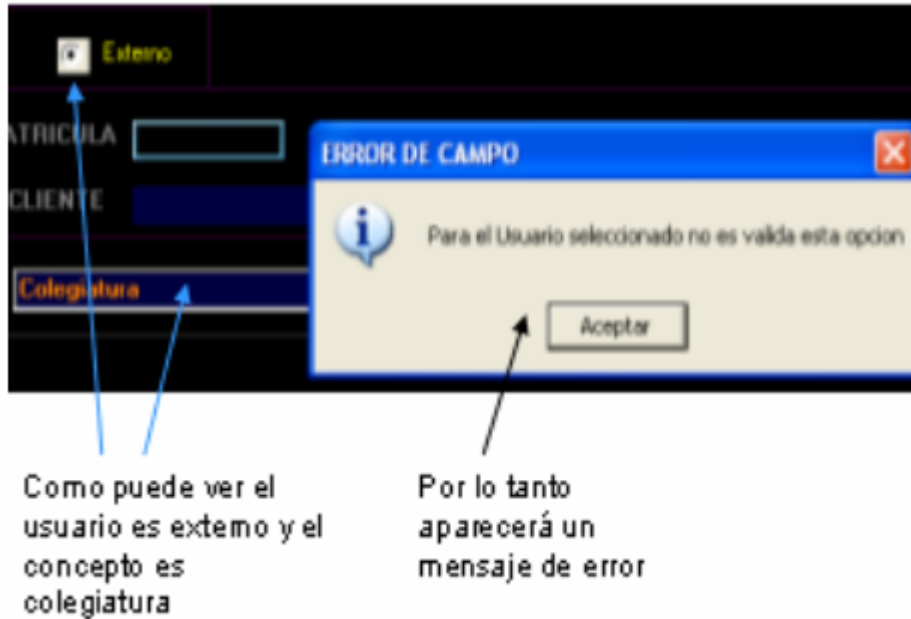
Una vez seleccionado el servicio aparecerá la pantalla de totales con el total dependiendo el costo del servicio el pago es realizado directamente sin aplicar ningún descuento RECUERDE que para los usuarios externos no existen recargos

Figura 35.- Pasos para Pagos Externos (paso 2)

Como pudo ver el proceso es más sencillo ya que solo necesita seleccionar el servicio

NOTA:

Un usuario externo no puede realizar un pago por concepto de COLEGIATURA, el sistema no lo permitirá ya que es un usuario externo a la institución, y por lo tanto mostrará mensaje de error como se muestra continuación.



11

Figura 36.- Mensaje de Error en Pagos Externos.

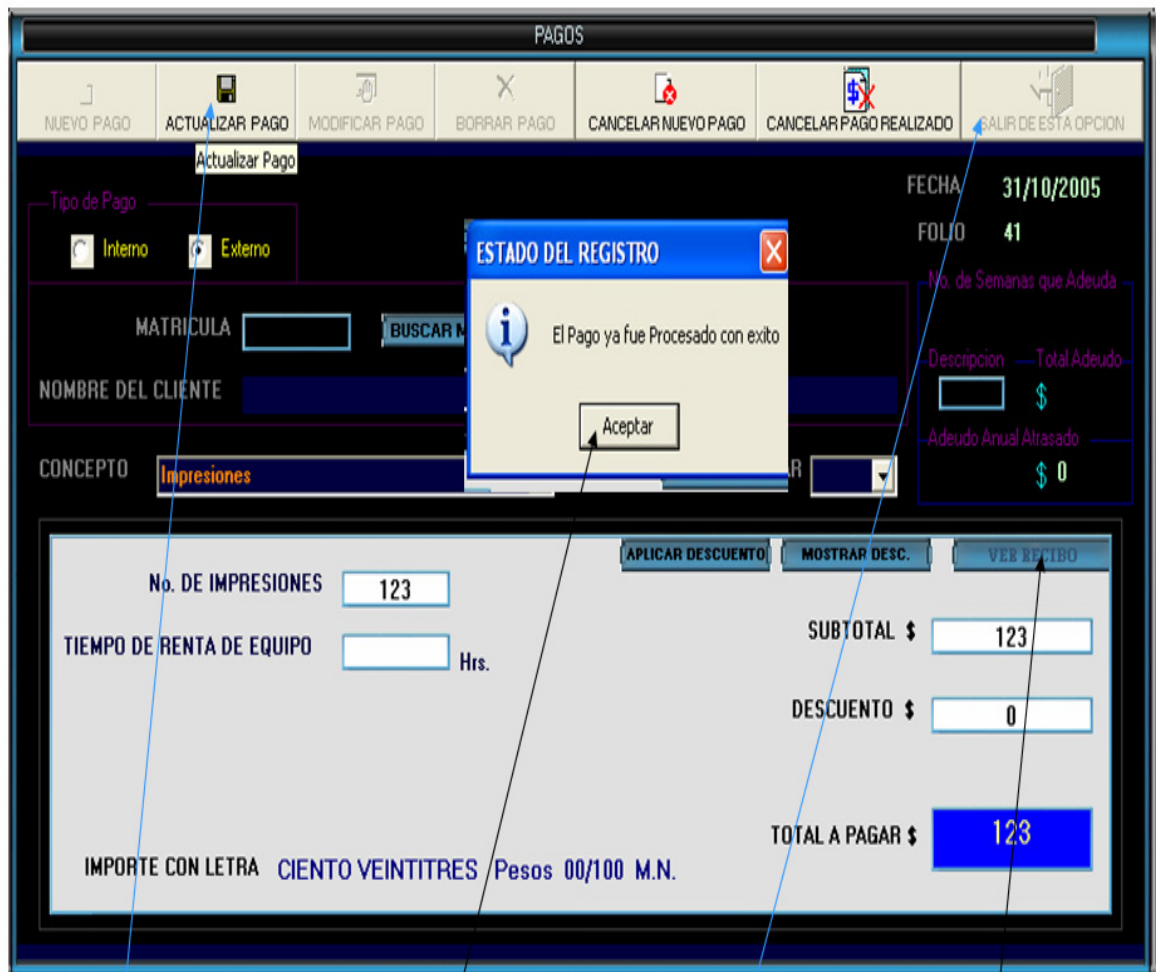
Hasta ahora todo lo que hemos hecho es el explicar como realizar un pago tanto INTERNO como EXTERNO pero ahora vamos a explicar la parte posterior al pago.

Una vez que sabe usted como usuario que el tipo de pago y que los datos están correctos entonces, es la hora de actualizar los pagos (anexarlo a la base de datos) debemos guardar los datos.

Nota:

Una vez actualizados los datos ya no hay manera de borrarlos solo hay una manera de que se pueda restar el dinero de la cantidad total y es cancelando el pago, pero se explicará más adelante.

Veamos el proceso de grabación, este proceso es igual para los dos tipos de pagos tanto INTERNOS CONO EXTERNOS, es lo único que no cambia:



Una vez que el pago es correcto proseguimos a grabar datos, damos clic en el botón ACTUALIZAR PAGO

Después de presionar el botón de GRABAR aparecerá el siguiente mensaje indicando que los datos fueron actualizados con éxito y le damos aceptar

Si nos damos cuenta no podemos salirnos de la pantalla de pagos esto es por que no hemos impreso el resibo

Una vez que le dimos ACEPTAR al botón del mensaje se activara el botón de VER RESIBO Y al darle clic podremos ver el resibo

Figura 37.- Pasos para el Procesamiento de Grabado de un Pago.

La siguiente figura muestra el modelo de recibo manejado en el sistema:

| CONCEPTO: | SUBTOTAL \$: | DESCUENTOS \$ |
|--------------------|-------------------------------------|----------------------------|
| Impresiones | 123 | 0 |
| RECARGOS \$: | 0 | TOTAL EN SERVICIOS \$: 123 |
| IMPORTE CON LETRA: | CIENTO VEINTITRES Pesos 00/100 M.N. | TOTAL EN RECARGOS \$: 0 |
| | | TOTAL A PAGAR \$: 123 |

Una vez presionado el botón VER RECIBO veremos el recibo

Como podemos ver el sistema esta dividido en dos partes de las cuales la que esta en la parte izquierda es para el cliente y la derecha es un comprobante de que se entrego ese recibo para el cajero

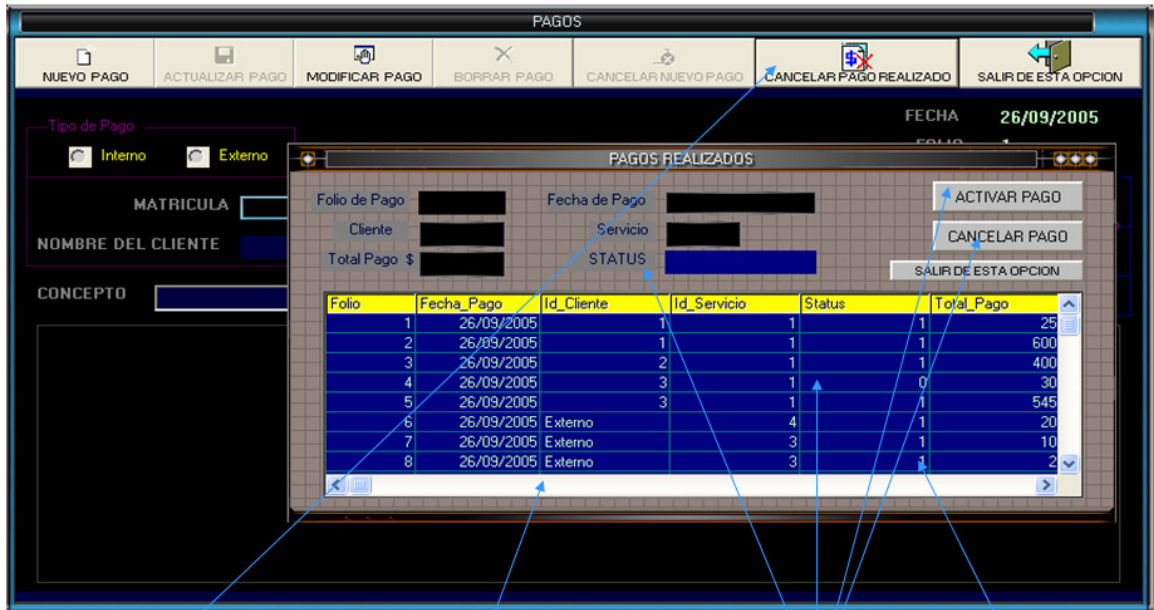
Presione el siguiente botón para imprimir el recibo y regresar a la pantalla de PAGOS

Figura 38.- Modelo de Recibo (Comprobantes de Pago)

Una vez impreso el recibo regresaremos a la pantalla de pagos y podremos realizar otro pago.

Otro punto importante es cuando ya realizamos un pago y ya lo actualizamos (se encuentra almacenado en la base de datos) y por algún motivo se requiere cancelar el pago, por que como se explico anteriormente los pagos que ya fueron actualizados no pueden ser borrados, SISTEMA DE CONTROL DE PAGOS (SICEP) cuenta con una opción para cancelar pagos que ya fueron realizados esto con el fin de que puedan ser restados del corte de caja una vez que el cajero responsable lo realice.

A Continuación se muestra como acceder a esta opción:



La opción se encuentra dentro de la pantalla de pagos haciendo clic en el botón CANCELAR PAGO REALIZADO

Usted accederá a la siguiente pantalla

En donde usted podrá recorrer la lista de pagos que fueron realizados y podrá seleccionar el que le interese CANCELAR O ACTIVAR, la información del registro que selecciono se colocara en las casillas de arriba

Podemos darnos cuenta que el pago fue cancelado o activado con el STATUS si es "1" esta activado si es "0" esta cancelado

Figura 39.- Cancelación de Pagos Previamente Registrados.

El pago puede ser activado o cancelado y esto lo podemos visualizar en el STATUS como se muestra en la figura anterior (ver figura 39).

NOTA:

Los pagos cancelados aparecerán en el TICKET DE CORTE como total en recibos cancelados, pero esto no quiere decir que por el hecho de estar cancelado el dinero correspondiente a dicha cancelación ya se perdió y no aparecerá en los INGRESOS ni el TICKET aparecerá como dinero cancelado y estará sujeto a verificación (las razones por las cuales se canceló).

Hasta aquí ya se explicó todo lo relacionado a pagos, a continuación se muestra cómo manipular el catalogo de CONCEPTOS DE COBRO.

DANDO DE ALTA CONCEPTOS DE COBRO

El SISTEMA DE CONTROL DE PAGOS (SICEP) cuenta con un catálogo de conceptos de cobro, el cual permite dar de alta nuevos CONCEPTOS DE COBRO sólo a personas autorizadas, asimismo, dar de baja conceptos de COBRO que ya no se utilicen, a continuación se muestra este procedimiento y los pasos a seguir:

1.- Usted puede ingresar al catálogo de CONCEPTOS DE COBRO, ésto debe realizarse desde el menú principal en el menú ARCHIVO y CONFIGURAR CONCEPTOS DE COBRO o si lo desea desde la barra de iconos dando click en el icono CONCEPTOS DE COBRO.

Aparecerá la siguiente ventana:



Esta es la pantalla de CONCEPTOS DE COBRO la cual permite dar de alta nuevos conceptos de cobro

En esta parte muestra el listado de todos los conceptos de cobro dados de alta, el cual podemos ver su descripción dando doble clic sobre el registro con el Mouse y los datos aparecerán en la parte de arriba

En la parte superior derecha aparecen los botones necesarios para realizar la manipulación de los conceptos de cobro

Figura 40.--Altas y bajas de los Conceptos de Cobro.¶

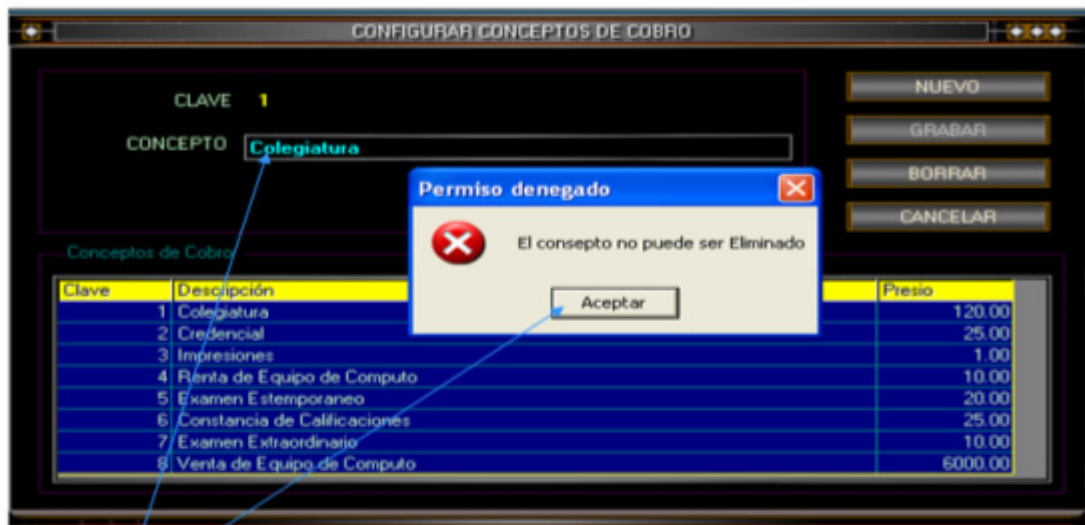
Todos los conceptos de cobro que se encuentren dados de alta, son utilizados para realizar los pagos respectivos, usted como usuario tiene la libertad de dar de alta cualquier cantidad de conceptos de cobro de acuerdo a sus necesidades, pero como en todo sistema la seguridad es primero existen conceptos que son parte

importante y predeterminada del sistema y por lo tanto no pueden ser borrados porque sin ellos el sistema dejaría de realizar algunas funciones vitales.

Entre los conceptos que se encuentran predeterminados y que no pueden ser borrados se encuentran:

- ✓ Colegiatura
- ✓ Renta de Equipo de Computo
- ✓ Impresiones

Estos conceptos no pueden ser borrados, si intenta borrarlos aparecerá mensaje de error como se muestra a continuación:



En el momento que seleccione algún concepto que sea protegido por el sistema, y desee borrarlo aparecerá mensaje de error indicando que es un concepto protegido

Figura 41.- Error de Borrado de Conceptos Predeterminados.

Pasemos ahora a dar de alta los descuentos en el CATÁLOGO DE DESCUENTOS.

CÓMO DAR DE ALTA DESCUENTOS AUTORIZADOS EN EL CATÁLOGO DE DESCUENTOS.

Como ya vimos que el catálogo de conceptos de cobro es importante para poder llevar a cabo los pagos, también el contar con un catalogo de DESCUENTOS es importante para poder aplicar los DESCUENTOS (solo cuando sean necesarios) a los pagos, esta opción es importante ya que de alguna manera cualquier tipo de pago por muy pequeño que sea alguna que otra vez es necesario aplicarle algún descuento ya sea autorizado o no autorizado.

La forma de acceder al CATÁLOGO DE DESCUENTOS tiene 2 formas una es que puede acceder desde el MENÚ ARCHIVO y CONFIGURAR DESCUENTOS ó si lo prefiere puede acceder desde una manera mas rápida esto en la barra de iconos dando click en el icono Descuentos y listo usted abrirá la pantalla de Descuentos la cual se muestra a continuación:

| Clave | Descripción | Estado | Descuento % |
|-------|----------------|--------|-------------|
| 1 | Sin descuentos | 0 | 0 |
| 2 | Colegiatura | 0 | 10 |

Esta es la pantalla de descuentos la cual cuenta con el desglose detallado de cada uno de los descuentos dados de alta

Usted puede ver de forma detallada cada uno de los registros con dar doble clic con el Mouse y los datos aparecerán en la parte de arriba los cuales usted puede manipular ya sea borrar, nuevo, etc.

Figura 42.- Formulario del Catálogo de Descuentos.

Como puede ver para manipular el catálogo de DESCUENTOS es similar al catálogo de conceptos de cobro.

COMO REALIZAR EL CORTE DE CAJA

El corte de caja es una medida de seguridad para la persona responsable en ese momento, es necesario realizar un corte de caja aún si el usuario es el único que maneja el sistema ya que alguna manera se debe comprobar cuánto vendió y debe coincidir con el capital físico existente en caja.

Cuando el cajero es más de uno, el corte de caja es indispensable. El SISTEMA permite realizar cortes de caja diariamente, con la finalidad de que la información sea correcta y el cajero este respaldado (con su respectiva impresión del corte), al mismo tiempo se sepa quién cobró un determinado recibo, el motivo del por qué realizar corte de caja diario, es porque, los folios de los recibos deben ser continuos y si el cajero no respetara el número de folio del recibo y normaliza el corte de caja hasta el día siguiente, habrá otra persona intermedia (cajero de segundo turno) que realizará otros cobros, por lo tanto, los folios inicial y final ya no coincidirán.

Por ejemplo:

Supogamos que en la empresa contamos con 2 cajeros el primero (cajero1) tiene el turno de 7:30 a.m. a 2:00 p.m. y el segundo cajero (cajero2) tiene el turno de 2:00 p.m. a 8:00 p.m.

Si cajero1 inicia el día con el folio inicial 35 y el folio final es 70 y no realiza corte de caja ese día al terminar su turno de trabajo, entra un segundo cajero2 inicia en el folio 71 y el final es 80, y si realiza corte de caja el cajero 2 no tendrá problemas con el corte pero sigamos analizando el caso del cajero1.

Al día siguiente llega el cajero1 y **si realiza su corte de caja del día anterior ANTES DE INICIAR A UTILIZAR NUEVOS FOLIOS (ANTES DE EMPEZAR A COBRAR)** no habrá ningún problema, porque cuando realice el corte de caja los folios no estarán alterados y se imprimirán como folio inicial 35 y final 70.

De lo contrario, si cajero1 llega al día siguiente **y no realiza su corte de caja del día anterior y comienza a trabajar utilizando nuevos folios, folio inicial 81 y final 95, CUANDO QUIERA REALIZAR SU CORTE DE CAJA LOS FOLIOS DE LOS DOS DÍAS SE APARECERÁN JUNTOS EN EL MISMO CORTE Y MANDARÁ INFORMACION ERRÓNEA**, en éste caso, se mandará el folio inicial 35 hasta el folio 95 y si nos damos cuenta entre el folio 35 y 95 se encuentra el cajero2 con sus folios del 71 al 80, por lo tanto, los folios con todos sus datos serán erróneos por eso el SICEP permite realizar los CORTES DE CAJA DIARIOS ó como ya se explicó, si se olvidó realizar el corte de caja del día anterior puede realizarlo al día siguiente o el día que se requiera (por si no trabaja sábados o domingos) PERO ANTES DE EMPEZAR A TRABAJAR CON PAGOS.

NOTA:

PRIMERO QUE NADA REALICE EL CORTE DEL DIA ANTERIOR antes de comenzar a trabajar con nuevos folios a fin de no tener problemas.

A continuación se muestra la pantalla de CORTE DE CAJA.

| No. PAGO | FECHA DE PAGO | CLAVE DEL CUENTE | CLAVE DEL SERVICIO | ESTADO DEL PAGO | TOTAL PAGADO |
|----------|---------------|------------------|--------------------|-----------------|--------------|
| | | | | | |

Figura 43.- Formulario de Corte de Caja.

Para realizar su corte de caja de lo cobrado durante el día es muy sencillo, usted puede ingresar a la pantalla que se muestra en la figura 43 y, con sólo dar click en el botón CORTE DE CAJA ubicado en la barra de iconos, ingresará en la pantalla de corte de caja el nombre del cajero responsable (el sistema lo tendrá cargado automáticamente en la sección de la lista).

Al dar click en CAJERO RESPONSABLE aparecerá el listado de todos los cajeros activos, seleccione el nombre y será lo único que necesitará hacer para realizar su corte de caja ya que el sistema arroja los datos de lo cobrado automáticamente, entonces usted dará click en el BOTON REALIZAR CORTE y posterior al click, en

el botón TICKET DE CORTE donde aparecerá el resumen de lo cobrado listo para imprimir y automáticamente usted habrá realizado su corte de caja exitosamente.

NOTA:

Si usted intenta realizar el corte de caja de un cajero "x", el sistema SIGEP no se lo permitirá ya que por seguridad de los demás cajeros y usted mismo solo el cajero en uso puede realizar su corte de caja de lo contrario aparecerá un mensaje de error indicando que no es el cajero en uso.

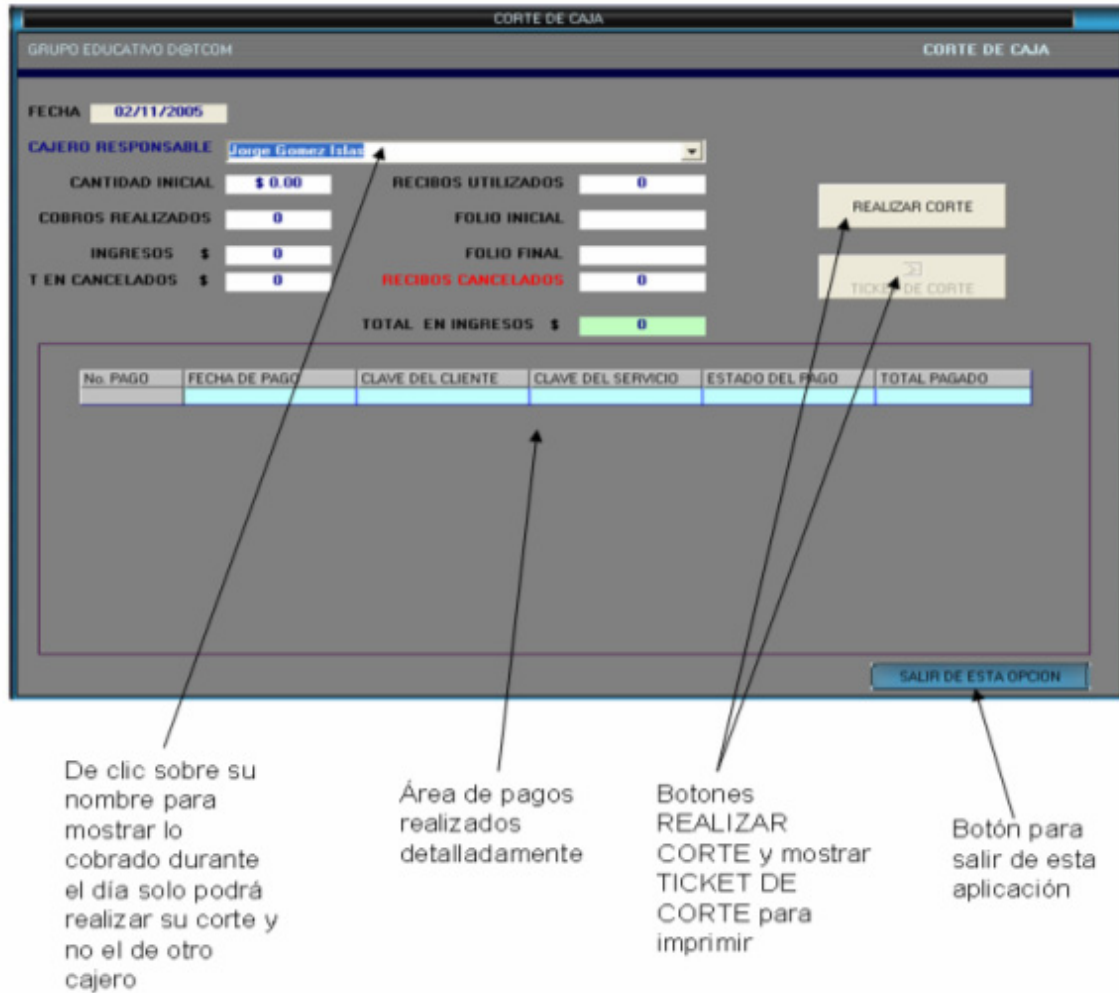


Figura 44.- Formulario de Corte de Caja.

Hasta este momento ya vimos desde cómo se realizan los pagos hasta realizar un corte de caja, ahora toca el turno de los reportes que arroja el sistema como se muestra a continuación.

CONSULTAS DE LA INFORMACION

El SICEP también cuenta con consultas de información las cuales son de los siguientes tipos:

- ✓ Adeudos por concepto de COLEGIATURA (semanal)
- ✓ Ingresos generales (todos los conceptos) semanales
- ✓ Ingresos por concepto de COLEGIATURA (de forma desglosada)
- ✓ Consulta de cortes realizados (todos)

Veamos el primero **ADEUDOS POR CONCEPTO DE COLEGIATURA**

Usted puede acceder a este tipo de consulta si lo desea desde el menú CONSULTAS y ADEUDOS SEMANALES, ó si lo desea desde la barra de iconos y ADEUDOS, de ésta forma usted ingresará a la siguiente ventana:

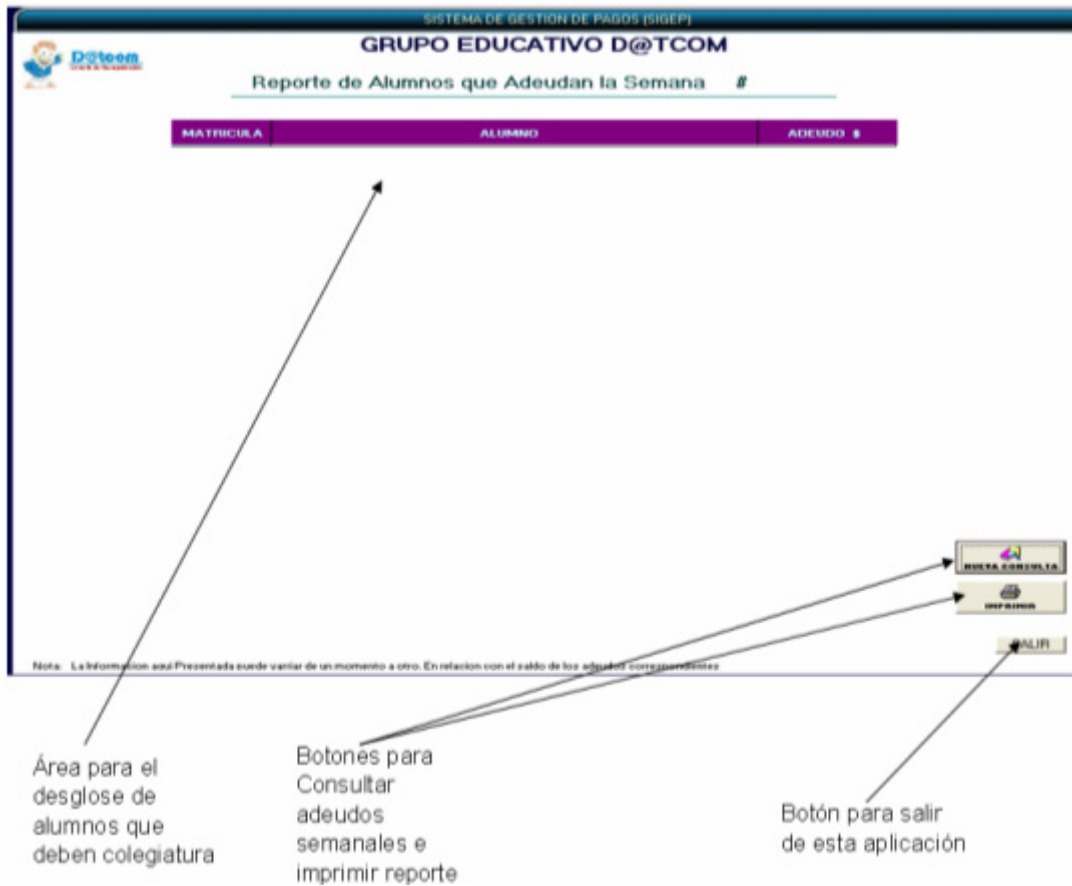
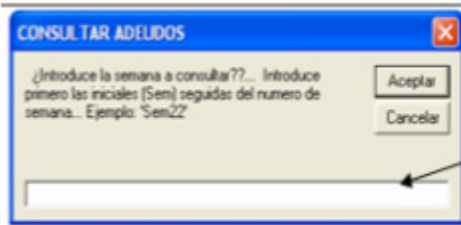


Figura 45.- Formulario de Consultas.

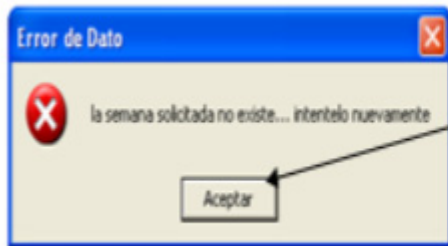
En el momento que usted realiza una nueva consulta debe colocar "Sem" y seguido el numero de semana que desea consultar sin dejar espacios en la sección que se muestra como sigue:



Área para introducir el numero de semana a consultar debe introducir "Sem" seguido del numero de semana a consultar

Figura 46.- Ventana de Introducción de Semana.

Si la semana es introducida incorrectamente ya sea, porque no exista o porque el formato no lo introdujo como debe ser, aparecerá el siguiente mensaje de error:



Mensaje de error en caso de que la semana se incorpore de forma incorrecta dar clic en ACEPTAR

Figura 47.- Mensaje de error.

De lo contrario, si la semana es correcta se desglosará el listado de los alumnos que adeudan la semana solicitada como se puede ver en la figura 48.

| SISTEMA DE GESTION DE PAGOS (SIGEP) | | |
|--|----------------------------|-----------|
| GRUPO EDUCATIVO D@TCOM | | |
| Reporte de Alumnos que Adeudan la Semana | | # 43 |
| MATRICULA | ALUMNO | ADEUDO \$ |
| 1 | Carlos Escorcía Díaz | 120 |
| 3 | Victor Telles Peralta | 115 |
| 4 | Leonardo Guevara Ostría | 80 |
| 5 | Osbaldo Fernandez Hrnandez | 90 |

Figura 48.- Resultados de la Consulta.

INGRESOS GENERALES

Usted puede acceder a este tipo de consulta si lo desea desde el menú CONSULTAS e INGRESOS SEMANALES o si lo desea desde la barra de iconos e INGRESOS y listo usted ingresará a la siguiente ventana (ver figura 49):

| CLAVE SERV. | SERVICIO | INGRESO TOTAL \$ | ESTADISTICA DE INGRESOS | |
|-------------|----------|------------------|---------------------------|----------|
| | | | No. DE RECIBOS: | |
| | | | DEL FOLIO | AL FOLIO |
| | | | | |
| | | | INGRESOS: \$ | |
| | | | No. DE RECIBOS CANCELADOS | |
| | | | T. EN PAGOS CANCELADOS: | |
| | | | \$ | |
| | | | TOTAL DESCUENTOS: | |
| | | | \$ | |
| | | | TOTAL RECARGOS: | |
| | | | \$ | |
| | | | TOTAL INGRESOS: | |
| | | | \$ | |

Área para el desglose de conceptos de cobro

Resumen de los ingresos realizados en la semana

Botón para consultar detalladamente los ingresos por concepto de COLEGIATURA

Botón para consultar los ingresos generales por todos los conceptos

Figura 49.- Reporte de Ingresos Semanales.

En el momento que usted requiera consultar los ingresos generales, aparecerá la siguiente ventana (ver figura 50) donde debe ingresar el numero de semana a consultar (solo el numero).

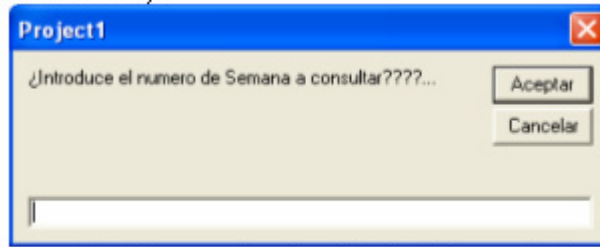


Figura 50.- Ventana Introducción del Número de Semana.

Si en el número de semana introducido no fue realizado ningún pago, el sistema mandará un mensaje de información (ver figura 51) indicando que no hay datos que mostrar.

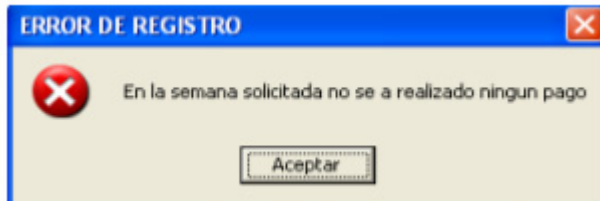


Figura 51.- Mensaje de error por la inexistencia de datos.

De lo contrario mostrara la información solicitada (ver figura 52):



| CLAVE SEM | SERVICIO | INGRESO TOTAL | ESTADISTICA DE INGRESOS | |
|-----------|------------------------------|---------------|----------------------------|----------|
| 1 | Colegiatura | 1055 | No. DE RECIBOS: | 5 |
| 2 | Credencial | 0 | DEL FOLIO | AL FOLIO |
| 3 | Impresiones | 0 | 36 | 40 |
| 4 | Renta de Equipo de Computo | 0 | INGRESOS: | \$ 1055 |
| 5 | Examen Estemporaneo | 0 | No. DE RECIBOS CANCELADOS: | 0 |
| 6 | Constancia de Calificaciones | 0 | T. EN PAGOS CANCELADOS: | \$ 0 |
| 7 | Examen Extraordinario | 0 | TOTAL DESCUENTOS: | \$ 56 |
| 8 | Venta de Equipo de Computo | 0 | TOTAL RECARGOS: | \$ 58 |
| | | | TOTAL INGRESOS: | \$ 1055 |

Figura 52.- Reporte de Ingresos.

Si usted desea consultar los ingresos semanales por concepto de colegiatura, por medio del icono aparecerá la siguiente ventana (ver figura 52), solicitando el número de semana a consultar.

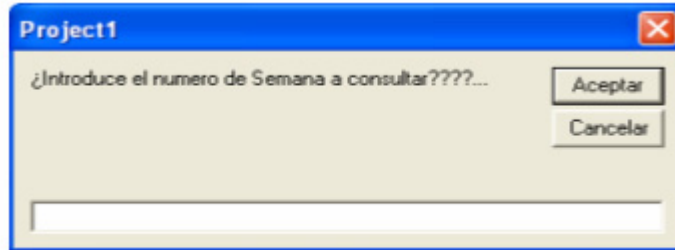
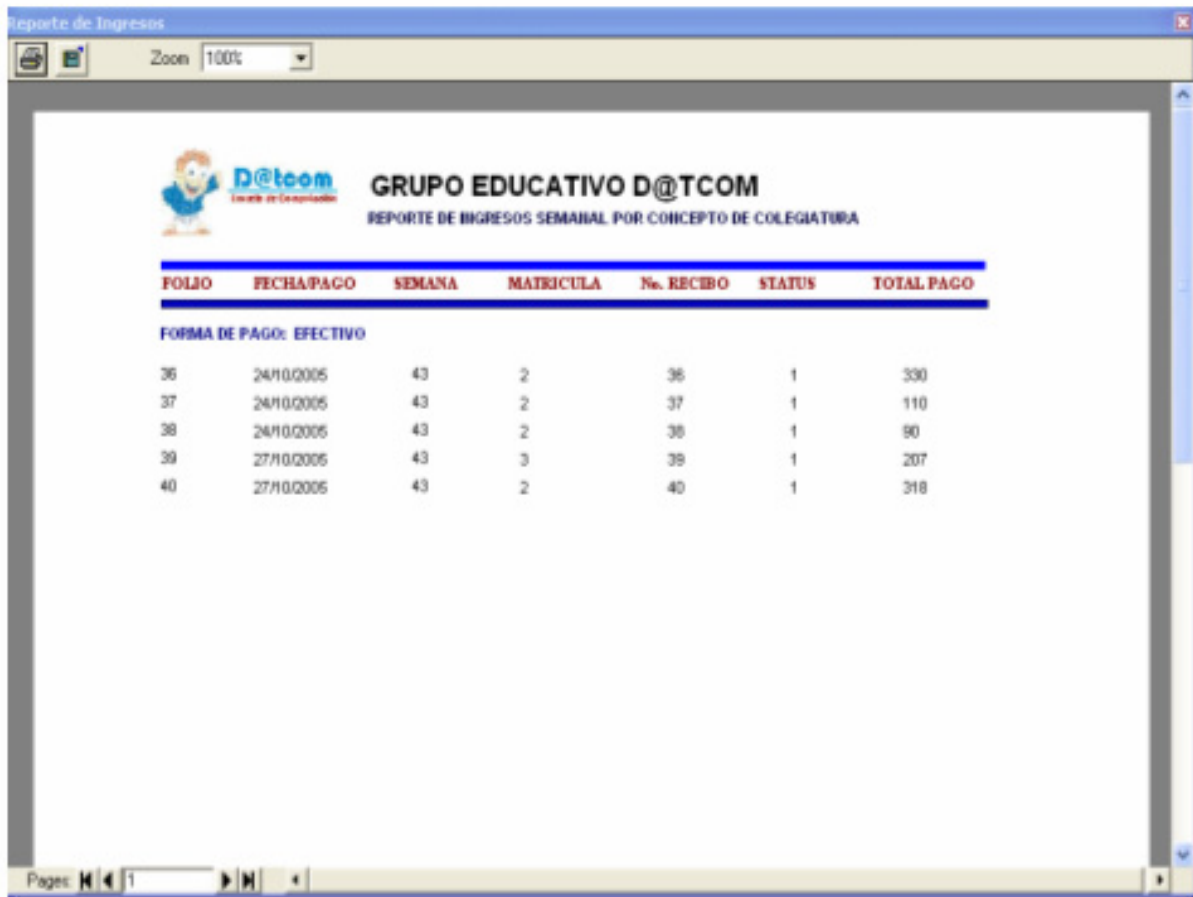



Figura 52.- Error de Borrado de Conceptos Predeterminados.

En el momento que introduzca el número de semana a consultar (sólo el número) la información (reporte) será desglosada como sigue (ver figura 53):



Reporte de Ingresos

Zoom: 100%

 **GRUPO EDUCATIVO D@TCOM**
REPORTE DE INGRESOS SEMANAL POR CONCEPTO DE COLEGIATURA

| FOLIO | FECHA PAGO | SEMANA | MATRICULA | No. RECIBO | STATUS | TOTAL PAGO |
|-------------------------|------------|--------|-----------|------------|--------|------------|
| FORMA DE PAGO: EFECTIVO | | | | | | |
| 36 | 24/10/2005 | 43 | 2 | 36 | 1 | 330 |
| 37 | 24/10/2005 | 43 | 2 | 37 | 1 | 110 |
| 38 | 24/10/2005 | 43 | 2 | 38 | 1 | 90 |
| 39 | 27/10/2005 | 43 | 3 | 39 | 1 | 207 |
| 40 | 27/10/2005 | 43 | 2 | 40 | 1 | 318 |

Pages: 1

Figura 53.- Reporte de Ingresos Semanales.

CONSULTAR CORTES REALIZADOS

Este tipo de consulta se creó para los casos en que el cajero haya extraviado su comprobante de corte de caja o para cualquier tipo de aclaración.

Este tipo de consulta muestra en forma de lista todos los cortes de caja realizados y usted puede acceder a ellos al dar click con el mouse en el registro correspondiente y después en el botón VER CORTE el cual le permitirá ver el corte previamente seleccionado para reimprimir el ticket correspondiente como se muestra a continuación (ver figura 54):

CORTES DE CAJA REALIZADOS

CAJERO : Jorge Gomez Islas

| Fecha Corte | Hora | Cajero | Cobr. Realizados | Rec. Cancelados | Total Pago |
|-------------|---------------|-------------------|------------------|-----------------|------------|
| 02/11/2005 | 07:42:01 a.m. | Jorge Gomez Islas | 0 | 0 | 0 |
| 17/10/2005 | 09:11:29 a.m. | Jorge Gomez Islas | 14 | 2 | 4996.18 |
| 27/10/2005 | 03:58:14 p.m. | teresa | 2 | 0 | 525 |

VER CORTE SALIR

Listado de cortes realizados

En el momento que aparezca el nombre del cajero después de dar clic en esta parte indica que esta activo

Y podemos ver el corte seleccionado

Figura 54.- Formulario de Cortes de Caja Realizados.

Una vez que fué seleccionado el corte de interés y se dé clic el botón VER CORTE aparecerá el TICKET DE CORTE de ese usuario listo para imprimir como se muestra a continuación (ver figura 55):

| TICKET DE CORTE | | |
|-----------------------------------|--------------------------------|-------------------------------|
| CAJERO : <u>Jorge Gomez Islas</u> | FECHA : 17/10/2005 | |
| | HORA : 09:11:29 a.m. | |
| FONDO INICIAL : \$0.00 | TOTAL EN DESCUENTOS \$ 2824.82 | |
| COBROS REALIZADOS : 14 | FOLIO INICIAL : 22 | INGRESOS \$ 5235.18 |
| RECIBOS CANCELADOS : 2 | FOLIO FINAL : 35 | TOTAL EN R. CANCELADOS \$ 239 |
| FIRMA DEL EXPEDIDOR | | TOTAL EN INGRESOS \$ 4996.18 |

Grupo Educativo CD@T.COM [Admon.y Finanzas]

Imprimir Ticket

Figura 55.- Formulario Ticket de Corte.

3.3.- Implementación

Para comenzar vamos a describir las características del equipo de cómputo que se utilizará para el desempeño de las funciones del software, entre las más importantes se encuentran las siguientes:

- ❖ Procesador Intel Celeron a 2.4GHz con 533MHz FSB
- ❖ 512 MB de Memoria RAM DDR 2100
- ❖ Motherboard Intel
- ❖ Tarjeta de Video de 32MB

El sistema desarrollado tendrá su instalación principal en el disco duro, específicamente en el directorio raíz C, así como las bases de datos necesarias para el adecuado control de los datos.

En general, en ésta fase se desarrolla el proceso de captura de todos los datos necesarios para el control de los ingresos, lo que implica el registro de todos y cada uno de los alumnos que integran el plantel en la sección correspondiente del sistema creado, con la finalidad de que en la sección de pruebas se puedan analizar los resultados y compararlos con la realidad.

Desafortunadamente por políticas de seguridad de la información, no se puede publicar un desglose detallado de los datos capturados para cada alumno, pero si podemos decir que los datos mencionados son la base principal para el buen funcionamiento de nuestro sistema ya que, aquí será en donde el sistema deberá trabajar de forma precisa con los registros de control de pago.

Por seguridad de los datos que contiene la computadora principal el sistema será instalado en un equipo alterno el cual funcionará paralelamente a las actividades que se realizan en el departamento de recepción con el objetivo de realizar una comparación de resultados del sistema con los resultados reales de la sección de la institución bajo éste análisis.

3.4.- Pruebas finales.

Para comenzar cabe decir que las pruebas comienzan desde el momento en que se inician los trabajos de implementación y captura de los datos necesarios para el funcionamiento del sistema, como las matrículas y datos generales del alumno, ya que el sistema debe iniciar el envío de los datos hacia las bases de datos correspondientes.

Pero éstas no son las únicas pruebas realizadas, también se debe realizar la verificación de funcionamiento de cada una de las opciones que el sistema tenga programadas, entre ellas podemos destacar:

- ❖ Todas las opciones de pago como:
 - Colegiaturas (ya sea, con descuento, con recargo o simples)
 - Pago de impresiones
 - Pago de consumibles
 - Pago de renta de equipo, etc.
- ❖ Cancelaciones de pagos
- ❖ Modificaciones
- ❖ Conceptos de cobro
- ❖ Conceptos de descuentos
- ❖ Reportes de:
 - Adeudos por semana e
 - Ingresos totales de la institución

Obviamente, ésta fase es una de las más importantes porque aquí podemos darnos cuenta si el sistema esta cumpliendo la funciones para las cuales fue creado (evaluación del cliente) y, si existiera algún error cumplir con la metodología que se está siguiendo, que para el proyecto actual dicha metodología regresará a la fase de planeación para corregir los problemas que pudieran surgir

durante las pruebas necesarias no perdiendo de vista el estándar de calidad que el cliente persiga.

NOTA:

Para no extender demasiado el presente documento se omiten las iteraciones que marca la metodología en espiral.

Conclusiones

Todas las empresas en la gran necesidad de controlar la información se ven inmersas en los avances tecnológicos que puedan dar solución a la agilidad, veracidad, sencillez y economía en el manejo de los datos que las mismas producen.

Es por ello que, cada día se abren nuevas oportunidades para el desarrollo de sistemas que puedan resolver tales problemáticas porque las empresas son tan diversas que en la mayoría de los casos no se puede encontrar la estandarización de los mismos.

El caso de “**D@tcom** Escuela de Computación” es uno de ellos, debido a que son muchas las variables que se manejan en el control de los datos internos de la misma. En el estudio del caso específico de ésta escuela encontramos muy diversos problemas que mediante un poco de razonamiento y estudio se han solucionado demostrando que el dueto inseparable de la teoría en conjunto con la práctica son la solución de innumerables obstáculos.

Como se ha demostrado en el proyecto actual en nuestro país existe gran potencial para desarrollar sistemas de gran calidad teniendo unas bases del conocimiento sumamente sólidas y que solamente hace falta dar el cause que señale los grandes beneficios que se pueden lograr.

Desde hace algunas décadas el hombre ha intentado establecer diversas alternativas de descripción de los sistemas, para de ésta forma mantener un mejor control de los comportamientos de dichos sistemas haciendo uso de las tecnologías que se encuentren a su alcance.

Hoy en día existen métodos distintos que simplifican la tarea, y más aún si la tecnología puede hacer mucho más fácil el manejo de las empresas, es difícil creer que a éstas alturas existan empresas que se resistan al uso de una herramienta tan útil y potente como la computadora.

Más aún cuando sabemos que el mundo se rige por sistemas, debemos darle el lugar que se merece al monstruo voráz de la información, de modo que siempre busquemos la optimalidad del funcionamiento de dichos sistemas.

México, se preocupa cada vez más por tener y crear mejores sistemas que simplifiquen esas tareas, por lo que en el país respecta, debemos aprovechar el potencial de todas aquellas personas que tengan el interés por desarrollar una solución viable para cualquier empresa, negocio ó institución impulsando el desarrollo integral de nuestra nación y lograr tener el poder de la información en la manos de los mexicanos como lo han hecho grandes potencias en el mundo.

Glosario

A.

Administrador de Bases de Datos. (DBA) La persona encargada del mantenimiento de la base, tanto de los datos, su integridad, manejo del gestor de datos, los índices (por corrupción o cualquier motivo), de directorios y alias, consultas y sentencias desde fuera de programas, igual que conexiones o transacciones, etc., como del acceso de usuarios, autorizaciones, palabras de paso. Depende totalmente de la base de datos de que se trate

Análisis Distintos procesos dentro del desarrollo de una aplicación. Generalmente se suele dividir en tres tipos distintos: Análisis de Oportunidad o Previo, donde tan sólo se globaliza el problema y se estudia viabilidad. Análisis Funcional, donde se estudian los objetivos a conseguir bajo un prisma informático y los pasar a dar a nivel de datos, tiempos de procesos, etc. Análisis Orgánico, que define exactamente la aplicación basándose en el análisis funcional, y se crean los diagramas necesarios, generando ya, con los estudios previos, el denominado "cuaderno de carga" con el material suficiente para el desarrollo por los programadores.

Archivo Sinónimo de fichero y esencia del almacenamiento informático. Delimitado por una cabecera y una marca de final, lugar donde puede estar ubicada cualquier tipo de información, ya sea texto, programas, imágenes, sonidos, etc.

B.

Base de Datos Conjunto de ficheros dedicados a guardar información relacionada entre sí, con referencias entre ellos de manera que se complementen con el principio de no duplicidad de datos. Dependiendo de cómo se vinculen dan lugar a B.D. jerárquicas, relacionales, etc. Un caso especial de éstas son las documentales, que, como su nombre indica, están diseñadas para almacenar volúmenes grandes de documentos, lo que genera una problemática distinta por los sistemas de búsqueda.

Base de Datos Orientada a Objetos Las BDOO almacenan y manipulan información que puede ser representada por objetos. Es la evolución de las Bases de Datos para soportar el análisis, diseño y programación Orientado a Objetos. Ver Artículo sobre el tema.

Basic Uno de los lenguajes de programación pioneros en ordenadores domésticos. Con múltiples transformaciones por fabricantes y versiones, en la actualidad ha desembocado en el popular Visual Basic.

C.

Campo Tradicionalmente es la unidad lógica de información. En términos modernos, si un fichero relacional es una tabla, los campos forman las columnas de la misma.

Ciclo de vida de un sistema Desde que se inicia la necesidad de creación de un sistema informático, hasta que se decide que esta obsoleto, los pasos posibles. Como ejemplo, en España, patrocina el Consejo Superior de Informática el llamado Métrica, en él se divide el ciclo en distintas fases: planificación, análisis, diseño, construcción e implementación. Cada una de ellas está subdividida en múltiples subfases o actividades

Clase En programación orientada a objetos, una clase es un nivel más alto que un objeto, de hecho la relación entre ambos es el concepto principal y base de todo el resto. Por decirlo de alguna manera, un objeto es más tangible que una clase, que es un nivel más abstracto (sin ninguna relación con lo que son las clases abstractas, pues tipos de clases hay varios).

Código Es la sucesión de sentencias que dan lugar al programa fuente.

D.

DBA Database Administrator. Ver Administrador de Bases de Datos.

DBMS DataBase Management System, Sistema de utilización de Bases de Datos. Ver SGBD.

Diagrama Es una representación gráfica que sirve para esquematizar un diseño cuando se trata la información. El más común es el diagrama de flujo, que se divide en diagramas de flujo de datos (u organigramas) o diagrama de flujo de programa (u ordinograma).

E.

Encapsulación Una de las tres bases de la Programación Orientada a Objetos y que además diferencia lo que es un Objeto de cualquier otro tipo de código, como una función o un procedimiento. Lo que significa, es casi lo que su nombre indica,

encapsula o aísla lo que el objeto contiene, de manera que sólo es accesible de unas determinadas formas que se han creado en el mismo, unos parámetros o unos métodos precisos.

Estructuras de datos No se debe de confundir con el concepto del mismo nombre de programación estructurada, pues en este caso son propias de lenguajes de programación, principalmente del lenguaje C, y son de especial importancia pues sobre ellas se basan gran parte de las API de Sistemas Operativos como Windows o Unix. Se podría definir como un conjunto de variables de cualquier tipo, "encapsulados" en un nombre de estructura, del tipo: Struct nombre_estructura { Variables que la componen.

F.

Fuente Lenguaje fuente es el que utiliza el programador, con la sintaxis que corresponda y perfectamente legible. A partir de él, mediante los compiladores, salvo en lenguajes de programación concretos, se genera el código que el ordenador es capaz de entender.

G.

GUI Graphic User Interface. Proporciona el cambio de las pantallas conocidas como "de modo texto" a las gráficas, con manejo de puntos en lugar de caracteres.

H.

Herencia Termino utilizado en Programación Orienta a Objetos. Teniendo un objeto con unas características concretas (ascendiente), se podría definir como herencia a la derivación de éste en otro nuevo que incluye sus características mas las que se incluyan (descendiente) Hay una herencia simple. Imagínate un diagrama en forma de árbol, de una clase pueden descender un montón de otras clases, pero lo que normalmente se ve es que una clase siempre parte de otra, o que de una partan distintas o de la originaria partan todas las demás. Estaríamos en herencia simple, de manera que al generar un componente nuevo tienes absolutamente claro de qué otro descende. Pero hay lenguajes, como ocurre en C++, que esto puede no ser así, puede haber una herencia múltiple cuando a una clase le pueden corresponder distintos ancestros. ¿Que ocurre con ello? pues resulta que no hereda de un solo "padre", sino de múltiples, por lo que puede tener propiedades tanto de unos como de otros.

Heurística Es una técnica que se basa en la experiencia conseguida después de realizar intentos repetidos, normalmente por medio de algoritmos concretos. Se van realizando pruebas, aún sin conocer exactamente todos los datos a tener en cuenta, hasta conseguir una solución al problema. Es uno de los sistemas de funcionamiento de los antivirus.

I.

Icono Representación gráfica de un elemento, generalmente una opción a elegir, que sustituye o complementa al texto escrito. Su característica es que ha de ser lo suficientemente significativo, y emplean bien ficheros .ico o .bmp normalmente de 16x16 o de 32x32 bits.

Información Elemento base del concepto informática, susceptible de ser tratado por los ordenadores. A causa de la proliferación de los medios computacionales y su universalización, se introduce el término "Sociedad de la información" refiriéndose a la apertura del conocimiento a todas las clases sociales.

Integridad Referencial Un gestor de base de datos debe de estar preparado para poder asegurar que los valores introducidos por el usuario son válidos. Esto se puede hacer a través de rangos, o de distintas operaciones (ver Restricciones de Dominio) y entre ellas la que probablemente sea más compleja, la referencia, que se da cuando a priori o en el momento de crear los ficheros o tablas no se conocen los valores que puede tomar un campo, por lo que es preciso una segunda tabla de manera que vaya almacenando los posibles, y que será la referencia de aquella en la que se introducen los datos, para asegurar que son correctos.

L.

Lenguaje de Programación Son secuencias de instrucciones, mas o menos parecidas al lenguaje humano, que permiten, una vez traducidas o compiladas, ejecutar las órdenes incluidas en el programa.

Lenguaje máquina Se denomina así a las instrucciones con las cuales puede trabajar el procesador, a lo que se denominaría código máquina, o aquellos ficheros que sin sin ser codificaciones u órdenes, se guardan en ese nivel.

Lenguajes de Prog Orientada Objetos (LPOO). Se les denomina así a aquellos lenguajes de programación capaces de trabajar con objetos en un sentido amplio, es decir, un lenguaje puede manejar y basarse en objetos porque los pueden crear según un patrón o porque ya estén creados, pero no se considera un LPOO en tanto no admite herencia, es decir, que de un objeto pueda crearse otro distinto,

que heredará parte de las propiedades del primero. En este sentido hay polémica, puesto que dos tipos de lenguaje cumplen con esas características, los denominados *puros* y los *híbridos*. Los puros rompen con la programación anterior y cualquier cosa a utilizar es un objeto. Los híbridos soportan igual los objetos o métodos o clases, pero además se traen las estructuras de programación tradicionales. Entre los primeros se encuentran Smalltalk o Eiffel, entre los segundos están los más utilizados y conocidos en este momento, como C++, Delphi (basado en Object Pascal), Java, etc.

M.

Modularidad Es la capacidad que tiene o se le confiere, a una aplicación informática, de manera que pueda ser dividida en partes más pequeñas, cuanto más, mejor, pero que deben de tener todas las características en código y datos como para poder ser independientes y a la vez, ensambladas, dar lugar al funcionamiento de un todo. Si la programación modular ha sido un clásico dentro de los sistemas tradicionales, ha vuelto a despertar el interés con la programación dirigida a objetos, donde los módulos, como entes cerrados, son una de sus características.

Motor de almacenamiento En un SGBD, es el software que se encarga de lo referente a datos, como se organizan y que relaciones tienen, como se almacenan y de qué forma aceptan las interrogaciones (accesos a datos o preguntas), como gestionan el acceso de distintos usuarios y los bloqueos pertinentes, sus medidas de seguridad y la integridad.

O.

Objeto Término base de la Programación Orientada a Objetos. En pura teoría podríamos pensar en lo que el hombre conoce, una mesa, un mechero, son cosas que nuestra inteligencia sabe comprender y explicar porque tienen unas características comunes, a pesar de que las diferencien muchas otras, pues hay formas y modelos muy dispares. A nivel informático es similar, un objeto es una parte de código con unas peculiaridades, de manera que podemos crear el objeto "subMesa", que contendrá las básicas de su ancestro pero modificadas, es decir, todo lo que puede hacer "Mesa" con su código, llamadas a funciones, punteros, etc., mas las particularidades que proporcionemos a este "hijo" suyo.

ODBC Open Data Base Connectivity. Ha sido la base de Windows en sistemas abiertos, es decir que permiten una conectividad entre distintos lenguajes de programación con distintas bases de datos.

OOO Object Oriented Programing. Programación orientada a objetos. Es la forma de programar de los últimos años (en aquellos lenguajes que lo permitan), a

diferencia de la programación clásica en que se leía línea a línea y se ejecutaba cada una. Se basa en los objetos, que en teoría deben ser más cercanos a la forma de pensar del hombre en el sentido abstracto. En realidad un objeto es una cantidad de código capaz de recibir unas ordenes y devolver unas respuestas relacionadas, y con unas características especiales, principalmente la herencia que permite generar hijos de los anteriores.

Organigramas Ver Diagrama.

P.-

Polimorfismo Uno de los tres pilares de la Programación Orientada a Objetos. Si consideramos un Objeto, de manera simbólica, como un ente sobre el que tenemos capacidad de conocimiento, como una mesa, de manera que podemos definir sus propiedades con independencia de que se trate en concreto de una u otra, el polimorfismo permite crear identidades propias, es decir, una *subMesa* podremos crearla de madera de caoba, pero seguirá teniendo las características básicas de su ancestro "Mesa", de manera que podremos refiriéndonos a éstas sin tener que preocuparnos en definirla en su heredero.

Procesamiento por lotes Suelen denominarse "batch" procedente de batch processing. Es un método en el cual el trabajo se prepara para ser procesado por el ordenador de manera que todas las operaciones elegidas se realicen al mismo tiempo, generalmente de forma diferida. Es lo contrario de lo que se denomina "procesamiento en línea".

Programa Instrucciones que varían según el lenguaje que se utiliza, pero cuyo fin es el de controlar las acciones que tiene que llevar a cabo el ordenador y sus periféricos.

Programación Estructurada Es una técnica de programación que se inició hacia los años 1.970 y que ha tenido distintos métodos según el autor. Básicamente se divide el problema en partes más pequeñas, que son "funciones" que realizan una tarea concreta cada una, es lo que se denomina "abstracción". También utilizan unas reglas para compartir las variables, ya sean locales o globales. El problema de la programación estructurada aumenta según es mayor la aplicación.

Programación Orientada a Objetos (POO) Es un sistema de programación, procedente de la evolución de la programación estructurada, cuyos pilares son la reutilización del código y la facilidad (teórica) de simplificarle. Su base es lo que se denomina Objeto y que es el fundamento de los lenguajes de este tipo o LPOO.

R.

RDBMS Relational Database Management System. Sistema de Gestión de Bases de Datos Relacionales. También conocidas como bases SQL, son las típicas relacionales, donde la organización en tablas con filas y columnas se relaciona en alguna o algunas celdas con la correspondiente de otras tablas distintas.

S.

SGBD Sistema Gestor de Bases de Datos. Conjunto de programas que hacen posible la creación y mantenimiento de una base de datos. En estos momentos la tendencia es a las Bases de Datos relacionales basadas en lenguaje de interrogación SQL, y aunque se utiliza uno de sus estándares cada fabricante introduce sus modificaciones. Pero no tiene por qué ser así, un SGBD o DBMS puede ser interrogado desde muy distintos lenguajes de programación e incluso por combinaciones entre estos y SQL, o tener los suyos propios. En cualquier caso todos funcionan a través de lo que denominan "motores" de datos. Algunos de ellos, aún siendo pequeños, tienen capacidad gracias a su *Administrador de datos* de poder generar archivos, editarlos, imprimirlos, etc. sin necesidad de programación.

Sistemas de Información Se debe considerar un sistema de computación e información como el conjunto de componentes físicos (hardware), lógicos (software), de comunicación (bien redes de cualquier tipo o Internet) y medios humanos (lo que ahora llaman orgware), todo ello unido permite el tratamiento de la información.

T.

Tabla Concretando en la versión "moderna", cada uno de los ficheros que forman una base de datos relacional. Ver RDBMS. Aunque tradicionalmente el concepto tabla ha sido sinónimo de matriz (generalmente de dos dimensiones).

V.

Variables En programación, una parte de la memoria que es representada por unos caracteres a discreción del programador, y que tiene un contenido cambiante, generalmente por las operaciones realizadas en el programa.

Ventana (Window). Aunque se iniciaron con los sistemas de Mac, en el día de hoy, y al menos en Europa es el sistema operativo Windows quien encabeza su

manejo. En la pantalla general es un cuadrado en el que se superpone una franja y dentro de él se están ejecutando programas de cualquier tipo. Constituyen las zonas Clientes dentro de las pantallas Windows, que pueden perfectamente ocupar la totalidad de la zona de visión.

Apéndice

Aquí mostraremos parte del código que se escribió para el desarrollo del sistema.

Para comenzar daremos una descripción del formulario principal de nuestro sistema:

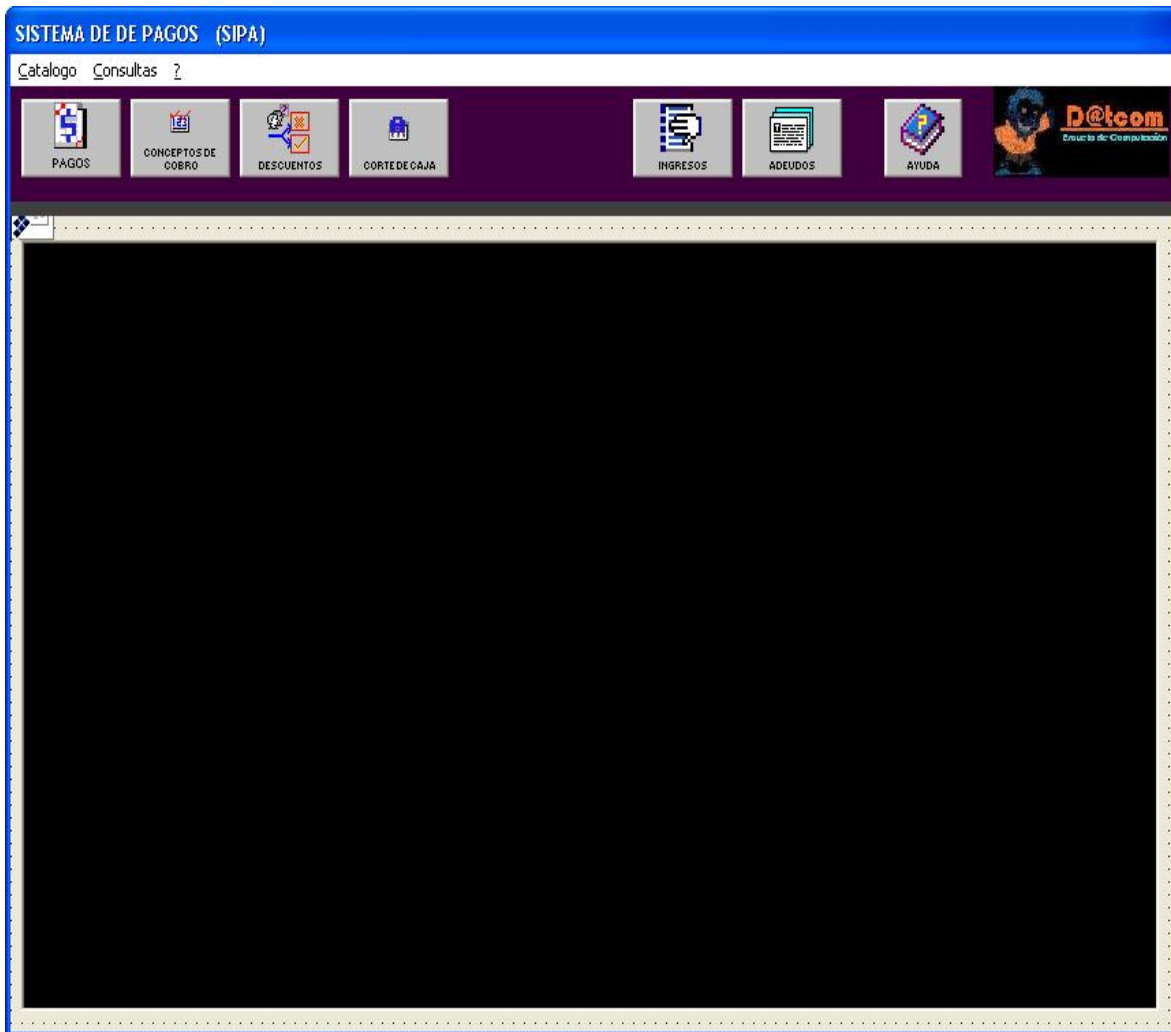


Figura 13.- Formulario principal.

Apéndice

En este formulario se centralizan las funciones de todo el sistema, tales como alta de alumnos, pagos, las configuraciones de los conceptos de cobro, y las configuraciones de los descuentos, así como las consultas de adeudos ingresos y recibos, sin olvidar el reporte de adeudos.

El código correspondiente es el siguiente:

```
Private Sub Command1_Click()  
PAGOS.Show 1  
End Sub
```

```
Private Sub Command2_Click()  
CONCEPTOS_COBRO.Show 1  
End Sub
```

```
Private Sub Command3_Click()  
ADEUDOS_DIA.Show 1  
End Sub
```

```
Private Sub Command4_Click()  
DESCUENTOS.Show 1
```

```
End Sub
```

```
Private Sub Command5_Click()  
CORTE.Show 1  
End Sub
```

```
Private Sub Command6_Click()  
ACCESO.Show 1  
End Sub
```

```
Private Sub Command7_Click()  
CON_INGRESOS.Show 1  
End Sub
```

```
Private Sub Form_Load()  
Skin1.LoadSkin ("Web-II.skn")  
Skin1.ApplySkin PRINCIPAL.hWnd
```

```
'Command5.Enabled = False
```

```
End Sub
```

```
Private Sub mnuAdescuentos_Click()  
AUTORIZAR_DE.Show 1  
End Sub
```

```
Private Sub mnuAdeudos_Click()  
ADEUDOS_DIA.Show 1  
End Sub
```

```
Private Sub mnuAlum_Click()  
ALUMNOS.Show 1  
End Sub
```

```
Private Sub mnuCcobro_Click()  
CONCEPTOS_COBRO.Show 1  
End Sub
```

```
Private Sub mnuDesc_Click()  
DESCUENTOS.Show 1  
End Sub
```

```
Private Sub mnuIngresos_Click()  
CON_INGRESOS.Show 1  
End Sub
```

Apéndice

```
Private Sub mnuPagos_Click()  
PAGOS.Show 1  
End Sub
```

```
Private Sub mnuSalir_Click()  
Unload FONDO  
Unload Me  
End Sub
```

Con respecto a los objetos utilizados en el formulario.

Cabe hacer mención que este programa es la primera parte del proyecto final, debido a que se pretende que el sistema pueda llevar el control escolar de la institución, en donde se integren calificaciones de alumnos, inasistencias, y notas especiales tales como exámenes extraordinarios y/o especiales, todo esto a través de la matrícula del alumno.

La opción de alumnos nos permite dar de alta alumnos proporcionando sus datos generales (obtenidos mediante previa inscripción) y asignando al mismo un número de control (matrícula).

En el formulario de pagos se encuentra el ingreso de capital económico del plantel, en cuyo caso, se registrará el pago de la colegiatura correspondiente a la semana en curso de cada uno de los alumnos inscritos en dicho plantel, denotando que:

1. Se podrán realizar pagos por adelantado de colegiatura de semanas posteriores considerando un descuento por pago adelantado de colegiatura.
2. En casos de retraso por este concepto, también se tendrá que considerar un porcentaje adicional que incrementará dicho pago de colegiatura (recargos).
3. Si el alumno se encuentra regular en sus pagos de colegiatura no se hará acreedor a ninguno de los puntos anteriores.

El formulario mencionado es el siguiente:

Figura 14.- Formulario de pagos.

El manejo del formulario de pagos es realmente sencillo pero, hay que aclarar que no solamente se perciben colegiaturas, sino también, se puede producir una entrada por un concepto diferente como son:

1. Pago por impresiones
2. Pago por renta de equipo
3. Adquisición de accesorios
4. Adquisición de equipos de cómputo
5. Adquisición de Consumibles
6. Reparación de equipo, etc.

Por esta causa el formulario cuenta con dos opciones para controlar tanto los pagos internos (que corresponden únicamente a los pagos de colegiatura de los alumnos), como los pagos externos (que se dan por un concepto diferente al de colegiaturas)

Asímismo, dicho formulario imprimirá en pantalla un pequeño reporte del estado actual de los alumnos para asegurarse de la regularidad del alumno por concepto de colegiaturas, permitiendo en todo caso comentarlo con el interesado, sin olvidar que en algún momento dado se puede cancelar un pago con las opciones correspondientes.

El código para éste formulario se muestra a continuación:

Dim servicios As Recordset

Dim Ingreso As Database

Dim van2 As Boolean

Dim resul, var1, control, cadeu, cont As Integer

Dim jas, jas1 As Integer

Dim modisem As Double

Apéndice

Dim var2 As Double

```
Private Sub cmdCancelar_Click()  
Data3.Recordset.CancelUpdate  
cmdGrabar.Enabled = False  
cmdCancelar.Enabled = False  
cmdModificar.Enabled = True  
cmdNuevo.Enabled = True  
cmdSalir.Enabled = True
```

```
'limpiar todo  
Picture2.Visible = False  
Picture4.Visible = False  
Label34.Visible = False  
Combo1.Text = ""  
Combo2.Text = ""  
Usuario.Text = ""  
Adeudo.Caption = ""  
List1.Clear  
Combo1.Locked = True  
Combo2.Locked = True  
Option1.Enabled = False  
Option2.Enabled = False
```

End Sub

```
Private Sub cmdGrabar_Click()  
On Error GoTo RUTINA  
Dim otra As Integer
```

```
Text73.Text = Text14.Text
```

```
If (Text12.Text = "Externo") Then  
Matricula.Text = Text12.Text  
lolo = 1  
End If  
If (Matricula.Text = "") Then  
MsgBox "No se puede Actualizar el registro Introduce los datos...?", vbInformation, "ERROR AL GRABAR"  
Option1.Enabled = True  
Option2.Enabled = True  
GoTo BRINCO  
End If  
If (Text20.Text = "" Or Label5.Caption = "" Or Matricula.Text = "" Or Text73.Text = "" Or Text13.Text = "" Or Text75.Text  
= "" Or Text3.Text = "") Then  
MsgBox "ERROR AL GRABAR ALGUN CAMPO VACIO", vbInformation, "ERROR AL GRABAR"  
  
txt2.SetFocus  
  
GoTo BRINCO  
End If
```

```
Data3.Recordset.Requery  
MsgBox "El Registro ya fue Actualizado con exito", vbInformation, "ESTADO DEL REGISTRO"  
Data3.Refresh
```

RUTINA:

```
If Err.Number = 3022 Then  
MsgBox "El folio de pago ya existe si decea reemplazarlo borre el registro y actualice los datos nuevamente o si decea  
agregar una nueva clave presione el boton CANCELAR, NUEVO y despues BUSCAR CLAVE para asignar una nueva",  
vbInformation, "ERROR AL GRABAR"  
Option1.Enabled = True  
Option2.Enabled = True  
End If
```

Apéndice

```
'
If (lolo = 0) Then
sustituir_pago
cambiar_pago
End If

cmdGrabar.Enabled = False
cmdCancelar.Enabled = False
cmdNuevo.Enabled = True
cmdModificar.Enabled = True
'cmdSalir.Enabled = True

BRINCO:
control = 0

Command2.Enabled = True
Command3.Enabled = True
End Sub

Private Sub cmdNuevo_Click()

Picture2.Visible = False
Picture4.Visible = False
Label34.Visible = False
Combo1.Text = ""
Combo2.Text = ""
Usuario.Text = ""
Adeudo.Caption = ""
List1.Clear
Combo1.Locked = True
Combo2.Locked = True

Data6.Refresh
Option1.Enabled = True
Option2.Enabled = True
Label13.Caption = ""
sem_cero
vandera = 0
vanderita = 0
oso = 0
presio = 0
toto = 0

'codigo para recorrer la lista
If Data3.Recordset.EOF Then
cont = 1
GoTo algo
End If
Data3.Recordset.MoveLast
If Data3.Recordset.EOF Then
Data3.Recordset.MoveLast
End If
cont = Text20.Text

cont2 = 1
Data3.Recordset.MoveFirst
If (Data3.Recordset.Fields("Folio_Pago") = cont2) Then
Do While (Data3.Recordset.Fields("Folio_Pago") < cont And Not Data3.Recordset.EOF)
If (cont2 <> Data3.Recordset.Fields("Folio_Pago")) Then
cont = cont2
GoTo j
End If
Data3.Recordset.MoveNext

cont2 = cont2 + 1

j:
```

Apéndice

Loop

Else

```
If (Data3.Recordset.Fields("Folio_Pago") <> cont2) Then
    cont2 = cont2 - 1
    cont = cont2
End If
```

End If

```
If (cont2 < Data3.Recordset.Fields("Folio_Pago")) Then
    cont2 = cont2 - 1
    cont = cont2
```

End If

```
If (cont2 = Data3.Recordset.Fields("Folio_Pago")) Then
    cont = cont2
```

End If

```
cont = cont + 1
```

```
If (cont = 0) Then
```

```
    cont = cont + 1
```

End If

```
'codigo de nuevo
```

```
'habilitar_textbox
```

```
Data3.Recordset.MoveLast
```

```
'habilitartextbox
```

algo:

```
cmdCancelar.Enabled = True
```

```
Data3.Recordset.AddNew
```

```
'control = 1
```

```
Text20.Text = cont
```

```
Label5.Caption = Date
```

```
Text13.Text = 1
```

```
num_recibo = Val(Text20.Text)
```

```
Text20.Locked = True
```

```
cmdGrabar.Enabled = False
```

```
cmdModificar.Enabled = False
```

```
cmdNuevo.Enabled = False
```

```
cmdSalir.Enabled = False
```

End Sub

```
Private Sub cmdSalir_Click()
```

```
Unload Me
```

End Sub

```
Private Sub Combo1_Click()
```

```
If (Text12.Text = "Externo") Then
```

```
If (Combo1.Text = "Colegiatura") Then
```

```
    MsgBox "Para el Usuario seleccionado no es valida esta opcion", vbInformation, "ERROR DE CAMPO"
```

```
    Combo1.SetFocus
```

```
    GoTo asis
```

End If

End If

```
If (Combo1.Text = "Colegiatura") Then
```

```
    Picture2.Visible = True
```

```
    Picture4.Visible = False
```

```
    Combo2.Locked = False
```

```
    Combo2.SetFocus
```

```
    control = 1
```

```
    Text75.Text = 1
```

```
    Combo1.Locked = True
```

```
    Text3.Text = Text18.Text
```

```
    ste = Text18.Text
```

End If

```
If (Combo1.Text = "Impresiones") Then
```

```
    Picture4.Visible = True
```

```
    Text4.Locked = False
```

```
    Text5.Enabled = False
```

```
    Text4.SetFocus
```

```
    control = 1
```

```
    Combo1.Locked = True
```

Apéndice

```
Combo2.Locked = True

Text75.Text = 1

End If
If (Combo1.Text = "Renta de Equipo de Computo") Then
    Picture4.Visible = True
    Picture2.Visible = False
    Text5.Locked = False
    Text4.Enabled = False
    Text5.SetFocus
    control = 1
    Combo1.Locked = True
    Combo2.Locked = True
    Text75.Text = 1
End If

llamada
'lamada2

If (control = 0) Then
    Picture4.Visible = True
    Picture2.Visible = False
    Text5.Locked = True
    Text4.Locked = True
    Text75.Text = 1
    Combo1.Locked = True
    Combo2.Locked = True

    Text9.Text = Text1.Text
    Text10.Text = Text11.Text
    var2 = Val(Text9.Text) - Val(Text10.Text)
    'con letra
    ' con letra
        'valor con letra
        Dim Numeros As New clsNumeros

        If IsNumeric(var2) Then
            Letras.Caption = Numeros.NroEnLetras(var2)
            Letras.Caption = Letras.Caption & Label18.Caption
        Else
            '
        End If

    'resultado
    Text7.Text = var2
    Text76.Text = var2
    Text7.Locked = True
    Text7.SetFocus
    var2 = 0

    cmdNuevo.Enabled = False
    cmdGrabar.Enabled = True
    cmdCancelar.Enabled = True
    cmdModificar.Enabled = False
    cmdBorrar.Enabled = False
    cmdSalir.Enabled = False
End If

asis:
cmdNuevo.Enabled = False
cmdGrabar.Enabled = True
cmdCancelar.Enabled = True
cmdModificar.Enabled = False
cmdBorrar.Enabled = False
cmdSalir.Enabled = False

End Sub
```

Apéndice

```
Private Sub Combo2_Click()
Dim cam_adeu As Integer
cam_adeu = Adeudo.Caption
llamada
'llamada2
If (Combo2.Text = "1") Then
resul = 1 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
Adeudo.Caption = Val(Adeudo.Caption - 1)
If (Val(Adeudo.Caption) < 0) Then
Label34.Visible = True
If (Val(Adeudo.Caption) = -1) Then
Adeudo.Caption = "1"
End If
If (Val(Adeudo.Caption) = -2) Then
Adeudo.Caption = "2"
End If
If (Val(Adeudo.Caption) = -3) Then
Adeudo.Caption = "3"
End If
If (Val(Adeudo.Caption) = -4) Then
Adeudo.Caption = "4"
End If
If (Val(Adeudo.Caption) = -5) Then
Adeudo.Caption = "5"
End If
If (Val(Adeudo.Caption) = -6) Then
Adeudo.Caption = "6"
End If
End If

End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "2") Then
resul = 2 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
Adeudo.Caption = Val(Adeudo.Caption - 2)
If (Val(Adeudo.Caption) < 0) Then
Label34.Visible = True
If (Val(Adeudo.Caption) = -1) Then
Adeudo.Caption = "1"
End If
If (Val(Adeudo.Caption) = -2) Then
Adeudo.Caption = "2"
End If
If (Val(Adeudo.Caption) = -3) Then
Adeudo.Caption = "3"
End If
If (Val(Adeudo.Caption) = -4) Then
Adeudo.Caption = "4"
End If
If (Val(Adeudo.Caption) = -5) Then
Adeudo.Caption = "5"
End If
If (Val(Adeudo.Caption) = -6) Then
Adeudo.Caption = "6"
End If
End If

End If
```

Apéndice

```
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "3") Then
    resul = 3 * Val(Text74.Text)
    SUBTOTAL_TXT.Text = resul
    var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
    Adeudo.Caption = Val(Adeudo.Caption - 3)
    If (Val(Adeudo.Caption) < 0) Then
        Label34.Visible = True
        If (Val(Adeudo.Caption) = -1) Then
            Adeudo.Caption = "1"
        End If
        If (Val(Adeudo.Caption) = -2) Then
            Adeudo.Caption = "2"
        End If
        If (Val(Adeudo.Caption) = -3) Then
            Adeudo.Caption = "3"
        End If
        If (Val(Adeudo.Caption) = -4) Then
            Adeudo.Caption = "4"
        End If
        If (Val(Adeudo.Caption) = -5) Then
            Adeudo.Caption = "5"
        End If
        If (Val(Adeudo.Caption) = -6) Then
            Adeudo.Caption = "6"
        End If
    End If
End If

End If
'descuento
Text6.Text = 0
End If

If (Combo2.Text = "4") Then
    resul = 4 * Val(Text74.Text)
    SUBTOTAL_TXT.Text = resul
'descuento
Text6.Text = Text11.Text
resul = Val(SUBTOTAL_TXT.Text) - Val(Text6.Text)
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
    Adeudo.Caption = Val(Adeudo.Caption - 4)
    If (Val(Adeudo.Caption) < 0) Then
        Label34.Visible = True
        If (Val(Adeudo.Caption) = -1) Then
            Adeudo.Caption = "1"
        End If
        If (Val(Adeudo.Caption) = -2) Then
            Adeudo.Caption = "2"
        End If
        If (Val(Adeudo.Caption) = -3) Then
            Adeudo.Caption = "3"
        End If
        If (Val(Adeudo.Caption) = -4) Then
            Adeudo.Caption = "4"
        End If
        If (Val(Adeudo.Caption) = -5) Then
            Adeudo.Caption = "5"
        End If
        If (Val(Adeudo.Caption) = -6) Then
            Adeudo.Caption = "6"
        End If
    End If
End If
```

Apéndice

```
End If
End If
If (Combo2.Text = "5") Then
resul = 5 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
  Adeudo.Caption = Val(Adeudo.Caption - 5)
  If (Val(Adeudo.Caption) < 0) Then
    Label34.Visible = True
    If (Val(Adeudo.Caption) = -1) Then
      Adeudo.Caption = "1"
    End If
    If (Val(Adeudo.Caption) = -2) Then
      Adeudo.Caption = "2"
    End If
    If (Val(Adeudo.Caption) = -3) Then
      Adeudo.Caption = "3"
    End If
    If (Val(Adeudo.Caption) = -4) Then
      Adeudo.Caption = "4"
    End If
    If (Val(Adeudo.Caption) = -5) Then
      Adeudo.Caption = "5"
    End If
    If (Val(Adeudo.Caption) = -6) Then
      Adeudo.Caption = "6"
    End If
  End If
End If
```

```
End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "6") Then
resul = 6 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
  Adeudo.Caption = Val(Adeudo.Caption - 5)
  If (Val(Adeudo.Caption) < 0) Then
    Label34.Visible = True
    If (Val(Adeudo.Caption) = -1) Then
      Adeudo.Caption = "1"
    End If
    If (Val(Adeudo.Caption) = -2) Then
      Adeudo.Caption = "2"
    End If
    If (Val(Adeudo.Caption) = -3) Then
      Adeudo.Caption = "3"
    End If
    If (Val(Adeudo.Caption) = -4) Then
      Adeudo.Caption = "4"
    End If
    If (Val(Adeudo.Caption) = -5) Then
      Adeudo.Caption = "5"
    End If
    If (Val(Adeudo.Caption) = -6) Then
      Adeudo.Caption = "6"
    End If
  End If
End If
```

```
End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "7") Then
```


Apéndice

```
resul = 7 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
  Adeudo.Caption = Val(Adeudo.Caption - 7)
  If (Val(Adeudo.Caption) < 0) Then
    Label34.Visible = True
    If (Val(Adeudo.Caption) = -1) Then
      Adeudo.Caption = "1"
    End If
    If (Val(Adeudo.Caption) = -2) Then
      Adeudo.Caption = "2"
    End If
    If (Val(Adeudo.Caption) = -3) Then
      Adeudo.Caption = "3"
    End If
    If (Val(Adeudo.Caption) = -4) Then
      Adeudo.Caption = "4"
    End If
    If (Val(Adeudo.Caption) = -5) Then
      Adeudo.Caption = "5"
    End If
    If (Val(Adeudo.Caption) = -6) Then
      Adeudo.Caption = "6"
    End If
    If (Val(Adeudo.Caption) = -7) Then
      Adeudo.Caption = "7"
    End If
  End If
End If

End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "8") Then
  resul = 8 * Val(Text74.Text)
  SUBTOTAL_TXT.Text = resul
  var1 = resul
  'resta de adeudo
  If (cam_adeu >= 0) Then
    Adeudo.Caption = Val(Adeudo.Caption - 8)
    If (Val(Adeudo.Caption) < 0) Then
      Label34.Visible = True
      If (Val(Adeudo.Caption) = -1) Then
        Adeudo.Caption = "1"
      End If
      If (Val(Adeudo.Caption) = -2) Then
        Adeudo.Caption = "2"
      End If
      If (Val(Adeudo.Caption) = -3) Then
        Adeudo.Caption = "3"
      End If
      If (Val(Adeudo.Caption) = -4) Then
        Adeudo.Caption = "4"
      End If
      If (Val(Adeudo.Caption) = -5) Then
        Adeudo.Caption = "5"
      End If
      If (Val(Adeudo.Caption) = -6) Then
        Adeudo.Caption = "6"
      End If
      If (Val(Adeudo.Caption) = -7) Then
        Adeudo.Caption = "7"
      End If
      If (Val(Adeudo.Caption) = -8) Then
        Adeudo.Caption = "8"
      End If
    End If
  End If
End If
```

Apéndice

```
End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "9") Then
resul = 9 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
  Adeudo.Caption = Val(Adeudo.Caption - 9)
  If (Val(Adeudo.Caption) < 0) Then
    Label34.Visible = True
    If (Val(Adeudo.Caption) = -1) Then
      Adeudo.Caption = "1"
    End If
    If (Val(Adeudo.Caption) = -2) Then
      Adeudo.Caption = "2"
    End If
    If (Val(Adeudo.Caption) = -3) Then
      Adeudo.Caption = "3"
    End If
    If (Val(Adeudo.Caption) = -4) Then
      Adeudo.Caption = "4"
    End If
    If (Val(Adeudo.Caption) = -5) Then
      Adeudo.Caption = "5"
    End If
    If (Val(Adeudo.Caption) = -6) Then
      Adeudo.Caption = "6"
    End If
    If (Val(Adeudo.Caption) = -7) Then
      Adeudo.Caption = "7"
    End If
    If (Val(Adeudo.Caption) = -8) Then
      Adeudo.Caption = "8"
    End If
    If (Val(Adeudo.Caption) = -9) Then
      Adeudo.Caption = "9"
    End If
  End If
End If
```

End If

```
End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "10") Then
resul = 10 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
  Adeudo.Caption = Val(Adeudo.Caption - 10)
  If (Val(Adeudo.Caption) < 0) Then
    Label34.Visible = True
    If (Val(Adeudo.Caption) = -1) Then
      Adeudo.Caption = "1"
    End If
    If (Val(Adeudo.Caption) = -2) Then
      Adeudo.Caption = "2"
    End If
    If (Val(Adeudo.Caption) = -3) Then
      Adeudo.Caption = "3"
    End If
    If (Val(Adeudo.Caption) = -4) Then
      Adeudo.Caption = "4"
    End If
    If (Val(Adeudo.Caption) = -5) Then
      Adeudo.Caption = "5"
    End If
  End If
End If
```

Apéndice

```
End If
If (Val(Adeudo.Caption) = -6) Then
    Adeudo.Caption = "6"
End If
If (Val(Adeudo.Caption) = -7) Then
    Adeudo.Caption = "7"
End If
If (Val(Adeudo.Caption) = -8) Then
    Adeudo.Caption = "8"
End If
If (Val(Adeudo.Caption) = -9) Then
    Adeudo.Caption = "9"
End If
If (Val(Adeudo.Caption) = -10) Then
    Adeudo.Caption = "10"
End If
End If

End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "11") Then
    resul = 11 * Val(Text74.Text)
    SUBTOTAL_TXT.Text = resul
    var1 = resul
    'resta de adeudo
    If (cam_adeu >= 0) Then
        Adeudo.Caption = Val(Adeudo.Caption - 11)
        If (Val(Adeudo.Caption) < 0) Then
            Label34.Visible = True
            If (Val(Adeudo.Caption) = -1) Then
                Adeudo.Caption = "1"
            End If
            If (Val(Adeudo.Caption) = -2) Then
                Adeudo.Caption = "2"
            End If
            If (Val(Adeudo.Caption) = -3) Then
                Adeudo.Caption = "3"
            End If
            If (Val(Adeudo.Caption) = -4) Then
                Adeudo.Caption = "4"
            End If
            If (Val(Adeudo.Caption) = -5) Then
                Adeudo.Caption = "5"
            End If
            If (Val(Adeudo.Caption) = -6) Then
                Adeudo.Caption = "6"
            End If
            If (Val(Adeudo.Caption) = -7) Then
                Adeudo.Caption = "7"
            End If
            If (Val(Adeudo.Caption) = -8) Then
                Adeudo.Caption = "8"
            End If
            If (Val(Adeudo.Caption) = -9) Then
                Adeudo.Caption = "9"
            End If
            If (Val(Adeudo.Caption) = -10) Then
                Adeudo.Caption = "10"
            End If
            If (Val(Adeudo.Caption) = -11) Then
                Adeudo.Caption = "11"
            End If
        End If
    End If

End If
'descuento
Text6.Text = 0
End If
```

Apéndice

```
If (Combo2.Text = "12") Then
resul = 12 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
  Adeudo.Caption = Val(Adeudo.Caption - 12)
  If (Val(Adeudo.Caption) < 0) Then
    Label34.Visible = True
    If (Val(Adeudo.Caption) = -1) Then
      Adeudo.Caption = "1"
    End If
    If (Val(Adeudo.Caption) = -2) Then
      Adeudo.Caption = "2"
    End If
    If (Val(Adeudo.Caption) = -3) Then
      Adeudo.Caption = "3"
    End If
    If (Val(Adeudo.Caption) = -4) Then
      Adeudo.Caption = "4"
    End If
    If (Val(Adeudo.Caption) = -5) Then
      Adeudo.Caption = "5"
    End If
    If (Val(Adeudo.Caption) = -6) Then
      Adeudo.Caption = "6"
    End If
    If (Val(Adeudo.Caption) = -7) Then
      Adeudo.Caption = "7"
    End If
    If (Val(Adeudo.Caption) = -8) Then
      Adeudo.Caption = "8"
    End If
    If (Val(Adeudo.Caption) = -9) Then
      Adeudo.Caption = "9"
    End If
    If (Val(Adeudo.Caption) = -10) Then
      Adeudo.Caption = "10"
    End If
    If (Val(Adeudo.Caption) = -11) Then
      Adeudo.Caption = "11"
    End If
    If (Val(Adeudo.Caption) = -12) Then
      Adeudo.Caption = "12"
    End If
  End If
End If
```

```
End If
'descuento
Text6.Text = 0
End If
If (Combo2.Text = "13") Then
resul = 13 * Val(Text74.Text)
SUBTOTAL_TXT.Text = resul
var1 = resul
'resta de adeudo
If (cam_adeu >= 0) Then
  Adeudo.Caption = Val(Adeudo.Caption - 13)
  If (Val(Adeudo.Caption) < 0) Then
    Label34.Visible = True
    If (Val(Adeudo.Caption) = -1) Then
      Adeudo.Caption = "1"
    End If
    If (Val(Adeudo.Caption) = -2) Then
      Adeudo.Caption = "2"
    End If
    If (Val(Adeudo.Caption) = -3) Then
      Adeudo.Caption = "3"
    End If
    If (Val(Adeudo.Caption) = -4) Then
```

Apéndice

```
        Adeudo.Caption = "4"
    End If
    If (Val(Adeudo.Caption) = -5) Then
        Adeudo.Caption = "5"
    End If
    If (Val(Adeudo.Caption) = -6) Then
        Adeudo.Caption = "6"
    End If
    If (Val(Adeudo.Caption) = -7) Then
        Adeudo.Caption = "7"
    End If
    If (Val(Adeudo.Caption) = -8) Then
        Adeudo.Caption = "8"
    End If
    If (Val(Adeudo.Caption) = -9) Then
        Adeudo.Caption = "9"
    End If
    If (Val(Adeudo.Caption) = -10) Then
        Adeudo.Caption = "10"
    End If
    If (Val(Adeudo.Caption) = -11) Then
        Adeudo.Caption = "11"
    End If
    If (Val(Adeudo.Caption) = -12) Then
        Adeudo.Caption = "12"
    End If
    If (Val(Adeudo.Caption) = -12) Then
        Adeudo.Caption = "12"
    End If
End If

End If
'descuento
Text6.Text = 0
End If

Text2.Locked = False
Text2.Text = var1
toto = var1
Text76.Text = Text2.Text
'con letra
' con letra
    'valor con letra
    Dim Numeros As New clsNumeros

    If IsNumeric(var1) Then
        Label10.Caption = Numeros.NroEnLetras(var1)
        Label10.Caption = Label10.Caption & Label18.Caption
    Else
        '
    End If

cmdGrabar.Enabled = True
Text2.Locked = True
End Sub

Private Sub Combo2_KeyPress(KeyAscii As Integer)
If (KeyAscii = 27) Then
    Picture2.Visible = False
    Combo2.Text = ""
    Combo1.Locked = False
    Combo1.SetFocus
    control = 0
    Label34.Visible = False
    funcion_adeu
End If

End Sub

Private Sub Command2_Click()
```

Apéndice

```
barbrinco = Combo1.Text
barnumeric = Text9.Text
bardesc = Text10.Text
tusuario = Text12.Text
barusuario = Usuario.Text
num_recibo = Val(Text20.Text)
Unload Me
RECIBO.Show 1
End Sub
```

```
Private Sub Command3_Click()
barbrinco = Combo1.Text
barnumeric = SUBTOTAL_TXT.Text
tusuario = Text12.Text
barusuario = Usuario.Text
Unload Me
RECIBO.Show 1
End Sub
```

```
Private Sub Command5_Click()
CANCELA.Show 1
End Sub
```

```
Private Sub Command7_Click()
SUBTOTAL_TXT.Text = ""
Text6.Text = ""
Text8.Text = ""
Text2.Text = ""
presio = 0
funcion_adeu
```

```
SUBTOTAL_TXT.Locked = False
SUBTOTAL_TXT.SetFocus
Text6.Locked = False
Text7.Locked = False
Text6.Text = 0
Text7.Text = 0
Text6.Locked = True
Text7.Locked = True
```

```
End Sub
```

```
Private Sub Form_Load()
Command2.Enabled = False
Command3.Enabled = False
cmdBorrar.Enabled = False
Skin1.LoadSkin ("chizh.skn")
Skin1.ApplySkin PAGOS.hWnd
```

```
'buscar base de datos
Set Ingreso = OpenDatabase("C:\CONTROL DE PAGOS (DATCOM)\base\COBROS.mdb")
```

```
'Buscar servicio
Set servicios = Ingreso.OpenRecordset("SERVICIOS")
van2 = False
sem_cero
funcion_servicios
Label5.Caption = Date
control = 0
cadeu = 0
lolo = 0
'
```

```
Text9.Locked = True
Text10.Locked = True
```

```
Option1.Enabled = False
Option2.Enabled = False
```

Apéndice

```
cmdGrabar.Enabled = False
cmdCancelar.Enabled = False
Combo1.Locked = True
Combo2.Locked = True
Matricula.Locked = True
```

```
End Sub
```

```
Private Sub funcion_servicios()
Dim con As Integer
con = 0
servicios.MoveFirst
While van2 = False
Combo1.List(con) = servicios.Fields("Descripcion")
con = con + 1
servicios.MoveNext
If servicios.EOF Then
van2 = True
End If
Wend
van2 = False
End Sub
```

```
Private Sub Matricula_KeyPress(KeyAscii As Integer)
If InStr("0123456789", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then
KeyAscii = 0
End If
If (KeyAscii = 13) Then
If (Matricula.Text = "") Then
MsgBox "Introduce la clave de Alumno", vbInformation, "ERROR DE CAMPO"
Matricula.SetFocus
Else
alumno
Text3.Text = Text18.Text
'adeu
'Matricula.Locked = True
'Combo1.SetFocus
End If
End If

If (KeyAscii = 27) Then
Picture2.Visible = False
Picture4.Visible = False
Option1.Enabled = True
Option2.Enabled = True
Matricula.Locked = True
End If
```

```
Text12.Text = Matricula.Text
```

```
End Sub
```

```
Private Sub Option1_Click()
presio = 0
Matricula.Enabled = True
Matricula.Locked = False
Matricula.Text = ""
Usuario.Text = ""
Usuario.Enabled = False
Matricula.SetFocus
'Text12.Text = "Interno"
End Sub
```

```
Private Sub Option2_Click()
presio = 0
Matricula.Text = ""
Usuario.Text = ""
Adeudo.Caption = ""
List1.Clear
```

Apéndice

```
Matricula.Enabled = False
Usuario.Enabled = True
Usuario.Locked = True
Combo1.SetFocus
Combo1.Locked = False
Text12.Text = "Externo"
actual_fecha
End Sub
```

```
'codigo para Buscar los servicios
Private Sub llamada()
    Dim jor As String
    jor = Combo1.Text
    Data1.Recordset.MoveFirst
    Do While (Data1.Recordset.Fields("Descripcion") <> UCase(jor) And Not Data1.Recordset.EOF)
        If (Data1.Recordset.Fields("Descripcion") = Combo1.Text) Then
            GoTo jorgebr
        End If
        Data1.Recordset.MoveNext
    If Data1.Recordset.EOF Then
        MsgBox "La clave del servicio no existe..., Verifique la Informacion", vbInformation, "ERROR DE LECTURA"
        Data1.Recordset.MoveFirst
        Combo1.Locked = False
        Combo1.SetFocus
        Combo1.Text = ""

        Exit Sub
    End If
Loop
jorgebr:
If (Data1.Recordset.Fields("Descripcion") = UCase(jor)) Then
    Text1.Text = Data1.Recordset.Fields("Presio")
    Text3.Text = Data1.Recordset.Fields("Clave_Descuento")
    Text14.Text = Data1.Recordset.Fields("Clave_Servicio")
End If
End Sub
```

```
'codigo para Buscar el descuento
Private Sub llamada2()
    Dim jor2 As Integer
    jor = Text3.Text
    Data2.Recordset.MoveFirst
    Do While (Data2.Recordset.Fields("Clave_Descuento") <> (jor2) And Not Data2.Recordset.EOF)
        If (Data2.Recordset.Fields("Clave_Descuento") = Text3.Text) Then
            GoTo jorgeb
        End If
        Data2.Recordset.MoveNext
    If Data2.Recordset.EOF Then
        MsgBox "La clave del Descuento no existe..., Verifique la Informacion", vbInformation, "ERROR DE LECTURA"
        Data2.Recordset.MoveFirst
        Combo1.Locked = False
        Combo1.SetFocus
        Combo1.Text = ""

        Exit Sub
    End If
Loop
jorgeb:
If (Data2.Recordset.Fields("Clave_Descuento") = (jor2)) Then
    Text11.Text = Data1.Recordset.Fields("Total_Descuento")
End If
End Sub
```

```
Private Sub SUBTOTAL_TXT_KeyPress(KeyAscii As Integer)
```


Apéndice

```
If InStr("0123456789.", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then
    KeyAscii = 0
End If
If (KeyAscii = 13) Then
    If (SUBTOTAL_TXT.Text = "") Then
        MsgBox "Introduce el pago", vbInformation, "ERROR DE CAMPO"
        SUBTOTAL_TXT.SetFocus
    Else
        modisem = Val(SUBTOTAL_TXT) / Val(Text74.Text)

        Text2.Text = Val(SUBTOTAL_TXT.Text)
        toto = Val(SUBTOTAL_TXT.Text)
        Text76.Text = Val(SUBTOTAL_TXT.Text)

        Combo2.Locked = False
        Combo2.Text = modisem
        Combo2.Locked = True

        'valor con letra
        Dim Numeros As New clsNumeros

        If IsNumeric(SUBTOTAL_TXT.Text) Then
            Label10.Caption = Numeros.NroEnLetras(SUBTOTAL_TXT.Text)
            Label10.Caption = Label10.Caption & Label18.Caption
        Else
            '
        End If

        SUBTOTAL_TXT.Locked = True
        Text2.SetFocus

    End If
End If
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
If (KeyAscii = 27) Then
    Picture2.Visible = False
End If
End Sub
```

```
Private Sub text4_KeyPress(KeyAscii As Integer)
If (KeyAscii = 27) Then
    Picture4.Visible = False
    Text5.Enabled = True
    Text4.Text = ""
    Text5.Text = ""
    Text9.Text = ""
    Text10.Text = ""
    Text7.Text = ""
    Combo1.Locked = False
    Combo1.SetFocus
    var2 = 0
    control = 0
    Combo1.SetFocus
End If
```

```
If InStr("1234567890", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then
    KeyAscii = 0
End If
If (KeyAscii = 13) Then
    If (Text4.Text = "") Then
        MsgBox "Introduce el Numero Impresiones", vbInformation, "ERROR DE CAMPO"
        Text4.SetFocus
    Else
        var2 = Val(Text1.Text) * Val(Text4.Text)
        Text9.Text = var2
        var2 = var2 - Val(Text11.Text)
    End If
End If
```

Apéndice

```
Text10.Text = Text11.Text
Text7.Text = var2
Text7.Locked = True
Text7.SetFocus
' con letra
'valor con letra
Dim Numeros As New clsNumeros

If IsNumeric(var2) Then
    Letras.Caption = Numeros.NroEnLetras(var2)
    Letras.Caption = Letras.Caption & Label18.Caption
Else
    '
End If

End If
End If
End Sub
Private Sub text5_KeyPress(KeyAscii As Integer)
If (KeyAscii = 27) Then
    Picture4.Visible = False
    Text4.Enabled = True
    Text4.Text = ""
    Text5.Text = ""
    Text9.Text = ""
    Text10.Text = ""
    Text7.Text = ""
    Combo1.Locked = False
    Combo1.SetFocus
    control = 0
    var2 = 0
    Combo1.SetFocus
End If

If InStr("1234567890.", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then
    KeyAscii = 0
End If
If (KeyAscii = 13) Then
    If (Text5.Text = "") Then
        MsgBox "Introduce el tiempo de Renta de Equipo", vbInformation, "ERROR DE CAMPO"
        Text5.SetFocus
    Else
        var2 = Val(Text1.Text) * Val(Text5.Text)
        Text9.Text = var2
        var2 = var2 - Val(Text11.Text)
        Text10.Text = Text11.Text
        Text7.Text = var2
        Text7.Locked = True
        Text7.SetFocus
        ' con letra
        'valor con letra
        Dim Numeros As New clsNumeros

        If IsNumeric(var2) Then
            Letras.Caption = Numeros.NroEnLetras(var2)
            Letras.Caption = Letras.Caption & Label18.Caption
        Else
            '
        End If

    End If
End If
Var = 0
End Sub
Private Sub Text7_KeyPress(KeyAscii As Integer)
If (Combo1.Text = "Impresiones") Then
    If (KeyAscii = 27) Then
        Text4.SetFocus
    End If
End If
```

Apéndice

```
End If
If (Combo1.Text = "Renta de Equipo de Computo") Then
  If (KeyAscii = 27) Then
    Text5.SetFocus
    Letras.Caption = ""
  End If
End If
If (control = 0) Then
  If (KeyAscii = 27) Then
    Text4.Text = ""
    Text5.Text = ""
    Text9.Text = ""
    Text10.Text = ""
    Text7.Text = ""
    Letras.Caption = ""
    Picture4.Visible = False
    Combo1.Locked = False
    Combo1.SetFocus
  End If
End If

End Sub

'codigo para Buscar el alumno
Private Sub alumno()
  Dim jor3 As Integer
  jor3 = Matricula.Text
  Data4.Recordset.MoveFirst
  Do While (Data4.Recordset.Fields("Clave_Alumno") <> (jor3) And Not Data4.Recordset.EOF)
    If (Data4.Recordset.Fields("Clave_Alumno") = Matricula.Text) Then
      GoTo jof
    End If
    Data4.Recordset.MoveNext
  If Data4.Recordset.EOF Then
    MsgBox "La clave del Alumno no existe..., Verifique la Informacion", vbInformation, "ERROR DE LECTURA"
    Data4.Recordset.MoveFirst
    Matricula.Locked = False
    Matricula.SetFocus
    Matricula.Text = ""

    Exit Sub
  End If
Loop
jof:
If (Data4.Recordset.Fields("Clave_Alumno") = (jor3)) Then
  Usuario.Text = Data4.Recordset.Fields("Nombre_Alumno")
  Text15.Text = Data4.Recordset.Fields("Fecha_Entrada")
  Text74.Text = Data4.Recordset.Fields("Colegiatura")

  funcion_adeu 'buscar adeudos

  Matricula.Locked = True
  Combo1.SetFocus
  Combo1.Locked = False
End If

End Sub

'verificar adeudos
Private Sub funcion_adeu()

Dim fe, fecha_acu, fe_pas, menor, mayor As String
Dim acum, n, n4, m4, m, csemana, acum2, acum3, ter, ter2, conta_fe, conta_fe2, dato As Integer
Dim prueba, pruebab, yo As Date
Dim pruebac As Date
Dim pruebac4, pruebab4 As Date
Dim jortere As Integer
Dim sem_entrada As Date
Dim dedo As Date
'
```

Apéndice

```
Dim jor4 As Integer
cadeu = 0
jor4 = Matricula.Text
Data5.Recordset.MoveFirst
Do While (Data5.Recordset.Fields("Clave_Alumno") <> (jor4) And Not Data5.Recordset.EOF)
  If (Data5.Recordset.Fields("Clave_Alumno") = Matricula.Text) Then
    GoTo jof
  End If
  Data5.Recordset.MoveNext
  If Data5.Recordset.EOF Then
    ' MsgBox "La clave del Alumno no existe..., Verifique la Informacion", vbInformation, "ERROR DE LECTURA"
    Data5.Recordset.MoveFirst
    Matricula.Locked = False
    Matricula.SetFocus
    Matricula.Text = ""

    Exit Sub
  End If
Loop
jof:
'buscar semanas
prueba = Date
sem_entrada = Text15.Text
'fechas de adeudos
jortere = Val(Text74.Text)
vandera = 0
If (jorgito = 0) Then
  acum = 0
End If
acum2 = 0
conta_fe = 247
ter = 0
ter2 = Val(Text74.Text) - 1
n = 6
m = 0
'
m4 = 0
n4 = 6
conta_fe4 = 247
acum4 = 0
'

yo = "07/09/2005"
dedo = yo
Do While (yo < Date)

If (Date > dedo) Then
  conta_fe = conta_fe + 1
  yo = yo + 1
  fe = Date - conta_fe

End If
Loop

If (Date = dedo) Then
  fe = conta_fe
End If

fe_pas = fe
Do While (acum <= 52)
  acum = acum + 1
  m = m + 1
  If (acum > 1) Then
    n = n + 7
  End If
```

Apéndice

```
dato = conta_fe
fe_pas = Date - (dato) + n
menor = Date - (dato) + n - 6
mayor = Date - (dato) + n
pruebab = menor
pruebac = mayor
```

```
'buscar semana actual
```

```
Do While (acum4 <= 52)
acum4 = acum4 + 1
m4 = m4 + 1
If (acum4 > 1) Then
    n4 = n4 + 7
End If
```

```
dato4 = conta_fe
fe_pas4 = Date - (dato4) + n4
menor4 = Date - (dato4) + n4 - 6
mayor4 = Date - (dato4) + n4
pruebab4 = menor4
pruebac4 = mayor4
```

```
If (jorgito = 0) Then
If (prueba >= pruebab4) Then
    If (prueba <= pruebac4) Then
        Text18.Text = acum4 'semana actual de pago
        acum3 = acum4 'acumular semana actual
        sactual = acum4
        acum3 = sactual
    End If
End If
End If
Loop
```

```
If (jorgito = 0) Then
If (vanderita = 0) Then
If (sem_entrada >= pruebab) Then
    If (sem_entrada <= pruebac) Then
        Text19.Text = acum 'semana de inicio del alumno
        jinicio = acum
        acum2 = acum 'acumular semana de inicio
        jas = acum 'aculular el valor para otra funcion
        vandra = 1
        vanderita = 1
    End If
End If
End If
End If
```

```
csemana = csemana + 1
>List1.AddItem csemana
fecha_acu = fe_pas
```

```
'consultar adeudos
If (oso = 0) Then
If (jorgito = 1) Then
    acum3 = sactual2
    acum = jinicio
    vandra = 1
    oso = 1
End If
End If
```

```
If (vandra = 1) Then
```

Apéndice

```
If (acum <= acum3) Then 'limite
  If (Data5.Recordset.Fields("Clave_Alumno") = (jor4)) Then

    If (acum = 1) Then

      If (Data5.Recordset.Fields("Sem1") >= ter) Then
        If (Data5.Recordset.Fields("Sem1") <= ter2) Then
          If (Data5.Recordset.Fields("Sem1") = 0) Then
            precio = precio + Data5.Recordset.Fields("Sem1")
            precio = precio + Val(Text74.Text)
          Else
            precio = precio + (jortere - Data5.Recordset.Fields("Sem1"))
          End If
          cadeu = cadeu + 1
          s1 = 1
          List1.AddItem acum
        Else
          Adeudo.Caption = "0"
        End If
      End If
    End If

    If (acum = 2) Then
      If (Data5.Recordset.Fields("Sem2") >= ter) Then
        If (Data5.Recordset.Fields("Sem2") <= ter2) Then
          If (Data5.Recordset.Fields("Sem2") = 0) Then
            precio = precio + Data5.Recordset.Fields("Sem2")
            precio = precio + Val(Text74.Text)
          Else
            precio = precio + (jortere - Data5.Recordset.Fields("Sem2"))
          End If
          cadeu = cadeu + 1
          s2 = 1
          List1.AddItem acum
        Else
          Adeudo.Caption = "0"
        End If
      End If
    End If

    If (acum = 3) Then
      If (Data5.Recordset.Fields("Sem3") >= ter) Then
        If (Data5.Recordset.Fields("Sem3") <= ter2) Then
          If (Data5.Recordset.Fields("Sem3") = 0) Then
            precio = precio + Data5.Recordset.Fields("Sem3")
            precio = precio + Val(Text74.Text)
          Else
            precio = precio + (jortere - Data5.Recordset.Fields("Sem3"))
          End If
          cadeu = cadeu + 1
          s3 = 1
          List1.AddItem acum
        Else
          Adeudo.Caption = "0"
        End If
      End If
    End If

    If (acum = 4) Then
      If (Data5.Recordset.Fields("Sem4") >= ter) Then
        If (Data5.Recordset.Fields("Sem4") <= ter2) Then
          If (Data5.Recordset.Fields("Sem4") = 0) Then
            precio = precio + Data5.Recordset.Fields("Sem4")
            precio = precio + Val(Text74.Text)
          Else
            precio = precio + (jortere - Data5.Recordset.Fields("Sem4"))
          End If
          cadeu = cadeu + 1
          s4 = 1
          List1.AddItem acum
        Else
          Adeudo.Caption = "0"
        End If
      End If
    End If
  End If
End If
```

```
Adeudo.Caption = "0"
End If
End If
End If
If (acum = 5) Then
  If (Data5.Recordset.Fields("Sem5") >= ter) Then
    If (Data5.Recordset.Fields("Sem5") <= ter2) Then
      If (Data5.Recordset.Fields("Sem5") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem5")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem5"))
      End If
      cadeu = cadeu + 1
      s5 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 6) Then
  If (Data5.Recordset.Fields("Sem6") >= ter) Then
    If (Data5.Recordset.Fields("Sem6") <= ter2) Then
      If (Data5.Recordset.Fields("Sem6") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem6")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem6"))
      End If
      cadeu = cadeu + 1
      s6 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 7) Then
  If (Data5.Recordset.Fields("Sem7") >= ter) Then
    If (Data5.Recordset.Fields("Sem7") <= ter2) Then
      If (Data5.Recordset.Fields("Sem7") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem7")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem7"))
      End If
      cadeu = cadeu + 1
      s7 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 8) Then
  If (Data5.Recordset.Fields("Sem8") >= ter) Then
    If (Data5.Recordset.Fields("Sem8") <= ter2) Then
      If (Data5.Recordset.Fields("Sem8") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem8")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem8"))
      End If
      cadeu = cadeu + 1
      s8 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
```

```
End If
End If
If (acum = 9) Then
  If (Data5.Recordset.Fields("Sem9") >= ter) Then
    If (Data5.Recordset.Fields("Sem9") <= ter2) Then
      If (Data5.Recordset.Fields("Sem9") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem9")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem9"))
      End If
      cadeu = cadeu + 1
      s9 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 10) Then
  If (Data5.Recordset.Fields("Sem10") >= ter) Then
    If (Data5.Recordset.Fields("Sem10") <= ter2) Then
      If (Data5.Recordset.Fields("Sem10") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem10")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem10"))
      End If
      cadeu = cadeu + 1
      s10 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 11) Then
  If (Data5.Recordset.Fields("Sem11") >= ter) Then
    If (Data5.Recordset.Fields("Sem11") <= ter2) Then
      If (Data5.Recordset.Fields("Sem11") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem11")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem11"))
      End If
      cadeu = cadeu + 1
      s11 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 12) Then
  If (Data5.Recordset.Fields("Sem12") >= ter) Then
    If (Data5.Recordset.Fields("Sem12") <= ter2) Then
      If (Data5.Recordset.Fields("Sem12") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem12")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem12"))
      End If
      cadeu = cadeu + 1
      s12 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
End If
```


Apéndice

```
If (acum = 13) Then
  If (Data5.Recordset.Fields("Sem13") >= ter) Then
    If (Data5.Recordset.Fields("Sem13") <= ter2) Then
      If (Data5.Recordset.Fields("Sem13") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem13")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem13"))
      End If
      cadeu = cadeu + 1
      s13 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 14) Then
  If (Data5.Recordset.Fields("Sem14") >= ter) Then
    If (Data5.Recordset.Fields("Sem14") <= ter2) Then
      If (Data5.Recordset.Fields("Sem14") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem14")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem14"))
      End If
      cadeu = cadeu + 1
      s14 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 15) Then
  If (Data5.Recordset.Fields("Sem15") >= ter) Then
    If (Data5.Recordset.Fields("Sem15") <= ter2) Then
      If (Data5.Recordset.Fields("Sem15") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem15")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem15"))
      End If
      cadeu = cadeu + 1
      s15 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 16) Then
  If (Data5.Recordset.Fields("Sem16") >= ter) Then
    If (Data5.Recordset.Fields("Sem16") <= ter2) Then
      If (Data5.Recordset.Fields("Sem16") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem16")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem16"))
      End If
      cadeu = cadeu + 1
      s16 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 17) Then
  If (Data5.Recordset.Fields("Sem17") >= ter) Then
```

```
If (Data5.Recordset.Fields("Sem17") <= ter2) Then
  If (Data5.Recordset.Fields("Sem17") = 0) Then
    presio = presio + Data5.Recordset.Fields("Sem17")
    presio = presio + Val(Text74.Text)
  Else
    presio = presio + (jortere - Data5.Recordset.Fields("Sem17"))
  End If
  cadeu = cadeu + 1
  s17 = 1
  List1.AddItem acum
Else
  Adeudo.Caption = "0"
End If
End If
End If
If (acum = 18) Then
  If (Data5.Recordset.Fields("Sem18") >= ter) Then
    If (Data5.Recordset.Fields("Sem18") <= ter2) Then
      If (Data5.Recordset.Fields("Sem18") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem18")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem18"))
      End If
      cadeu = cadeu + 1
      s18 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 19) Then
  If (Data5.Recordset.Fields("Sem19") >= ter) Then
    If (Data5.Recordset.Fields("Sem19") <= ter2) Then
      If (Data5.Recordset.Fields("Sem19") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem19")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem19"))
      End If
      cadeu = cadeu + 1
      s19 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 20) Then
  If (Data5.Recordset.Fields("Sem20") >= ter) Then
    If (Data5.Recordset.Fields("Sem20") <= ter2) Then
      If (Data5.Recordset.Fields("Sem20") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem20")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem20"))
      End If
      cadeu = cadeu + 1
      s20 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 21) Then
  If (Data5.Recordset.Fields("Sem21") >= ter) Then
    If (Data5.Recordset.Fields("Sem21") <= ter2) Then
      If (Data5.Recordset.Fields("Sem21") = 0) Then
```

```
        presio = presio + Data5.Recordset.Fields("Sem21")
        presio = presio + Val(Text74.Text)
    Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem21"))
    End If
    cadeu = cadeu + 1
    s21 = 1
    List1.AddItem acum
Else
    Adeudo.Caption = "0"
End If
End If
End If
If (acum = 22) Then
    If (Data5.Recordset.Fields("Sem22") >= ter) Then
        If (Data5.Recordset.Fields("Sem22") <= ter2) Then
            If (Data5.Recordset.Fields("Sem22") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem22")
                presio = presio + Val(Text74.Text)
            Else
                presio = presio + (jortere - Data5.Recordset.Fields("Sem22"))
            End If
            cadeu = cadeu + 1
            s22 = 1
            List1.AddItem acum
        Else
            Adeudo.Caption = "0"
        End If
    End If
End If
If (acum = 23) Then
    If (Data5.Recordset.Fields("Sem23") >= ter) Then
        If (Data5.Recordset.Fields("Sem23") <= ter2) Then
            If (Data5.Recordset.Fields("Sem23") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem23")
                presio = presio + Val(Text74.Text)
            Else
                presio = presio + (jortere - Data5.Recordset.Fields("Sem23"))
            End If
            cadeu = cadeu + 1
            s23 = 1
            List1.AddItem acum
        Else
            Adeudo.Caption = "0"
        End If
    End If
End If
If (acum = 24) Then
    If (Data5.Recordset.Fields("Sem24") >= ter) Then
        If (Data5.Recordset.Fields("Sem24") <= ter2) Then
            If (Data5.Recordset.Fields("Sem24") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem24")
                presio = presio + Val(Text74.Text)
            Else
                presio = presio + (jortere - Data5.Recordset.Fields("Sem24"))
            End If
            cadeu = cadeu + 1
            s24 = 1
            List1.AddItem acum
        Else
            Adeudo.Caption = "0"
        End If
    End If
End If
If (acum = 25) Then
    If (Data5.Recordset.Fields("Sem25") >= ter) Then
        If (Data5.Recordset.Fields("Sem25") <= ter2) Then
            If (Data5.Recordset.Fields("Sem25") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem25")
                presio = presio + Val(Text74.Text)
            End If
        End If
    End If
End If
```

```
Else
    presio = presio + (jortere - Data5.Recordset.Fields("Sem25"))
End If
cadeu = cadeu + 1
s25 = 1
List1.AddItem acum
Else
    Adeudo.Caption = "0"
End If
End If
End If
If (acum = 26) Then
    If (Data5.Recordset.Fields("Sem26") >= ter) Then
        If (Data5.Recordset.Fields("Sem26") <= ter2) Then
            If (Data5.Recordset.Fields("Sem26") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem26")
                presio = presio + Val(Text74.Text)
            Else
                presio = presio + (jortere - Data5.Recordset.Fields("Sem26"))
            End If
            cadeu = cadeu + 1
            s26 = 1
            List1.AddItem acum
        Else
            Adeudo.Caption = "0"
        End If
    End If
End If
If (acum = 27) Then
    If (Data5.Recordset.Fields("Sem27") >= ter) Then
        If (Data5.Recordset.Fields("Sem27") <= ter2) Then
            If (Data5.Recordset.Fields("Sem27") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem27")
                presio = presio + Val(Text74.Text)
            Else
                presio = presio + (jortere - Data5.Recordset.Fields("Sem27"))
            End If
            cadeu = cadeu + 1
            s27 = 1
            List1.AddItem acum
        Else
            Adeudo.Caption = "0"
        End If
    End If
End If
If (acum = 28) Then
    If (Data5.Recordset.Fields("Sem28") >= ter) Then
        If (Data5.Recordset.Fields("Sem28") <= ter2) Then
            If (Data5.Recordset.Fields("Sem28") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem28")
                presio = presio + Val(Text74.Text)
            Else
                presio = presio + (jortere - Data5.Recordset.Fields("Sem28"))
            End If
            cadeu = cadeu + 1
            s28 = 1
            List1.AddItem acum
        Else
            Adeudo.Caption = "0"
        End If
    End If
End If
If (acum = 29) Then
    If (Data5.Recordset.Fields("Sem29") >= ter) Then
        If (Data5.Recordset.Fields("Sem29") <= ter2) Then
            If (Data5.Recordset.Fields("Sem29") = 0) Then
                presio = presio + Data5.Recordset.Fields("Sem29")
                presio = presio + Val(Text74.Text)
            Else
                presio = presio + (jortere - Data5.Recordset.Fields("Sem29"))
            End If
        End If
    End If
End If
```

```
End If
cadeu = cadeu + 1
s29 = 1
List1.AddItem acum
Else
Adeudo.Caption = "0"
End If
End If
End If
If (acum = 30) Then
If (Data5.Recordset.Fields("Sem30") >= ter) Then
If (Data5.Recordset.Fields("Sem30") <= ter2) Then
If (Data5.Recordset.Fields("Sem30") = 0) Then
presio = presio + Data5.Recordset.Fields("Sem30")
presio = presio + Val(Text74.Text)
Else
presio = presio + (jortere - Data5.Recordset.Fields("Sem30"))
End If
cadeu = cadeu + 1
s30 = 1
List1.AddItem acum
Else
Adeudo.Caption = "0"
End If
End If
End If
If (acum = 31) Then
If (Data5.Recordset.Fields("Sem31") >= ter) Then
If (Data5.Recordset.Fields("Sem31") <= ter2) Then
If (Data5.Recordset.Fields("Sem31") = 0) Then
presio = presio + Data5.Recordset.Fields("Sem31")
presio = presio + Val(Text74.Text)
Else
presio = presio + (jortere - Data5.Recordset.Fields("Sem31"))
End If
cadeu = cadeu + 1
s31 = 1
List1.AddItem acum
Else
Adeudo.Caption = "0"
End If
End If
End If
If (acum = 32) Then
If (Data5.Recordset.Fields("Sem32") >= ter) Then
If (Data5.Recordset.Fields("Sem32") <= ter2) Then
If (Data5.Recordset.Fields("Sem32") = 0) Then
presio = presio + Data5.Recordset.Fields("Sem32")
presio = presio + Val(Text74.Text)
Else
presio = presio + (jortere - Data5.Recordset.Fields("Sem32"))
End If
cadeu = cadeu + 1
s32 = 1
List1.AddItem acum
Else
Adeudo.Caption = "0"
End If
End If
End If
If (acum = 33) Then
If (Data5.Recordset.Fields("Sem33") >= ter) Then
If (Data5.Recordset.Fields("Sem33") <= ter2) Then
If (Data5.Recordset.Fields("Sem33") = 0) Then
presio = presio + Data5.Recordset.Fields("Sem33")
presio = presio + Val(Text74.Text)
Else
presio = presio + (jortere - Data5.Recordset.Fields("Sem33"))
End If
cadeu = cadeu + 1
```

```
s33 = 1
List1.AddItem acum
Else
  Adeudo.Caption = "0"
End If
End If
End If
If (acum = 34) Then
  If (Data5.Recordset.Fields("Sem34") >= ter) Then
    If (Data5.Recordset.Fields("Sem34") <= ter2) Then
      If (Data5.Recordset.Fields("Sem34") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem34")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem34"))
      End If
      cadeu = cadeu + 1
      s34 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 35) Then
  If (Data5.Recordset.Fields("Sem35") >= ter) Then
    If (Data5.Recordset.Fields("Sem35") <= ter2) Then
      If (Data5.Recordset.Fields("Sem35") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem35")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem35"))
      End If
      cadeu = cadeu + 1
      s35 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 36) Then
  If (Data5.Recordset.Fields("Sem36") >= ter) Then
    If (Data5.Recordset.Fields("Sem36") <= ter2) Then
      If (Data5.Recordset.Fields("Sem36") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem36")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem36"))
      End If
      cadeu = cadeu + 1
      s36 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 37) Then
  If (Data5.Recordset.Fields("Sem37") >= ter) Then
    If (Data5.Recordset.Fields("Sem37") <= ter2) Then
      If (Data5.Recordset.Fields("Sem37") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem37")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem37"))
      End If
      cadeu = cadeu + 1
      s37 = 1
      List1.AddItem acum
    End If
  End If
End If
```

```
Else
  Adeudo.Caption = "0"
End If
End If
End If
If (acum = 38) Then
  If (Data5.Recordset.Fields("Sem38") >= ter) Then
    If (Data5.Recordset.Fields("Sem38") <= ter2) Then
      If (Data5.Recordset.Fields("Sem38") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem38")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem38"))
      End If
      cadeu = cadeu + 1
      s38 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 39) Then
  If (Data5.Recordset.Fields("Sem39") >= ter) Then
    If (Data5.Recordset.Fields("Sem39") <= ter2) Then
      If (Data5.Recordset.Fields("Sem39") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem39")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem39"))
      End If
      cadeu = cadeu + 1
      s39 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 40) Then
  If (Data5.Recordset.Fields("Sem40") >= ter) Then
    If (Data5.Recordset.Fields("Sem40") <= ter2) Then
      If (Data5.Recordset.Fields("Sem40") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem40")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem40"))
      End If
      cadeu = cadeu + 1
      s40 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 41) Then
  If (Data5.Recordset.Fields("Sem41") >= ter) Then
    If (Data5.Recordset.Fields("Sem41") <= ter2) Then
      If (Data5.Recordset.Fields("Sem41") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem41")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem41"))
      End If
      cadeu = cadeu + 1
      s41 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
```

```
End If
End If
End If
If (acum = 42) Then
  If (Data5.Recordset.Fields("Sem42") >= ter) Then
    If (Data5.Recordset.Fields("Sem42") <= ter2) Then
      If (Data5.Recordset.Fields("Sem42") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem42")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem42"))
      End If
      cadeu = cadeu + 1
      s42 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 43) Then
  If (Data5.Recordset.Fields("Sem43") >= ter) Then
    If (Data5.Recordset.Fields("Sem43") <= ter2) Then
      If (Data5.Recordset.Fields("Sem43") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem43")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem43"))
      End If
      cadeu = cadeu + 1
      s43 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 44) Then
  If (Data5.Recordset.Fields("Sem44") >= ter) Then
    If (Data5.Recordset.Fields("Sem44") <= ter2) Then
      If (Data5.Recordset.Fields("Sem44") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem44")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem44"))
      End If
      cadeu = cadeu + 1
      s44 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 45) Then
  If (Data5.Recordset.Fields("Sem45") >= ter) Then
    If (Data5.Recordset.Fields("Sem45") <= ter2) Then
      If (Data5.Recordset.Fields("Sem45") = 0) Then
        presio = presio + Data5.Recordset.Fields("Sem45")
        presio = presio + Val(Text74.Text)
      Else
        presio = presio + (jortere - Data5.Recordset.Fields("Sem45"))
      End If
      cadeu = cadeu + 1
      s45 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
```



```
End If
If (acum = 46) Then
  If (Data5.Recordset.Fields("Sem46") >= ter) Then
    If (Data5.Recordset.Fields("Sem46") <= ter2) Then
      If (Data5.Recordset.Fields("Sem46") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem46")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem46"))
      End If
      cadeu = cadeu + 1
      s46 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 47) Then
  If (Data5.Recordset.Fields("Sem47") >= ter) Then
    If (Data5.Recordset.Fields("Sem47") <= ter2) Then
      If (Data5.Recordset.Fields("Sem47") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem47")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem47"))
      End If
      cadeu = cadeu + 1
      s47 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 48) Then
  If (Data5.Recordset.Fields("Sem48") >= ter) Then
    If (Data5.Recordset.Fields("Sem48") <= ter2) Then
      If (Data5.Recordset.Fields("Sem48") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem48")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem48"))
      End If
      cadeu = cadeu + 1
      s48 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 49) Then
  If (Data5.Recordset.Fields("Sem49") >= ter) Then
    If (Data5.Recordset.Fields("Sem49") <= ter2) Then
      If (Data5.Recordset.Fields("Sem49") = 0) Then
        precio = precio + Data5.Recordset.Fields("Sem49")
        precio = precio + Val(Text74.Text)
      Else
        precio = precio + (jortere - Data5.Recordset.Fields("Sem49"))
      End If
      cadeu = cadeu + 1
      s49 = 1
      List1.AddItem acum
    Else
      Adeudo.Caption = "0"
    End If
  End If
End If
If (acum = 50) Then
```

Apéndice

```
If (Data5.Recordset.Fields("Sem50") >= ter) Then
If (Data5.Recordset.Fields("Sem50") <= ter2) Then
  If (Data5.Recordset.Fields("Sem50") = 0) Then
    presio = presio + Data5.Recordset.Fields("Sem50")
    presio = presio + Val(Text74.Text)
  Else
    presio = presio + (jortere - Data5.Recordset.Fields("Sem50"))
  End If
  cadeu = cadeu + 1
  s50 = 1
  List1.AddItem acum
Else
  Adeudo.Caption = "0"
End If
End If
End If
If (acum = 51) Then
If (Data5.Recordset.Fields("Sem51") >= ter) Then
If (Data5.Recordset.Fields("Sem51") <= ter2) Then
  If (Data5.Recordset.Fields("Sem51") = 0) Then
    presio = presio + Data5.Recordset.Fields("Sem51")
    presio = presio + Val(Text74.Text)
  Else
    presio = presio + (jortere - Data5.Recordset.Fields("Sem51"))
  End If
  cadeu = cadeu + 1
  s51 = 1
  List1.AddItem acum
Else
  Adeudo.Caption = "0"
End If
End If
End If
If (acum = 52) Then
If (Data5.Recordset.Fields("Sem52") >= ter) Then
If (Data5.Recordset.Fields("Sem52") <= ter2) Then
  If (Data5.Recordset.Fields("Sem52") = 0) Then
    presio = presio + Data5.Recordset.Fields("Sem52")
    presio = presio + Val(Text74.Text)
  Else
    presio = presio + (jortere - Data5.Recordset.Fields("Sem52"))
  End If
  cadeu = cadeu + 1
  s52 = 1
  List1.AddItem acum
Else
  Adeudo.Caption = "0"
End If
End If
End If

End If
End If 'de vandra
End If 'limite

Adeudo.Caption = cadeu
Label13.Caption = presio

Matricula.Locked = True
Combo1.SetFocus
'End If
Loop

Text17.Text = fecha_acu

End Sub

Function sustituir_pago()
```

Apéndice

```
Dim acum2 As Integer
acum2 = 0
Do While (acum2 <= 52)
    If (acum2 = 0) Then
        acum2 = jas
    End If
    acum2 = acum2 + 1

Loop
m_pago
End Function

'codigo para Buscar los servicios
Private Sub m_pago()
    Dim jor8 As Integer
    jor8 = Matricula.Text
    Data6.Recordset.MoveFirst
    Do While (Data6.Recordset.Fields("Clave_Alumno") <> (jor8) And Not Data6.Recordset.EOF)
        If (Data6.Recordset.Fields("Clave_Alumno") = Matricula.Text) Then
            GoTo jorgebrs
        End If
        Data6.Recordset.MoveNext
    If Data6.Recordset.EOF Then
        ' MsgBox "La clave del servicio no existe..., Verifique la Informacion", vbInformation, "ERROR DE LECTURA"
        Data6.Recordset.MoveFirst

    Exit Sub
    End If

Loop
jorgebrs:
If (Data6.Recordset.Fields("Clave_Alumno") = (jor8)) Then
    Text21.Text = Data6.Recordset.Fields("Sem1")
    Text22.Text = Data6.Recordset.Fields("Sem2")
    Text23.Text = Data6.Recordset.Fields("Sem3")
    Text24.Text = Data6.Recordset.Fields("Sem4")
    Text25.Text = Data6.Recordset.Fields("Sem5")
    Text26.Text = Data6.Recordset.Fields("Sem6")
    Text27.Text = Data6.Recordset.Fields("Sem7")
    Text28.Text = Data6.Recordset.Fields("Sem8")
    Text29.Text = Data6.Recordset.Fields("Sem9")
    Text30.Text = Data6.Recordset.Fields("Sem10")
    Text31.Text = Data6.Recordset.Fields("Sem11")
    Text32.Text = Data6.Recordset.Fields("Sem12")
    Text33.Text = Data6.Recordset.Fields("Sem13")
    Text34.Text = Data6.Recordset.Fields("Sem14")
    Text35.Text = Data6.Recordset.Fields("Sem15")
    Text36.Text = Data6.Recordset.Fields("Sem16")
    Text37.Text = Data6.Recordset.Fields("Sem17")
    Text38.Text = Data6.Recordset.Fields("Sem18")
    Text39.Text = Data6.Recordset.Fields("Sem19")
    Text40.Text = Data6.Recordset.Fields("Sem20")
    Text41.Text = Data6.Recordset.Fields("Sem21")
    Text42.Text = Data6.Recordset.Fields("Sem22")
    Text43.Text = Data6.Recordset.Fields("Sem23")
    Text44.Text = Data6.Recordset.Fields("Sem24")
    Text45.Text = Data6.Recordset.Fields("Sem25")
    Text46.Text = Data6.Recordset.Fields("Sem26")
    Text47.Text = Data6.Recordset.Fields("Sem27")
    Text48.Text = Data6.Recordset.Fields("Sem28")
    Text49.Text = Data6.Recordset.Fields("Sem29")
    Text50.Text = Data6.Recordset.Fields("Sem30")
    Text51.Text = Data6.Recordset.Fields("Sem31")
    Text52.Text = Data6.Recordset.Fields("Sem32")
    Text53.Text = Data6.Recordset.Fields("Sem33")
    Text54.Text = Data6.Recordset.Fields("Sem34")
    Text55.Text = Data6.Recordset.Fields("Sem35")
    Text56.Text = Data6.Recordset.Fields("Sem36")
    Text57.Text = Data6.Recordset.Fields("Sem37")
    Text58.Text = Data6.Recordset.Fields("Sem38")
```

Apéndice

```
Text59.Text = Data6.Recordset.Fields("Sem39")
Text60.Text = Data6.Recordset.Fields("Sem40")
Text61.Text = Data6.Recordset.Fields("Sem41")
Text62.Text = Data6.Recordset.Fields("Sem42")
Text63.Text = Data6.Recordset.Fields("Sem43")
Text64.Text = Data6.Recordset.Fields("Sem44")
Text65.Text = Data6.Recordset.Fields("Sem45")
Text66.Text = Data6.Recordset.Fields("Sem46")
Text67.Text = Data6.Recordset.Fields("Sem47")
Text68.Text = Data6.Recordset.Fields("Sem48")
Text69.Text = Data6.Recordset.Fields("Sem49")
Text70.Text = Data6.Recordset.Fields("Sem50")
Text71.Text = Data6.Recordset.Fields("Sem51")
Text72.Text = Data6.Recordset.Fields("Sem52")
End If
```

```
End Sub
```

```
Function sem_cero()
```

```
s1 = 0
s2 = 0
s3 = 0
s4 = 0
s5 = 0
s6 = 0
s7 = 0
s8 = 0
s9 = 0
s10 = 0
s11 = 0
s12 = 0
s13 = 0
s14 = 0
s15 = 0
s16 = 0
s17 = 0
s18 = 0
s19 = 0
s20 = 0
s21 = 0
s22 = 0
s23 = 0
s24 = 0
s25 = 0
s26 = 0
s27 = 0
s28 = 0
s29 = 0
s30 = 0
s31 = 0
s32 = 0
s33 = 0
s34 = 0
s35 = 0
s36 = 0
s37 = 0
s38 = 0
s39 = 0
s40 = 0
s41 = 0
s42 = 0
s43 = 0
s44 = 0
s45 = 0
s46 = 0
s47 = 0
s48 = 0
s49 = 0
s50 = 0
s51 = 0
```

Apéndice

```
s52 = 0
End Function
```

```
Private Sub cambiar_pago()
Dim jo10 As Integer
Dim bet, rest1 As Integer
```

```
'jo10 = Matricula.Text
jo10 = Text12.Text
Data6.Recordset.MoveFirst
Do While (Data6.Recordset.Fields("Clave_Alumno") <> (jo10) And Not Data6.Recordset.EOF)
  If (Data6.Recordset.Fields("Clave_Alumno") = Text12.Text) Then
    GoTo jofw
  End If
  Data6.Recordset.MoveNext
  If Data5.Recordset.EOF Then
    Data6.Recordset.MoveFirst
    Exit Sub
  End If
Loop
jofw:
'recorrer adeudos
If (Label13.Caption = "0") Then
  jorgito = 1
  acum3 = sactual + Val(Adeudo.Caption)
  sactual2 = acum3
  funcion_adeu
End If
```

```
If (Data6.Recordset.Fields("Clave_Alumno") = (jo10)) Then
  ac1 = Val(Text76.Text)
  'numr = Val(Text2.Text)
  numr = toto
  If (s1 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text21.Text
    If (bet <= ac1) Then

      numr = numr - bet

      If (ac1 > 0) Then
        Text21.Text = Val(Text21) + bet
        If (numr > 0) Then
          s2 = 1
        End If
      Else
        Text21.Text = Val(Text21) + bet
        GoTo salte
      End If
    Else
      Text21.Text = Val(Text21.Text) + ac1
      ac1 = ac1 - ac1
      numr = ac1
    End If
  End If
End If
,
```

```
If (s2 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text22.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text22.Text = Val(Text22) + bet
      If (numr > 0) Then
```

Apéndice

```
        s3 = 1
    End If
    Else
        Text22.Text = Val(Text22) + bet
        GoTo salte
    End If
    Else
        Text22.Text = Val(Text22.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s3 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text23.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text23.Text = Val(Text23) + bet
            If (numr > 0) Then
                s4 = 1
            End If
        Else
            Text23.Text = Val(Text23) + bet
            GoTo salte
        End If
    Else
        Text23.Text = Val(Text23.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s4 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text24.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text24.Text = Val(Text24) + bet
            If (numr > 0) Then
                s5 = 1
            End If
        Else
            Text24.Text = Val(Text24) + bet
            GoTo salte
        End If
    Else
        Text24.Text = Val(Text24.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s5 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text25.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
```

Apéndice

```
Text25.Text = Val(Text25) + bet
If (numr > 0) Then
    s6 = 1
End If
Else
    Text25.Text = Val(Text25) + bet
    GoTo salte
End If
Else
    Text25.Text = Val(Text25.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If
,
```

```
If (s6 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text26.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text26.Text = Val(Text26) + bet
            If (numr > 0) Then
                s7 = 1
            End If
        Else
            Text26.Text = Val(Text26) + bet
            GoTo salte
        End If
    Else
        Text26.Text = Val(Text26.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s7 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text27.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text27.Text = Val(Text27) + bet
            If (numr > 0) Then
                s8 = 1
            End If
        Else
            Text27.Text = Val(Text27) + bet
            GoTo salte
        End If
    Else
        Text27.Text = Val(Text27.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s8 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text28.Text
    If (bet <= ac1) Then

        numr = numr - bet
```

Apéndice

```
If (ac1 > 0) Then
  Text28.Text = Val(Text28) + bet
  If (numr > 0) Then
    s9 = 1
  End If
Else
  Text28.Text = Val(Text28) + bet
  GoTo salte
End If
Else
  Text28.Text = Val(Text28.Text) + ac1
  ac1 = ac1 - ac1
  numr = ac1
End If
End If
,
```

```
If (s9 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text29.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text29.Text = Val(Text29) + bet
      If (numr > 0) Then
        s10 = 1
      End If
    Else
      Text29.Text = Val(Text29) + bet
      GoTo salte
    End If
  Else
    Text29.Text = Val(Text29.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,
```

```
If (s10 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text30.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text30.Text = Val(Text30) + bet
      If (numr > 0) Then
        s11 = 1
      End If
    Else
      Text30.Text = Val(Text30) + bet
      GoTo salte
    End If
  Else
    Text30.Text = Val(Text30.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,
```

```
If (s11 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text31.Text
  If (bet <= ac1) Then
```


Apéndice

```
    numr = numr - bet

    If (ac1 > 0) Then
        Text31.Text = Val(Text31) + bet
        If (numr > 0) Then
            s12 = 1
        End If
    Else
        Text31.Text = Val(Text31) + bet
        GoTo salte
    End If
Else
    Text31.Text = Val(Text31.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If
,
```

```
If (s12 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text32.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text32.Text = Val(Text32) + bet
            If (numr > 0) Then
                s13 = 1
            End If
        Else
            Text32.Text = Val(Text32) + bet
            GoTo salte
        End If
    Else
        Text32.Text = Val(Text32.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s13 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text33.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text33.Text = Val(Text33) + bet
            If (numr > 0) Then
                s14 = 1
            End If
        Else
            Text33.Text = Val(Text33) + bet
            GoTo salte
        End If
    Else
        Text33.Text = Val(Text33.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s14 = 1) Then
    ac1 = numr
```

Apéndice

```
bet = Val(Text74.Text) - Text34.Text
If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
        Text34.Text = Val(Text34) + bet
        If (numr > 0) Then
            s15 = 1
        End If
    Else
        Text34.Text = Val(Text34) + bet
        GoTo salte
    End If
Else
    Text34.Text = Val(Text34.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If
,
```

```
If (s15 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text35.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text35.Text = Val(Text35) + bet
            If (numr > 0) Then
                s16 = 1
            End If
        Else
            Text35.Text = Val(Text35) + bet
            GoTo salte
        End If
    Else
        Text35.Text = Val(Text35.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

```
If (s16 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text36.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text36.Text = Val(Text36) + bet
            If (numr > 0) Then
                s17 = 1
            End If
        Else
            Text36.Text = Val(Text36) + bet
            GoTo salte
        End If
    Else
        Text36.Text = Val(Text36.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
```

Apéndice

```
If (s17 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text37.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text37.Text = Val(Text37) + bet
      If (numr > 0) Then
        s18 = 1
      End If
    Else
      Text37.Text = Val(Text37) + bet
      GoTo salte
    End If
  Else
    Text37.Text = Val(Text37.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,
```

```
If (s18 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text38.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text38.Text = Val(Text38) + bet
      If (numr > 0) Then
        s19 = 1
      End If
    Else
      Text38.Text = Val(Text38) + bet
      GoTo salte
    End If
  Else
    Text38.Text = Val(Text38.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,
```

```
If (s19 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text39.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text39.Text = Val(Text39) + bet
      If (numr > 0) Then
        s20 = 1
      End If
    Else
      Text39.Text = Val(Text39) + bet
      GoTo salte
    End If
  Else
    Text39.Text = Val(Text39.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
```

Apéndice

```
End If
,

If (s20 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text40.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text40.Text = Val(Text40) + bet
      If (numr > 0) Then
        s21 = 1
      End If
    Else
      Text40.Text = Val(Text40) + bet
      GoTo salte
    End If
  Else
    Text40.Text = Val(Text40.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,

If (s21 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text41.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text41.Text = Val(Text41) + bet
      If (numr > 0) Then
        s22 = 1
      End If
    Else
      Text41.Text = Val(Text41) + bet
      GoTo salte
    End If
  Else
    Text41.Text = Val(Text41.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,

If (s22 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text42.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text42.Text = Val(Text42) + bet
      If (numr > 0) Then
        s23 = 1
      End If
    Else
      Text42.Text = Val(Text42) + bet
      GoTo salte
    End If
  End If
```

Apéndice

```
Else
  Text42.Text = Val(Text42.Text) + ac1
  ac1 = ac1 - ac1
  numr = ac1
End If
End If
```

```

,

If (s23 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text43.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text43.Text = Val(Text43) + bet
      If (numr > 0) Then
        s24 = 1
      End If
    Else
      Text43.Text = Val(Text43) + bet
      GoTo salte
    End If
  Else
    Text43.Text = Val(Text43.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
```

```

,

If (s24 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text44.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text44.Text = Val(Text44) + bet
      If (numr > 0) Then
        s25 = 1
      End If
    Else
      Text44.Text = Val(Text44) + bet
      GoTo salte
    End If
  Else
    Text44.Text = Val(Text44.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
```

```

,

If (s25 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text45.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text45.Text = Val(Text45) + bet
      If (numr > 0) Then
        s26 = 1
      End If
    End If
  End If
```

Apéndice

```
    End If
    Else
        Text45.Text = Val(Text45) + bet
        GoTo salte
    End If
Else
    Text45.Text = Val(Text45.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If
```

,

```
If (s26 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text46.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text46.Text = Val(Text46) + bet
            If (numr > 0) Then
                s27 = 1
            End If
        Else
            Text46.Text = Val(Text46) + bet
            GoTo salte
        End If
    Else
        Text46.Text = Val(Text46.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
```

,

```
If (s27 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text47.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text47.Text = Val(Text47) + bet
            If (numr > 0) Then
                s28 = 1
            End If
        Else
            Text47.Text = Val(Text47) + bet
            GoTo salte
        End If
    Else
        Text47.Text = Val(Text47.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
```

,

```
If (s28 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text48.Text
    If (bet <= ac1) Then

        numr = numr - bet
```

Apéndice

```
If (ac1 > 0) Then
  Text48.Text = Val(Text48) + bet
  If (numr > 0) Then
    s29 = 1
  End If
Else
  Text48.Text = Val(Text48) + bet
  GoTo salte
End If
Else
  Text48.Text = Val(Text48.Text) + ac1
  ac1 = ac1 - ac1
  numr = ac1
End If
End If
```

```

'
If (s29 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text49.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text49.Text = Val(Text49) + bet
      If (numr > 0) Then
        s30 = 1
      End If
    Else
      Text49.Text = Val(Text49) + bet
      GoTo salte
    End If
  Else
    Text49.Text = Val(Text49.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
```

```

'
If (s30 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text50.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text50.Text = Val(Text50) + bet
      If (numr > 0) Then
        s31 = 1
      End If
    Else
      Text50.Text = Val(Text50) + bet
      GoTo salte
    End If
  Else
    Text50.Text = Val(Text50.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
```

```

'
If (s31 = 1) Then
```

Apéndice

```
ac1 = numr
bet = Val(Text74.Text) - Text51.Text
If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
        Text51.Text = Val(Text51) + bet
        If (numr > 0) Then
            s32 = 1
        End If
    Else
        Text51.Text = Val(Text51) + bet
        GoTo salte
    End If
Else
    Text51.Text = Val(Text51.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If
```

```

,

If (s32 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text52.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text52.Text = Val(Text52) + bet
            If (numr > 0) Then
                s33 = 1
            End If
        Else
            Text52.Text = Val(Text52) + bet
            GoTo salte
        End If
    Else
        Text52.Text = Val(Text52.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If

,
```

```


If (s33 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text53.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text53.Text = Val(Text53) + bet
            If (numr > 0) Then
                s34 = 1
            End If
        Else
            Text53.Text = Val(Text53) + bet
            GoTo salte
        End If
    Else
        Text53.Text = Val(Text53.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
```


Apéndice

```
End If
End If
'

If (s34 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text54.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text54.Text = Val(Text54) + bet
      If (numr > 0) Then
        s35 = 1
      End If
    Else
      Text54.Text = Val(Text54) + bet
      GoTo salte
    End If
  Else
    Text54.Text = Val(Text54.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
'

If (s35 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text55.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text55.Text = Val(Text55) + bet
      If (numr > 0) Then
        s36 = 1
      End If
    Else
      Text55.Text = Val(Text55) + bet
      GoTo salte
    End If
  Else
    Text55.Text = Val(Text55.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
'

If (s36 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text56.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text56.Text = Val(Text56) + bet
      If (numr > 0) Then
        s37 = 1
      End If
    Else
      Text56.Text = Val(Text56) + bet
      GoTo salte
    End If
  End If
End If
```

```
Else
  Text56.Text = Val(Text56.Text) + ac1
  ac1 = ac1 - ac1
  numr = ac1
End If
End If
,
If (s37 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text57.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text57.Text = Val(Text57) + bet
      If (numr > 0) Then
        s38 = 1
      End If
    Else
      Text57.Text = Val(Text57) + bet
      GoTo salte
    End If
  Else
    Text57.Text = Val(Text57.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,
If (s38 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text58.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text58.Text = Val(Text58) + bet
      If (numr > 0) Then
        s39 = 1
      End If
    Else
      Text58.Text = Val(Text58) + bet
      GoTo salte
    End If
  Else
    Text58.Text = Val(Text58.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
,
If (s39 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text59.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text59.Text = Val(Text59) + bet
      If (numr > 0) Then
        s40 = 1
      End If
    Else
      Text59.Text = Val(Text59) + bet
      GoTo salte
    End If
  End If
End If
```

Apéndice

```
Else
  Text59.Text = Val(Text59.Text) + ac1
  ac1 = ac1 - ac1
  numr = ac1
End If
End If
,
If (s40 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text60.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text60.Text = Val(Text60) + bet
      If (numr > 0) Then
        s41 = 1
      End If
    Else
      Text60.Text = Val(Text60) + bet
      GoTo salte
    End If
  Else
    Text60.Text = Val(Text60.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If

,
If (s41 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text61.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text61.Text = Val(Text61) + bet
      If (numr > 0) Then
        s42 = 1
      End If
    Else
      Text61.Text = Val(Text61) + bet
      GoTo salte
    End If
  Else
    Text61.Text = Val(Text61.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If

,
If (s42 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text62.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text62.Text = Val(Text62) + bet
      If (numr > 0) Then
        s43 = 1
      End If
    Else
```

Apéndice

```
        Text62.Text = Val(Text62) + bet
        GoTo salte
    End If
Else
    Text62.Text = Val(Text62.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If
,
If (s43 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text63.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text63.Text = Val(Text63) + bet
            If (numr > 0) Then
                s44 = 1
            End If
        Else
            Text63.Text = Val(Text63) + bet
            GoTo salte
        End If
    Else
        Text63.Text = Val(Text63.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
If (s44 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text64.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text64.Text = Val(Text64) + bet
            If (numr > 0) Then
                s45 = 1
            End If
        Else
            Text64.Text = Val(Text64) + bet
            GoTo salte
        End If
    Else
        Text64.Text = Val(Text64.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
If (s45 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text65.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text65.Text = Val(Text65) + bet
            If (numr > 0) Then
                s46 = 1
            End If
        End If
    End If
End If
```

Apéndice

```
    End If
  Else
    Text65.Text = Val(Text65) + bet
    GoTo salte
  End If
Else
  Text65.Text = Val(Text65.Text) + ac1
  ac1 = ac1 - ac1
  numr = ac1
End If
End If
```

```

'
If (s46 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text66.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text66.Text = Val(Text66) + bet
      If (numr > 0) Then
        s47 = 1
      End If
    Else
      Text66.Text = Val(Text66) + bet
      GoTo salte
    End If
  Else
    Text66.Text = Val(Text66.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
```

```

'
If (s47 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text67.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
      Text67.Text = Val(Text67) + bet
      If (numr > 0) Then
        s48 = 1
      End If
    Else
      Text67.Text = Val(Text67) + bet
      GoTo salte
    End If
  Else
    Text67.Text = Val(Text67.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
  End If
End If
```

```

'
If (s48 = 1) Then
  ac1 = numr
  bet = Val(Text74.Text) - Text68.Text
  If (bet <= ac1) Then

    numr = numr - bet

    If (ac1 > 0) Then
```

```
Text68.Text = Val(Text68) + bet
If (numr > 0) Then
    s49 = 1
End If
Else
    Text68.Text = Val(Text68) + bet
    GoTo salte
End If
Else
    Text68.Text = Val(Text68.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If
,
If (s49 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text69.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text69.Text = Val(Text69) + bet
            If (numr > 0) Then
                s50 = 1
            End If
        Else
            Text69.Text = Val(Text69) + bet
            GoTo salte
        End If
    Else
        Text69.Text = Val(Text69.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
If (s50 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text70.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text70.Text = Val(Text70) + bet
            If (numr > 0) Then
                s51 = 1
            End If
        Else
            Text70.Text = Val(Text70) + bet
            GoTo salte
        End If
    Else
        Text70.Text = Val(Text70.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If
,
If (s51 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text71.Text
    If (bet <= ac1) Then

        numr = numr - bet
```

Apéndice

```

    If (ac1 > 0) Then
        Text71.Text = Val(Text71) + bet
        If (numr > 0) Then
            s52 = 1
        End If
    Else
        Text71.Text = Val(Text71) + bet
        GoTo salte
    End If
Else
    Text71.Text = Val(Text71.Text) + ac1
    ac1 = ac1 - ac1
    numr = ac1
End If
End If

,

If (s52 = 1) Then
    ac1 = numr
    bet = Val(Text74.Text) - Text72.Text
    If (bet <= ac1) Then

        numr = numr - bet

        If (ac1 > 0) Then
            Text72.Text = Val(Text72) + bet
        Else
            Text72.Text = Val(Text72) + bet
            GoTo salte
        End If
    Else
        Text72.Text = Val(Text72.Text) + ac1
        ac1 = ac1 - ac1
        numr = ac1
    End If
End If

End If
salte:
End Sub

Function actual_fecha()

Dim w1, w2, w3, w4 As String
Dim w5, w6, w7, w8, w9, w10, w11, w12, w18 As Integer
Dim w13, w14, yo As Date
Dim w15, w19, w20 As Date
Dim w16, w17, denis As Date

w5 = 0
w11 = 247
w13 = Date
w6 = 6
w9 = 0
,

w8 = 0
w7 = 6
conta_fe4 = 247
w10 = 0

yo = "07/09/2005"
denis = yo
Do While (yo < Date)
If (Date > denis) Then
    w11 = w11 + 1
    yo = yo + 1

```

Apéndice

```
w1 = Date - w11
End If
Loop

If (Date = Label32.Caption) Then
    w1 = w11
End If

w2 = w1
Do While (w5 <= 52)
    w5 = w5 + 1
    w9 = w9 + 1
    If (w5 > 1) Then
        w6 = w6 + 7
    End If

    w12 = w11
    w2 = Date - (w12) + w6
    w3 = Date - (w12) + w6 - 6
    w4 = Date - (w12) + w6
    w14 = w3
    w15 = w4

    'buscar semana actual

    Do While (w10 <= 52)
        w10 = w10 + 1
        w8 = w8 + 1
        If (w10 > 1) Then
            w7 = w7 + 7
        End If

        w18 = w11
        fe_pas4 = Date - (w18) + w7
        w19 = Date - (w18) + w7 - 6
        w20 = Date - (w18) + w7
        w17 = w19
        w16 = w20

        If (w13 >= w17) Then
            If (w13 <= w16) Then
                Text3.Text = w10      'acumular semana actual
                ste = w10
            End If
        End If

    Loop
Loop
End Function
```


Apéndice

El formulario de conceptos de cobro se encarga de almacenar los datos correspondientes a los distintos rubros por los cuales la institución obtiene recursos económicos, lo cuales son diversos.

El formulario referido se muestra como sigue:

The screenshot shows a Windows-style window titled "CONFIGURAR CONCEPTOS DE COBRO". Inside the window, there are two main input areas. The first is labeled "CONCEPTO" and has a text box. The second is labeled "PRECIO \$" and has a text box. To the right of these inputs are four buttons stacked vertically: "NUEVO", "GRABAR", "BORRAR", and "CANCELAR". Below the input fields is a table with a header row in yellow and a data row in blue. The table is currently empty.

Figura 15.- Formulario de conceptos de cobro.

Este formulario nos dará la pauta para definir los costos de cada uno de los diferentes conceptos a cobrar.

El código utilizado es el siguiente:

```
Dim pos As Integer
Dim van As Boolean, van_b1 As Boolean
Dim van_b2 As Boolean
Dim filas As Double
Dim Cobros As Recordset
Dim Mensaje, Estilo, Título, respuesta, cadena
Dim error1 As Boolean

Private Sub cmdSalir_Click()
Unload Me
End Sub

Private Sub Busqueda_servicio()
Cobros.MoveFirst
Busqueda.Caption = Cobros.Fields("Clave_Servicio")
While van = False
If Busqueda.Caption <> Text2.Text Then
Cobros.MoveNext
If Cobros.EOF Then
MsgBox "El registro no existe", vbOKOnly + vbExclamation, ""
Clave_Txt.Text = ""
Clave_Txt.SetFocus
Servicio_Txt = ""
```

Apéndice

```
    UM_Txt = ""
    Precio_Txt = ""
    Cancelar_Click
    GoTo salto00
End If
Busqueda.Caption = Cobros.Fields("Clave_Servicio")
Else
txt1.Text = Cobros.Fields("Descripcion")
txt2.Text = Cobros.Fields("Presio")
van = True
***

Guardar.Enabled = False
Eliminar.Enabled = True
nuevo.Enabled = True
Cancelar.Enabled = True
***

permiso1 = False
End If
Wend
salto00:
van = False
End Sub

Private Sub Tabla_Llena()
Cobros.MoveFirst
While van = False
If Cobros.EOF Then
van = True
Else
Tabla_Con.Col = 1
Tabla_Con.Row = filas
Tabla_Con.Text = Cobros.Fields("Clave_Servicio")
Tabla_Con.Col = 2
Tabla_Con.Text = Cobros.Fields("Descripcion")
Tabla_Con.Col = 3
Tabla_Con.Text = Cobros.Fields("Presio")
filas = filas + 1
Tabla_Con.Rows = Tabla_Con.Rows + 1
Cobros.MoveNext
End If
Wend
Tabla_Con.Rows = Tabla_Con.Rows - 1
van = False
End Sub
Private Sub Cancelar_Click()
Text2.Text = ""
txt1.Text = ""
txt2.Text = ""
txt1.Locked = True
txt2.Locked = True

Cancelar.Enabled = True
Guardar.Enabled = False
Eliminar.Enabled = False
nuevo.Enabled = True
End Sub

Private Sub Clave_Txt_Click()
If permiso1 = False Then
Guardar.Enabled = True
Eliminar.Enabled = False
nuevo.Enabled = False
Cancelar.Enabled = True
End If
End Sub

Private Sub Clave_Txt_KeyPress(KeyAscii As Integer)
```

Apéndice

```
cadena = "0123456789" + Chr(8) + Chr(13)
If InStr(cadena, Chr(KeyAscii)) = 0 Then
    KeyAscii = 0
End If
If KeyAscii = 13 Then
    Busqueda_servicio
End If
If KeyAscii = 8 Then
    Clave_Txt_Click
End If
End Sub

Private Sub Eliminar_Click()
If (Cobros.Fields("Descripcion") = "Colegiatura") Then
    MsgBox "El concepto no puede ser Eliminado", vbCritical, "Permiso denegado"
    GoTo sg
End If

If (Cobros.Fields("Descripcion") = "Renta de Equipo de Computo") Then
    MsgBox "El concepto no puede ser Eliminado", vbCritical, "Permiso denegado"
    GoTo sg
End If
If (Cobros.Fields("Descripcion") = "Impresiones") Then
    MsgBox "El concepto no puede ser Eliminado", vbCritical, "Permiso denegado"
    GoTo sg
End If
If (Cobros.Fields("Descripcion") = "Credencial") Then
    MsgBox "El concepto no puede ser Eliminado", vbCritical, "Permiso denegado"
    GoTo sg
End If
If (Cobros.Fields("Descripcion") = "Constancia de Calificaciones") Then
    MsgBox "El concepto no puede ser Eliminado", vbCritical, "Permiso denegado"
    GoTo sg
End If

Mensaje = "¿Desea continuar?"
Estilo = vbYesNo + vbQuestion + vbDefaultButton2
Título = "Borrar"
respuesta = MsgBox(Mensaje, Estilo, Título)
If respuesta = vbYes Then
    Cobros.Delete
    '---
    Tabla_Con.Clear
    Tabla_Con.Rows = 2
    filas = 1
    Tabla
    Tabla_Llena
    Tabla_Con.Col = 1
    Tabla_Con.Sort = flexSortNumericAscending
    'Cobros.Requery
    '-----
    Cancelar.Enabled = True
    Guardar.Enabled = False
    Eliminar.Enabled = False
    nuevo.Enabled = True

    Text2.Text = ""
    txt1.Text = ""
    txt1.SetFocus
    txt2.Text = ""
End If
sg:
End Sub
Private Sub Form_Load()
Skin1.LoadSkin ("galaxy.skn")
Skin1.ApplySkin CONSEPTOS_COBRO.hWnd
```

filas = 1

Apéndice

```
van = False
van_b1 = True
van_b2 = False
permiso1 = True
error1 = True
'
Tabla
Set alumno = OpenDatabase("C:\CONTROL DE PAGOS (DATCOM)\base\COBROS.mdb")
Set Cobros = alumno.OpenRecordset("SERVICIOS")
'-----
Tabla_Llena
Tabla_Con.Col = 1
Tabla_Con.Sort = flexSortNumericAscending
'-----
Guardar.Enabled = False
Eliminar.Enabled = False
End Sub
Private Sub Tabla()
Tabla_Con.ColWidth(0) = 0
Tabla_Con.ColWidth(1) = 1000
Tabla_Con.Col = 1
Tabla_Con.Row = 0
Tabla_Con.Text = "Clave"
Tabla_Con.ColWidth(2) = 6200
Tabla_Con.Col = 2
Tabla_Con.Row = 0
Tabla_Con.Text = "Descripción"
Tabla_Con.ColWidth(3) = 1200
Tabla_Con.Col = 3
Tabla_Con.Row = 0
Tabla_Con.Text = "Presio"
End Sub
Private Sub Guardar_Click()
Mensaje = "¿Desea continuar?"
Estilo = vbYesNo + vbQuestion + vbDefaultButton2
Título = "Actualizar Registro"
If (txt1.Text = "" Or txt2.Text = "") Then
MsgBox "Error al grabar faltan datos", vbCritical + vbOKOnly, ""
GoTo jo
End If
respuesta = MsgBox(Mensaje, Estilo, Título)
If respuesta = vbYes Then
'On Error GoTo Salir

'Cobros.AddNew
Cobros.Fields("Clave_Servicio") = Text2.Text
Cobros.Fields("Descripcion") = txt1.Text
Cobros.Fields("Presio") = txt2.Text
Cobros.Update

*****
Tabla_Con.Clear
Tabla_Con.Rows = 2
filas = 1
Tabla
Tabla_Llena
Tabla_Con.Col = 1
Tabla_Con.Sort = flexSortNumericAscending
*****
Cancelar.Enabled = True
Guardar.Enabled = False
Eliminar.Enabled = False
nuevo.Enabled = True
'-----
Text2.Text = ""
txt1.Text = ""
txt1.SetFocus
txt2.Text = ""

'---
```

Apéndice

```
van_b2 = False
permiso1 = True
End If
Salir:

jo:
Cobros.Fields.Refresh
End Sub

Private Sub Nuevo_Click()
Cancelar.Enabled = True
Guardar.Enabled = True
Eliminar.Enabled = False
nuevo.Enabled = False

Text2.Text = ""
txt1.SetFocus
txt1.Text = ""
txt2.Text = ""
txt1.Locked = False
txt2.Locked = True

'codigo para recorrer la lista
If Cobros.BOF Then
cont = 1
GoTo algo
End If
Cobros.MoveLast
If Cobros.EOF Then
Cobros.MoveLast
End If
'cont = Text2.Text
Cobros.MoveLast
cont = Cobros.Fields("Clave_Servicio")

cont2 = 1
Cobros.MoveFirst
Cobros.Fields.Refresh
If (Cobros.Fields("Clave_Servicio") = cont2) Then
Do While (Cobros.Fields("Clave_Servicio") < cont And Not Cobros.EOF)
If (cont2 <> Cobros.Fields("Clave_Servicio")) Then
cont = cont2
GoTo j
End If
Cobros.MoveNext

cont2 = cont2 + 1

j:
Loop

Else

If (Cobros.Fields("Clave_Servicio") <> cont2) Then
cont2 = cont2 - 1
cont = cont2
End If

End If

If (cont2 < Cobros.Fields("Clave_Servicio")) Then
cont2 = cont2 - 1
cont = cont2
End If
If (cont2 = Cobros.Fields("Clave_Servicio")) Then
cont = cont2
End If
cont = cont + 1
If (cont = 0) Then
cont = cont + 1
```

Apéndice

```
End If  
'codigo de nuevo
```

```
    Cobros.MoveLast
```

```
algo:  
    Cobros.AddNew  
    'Data1.Recordset.AddNew  
    Text2.Text = cont  
    txt1.SetFocus
```

```
End Sub
```

```
Private Sub Precio_Txt_Click()  
If permiso1 = True Then  
    Busqueda_servicio  
    GoTo s2  
End If  
Guardar.Enabled = True  
Eliminar.Enabled = False  
nuevo.Enabled = False  
Cancelar.Enabled = True  
s2:  
End Sub
```

```
Private Sub Precio_Txt_KeyPress(KeyAscii As Integer)  
cadena = "0123456789," + Chr(8) + Chr(13)  
If InStr(cadena, Chr(KeyAscii)) = 0 Then  
    KeyAscii = 0  
End If  
End Sub
```

```
Private Sub Salir_Lbl_Click()  
Unload SIP_ServConsul  
End Sub
```

```
Private Sub Salir_Lbl_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
Salir_Lbl.ForeColor = &HE0E0E0  
End Sub
```

```
Private Sub Servicio_Txt_Click()  
If permiso1 = True Then  
    Busqueda_servicio  
    GoTo s3  
End If  
Guardar.Enabled = True  
Eliminar.Enabled = False  
nuevo.Enabled = False  
Cancelar.Enabled = True  
s3:  
End Sub
```

```
Private Sub Tabla_Con_DblClick()  
Tabla_Con.Col = 1  
Text2.Text = Tabla_Con.Text  
Busqueda_servicio  
End Sub
```

```
Private Sub UM_Txt_Click()  
If permiso1 = True Then  
    Busqueda_servicio  
    GoTo s1
```

Apéndice

```
End If
Guardar.Enabled = True
Eliminar.Enabled = False
nuevo.Enabled = False
Cancelar.Enabled = True
s1:
End Sub

'Codigo de validacion del campo clave del educando
Private Sub txt1_KeyPress(KeyAscii As Integer)
    If InStr("0123456789", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then
        KeyAscii = 0
    End If
    If (KeyAscii = 13) Then
        If (Text1.Text = "") Then
            MsgBox "Introduce el Descuento", vbInformation, "ERROR DE CAMPO"
            Text1.SetFocus
        Else
            txt2.SetFocus
            Text1.Locked = True
            txt2.Locked = False
        End If
    End If
End Sub

'Codigo de validacion del campo nombre del educando
Private Sub txt1_KeyPress(KeyAscii As Integer)
    If InStr("ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyzáéíóú", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then
        KeyAscii = 0
    End If
    If (KeyAscii = 13) Then
        If (txt1.Text = "") Then
            MsgBox "Introduce el concepto", vbInformation, "ERROR DE CAMPO"
            txt1.SetFocus
        Else
            txt2.SetFocus
            txt1.Locked = True
            txt2.Locked = False
        End If
    End If
End Sub

'Codigo de validacion del campo clave del educando
Private Sub txt2_KeyPress(KeyAscii As Integer)
    If InStr("0123456789.", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then
        KeyAscii = 0
    End If
    If (KeyAscii = 13) Then
        If (txt2.Text = "") Then
            MsgBox "Introduce el Precio", vbInformation, "ERROR DE CAMPO"
            txt2.SetFocus
        Else
            Guardar.Enabled = True
        End If
    End If
End Sub

End Sub
```

Para el formulario siguiente tenemos que decir que se parece muchísimo al anterior, porque aquí se almacenan los conceptos de descuento que se van a aplicar a nuestro sistema.

Veamos el formulario:

The screenshot shows a Windows application window titled "CONFIGURAR DESCUENTOS". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is a form with the following elements:

- CLAVE DEL DESC.:** A text input field.
- DESCRIPCION:** A text input field.
- VIGENCIA:** A date picker control showing "26/08/2005".
- TOTAL DE DESCUENTO \$:** A text input field.
- Buttons:** Four buttons are stacked vertically on the right: "NUEVO", "GRABAR", "BORRAR", and "CANCELAR".
- Descuentos:** A section below the form containing a table with two rows and five columns. The first row has yellow cells, and the second row has blue cells.

Figura 16.- Formulario de conceptos de descuento.

Y su código es el siguiente:

```
Dim pos As Integer
Dim van As Boolean, van_b1 As Boolean, permiso1 As Boolean
Dim van_b2 As Boolean
Dim filas As Double
Dim Cobros As Recordset
Dim Mensaje, Estilo, Título, respuesta, cadena
Dim error1 As Boolean
```

```
Private Sub cmdSalir_Click()
Unload Me
End Sub
```

```
Private Sub Busqueda_servicio()
Cobros.MoveFirst
Busqueda.Caption = Cobros.Fields("Clave_Descuento")
While van = False
If Busqueda.Caption <> Text2.Text Then
Cobros.MoveNext
If Cobros.EOF Then
MsgBox "El registro no existe", vbOKOnly + vbExclamation, ""
Clave_Txt.Text = ""
Clave_Txt.SetFocus
Servicio_Txt = ""
UM_Txt = ""
Precio_Txt = ""
Cancelar_Click
GoTo salto00
End If
Busqueda.Caption = Cobros.Fields("Clave_Descuento")
Else
txt1.Text = Cobros.Fields("Descripcion")
DTPicker1 = Cobros.Fields("Vigencia")
txt2.Text = Cobros.Fields("Total_Descuento")
van = True
****
Guardar.Enabled = False
```


Apéndice

```
Eliminar.Enabled = True
nuevo.Enabled = True
Cancelar.Enabled = True
****
    permiso1 = False
End If
Wend
salto00:
van = False
End Sub

Private Sub Tabla_Llena()
Cobros.MoveFirst
While van = False
If Cobros.EOF Then
    van = True
Else
    Tabla_Des.Col = 1
    Tabla_Des.Row = filas
    Tabla_Des.Text = Cobros.Fields("Clave_Descuento")
    Tabla_Des.Col = 2
    Tabla_Des.Text = Cobros.Fields("Descripcion")
    Tabla_Des.Col = 3
    Tabla_Des.Text = Cobros.Fields("Total_Descuento")
    Tabla_Des.Col = 4
    Tabla_Des.Text = Cobros.Fields("Vigencia")
    filas = filas + 1
    Tabla_Des.Rows = Tabla_Des.Rows + 1
    Cobros.MoveNext
End If
Wend
Tabla_Des.Rows = Tabla_Des.Rows - 1
van = False
End Sub

Private Sub Cancelar_Click()
Text2.Text = ""
txt1.Text = ""
txt2.Text = ""
txt1.Locked = True
txt2.Locked = True

Cancelar.Enabled = True
Guardar.Enabled = False
Eliminar.Enabled = False
nuevo.Enabled = True
End Sub

Private Sub Clave_Txt_Click()
If permiso1 = False Then
    Guardar.Enabled = True
    Eliminar.Enabled = False
    nuevo.Enabled = False
    Cancelar.Enabled = True
End If
End Sub

Private Sub Clave_Txt_KeyPress(KeyAscii As Integer)
cadena = "0123456789" + Chr(8) + Chr(13)
If InStr(cadena, Chr(KeyAscii)) = 0 Then
    KeyAscii = 0
End If
If KeyAscii = 13 Then
    Busqueda_servicio
End If
If KeyAscii = 8 Then
    Clave_Txt_Click
End If
```

Apéndice

End Sub

```
Private Sub Eliminar_Click()
Mensaje = "¿Desea continuar?"
Estilo = vbYesNo + vbQuestion + vbDefaultButton2
Título = "Borrar"
respuesta = MsgBox(Mensaje, Estilo, Título)
If respuesta = vbYes Then
Cobros.Delete
'---
Tabla_Des.Clear
Tabla_Des.Rows = 2
filas = 1
Tabla
Tabla_Llena
Tabla_Des.Col = 1
Tabla_Des.Sort = flexSortNumericAscending
'Cobros.Requery
'-----
Cancelar.Enabled = True
Guardar.Enabled = False
Eliminar.Enabled = False
nuevo.Enabled = True
```

```
Text2.Text = ""
txt1.Text = ""
txt1.SetFocus
txt2.Text = ""
```

End If

End Sub

```
Private Sub Form_Load()
Skin1.LoadSkin ("galaxy.skn")
Skin1.ApplySkin DESCUENTOS.hWnd
```

```
filas = 1
van = False
van_b1 = True
van_b2 = False
permiso1 = True
error1 = True
'
```

Tabla

```
Set alumno = OpenDatabase("C:\CONTROL DE PAGOS (DATCOM)\base\COBROS.mdb")
Set Cobros = alumno.OpenRecordset("DESCUENTOS")
'-----
```

```
Tabla_Llena
Tabla_Des.Col = 1
Tabla_Des.Sort = flexSortNumericAscending
'-----
```

```
Guardar.Enabled = False
Eliminar.Enabled = False
```

End Sub

Private Sub Tabla()

```
Tabla_Des.ColWidth(0) = 0
Tabla_Des.ColWidth(1) = 1000
Tabla_Des.Col = 1
Tabla_Des.Row = 0
Tabla_Des.Text = "Clave"
Tabla_Des.ColWidth(2) = 4550
Tabla_Des.Col = 2
Tabla_Des.Row = 0
Tabla_Des.Text = "Descripción"
Tabla_Des.ColWidth(3) = 1250
Tabla_Des.Col = 3
Tabla_Des.Row = 0
Tabla_Des.Text = "Total Desc."
```

Apéndice

```
Tabla_Des.ColWidth(4) = 1250
Tabla_Des.Col = 4
Tabla_Des.Row = 0
Tabla_Des.Text = "Vigencia"
End Sub
Private Sub Guardar_Click()
Mensaje = "¿Desea continuar?"
Estilo = vbYesNo + vbQuestion + vbDefaultButton2
Título = "Actualizar Registro"
If (txt1.Text = "" Or txt2.Text = "") Then
MsgBox "Error al grabar faltan datos", vbCritical + vbOKOnly, ""
GoTo jo
End If
respuesta = MsgBox(Mensaje, Estilo, Título)
If respuesta = vbYes Then
'On Error GoTo Salir

'Cobros.AddNew
Cobros.Fields("Clave_Descuento") = Text2.Text
Cobros.Fields("Descripcion") = txt1.Text
Cobros.Fields("Vigencia") = DTPicker1
Cobros.Fields("Total_Descuento") = txt2.Text

Cobros.Update

*****
Tabla_Des.Clear
Tabla_Des.Rows = 2
filas = 1
Tabla
Tabla_Llena
Tabla_Des.Col = 1
Tabla_Des.Sort = flexSortNumericAscending
*****

Cancelar.Enabled = True
Guardar.Enabled = False
Eliminar.Enabled = False
nuevo.Enabled = True
'----
Text2.Text = ""
txt1.Text = ""
txt1.SetFocus
txt2.Text = ""

'---
van_b2 = False
permiso1 = True
End If
Salir:

jo:
Cobros.Fields.Refresh
End Sub

Private Sub Nuevo_Click()
Cancelar.Enabled = True
Guardar.Enabled = True
Eliminar.Enabled = False
nuevo.Enabled = False

Text2.Text = ""
txt1.SetFocus
txt1.Text = ""
txt2.Text = ""
txt1.Locked = False
txt2.Locked = True

'codigo para recorrer la lista
If Cobros.BOF Then
cont = 1
```

Apéndice

```
    GoTo algo
End If
Cobros.MoveLast
If Cobros.EOF Then
    Cobros.MoveLast
End If
'cont = Text2.Text
Cobros.MoveLast
cont = Cobros.Fields("Clave_Descuento")

cont2 = 1
Cobros.MoveFirst
Cobros.Fields.Refresh
If (Cobros.Fields("Clave_Descuento") = cont2) Then
    Do While (Cobros.Fields("Clave_Descuento") < cont And Not Cobros.EOF)
        If (cont2 <> Cobros.Fields("Clave_Descuento")) Then
            cont = cont2
            GoTo j
        End If
        Cobros.MoveNext

        cont2 = cont2 + 1

j:
Loop

Else

    If (Cobros.Fields("Clave_Servicio") <> cont2) Then
        cont2 = cont2 - 1
        cont = cont2
    End If

End If

If (cont2 < Cobros.Fields("Clave_Descuento")) Then
    cont2 = cont2 - 1
    cont = cont2
End If
If (cont2 = Cobros.Fields("Clave_Descuento")) Then
    cont = cont2
End If
cont = cont + 1
If (cont = 0) Then
    cont = cont + 1
End If
'codigo de nuevo

    Cobros.MoveLast

algo:
    Cobros.AddNew
    'Data1.Recordset.AddNew
    Text2.Text = cont
    txt1.SetFocus

End Sub

Private Sub Precio_Txt_Click()
If permiso1 = True Then
    Busqueda_servicio
    GoTo s2
End If
Guardar.Enabled = True
Eliminar.Enabled = False
nuevo.Enabled = False
Cancelar.Enabled = True
```

Apéndice

```
s2:  
End Sub
```

```
Private Sub Precio_Txt_KeyPress(KeyAscii As Integer)  
cadena = "0123456789," + Chr(8) + Chr(13)  
If InStr(cadena, Chr(KeyAscii)) = 0 Then  
    KeyAscii = 0  
End If  
End Sub
```

```
Private Sub Salir_Lbl_Click()  
Unload SIP_ServConsul  
End Sub
```

```
Private Sub Salir_Lbl_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
Salir_Lbl.ForeColor = &HE0E0E0  
End Sub
```

```
Private Sub Servicio_Txt_Click()  
If permiso1 = True Then  
    Busqueda_servicio  
    GoTo s3  
End If  
Guardar.Enabled = True  
Eliminar.Enabled = False  
nuevo.Enabled = False  
Cancelar.Enabled = True  
s3:  
End Sub
```

```
Private Sub Tabla_Con_DbClick()  
Tabla_Con.Col = 1  
Text2.Text = Tabla_Con.Text  
Busqueda_servicio  
End Sub
```

```
Private Sub UM_Txt_Click()  
If permiso1 = True Then  
    Busqueda_servicio  
    GoTo s1  
End If  
Guardar.Enabled = True  
Eliminar.Enabled = False  
nuevo.Enabled = False  
Cancelar.Enabled = True  
s1:  
End Sub
```

'Codigo de validacion del campo clave del educando

```
Private Sub text1_KeyPress(KeyAscii As Integer)  
If InStr("0123456789", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then  
    KeyAscii = 0  
End If  
If (KeyAscii = 13) Then  
    If (Text1.Text = "") Then  
        MsgBox "Introduce el Descuento", vbInformation, "ERROR DE CAMPO"  
        Text1.SetFocus  
    Else  
        txt2.SetFocus  
        Text1.Locked = True  
        txt2.Locked = False  
    End If  
End If  
End If
```

Apéndice

End Sub

```
Private Sub Tabla_Des_DbClick()  
Tabla_Des.Col = 1  
Text2.Text = Tabla_Des.Text  
Busqueda_servicio  
End Sub
```

'Codigo de validacion del campo nombre del educando

```
Private Sub txt1_KeyPress(KeyAscii As Integer)  
If InStr("ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyzáéíóú", Chr(KeyAscii)) = 0 And KeyAscii <>  
13 And KeyAscii <> 8 And KeyAscii <> 32 Then  
KeyAscii = 0  
End If  
End If  
If (KeyAscii = 13) Then  
If (txt1.Text = "") Then  
MsgBox "Introduce el concepto", vbInformation, "ERROR DE CAMPO"  
txt1.SetFocus  
Else  
txt2.SetFocus  
txt1.Locked = True  
txt2.Locked = False  
  
End If  
End If
```

End Sub

'Codigo de validacion del campo clave del educando

```
Private Sub txt2_KeyPress(KeyAscii As Integer)  
If InStr("0123456789.", Chr(KeyAscii)) = 0 And KeyAscii <> 13 And KeyAscii <> 8 And KeyAscii <> 32 Then  
KeyAscii = 0  
End If  
End If  
If (KeyAscii = 13) Then  
If (txt2.Text = "") Then  
MsgBox "Introduce el Precio", vbInformation, "ERROR DE CAMPO"  
txt2.SetFocus  
Else  
Guardar.Enabled = True  
End If  
End If
```

End Sub

Ahora expondremos el formulario para los reportes de adeudos por semana:

Apéndice

```
Private Sub Command1_Click()  
Unload Me  
End Sub
```

```
Private Sub Command2_Click()  
PrintForm  
End Sub
```

```
Private Sub Form_Load()
```

```
Set base = OpenDatabase("C:\CONTROL DE PAGOS (DATCOM)\base\COBROS.mdb")
```

```
Set jor = base.OpenRecordset("ADEUDOS")  
Set alum = base.OpenRecordset("ADEUDOS")  
Set m = base.OpenRecordset("ADEUDOS")
```

```
al.ColWidth(0) = 1200  
al.ColWidth(1) = 5900  
al.ColWidth(2) = 1670
```

```
contad = 1  
End Sub
```

```
Private Sub Nuevo_Click()
```

```
violet = 0  
cal = 0  
reco = 0  
contad = 0  
Set base = OpenDatabase("C:\CONTROL DE PAGOS (DATCOM)\base\COBROS.mdb")  
Set base2 = OpenDatabase("C:\CONTROL ESCOLAR(DATCOM)\ALUMNOS.mdb")
```

```
Set jor = base.OpenRecordset("ADEUDOS")  
Set alum = base.OpenRecordset("ADEUDOS")  
Set m = base.OpenRecordset("ADEUDOS")
```

```
Set nombre = base2.OpenRecordset("ALUMNOS")
```

```
al.ColWidth(0) = 1200  
al.ColWidth(1) = 5900  
al.ColWidth(2) = 1670
```

```
re = InputBox(" ¿Introduce la semana a consultar??... Introduce primero las iniciales (Sem) seguidas del numero de  
semana... Ejemplo: 'Sem22'", "CONSULTAR ADEUDOS")
```

```
If (re = "Sem1") Then  
semanaaa = "1"  
Label2.Caption = semanaaa  
End If  
If (re = "Sem2") Then  
semanaaa = "2"  
Label2.Caption = semanaaa  
End If  
If (re = "Sem3") Then  
semanaaa = "3"  
Label2.Caption = semanaaa  
End If  
If (re = "Sem4") Then  
semanaaa = "4"  
Label2.Caption = semanaaa  
End If  
If (re = "Sem5") Then  
semanaaa = "5"  
Label2.Caption = semanaaa  
End If  
If (re = "Sem6") Then  
semanaaa = "6"
```


Apéndice

```
Label2.Caption = semanaaa
End If
If (re = "Sem7") Then
    semanaaa = "7"
    Label2.Caption = semanaaa
End If
If (re = "Sem8") Then
    semanaaa = "8"
    Label2.Caption = semanaaa
End If
If (re = "Sem9") Then
    semanaaa = "9"
    Label2.Caption = semanaaa
End If
If (re = "Sem10") Then
    semanaaa = "10"
    Label2.Caption = semanaaa
End If
If (re = "Sem11") Then
    semanaaa = "11"
    Label2.Caption = semanaaa
End If
If (re = "Sem12") Then
    semanaaa = "12"
    Label2.Caption = semanaaa
End If
If (re = "Sem13") Then
    semanaaa = "13"
    Label2.Caption = semanaaa
End If
If (re = "Sem14") Then
    semanaaa = "14"
    Label2.Caption = semanaaa
End If
If (re = "Sem15") Then
    semanaaa = "15"
    Label2.Caption = semanaaa
End If
If (re = "Sem16") Then
    semanaaa = "16"
    Label2.Caption = semanaaa
End If
If (re = "Sem17") Then
    semanaaa = "17"
    Label2.Caption = semanaaa
End If
If (re = "Sem18") Then
    semanaaa = "18"
    Label2.Caption = semanaaa
End If
If (re = "Sem19") Then
    semanaaa = "19"
    Label2.Caption = semanaaa
End If
If (re = "Sem20") Then
    semanaaa = "20"
    Label2.Caption = semanaaa
End If
If (re = "Sem21") Then
    semanaaa = "21"
    Label2.Caption = semanaaa
End If
If (re = "Sem22") Then
    semanaaa = "22"
    Label2.Caption = semanaaa
End If
If (re = "Sem23") Then
    semanaaa = "23"
    Label2.Caption = semanaaa
End If
```

Apéndice

```
If (re = "Sem24") Then
    semanaaa = "24"
    Label2.Caption = semanaaa
End If
If (re = "Sem25") Then
    semanaaa = "25"
    Label2.Caption = semanaaa
End If
If (re = "Sem26") Then
    semanaaa = "26"
    Label2.Caption = semanaaa
End If
If (re = "Sem27") Then
    semanaaa = "27"
    Label2.Caption = semanaaa
End If
If (re = "Sem28") Then
    semanaaa = "28"
    Label2.Caption = semanaaa
End If
If (re = "Sem29") Then
    semanaaa = "29"
    Label2.Caption = semanaaa
End If
If (re = "Sem30") Then
    semanaaa = "30"
    Label2.Caption = semanaaa
End If
If (re = "Sem31") Then
    semanaaa = "31"
    Label2.Caption = semanaaa
End If
If (re = "Sem32") Then
    semanaaa = "32"
    Label2.Caption = semanaaa
End If
If (re = "Sem33") Then
    semanaaa = "33"
    Label2.Caption = semanaaa
End If
If (re = "Sem34") Then
    semanaaa = "34"
    Label2.Caption = semanaaa
End If
If (re = "Sem35") Then
    semanaaa = "35"
    Label2.Caption = semanaaa
End If
If (re = "Sem36") Then
    semanaaa = "36"
    Label2.Caption = semanaaa
End If
If (re = "Sem37") Then
    semanaaa = "37"
    Label2.Caption = semanaaa
End If
If (re = "Sem38") Then
    semanaaa = "38"
    Label2.Caption = semanaaa
End If
If (re = "Sem39") Then
    semanaaa = "39"
    Label2.Caption = semanaaa
End If
If (re = "Sem40") Then
    semanaaa = "40"
    Label2.Caption = semanaaa
End If
If (re = "Sem41") Then
    semanaaa = "41"
```

Apéndice

```
Label2.Caption = semanaaa
End If
If (re = "Sem42") Then
    semanaaa = "42"
    Label2.Caption = semanaaa
End If
If (re = "Sem43") Then
    semanaaa = "43"
    Label2.Caption = semanaaa
End If
If (re = "Sem44") Then
    semanaaa = "44"
    Label2.Caption = semanaaa
End If
If (re = "Sem45") Then
    semanaaa = "45"
    Label2.Caption = semanaaa
End If
If (re = "Sem46") Then
    semanaaa = "46"
    Label2.Caption = semanaaa
End If
If (re = "Sem47") Then
    semanaaa = "47"
    Label2.Caption = semanaaa
End If
If (re = "Sem48") Then
    semanaaa = "48"
    Label2.Caption = semanaaa
End If
If (re = "Sem49") Then
    semanaaa = "49"
    Label2.Caption = semanaaa
End If
If (re = "Sem50") Then
    semanaaa = "50"
    Label2.Caption = semanaaa
End If
If (re = "Sem51") Then
    semanaaa = "51"
    Label2.Caption = semanaaa
End If
If (re = "Sem52") Then
    semanaaa = "52"
    Label2.Caption = semanaaa
End If
If (Label2.Caption = "") Then
    MsgBox "la semana solicitada no existe... intentelo nuevamente ", vbCritical, "Error de Dato"
    GoTo astal
End If

For i = 0 To (al.Rows + 1)
    With al
        .Col = 0
        .Row = .Rows - 1
        .Text = ""
        .Col = 1
        .Text = ""
        .Col = 2
        .Text = ""
    End With
Next i
al.Rows = 2

jo = 0
reco = 0

nombre.MoveFirst
```

Apéndice

```
alum.MoveFirst
Do While (Not alum.EOF)
  Tabla
  If alum.EOF Then
    alum.MoveFirst
  Exit Sub
End If
```

```
If (alum.Fields("Clave_Alumno")) = contad Then
```

```
  llamadax
  'adeudo
  If (re = "Sem1") Then
    manza = alum.Fields("Sem1")
  End If
  If (re = "Sem2") Then
    manza = alum.Fields("Sem2")
  End If
  If (re = "Sem3") Then
    manza = alum.Fields("Sem3")
  End If
  If (re = "Sem4") Then
    manza = alum.Fields("Sem4")
  End If
  If (re = "Sem5") Then
    manza = alum.Fields("Sem5")
  End If
  If (re = "Sem6") Then
    manza = alum.Fields("Sem6")
  End If
  If (re = "Sem7") Then
    manza = alum.Fields("Sem7")
  End If
  If (re = "Sem8") Then
    manza = alum.Fields("Sem8")
  End If
  If (re = "Sem9") Then
    manza = alum.Fields("Sem9")
  End If
  If (re = "Sem10") Then
    manza = alum.Fields("Sem10")
  End If
  If (re = "Sem11") Then
    manza = alum.Fields("Sem11")
  End If
  If (re = "Sem12") Then
    manza = alum.Fields("Sem12")
  End If
  If (re = "Sem13") Then
    manza = alum.Fields("Sem13")
  End If
  If (re = "Sem14") Then
    manza = alum.Fields("Sem14")
  End If
  If (re = "Sem15") Then
    manza = alum.Fields("Sem15")
  End If
  If (re = "Sem16") Then
    manza = alum.Fields("Sem16")
  End If
  If (re = "Sem17") Then
    manza = alum.Fields("Sem17")
  End If
  If (re = "Sem18") Then
    manza = alum.Fields("Sem18")
  End If
  If (re = "Sem19") Then
    manza = alum.Fields("Sem19")
```

Apéndice

```
End If
If (re = "Sem20") Then
    manza = alum.Fields("Sem20")
End If
If (re = "Sem21") Then
    manza = alum.Fields("Sem201")
End If
If (re = "Sem22") Then
    manza = alum.Fields("Sem22")
End If
If (re = "Sem23") Then
    manza = alum.Fields("Sem23")
End If
If (re = "Sem24") Then
    manza = alum.Fields("Sem24")
End If
If (re = "Sem25") Then
    manza = alum.Fields("Sem25")
End If
If (re = "Sem26") Then
    manza = alum.Fields("Sem26")
End If
If (re = "Sem27") Then
    manza = alum.Fields("Sem27")
End If
If (re = "Sem28") Then
    manza = alum.Fields("Sem28")
End If
If (re = "Sem29") Then
    manza = alum.Fields("Sem29")
End If
If (re = "Sem30") Then
    manza = alum.Fields("Sem30")
End If
If (re = "Sem31") Then
    manza = alum.Fields("Sem31")
End If
If (re = "Sem32") Then
    manza = alum.Fields("Sem32")
End If
If (re = "Sem33") Then
    manza = alum.Fields("Sem332")
End If
If (re = "Sem34") Then
    manza = alum.Fields("Sem34")
End If
If (re = "Sem35") Then
    manza = alum.Fields("Sem35")
End If
If (re = "Sem36") Then
    manza = alum.Fields("Sem36")
End If
If (re = "Sem37") Then
    manza = alum.Fields("Sem37")
End If
If (re = "Sem38") Then
    manza = alum.Fields("Sem38")
End If
If (re = "Sem39") Then
    manza = alum.Fields("Sem39")
End If
If (re = "Sem40") Then
    manza = alum.Fields("Sem40")
End If
If (re = "Sem41") Then
    manza = alum.Fields("Sem41")
End If
If (re = "Sem42") Then
    manza = alum.Fields("Sem42")
End If
```

Apéndice

```
If (re = "Sem43") Then
  manza = alum.Fields("Sem43")
End If
If (re = "Sem44") Then
  manza = alum.Fields("Sem44")
End If
If (re = "Sem45") Then
  manza = alum.Fields("Sem45")
End If
If (re = "Sem46") Then
  manza = alum.Fields("Sem46")
End If
If (re = "Sem47") Then
  manza = alum.Fields("Sem47")
End If
If (re = "Sem48") Then
  manza = alum.Fields("Sem48")
End If
If (re = "Sem49") Then
  manza = alum.Fields("Sem49")
End If
If (re = "Sem50") Then
  manza = alum.Fields("Sem50")
End If
If (re = "Sem51") Then
  manza = alum.Fields("Sem51")
End If
If (re = "Sem52") Then
  manza = alum.Fields("Sem52")
End If
'calculo de adeudo

tur = yo2 - manza
If (tur = 0) Then
  tur = yo2
End If

'termina adeudo buscado
If (violet = 1) Then
  GoTo astal
End If

If (alum.Fields(re) >= 0) Then
  If (alum.Fields(re) < yo2) Then
    r = alum.Fields("Clave_Alumno")
    cal = cal + jo

Busqueda = "Clave_Alumno like '*' & r & '*'"
'm.FindFirst Busqueda
If m.NoMatch Then
Else
  al.Rows = al.Rows + 1
  With al
    .Col = 0
    .Row = .Rows - 1
    .Text = m.Fields("Clave_Alumno")
    .Col = 1
    .Text = yo
    .Col = 2
    .Text = tur
  End With
End If
End If
End If

If alum.Fields("Clave_Alumno") <> re Then
'reco = reco - 1
GoTo anaa
End If
```

Apéndice

```
If jo <> 0 Then
    reco = reco + 1
End If
```

```
anaa:
If Not alum.OpenRecordset.BOF Then
alum.MoveNext
m.MoveNext
End If
    nombre.MoveNext
End If
contad = contad + 1
Loop
astal:
End Sub
```

,

```
Private Sub llamadax()
    Dim jor As Integer
    jor = alum.Fields("Clave_Alumno")
    nombre.MoveFirst
    Do While (nombre.Fields("Clave_Alumno") <> (jor) And Not nombre.EOF)
        If (nombre.Fields("Clave_Alumno") = alum.Fields("Clave_Alumno")) Then
            GoTo jorgebr2
        End If
        nombre.MoveNext
    If nombre.EOF Then
        nombre.MoveFirst
        violet = 1
    Exit Sub
    End If
Loop
jorgebr2:
If (nombre.Fields("Clave_Alumno") = (jor)) Then
    yo = nombre.Fields("Nombre_Alumno")
    yo2 = nombre.Fields("Colegiatura")
End If

End Sub
```

```
Private Sub Tabla()
'al.ColWidth(0) = 0
al.ColWidth(0) = 1200
al.Col = 0
al.Row = 0
al.Text = "Matricula"
al.ColWidth(1) = 5900
al.Col = 1
al.Row = 0
al.Text = "Nombre del Alumno"
al.ColWidth(2) = 1670
al.Col = 2
al.Row = 0
al.Text = "Adeudo"
End Sub
```

Por último tenemos el formulario del reporte de ingresos totales de la institución que normalmente se creará cada semana:

Figura 18.- Formulario de reporte de ingresos.

Este reporte visualizará la lista de percepciones de la institución llevando un control más preciso del existente económico, a continuación mostramos el código:

```
Dim base As Database
Dim alum As Recordset
Dim califi As Recordset
Dim m As Recordset
Dim servi As Recordset
```

```
Dim r, algunos As String
Dim jo As Double
Dim i, cont, re As Integer
Dim cal, general As Double
Dim medio As Double
Dim o, yola, yola2, yola3, seg, asique, cancelo As Integer
```

```
Private Sub cmdImprimir_Click()
If (lbl2.Caption = "") Then
MsgBox "NO HAY DATOS QUE IMPRIMIR", vbInformation, "ERROR DE IMPRESION"
Else
```

```
cmdSalir.Visible = False
cmdImprimir.Visible = False
Command1.Visible = False
nuevo.Visible = False
PrintForm
End If
End Sub
```

```
Private Sub cmdSalir_Click()
Unload Me
End Sub
```


Apéndice

```
Private Sub Form_Load()

Set base = OpenDatabase("C:\CONTROL DE PAGOS (DATCOM)\base\COBROS.mdb")
Set alum = base.OpenRecordset("PAGOS")

Set califi = base.OpenRecordset("PAGOS")
Set m = base.OpenRecordset("PAGOS")
Set servi = base.OpenRecordset("SERVICIOS")

al.ColWidth(0) = 1100
al.ColWidth(1) = 5650
al.ColWidth(2) = 1300

seg = 0
general = 0
cancelo = 0

End Sub

Private Sub Nuevo_Click()
cal = 0
general = 0
o = 0
cancelo = 0
Label6.Caption = Date

Set base = OpenDatabase("C:\CONTROL DE PAGOS (DATCOM)\base\COBROS.mdb")

Set alum = base.OpenRecordset("PAGOS")
Set califi = base.OpenRecordset("PAGOS")
Set m = base.OpenRecordset("PAGOS")

al.ColWidth(0) = 1100
al.ColWidth(1) = 5650
al.ColWidth(2) = 1300

a:
re = InputBox("¿Introduce el numero de Semana a consultar????...")

For i = 0 To (al.Rows + 1)
    With al
        .Col = 0
        .Row = .Rows - 1
        .Text = ""
        .Col = 1
        .Text = ""
        .Col = 2
        .Text = ""
    End With
Next i
al.Rows = 1

If Not alum.OpenRecordset.BOF Then

alum.MoveFirst
Do While (alum.Fields("Semana") <> re And Not alum.EOF)
    alum.MoveNext
If alum.EOF Then
    MsgBox "En la semana solicitada no se a realizado ningun pago ", vbCritical, "ERROR DE REGISTRO"
    alum.MoveFirst
Exit Sub
```

Apéndice

```
End If
Loop

Else
  MsgBox "La base de datos esta vacia", vbInformation, "ERROR DE DATOS"
End If 'de v

Label13.Caption = re

jo = 0

' CICLO
o = 1
Do While (Not seg = 1)

If m.EOF Then

Else
  m.MoveNext
  If m.EOF Then
    o = o + 1

    al.Rows = al.Rows + 1
    With al
      .Col = 0
      .Row = .Rows - 1
      .Text = yola3
      .Col = 1
      .Text = algunos
      .Col = 2
      .Text = cal
    End With

    m.MoveFirst
    jo = 0
    cal = 0
    'cancelo = 0
    GoTo yonofui
  End If

  If (m.Fields("Semana") = re) Then
  If (m.Fields("Clave_Servicio") = o) Then
  If (m.Fields("Status") = 0) Then
    cancelo = cancelo + m.Fields("Total_Pago")
    Text3.Text = cancelo
    GoTo trizte
  End If

  yola = m.Fields("Clave_Servicio")
  'yola2 = m.Fields("Clave_Cliente")

  jo = m.Fields("Total_Pago")
  cal = cal + jo
  general = general + jo
  medio = Val(cal)
  Text1.Text = general
  Text4.Text = Text1.Text

  'Text2.Text = general

End If
End If
End If
```

trizte:

Apéndice

```
servi.MoveFirst
Do While (servi.Fields("Clave_Servicio") <> 0 And Not servi.EOF)
  If (servi.Fields("Clave_Servicio") = 0) Then
    GoTo yonofui
  End If
  servi.MoveNext
  If servi.EOF Then
    seg = 1
    Exit Sub
  End If
Loop
yonofui:
If (servi.Fields("Clave_Servicio") = 0) Then
  algunos = servi.Fields("Descripcion")
  yola3 = servi.Fields("Clave_Servicio")
End If

Loop

Text3.Text = cancelo
End Sub
```

BIBLIOGRAFÍA

Publicaciones Impresas.

Análisis y Diseño de Sistemas

Autor: Henry F. Korth & Abraham Silberschatz

Segunda Edición.

Editorial Mc Graw Hill

Ingeniería del Software

Autor: Roger S. Pressman

Cuarta Edición.

Editorial Mc Graw Hill

Enciclopedia de Términos de Computación

Autor: Linda Gail/ John Christie

Editorial PHH, Pentice Hall

El enfoque de sistemas

Víctor Gerez y Manuel Grajalva

Editorial Limusa, 1980, México.

Introducción a la teoría general de sistemas

Oscar Jahansen Bertoglio

Editorial Limusa, 1998, México.

Teoría general de los sistemas

Ludwig Von Bertalanffy

Editorial Fondo de cultura económica, 1984, México.

La dirección por sistemas

Fernando del Pozo Navarro

Editorial Limusa, 1976, México.

Introducción a la teoría general de administración

Idalberto Chiavenato

Editorial Mc Graw Hill, 1999, México

Nuevas estructuras para la administración pública. Gem-UAEM

Santiago G. Velasco Monroy

1987, México

Estudio de sistemas y procedimientos administrativos
Joaquín Rodríguez Valencia
Editorial Ecasa, 1990, México

Estudio de sistemas y procedimientos.
José Morales Ramírez
UAEM., 1989, México

Sistemas y procedimientos administrativos
José Luis Kramis Joubanc
Editorial Pac, 1998, México

Sistemas de información para la dirección
James, C. Emery
Editorial Díaz de Santos, s.a., 1990

Management information systems
Kenneth, C. Laudon y Jane P. Laudon
Editorial Prentice-Hall, 1998.

Stallings, William SISTEMAS OPERATIVOS. Prentice Hall, 2da edición
Piattini Mario, Adoración de Miguel, Marcos Esperanza. DISEÑO DE BASES DE DATOS RELACIONALES.
Editorial Alfaomega

Análisis y diseño de sistemas de información
James A. Senn
Segunda Edición
Editorial Mc Graw – Hill

Enciclopedia de Microsoft® Visual Basic™ 6
Francisco Javier Ceballos Sierra
Editorial Alfaomega – Ra – Ma, 2000

Web.

<http://www.ur.mx/ur/faciya/carreras/cursos/sis/mod-dat1/graph.HTM>

<http://www.yudy.8m.com/Sistemasmanejador.htm>

<http://berzal.freesevers.com/freeware/dbms/spanish.html>

http://www.lafacu.com/apuntes/informatica/base_datos/default.htm#Introducción

<http://www.dbinternet.com.ar/metodo.htm>

<http://www.uas.mx/cursoswebct/Progsist/material.htm>

<http://www.programacionfacil.com/basic/cuatro4.htm>

<http://www.yudy.8m.com/Sistemasmanejador.htm>

<http://elizabethpeguero.8m.com/Eliza.htm>

<http://arraquis.dif.um.es/~rafa/bd1.htm>

<http://www.elriconcito.com/diccionario.php>