

Universidad Nacional Autónoma de Nicaragua - León
Departamento de Computación
Facultad de Ciencias



Monografía para optar al título de
Ingeniero en Sistemas de Información:

Sistema automatizado de cuentas por cobrar

Integrantes:

Anastacio Benito González Fúnes

Juan Carlos López Munguía

Milton Antonio Tórrez Soto

Tutor: MSc Aldo René Martínez

Octubre del 2005



Agradecimiento

A Dios nuestro creador por estar en los momentos difíciles y alegres de mi vida, ayudándome, guiándome y fortaleciéndome para salir adelante.

A mis padres Antonio González Loaisiga y Rosa Maria Funes por su gran apoyo incondicional que me brindaron durante todo este tiempo.

A mi esposa Jasmina Galán Somarriba por su magnifica devoción a la familia.

A mis hermanos Maria Jesús y Roger González Funes que han estado conmigo dándome consejo para salir adelante y superarme.

A mis tías Carmen, Mercedes González y su esposo Francisco Jiménez por soportarme todo este tiempo que he vivido en su casa.

A Marcio Briones por ser un amigo y asesor quien me brindó sus conocimientos y experiencia para poder culminar mi monografía.

Anastacio Benito González Funes.



Dedicatoria

Dedico esta Tesis :

A Dios nuestro Padre por que gracias a el he podido llegar a este día y sin su ayuda jamás hubiese podido ser el profesional que hoy soy.

A mi hijo Jeter Benito González Galán por ser el principal motivo que me inspiro confianza y fuerza de voluntad para salir adelante

A mis Padres Antonio González y Rosa Maria Funes por motivarme todo este tiempo, aconsejándome y apoyándome incondicionalmente para que hoy pueda ser un hombre de bien

A todas las personas que me apoyaron e hicieron posible de una u otra manera para que hoy pueda culminar este trabajo y una vez mas pueda realizar uno de mis sueños

Anastacio Benito González Funes.



Agradecimiento.

A Jehová Dios nuestro señor, por haberme dado el Gozo de la vida, darme a mis padres y hermanos, por darme siempre salud, sabiduría e inteligencia, y por estar siempre conmigo.

Agradezco a mis padres por haberme dado amor, comprensión, sabiduría, educación, tolerancia y sobre todo por haberme ayudado a culminar mis estudios, a mis hermanos por estar siempre conmigo animándome a salir adelante.

Agradezco a todas las personas que de una u otra manera colaboraron a terminar mi trabajo monográfico.

A todos los profesores que a lo largo de la carrera nos transmitieron su sabiduría, conocimiento y nos enseñaron la verdadera importancia de los libros, a todos ellos mis saludos, respetos y admiración.

A todos mis compañeros, que siempre estuvieron conmigo en el transcurso de la carrera en las buenas y en las malas demostrándome siempre su verdadera amistad y lealtad.

Juan Carlos López Munguia



Dedicatoria.

A Jehová Dios nuestro Señor, dador de vida, sabiduría, fé y esperanza. Quién como buen padre me ha guiado con sus consejos internos al camino del éxito, permitiéndome culminar mis estudios universitarios.

A mis padres, Carlos Marcial López López y Alicia María Munguía Munguía a los padres de mis padres a los que amo, quienes con muchos esfuerzos, amor y sacrificios han podido ayudarme a realizar mis estudios.

A mis hermanos José Denis y Meyling María López Munguía, los que siempre me han dado consejos sabios, amor, aliento y respeto.

A Yarenia del Carmen Toruño Blanco, a quien aprecio mucho, y siempre ha estado conmigo, apoyándome a salir adelante en mis estudios.

A mi gran amigo Marcio Briones Jirón, quién de una u otra forma ha estado presente en los momento de dudas, al realizar mi trabajo monográfico.

A todos ellos les agradezco mucho por haber hecho mis sueños realidad.

Juan Carlos López Muguia



Agradecimiento

A Dios y a la santísima virgen Maria que son los que me guiaron para poder finalizar mis estudios.

A mi familia que me apoyo y gracias a ellos he logrado coronar mi carrera

A mi profesor, tutor y amigo Aldo Martínez que me ha brindado conocimiento y ayuda en muchas etapas de mi carrera y culminación de mi monografía y que gracias a el he podido sobrepasar muchas situaciones difíciles.

Al Decano de la faculta de ciencias Octavio Guevara y a su esposa los cuales considero como mis segundos padres quienes con su amistad y apoyo he logrado finalizar mi monografía

A Marcio Briones asesor y amigo quien me ha brindado sus conocimientos y experiencia para poder culminar mi monografía

A todos los profesores que a lo largo de la carrera nos transmitieron su sabiduría, conocimiento y nos enseñaron la verdadera importancia de los libros, a todos ellos mis saludos, respetos y admiración.

A mis amigos y compañeros que de alguna u otra forma han estado con migo aconsejando y apoyando en las buenas y en las malas.

Milton Antonio Tórrez Soto



Dedicatoria

Dedico esta monografía primeramente a **Dios mi padre celestial** quien me dio la vida, fuerza y sabiduría para realizar con éxito mis estudios académicos.

A mi madre Lilliam Maria Soto Castillo la madre más humilde y bondadosa quien con todo su amor, sacrificio y atención me apoyo incondicionalmente para terminar mis estudios y lograr mis metas

A mi padre Sergio Torrez que con su apoyo y consejos me ha ayudado a formarme como hombre.

A mis hermanos Sergio José y Ana Lucia Tórrez Soto quienes son el motivo y los que me dan fuerza para seguir adelante.

A mis abuelos Bertha Baca y Gustavo Tórrez, por sus consejos, atención y apoyo quienes fueron un pilar muy importa para poder culminar mis estudios.

Milton Antonio Tórrez Soto



ÍNDICE DE CONTENIDOS

INTRODUCCION	1
ANTECEDENTES	2
JUSTIFICACIÓN	3
OBJETIVO	4
MARCO TEORICO	5
DISEÑO METODOLÓGICO	31
ESPECIFICACIÓN DE REQUISITOS SOFTWARE	34
DIAGRAMA DE FLUJO DE DATOS	96
DICCIONARIO DE DATOS	101
DISEÑO ENTIDAD RELACION	106
DISEÑO DE DATOS	107
DISEÑO ARQUITECTONICO	114
DISEÑO PROCEDIMENTAL	115
DISEÑO DE INTERFAZ	116
CONCLUSIONES	121
RECOMENDACIONES	122
ANEXOS	123
BIBLIGRAFIA	140



Introducción

Las Cuentas por Cobrar representan el crédito que concede el negocio a sus clientes, sin más garantía que la promesa de pago (oral o escrito) a corto plazo.

Las empresas invierten en cuentas por cobrar un alto porcentaje de sus activos; por lo que es de gran importancia para cualquier empresa mantener un control exacto de cada transacción de forma eficiente, así como también, es fundamental generar reportes, mostrando la situación real de las cuentas por cobrar.



Antecedentes

En el Departamento de Computación de la Facultad de Ciencias de nuestra Universidad, en lo que cuenta y por lo que hemos constatado, no se han realizado trabajos monográficos similares a la aplicación **en la que queremos** trabajar. Además en la empresa a la cual le pretendemos desarrollar esta aplicación, no posee un sistema de cuentas por cobrar que cumpla con las necesidades que hoy en día está demandando, ya que cuenta con un sistema hecho en el lenguaje de programación **Clipper**, el cual está obsoleto .

Desde años atrás hasta el presente, la forma en la que la empresa “**Comercial Eléctrica**” ha venido manipulando el comportamiento y proceso de las cuentas por cobrar no ha sido de forma eficiente. Dicha empresa cuenta con un sistema que no cumple con todos los requerimientos necesarios para una óptima administración de esta importante cuenta, que en la actualidad se maneja con mucha frecuencia en las empresas. La Empresa posee un software basado en MS-DOS que no posee una interfaz gráfica moderna, y esto provoca que la información de los diferentes clientes sea más difícil de gestionar para los empleados.

De acuerdo a lo anterior, hemos decidido elaborar un sistema que cumpla con todos los requisitos y que además sea atractivo para el usuario y seguro.



Justificación

Consideramos que la realización de nuestro sistema (aplicación) facilitará las tareas principales de esta empresa, como es la generación de reportes para los clientes y empleados de forma automática en base a la información que se le proporcione al sistema, mejorará la presentación de los documentos y emisión de informes que la empresa necesite.



Objetivos

General:

- Desarrollar un Sistema Automatizado basado en un entorno visual que permita manejar el ingreso de la información necesaria para una gestión óptima de las cuentas por cobrar.

Específicos:

- Generar reportes útiles para la toma de decisiones, con el fin de tener un correcto manejo de las cuentas por cobrar.
- Lograr una mejor disponibilidad de la información en base a las cuentas por cobrar a clientes.
- Mantener la información personalizada y actualizada de los clientes y de la empresa.
- Optimizar el sistema, minimizando el tiempo de trabajo y el costo en que incurre la empresa, al llevar el proceso de esta cuenta de forma manual.



Marco Teórico

En la actualidad los sistemas de información automatizados están tomando auge en el mundo empresarial, hoy buena parte de nuestra sociedad se apoya en la tecnología de sistema de información, ya sea directa o indirectamente para trabajar con mayor rapidez. Las computadoras y los sistemas de información ocupan un sitio especial en las empresas donde facilitan de forma eficiente las distintas operaciones.

En nuestra sociedad actual el análisis de sistemas significa computadoras y éstas, a su vez, sistemas de información. Todos estos sistemas necesitan, principalmente, software. La venta de software es tan común que éste es considerado como un artículo de consumo. Un **software** no es más que un conjunto de instrucciones que le indican al usuario la acción que debe ejecutar.

Con la diversidad de software existente y dado que éste es la esencia de un sistema de información, se puede encontrar casi cualquier aplicación que se necesite en forma de software preempaquetado o enlatado. Pero no cualquier paquete hará el trabajo y además tiene que reunir ciertos requerimientos, los cuales estamos tratando de cubrir.

Es probable que las empresas pequeñas trabajen con software disponibles en el mercado, pero las grandes organizaciones seguirán teniendo la necesidad del diseño de sistemas a la medida.



El desarrollo de sistema puede considerarse, en general, formado por dos grandes componentes:

- 1. El análisis de sistemas**
- 2. El diseño de sistemas.**

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

2. El diseño de sistema es el proceso de planificar, reemplazar o complementar un sistema organizacional existente.

Aunque la palabra “**sistema**” se ha utilizado en forma repetitiva, tiene un significado especial; en el sentido mas amplio, **un sistema** es un conjunto de componentes que se interrelacionan entre si para lograr un objetivo común. En este caso el objetivo común que pretendemos alcanzar es que la entidad encargada de llevar acabo los análisis (y por ende las decisiones) del comportamiento contable de las empresas, se mantenga al día con respecto al comportamiento contable de dicha empresa, y de esta manera cumpla correctamente con su función primordial que es llevar a la institución bajo su cuidado hacia mejores horizontes financieros.

Trataremos que la explicación del trabajo realizado en la elaboración de esta tesis de fin de carrera quede lo suficientemente claro para que la mayoría de las personas o usuarios finales queden bastante entendidas sobre el uso que se le puede dar al sistema de cuentas por cobrar.

De aquí en adelante explicaremos los aspectos concernientes a las ramas de estudios que de una forma u otra tuvieron que ver en el desarrollo de dicho trabajo monográfico, como son: la contabilidad, los sistemas de base de datos



relacionales, accesos a datos locales y remotos, diseño de interfaces de usuarios, etc.

BASE DE DATOS

Una base de datos es una recopilación de información relativa a un asunto o propósito particular; en forma de un fichero o archivo organizado por algún tipo de criterio organizativo y que se encuentren contenido dentro de cualquier medio de almacenamiento. Dicho medio de almacenamiento es por lo general algún tipo de medio electrónico.

Dicho de otra forma una base de datos es una colección integrada de datos almacenados en distintos tipos de registros (tablas), de forma que son accesibles para múltiples aplicaciones. La interrelación de los registros se obtiene de las relaciones entre los datos, no de su lugar de almacenamiento.

Una base de datos es similar a un archivo de datos en cuanto en que ambos son un almacenamiento de datos. Como en un archivo de datos, una base de datos presenta información directamente al usuario; el usuario ejecuta una aplicación que tiene acceso a los datos de la base de datos y los presenta al usuario en un formato claro.

Los sistemas de base de datos son más útiles que los archivos de datos. Los datos están mucho mejor organizados. En una base de datos bien diseñada, no hay elementos de datos duplicados que el usuario o la aplicación tenga que actualizar al mismo tiempo. Los elementos de datos están agrupados en una única estructura o registro, y se pueden definir relaciones entre dichas estructuras y registros.

Cuando se trabaja con archivos de datos, las aplicaciones se tienen que escribir para la estructura especificada de cada archivo de datos. En contraste, las



bases de datos contienen un catálogo que las aplicaciones puedan utilizar para determinar la organización de los datos. Las aplicaciones de bases de datos genéricas pueden utilizar el catálogo para presentar dinámicamente al usuario datos de distintas bases de datos, sin tener que depender de formatos de datos específicos.

Una base de datos suele tener dos componentes: los archivos que almacena la base de datos física el software del sistema de administración de la base de datos (DBMS, Database Management System), que las aplicaciones utilizan para tener acceso a los datos. El DBMS es el responsable de mantener la estructura de la base de datos, lo que incluye:

- ✓ El mantenimiento de las relaciones entre los datos de la base de datos.
- ✓ La garantía de que los datos estén correctamente almacenados y de que no se infrinjan las reglas que definen las relaciones entre los datos.
- ✓ La recuperación de datos hasta un punto coherente en caso de fallo del sistema.

Base de datos Relacionales

Hay varias formas de organizar los datos en la base de datos, pero las bases de datos relacionales son una de las formas más efectivas. Los sistemas de bases de datos relacionales son una aplicación de la teoría matemática de los conjuntos al problema de la organización de los datos. En una base de datos relacional, los datos están organizados en tabla (llamadas relaciones en la teoría relacional).



Una tabla representa una clase de objetos que tienen cierta importancia en una organización. Por ejemplo, una corporación puede tener una base de datos para los empleados, otra tabla para los clientes y otra para los productos del almacén. Las tablas están compuestas de columnas y filas (atributos y tuplas en la teoría relacional). Cada columna representa algún atributo del objeto representado por la tabla. Por ejemplo en una tabla cliente normalmente tendría columnas para el nombre, el apellido, el código del cliente, la ciudad. Cada fila representa una instancia del objeto representado por la tabla. Por ejemplo, una fila de la tabla cliente representa al cliente cuyo código es CLTE01.

Al organizar los datos en la tabla, se puede encontrar varias formas de definirlos. La teoría de la base de datos relacionales define un proceso, la normalización que asegura que el conjunto de tablas definido organizará los datos de manera eficaz.

Tabla

Una tabla es una colección de datos sobre un tema o entidad específicos, como productos o proveedores. La utilización de una tabla diferente para cada tema significa que se almacenan los datos solo una vez, lo cual hace aumentar la eficacia de las bases de datos, y reduce errores de entrada de datos.

Consultas

Es una secuencia de comandos que permite recuperar datos almacenados en tablas de una base de datos, de manera rápida y sencilla.

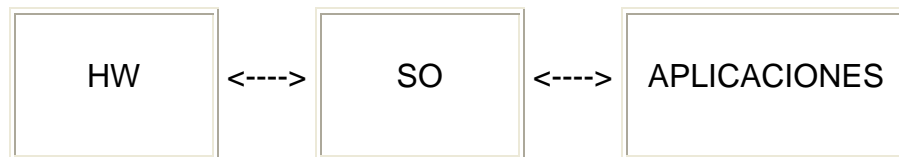


Informes o reportes

Un informe es un método eficaz de presentar los datos en formato impreso. Dado que tiene el control sobre el tamaño y el aspecto de todo el informe, puede mostrar la información en la manera que desea verla.

En un ordenador se puede distinguir tres partes: Hardware, Sistema Operativo y aplicaciones.

El hardware, es la parte física del ordenador: el microprocesador, la memoria, el disco duro y resto de dispositivos y periféricos (ratón, impresora, teclado, etc.). **El sistema operativo** es el software encargado de comunicarse con el hardware, permitiendo que los dispositivos del ordenador estén disponibles para ser utilizados. **Las aplicaciones** son el software que va a dar una funcionalidad específica y que es lo que le interesa realmente al usuario. Permite hacer cosas útiles con el ordenador que serían más difíciles o pesadas de hacer a mano. Por ejemplo un procesador de textos, un programa de dibujo o una hoja de cálculos. También hay aplicaciones de ocio como los juegos, de aprendizaje como los cursos de idiomas y a veces permiten hacer cosas que sería imposible hacer a mano, como por ejemplo, alinear automáticamente miles de secuencias.



Un programa recibe y presenta información al usuario (a través de un interfaz de usuario) lee y guarda información en el disco duro (como archivos) y realiza acciones pedidas por el usuario a través del interfaz sobre los datos cargados. El interfaz de usuario puede ser de texto, como en los antiguos programas de MS-DOS o gráfico (GUI) como en el caso de la mayoría de los



programas que actualmente se usan. Sin perjuicio de ello, el uso de la línea de comandos sigue siendo el método más potente de trabajar con un ordenador.

Modelo de Aplicación

Ejemplo: Programa que almacena información sobre los clones que posee un laboratorio: código del clon, secuencia, organismo, caja en la que está guardado, primero empleados para amplificar por PCR, etc. El usuario a través del interfaz administra los registros: añade clones, elimina alguno en caso necesario y modifica información de los existentes, acciones siempre a petición del usuario. La información se almacena en un fichero de base de datos de Access y el programa además puede presentar informes de que clones tenemos, ordenados siguiendo algún criterio determinado: por cajas, por organismo, etc.

En el ejemplo anteriormente expuesto, la misma aplicación se encarga de hacer todo, genera el interfaz para interactuar con el usuario, gestiona los datos usados almacenándolos en un fichero y cargándolos al iniciar, realiza sobre los datos todas las acciones pedidas por el usuario, etc. Este es un modelo monolítico de aplicación que como veremos tiene muchos inconvenientes. Por ejemplo si la aplicación la he creado en Visual Basic, que es un lenguaje de programación de Microsoft, esta solo va a funcionar en SO Windows y con ordenadores PC (microprocesadores Intel y semejantes). Además, el guardar los datos en un fichero me impide que el programa lo tengan instalado distintos usuarios en distintos PC's debido a que si alguien modifica algo con su copia habría que actualizarlo en los demás ficheros. En algunos casos, este modelo es suficiente y no presentaría ningún problema. En un entorno de varios usuarios potenciales de la aplicación que además comparten los datos y que pueden necesitar usarla desde ordenadores distintos, este modelo no es muy conveniente.



Modelo de tres capas.

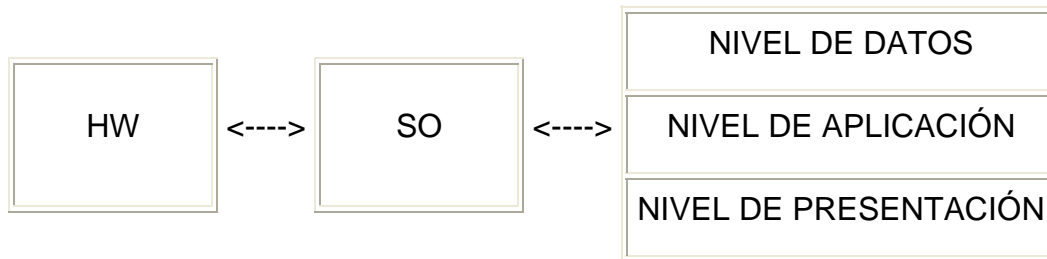
Lo ideal para empezar es que cada usuario no tuviese una copia local del fichero de la base de datos. Lo suyo sería que dicho fichero estuviese centralizado en algún lado y todos los posibles usuarios pudiesen acceder a él, ya sea desde el ordenador en el que está o desde otro. Esto se soluciona empleando un sistema gestor de bases de datos como MySQL, PostgreSQL y Oracle. Ese ordenador se convierte en un **servidor** de la base de datos y nuestra aplicación en un **cliente** de esta. La aplicación le pide los datos al servidor y este se los da. Entonces la aplicación ya puede hacer con los datos lo que el usuario quiera. Si hay alguna modificación de estos la aplicación le envía la información al servidor y este se encarga de actualizar, eliminar o añadir. Hemos separado de esta manera la gestión de los datos del resto de la aplicación.

Si nuestra aplicación tiene que hacer cosas más complicadas que añadir, modificar y eliminar registros de una base de datos puede ser que no nos valga cualquier ordenador de los que los usuarios van a usar. Por otro lado puede ser que los usuarios accedan desde ordenadores muy distintos con SO muy distintos. Otro paso que podemos realizar ahora es separar la parte operacional del programa del interfaz de usuario. La primera parte podría estar en un ordenador potente mientras que el interfaz puede estar en cualquier ordenador más limitado, idealmente, se debería poder emplear el interfaz desde cualquier ordenador con cualquier SO. Esto se puede hacer por ejemplo poniendo el programa con la parte operacional en un ordenador que además esté ejecutando un servidor Web. Entonces los usuarios pueden acceder al programa mediante un navegador Web y un sistema llamado CGI. Otra forma sería diseñar el propio programa como un servidor y que la parte del interfaz de usuario fuese un cliente de este servidor.



Nosotros preferimos el primer sistema en el que el interfaz de usuario es cliente de un servidor Web y este se comunica con el programa a través de los CGI.

Hemos pasado de esta manera de un modelo monolítico de aplicación a un **Modelo de Tres Capas** o niveles, el nivel de datos, el nivel de aplicación (u operacional) y el nivel de presentación (o interfaz de usuario). Este modelo permite aumentar el rendimiento y hacer la aplicación accesible para un numero casi ilimitado de plataformas de ordenadores, independientemente del microprocesador que usen o el SO que tengan instalado.



Arquitectura Cliente/Servidor.

Para que el modelo de tres capas sea útil tiene que basarse en una arquitectura cliente/servidor. Esto permite la comunicación entre los distintos niveles comportándose cada uno de ellos como cliente y/o servidor. Un **servidor** es un programa que sirve cosas, ya sean datos, utilidades, etc. Un cliente es el programa que pide los datos a ese servidor. El servidor solo esta a la espera de que algún cliente le pida algo y cuando le llega una petición, la realiza.

Ejemplo1: Navegar por Internet y ver páginas Web. **Ejemplo 2:** Recibir un e-mail en nuestro programa de correo.



Siguiendo con el ejemplo de la base de datos de clones, podemos almacenar en una base de datos la información de los clones. La base de datos estará funcionando en un ordenador que será el servidor de base de datos. En el mismo ordenador u en otro está corriendo un servidor Web, y en nuestro ordenador solo tendremos que abrir un navegador. En el navegador escribimos la dirección Web de la página de la aplicación y la pagina, mediante CGI ejecuta la aplicación en función de las opciones elegidas por nosotros en la página Web. Al ejecutarse la aplicación esta leerá datos de la base de datos, los modificará, los eliminara o añadirá otros nuevos si se lo pedimos.

Software de Base.

Con estos conceptos, se llega a la definición de un tipo de aplicaciones que en realidad se encuentran entre el SO y las aplicaciones de los usuarios. Es el llamado **software de base** o **middleware** y que proporcionan una funcionalidad específica pero que el usuario normalmente no interacciona demasiado con ellos. Serían por ejemplo los sistemas gestores de bases de datos o servidor de bases de datos (MySQL), el servidor Web (Apache), el servidor de correo (Sendmail), el servidor de archivos (Samba), etc. Como vemos, algunos de ellos están implicados en el nivel de datos de nuestra aplicación según el modelo de tres capas mientras que otros están implicados en el nivel de presentación. Nuestra aplicación los emplea pero el usuario no tiene porque saber nada de ellos.

Hoy en día existen muchas empresas y sitios Web que necesitan mantener de forma eficiente un gran volumen de datos. Muchos de ellos optan por soluciones comerciales, aunque muchas otras confían en el software libre optando por una solución como MySQL.



¿Qué es MySQL?

MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.MySQL

Es un sistema de administración para bases de datos relacionales (RDBMS) que provee una solución robusta a los usuarios con poderosas herramientas multi-usuario, soluciones de base de datos SQL (Structured Query Language) multi-threaded. Es rápido, robusto y fácil de utilizar. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.



¿Qué es lo que le falta?

MySQL surgió cómo una necesidad de un grupo de personas sobre un gestor de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. Es por ello que, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características:

1. Subconsultas: tal vez ésta sea una de las características que más se echan en falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
2. SELECT INTO TABLE: Esta característica propia de Oracle, todavía no está implementada.
3. Triggers y Procedures: Se tiene pensado incluir el uso de procedures almacenados en la base de datos, pero no el de triggers, ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
4. Transacciones: a partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).
5. Integridad referencial: aunque sí que admite la declaración de claves ajenas en la creación tablas, internamente no las trata de forma diferente al resto de campos.

Los desarrolladores comentan en la documentación que todas estas carencias no les resultaba un problema, ya que era lo que ellos necesitaban. De hecho, MySQL fue diseñada con estas características, debido a que lo que buscaban era un gestor de bases de datos con una gran rapidez de respuesta.



Pero ha sido con la distribución de MySQL por Internet, cuando más y más gente le está pidiendo estas funcionalidades, por lo que serán incluidas en futuras versiones del gestor.

Copias de seguridad en MySQL

Conceptos a tener en cuenta a la hora de hacer una copia de seguridad y métodos disponibles para ello.

El gestor de Bases de datos MySQL incluye varias herramientas para la realización de copias de seguridad de la base de datos. Mediante ellas podremos poner a salvo nuestros datos, para que, en el eventual caso de que se pierdan, poderlos recuperar.

A la hora de hacer una copia de seguridad, lo primero que se hay que tener en cuenta es la integridad de los datos que se estén guardando. En todos los casos es necesario que haya integridad en los datos de una tabla, con esto quiero decir que todos los datos de la tabla deberán estar escritos en la misma, esto puede sonar un poco raro, pero tal y como pasa con todos los gestores de bases

de datos, Mysql dispone de diversas "caches" en las que se almacenan datos temporalmente con el objetivo de mejorar en rendimiento, de forma que por ejemplo, una vez hecha una modificación en una tabla, puede ser que los datos no se guarden inmediatamente en disco, hasta que termine, por ejemplo, una consulta que se estaba ejecutando. Por esto, es necesario "forzar" a Mysql a escribir todos los datos en el disco, mediante la sentencia "Flush Tables".

Además es necesario que no se escriba en las tablas mientras se esta haciendo la copia de seguridad de la base de datos, que se consigue con el comando "lock tables", seguido del nombre de la tabla. Puede haber bases de



datos en las que sea necesario bloquear todas las tablas al mismo tiempo antes de hacer la copia de seguridad.

Existen varias opciones para realizar la copia de seguridad de una base de datos de MySQL:

- En primer lugar, se podría utilizar alguna herramienta comercial que gestione todo el proceso, esto tiene como ventaja la simplicidad del método, y como inconveniente, que no suelen ser gratis, sino que hay que pagar licencia por la utilización de los mismos. Estas herramientas no se cubrirán en este reportaje.
- En segundo lugar, y a partir de la versión 3.23.25 y posteriores, existe la posibilidad de realizar una copia de seguridad a través de la sentencia sql "**backup table**".
- Como tercera opción, es posible realizar copias de seguridad a través de las herramientas que nos proporciona el propio gestor de base de datos, como pueden ser **mysqldump** ó **mysqlhotcopy**.

Fundamentos de la Arquitecta MYSQL

La era de la arquitectura cliente servidor

“En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores” estos últimos responden a la demanda del cliente que la produjo.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet.



Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.

¿Qué es una arquitectura?

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización.

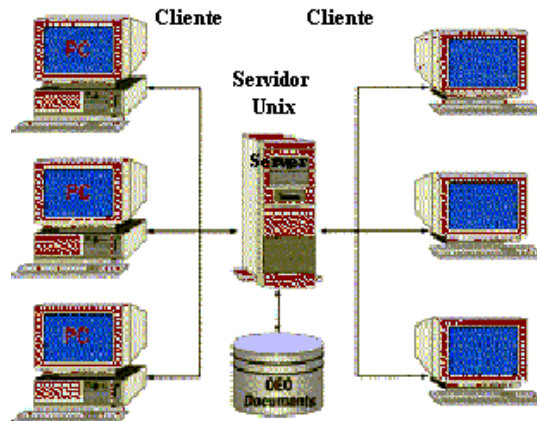
Debemos señalar que para seleccionar el modelo de una arquitectura, hay que partir del contexto tecnológico y organizativo del momento y, que la arquitectura Cliente/Servidor requiere una determinada especialización de cada uno de los diferentes componentes que la integran.

¿Qué es un cliente?

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

¿Qué es un servidor?

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc.



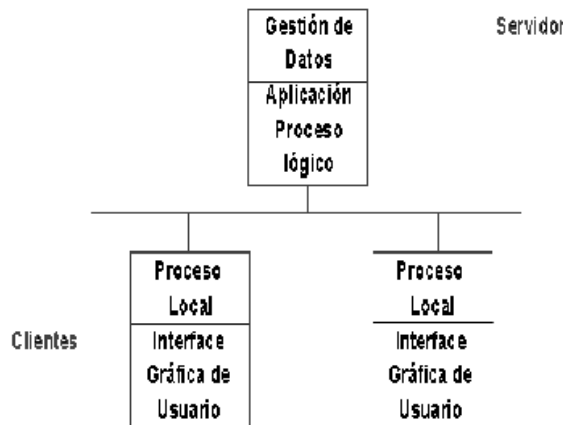
Este es el ejemplo gráfico de la arquitectura cliente servidor.

Elementos de la arquitectura cliente/servidor

En esta aproximación, y con el objetivo de definir y delimitar el modelo de referencia de una arquitectura Cliente/Servidor, debemos identificar los componentes que permitan articular dicha arquitectura, considerando que toda aplicación de un sistema de información está caracterizada por tres componentes básicos:

- ❖ Presentación/Captación de Información
- ❖ Procesos
- ❖ Almacenamiento de la Información

Los cuales se suelen distribuir tal como se presenta en la figura:

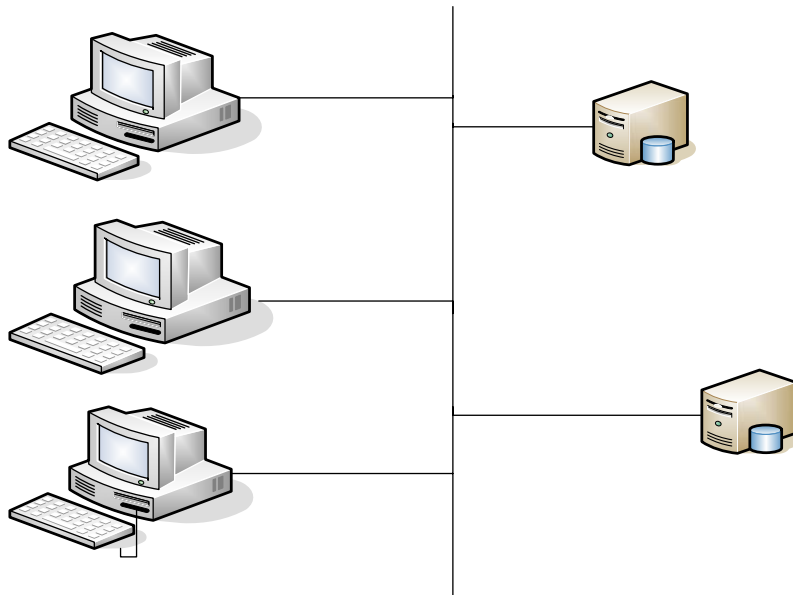




Y se integran en una arquitectura Cliente/Servidor en base a los elementos que caracterizan dicha arquitectura, es decir:

- Puestos de trabajo
- Comunicaciones
- Servidores

Tal como se presenta en la figura:



ARQUITECTURA

Puestos de trabajo

CLIENTE A



Estructura Cliente – Servidor

Características del modelo cliente/servidor

En el modelo CLIENTE/SERVIDOR podemos encontrar las siguientes características:

1. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
2. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
3. Un servidor da servicio a múltiples clientes en forma concurrente.
4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
5. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
6. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.



También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red. Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.

7. Además se constituye como el nexo de unión mas adecuado para reconciliar los sistemas de información basados en mainframes o mini computadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.

8. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

1. Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central y otros sistemas de la organización ponen a su servicio.

Funciones de un programa servidor

1. Espera las solicitudes de los clientes.
2. Ejecuta muchas solicitudes al mismo tiempo.
3. Atiende primero a los clientes VIP.
4. Emprende y opera actividades de tareas en segundo plano.
5. Se mantiene activa en forma permanente.



En los sistemas cliente – servidor, el servidor es un equipo relativamente grande situado en una ubicación central que administra recursos utilizados por varios individuos. Cuando los individuos tienen que utilizar un recurso, se conectan con el servidor desde sus equipos, o cliente, a través de la red.

El tener los datos almacenados y administrados en una ubicación central ofrece varias ventajas:

- Todos los elementos de datos están almacenados en una ubicación central en donde todos los usuarios pueden trabajar con ellos.
- No se almacenan copias separadas del elemento en cada cliente, lo que elimina los problemas de hacer que todos los usuarios trabajen con la misma información.
- Las reglas de la organización y las reglas de seguridad se pueden definir una sola vez en el servidor para todos los usuarios. Esto se puede hacer en una base de datos mediante el uso de restricciones, procedimientos almacenados y desencadenadores. También se pueden hacer en una aplicación de servidor.
- Los servidores de base de datos relacionales optimizan el tráfico de la red al devolver solo los datos que la aplicación necesita.
- Los gastos en Hardware se pueden minimizar.

Como los datos no están almacenados en los clientes, estos no tienen que dedicar espacio de disco a almacenarlo. Los clientes tampoco necesitan la capacidad de proceso para administrar los datos localmente y el servidor no tiene que dedicar capacidad de proceso para presentar los datos.

El servidor no se puede configurar para optimizar la capacidad de entrada salida de disco necesaria para obtener los datos y los clientes se pueden



configurar para optimizar el formato y presentación de los datos obtenidos desde el servidor.

El servidor puede estar situado en una ubicación relativamente segura y estar equipado con dispositivo como sistema de alimentación interrumpida (SAI), lo que resulta más económico que si se protegieran todos los clientes.

- Las tareas de mantenimiento como las copias de seguridad y restauración de los datos son más sencillas porque están concentradas en el servidor central.

ODBC

Conocido también como (**Open Database Connectivity**) o conectividad de base de datos abierta es una estándar internacional y su principal misión es proporcionar acceso a bases de datos relacionales de cualquier tipo utilizando como vehículo el lenguaje SQL. ODBC constituye una parte importante de la estrategia WOSA (Windows Open System Architecture) Arquitectura de sistemas abiertos para Windows, el cual es el principal medio de comunicación entre diferentes plataformas de la familia Microsoft ODBC permite la conectividad entre bases de datos diferentes con distintos gestores desde cualquier estación de

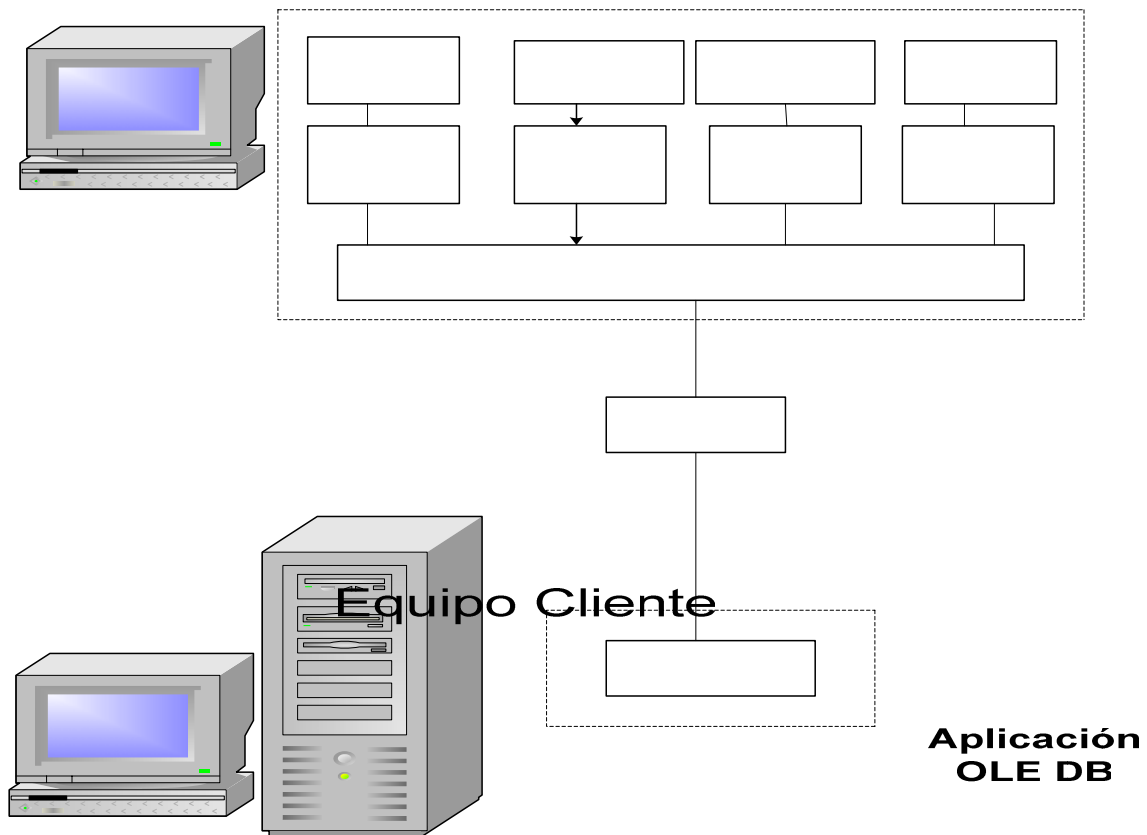
trabajo con un sistema operativo Windows, esto, a través del lenguaje de consulta SQL.

Tradicionalmente el sistema empleado para acceder modularmente a varios dispositivos está estipulado por un drive. Microsoft plantea un tipo de driver que se comunica con una base de datos en vez de hacerlo con un dispositivo periférico hardware. Incluso yendo más allá, puede ser el propio driver el que actúe como servidor de datos en el caso de que no exista más que un fichero físico de datos al



que deseamos acceder. Para poder realizar el enlace se define un conjunto de funciones comunes llamadas API/ODBC que cada sistema nativo debe implementar con sus instrucciones propias para compatibilizarse con el estándar.

Las capas de software que interactúan en el estándar ODBC son las siguientes:



Aplicación: realiza las llamadas a las API/ODBC para enviar las instrucciones SQL necesarias y obtener los resultados correspondientes.

Administrador de Driver ODBC: realiza dos tareas que corresponden al modo de funcionamiento oculto y al modo de funcionamiento de interfaz.



En el modo de funcionamiento oculto realiza la carga de los drivers necesarios por petición de cada aplicación, mientras que en el modo de trabajo como interfaz se encarga del mantenimiento de los drivers cargados y de las fuentes de datos.

Para el mantenimiento de dichos drivers y datos acudiremos al panel de control de Windows accediendo al icono de Driver Manager para ODBC.

Driver: recibe llamadas de la aplicación hacia las funciones de la API/ODBC y traduce dichas llamadas al lenguaje nativo del servidor en concreto.

Fuente de datos (DSN): es la asociación de un nombre de fuente con una información asociada en el servidor de la base de datos destino, esta información contiene puntos clave como el sistema operativo en el que se encuentra el servidor, el driver utilizado, el tipo de sistema gestor y la plataforma de red a la que necesitamos acceder para conectarnos a un servidor en concreto.

El administrador de driver utilizado en nuestro sistema es:

ODBC32.DLL: trabaja para aplicaciones de 32 bits.

Servidor: es el sistema gestor de bases de datos al que nos queremos conectar.



Lenguaje de programación utilizado

Visual Basic 6.0 Edición Empresarial

¿**Qué es Visual Basic**? La palabra “Visual” hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente puede agregar objetos prefabricados en su lugar dentro de la pantalla. Si ha utilizado alguna vez un programa de dibujo como Saint, ya tiene la mayor parte de las habilidades necesarias para crear una interfaz de usuario efectiva.

La palabra “Basic” hace referencia al lenguaje BASIC (Beginners AllPurpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows. Los principiantes pueden crear aplicaciones útiles con solo aprender unas pocas palabras clave, pero, al mismo tiempo, la eficacia del lenguaje permite a los profesionales a cometer cualquier objetivo que pueda alcanzarse mediante cualquier otro lenguaje de programación de Windows.

El lenguaje de programación Visual Basic no es exclusivo de Visual Basic. La Edición para aplicaciones del sistema de programación de Visual Basic, incluida en Microsoft Excel, Microsoft Access y muchas otras aplicaciones Windows, utilizan el mismo lenguaje. El sistema de programación de Visual Basic, Scripting Edition (VBScript) es un lenguaje de secuencias de comandos ampliamente difundido y un subconjunto del lenguaje Visual Basic.



Si su objetivo es crear un pequeño programa para su uso personal o para su grupo de trabajo, un sistema para una empresa o incluso aplicaciones distribuidas de alcance mundial a través de Internet, Visual Basic dispone de las herramientas que necesita:

- Las características de acceso a datos de permiten crear bases de datos, aplicaciones cliente, y componentes de servidor escalables para los formatos de las bases de datos mas conocidas, incluidos Microsoft SQL Server y otras bases de datos de ámbito empresarial.
- Las tecnologías ActiveX le permite usar la funcionalidad proporcionada por otras aplicaciones, como el procesador de textos Microsoft Word, la hoja de cálculo Microsoft Excel y otras aplicaciones Windows. Puede incluso automatizar las aplicaciones y los objetos creados con la Edición Profesional o la Edición Empresarial de Visual Basic.
- Las capacidades de Internet facilitan el acceso a documentos y aplicaciones a través de Internet o Intranet desde su propia aplicación, o la creación de aplicaciones de servidor para Internet.
- La aplicación terminada es un autentico archivo.exe que utiliza una maquina virtual de Visual Basic que puede distribuir con toda libertad.

Visual Basic proporciona varias formas de acceso a base de datos remotas como: ADO, RDO Y DAO. En nuestra aplicación optamos por utilizar el ADO, basados en las siguientes ventajas:



Ventajas de usar ADO (ADO comparado con RDO Y DAO):

Aunque ADO encapsula la funcionalidad proporcionada por DAO Y RDO, en las aplicaciones de acceso a datos existentes que utilizan DAO Y RDO, es necesario rescribir algunos de los elementos del lenguaje para que sean compatibles con la tecnología ADO.

ADO es el sucesor de DAO/FDO. Fundamentalmente ADO es muy similar a RDO. ADO proporciona un modelo mas simple que el modelo utilizado por DAO Y RDO, lo que se traduce en menos objetos y mas propiedades, métodos y eventos. Por ejemplo, ADO no tiene ningún objeto equivalente a rdoEngine ni a rdoEnvironment. Tampoco desde ADO es posible actualmente crear bases de datos ODBC, a pesar de tener acceso a ellas a través de los proveedores de servicios OLE DB a bases de datos ODBC.

ADO es la única interfaz que necesitamos conocer para cualquier solución de acceso a datos basada en Web o en tecnología cliente-servidor.



Diseño Metodológico.

Para llevar a cabo el desarrollo de nuestro trabajo utilizaremos el Ciclo de Vida Clásico del Software o Modelo de Cascada, ya que este modelo nos permite realizar cada una de las etapas del Software de forma independiente.

Las etapas que seguiremos para el desarrollo de nuestro sistema de Cuentas Por Cobrar son las siguientes:

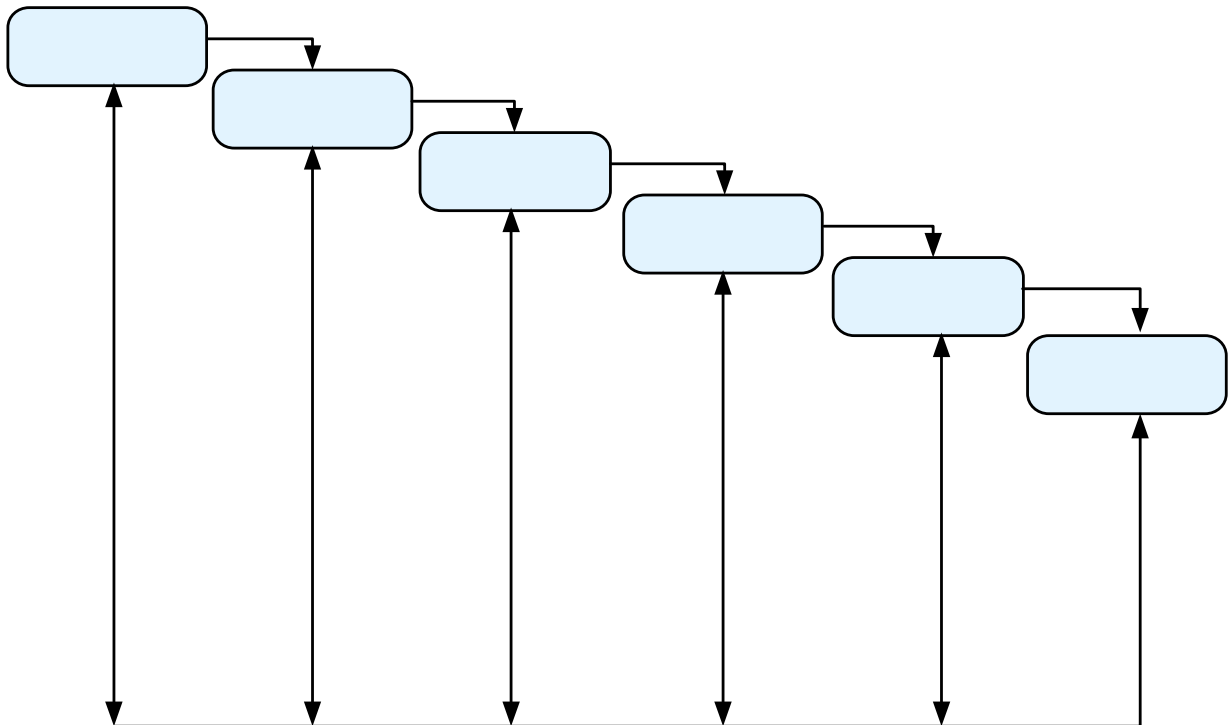
- ✓ **Ingeniería del sistema:** El trabajo de desarrollo de software inicia con la ingeniería del mantenimiento del sistema y consiste en hacer un estudio del ámbito del problema, estableciendo los requisitos generales a un nivel superiora además se aplican las técnicas de elaboración y evaluación del proyecto.
- ✓ **Análisis del sistema:** Los desarrolladores ponen su atención en la información que manejará el software, las funciones que ejecutara el rendimiento y las interfaces hombre maquina requerida. Como resultado del análisis del sistema se obtiene un documento llamado ERS (Especificación de Requisitos del Software).
- ✓ **Diseño:** Es un proceso de varios pasos que se centra en cuatro aspectos del Software:
 1. Estructura de datos.
 2. Arquitectura del Software.
 3. Detalle Procedimental.
 4. Caracterización de la interfaces.



- ✓ **Codificación:** Consiste en la traducción de un diseño en un formato legible para la maquina.
- ✓ **Pruebas:** Consiste en asegurar que las entradas del sistema produzcan los resultados deseados.
- ✓ **Mantenimiento:** Actividad de mantener el sistema de forma fiable.

Modelo en Cascada

Descompone el proceso de desarrollo en diferentes procesos, este genera una salida que será entrada del siguiente proceso. En este modelo comienza con el nivel de ingeniería de sistema que facilita al gestor controlar el proceso de desarrollo de software. Así va progresando a través del análisis, diseño, codificación prueba y mantenimiento.





El desarrollo de nuestro sistema esta basado en la utilización de fase o etapas que surgen como la combinación del método del ciclo de vida clásico y el método de canales estructurado, siendo estas las siguientes:

Planificación

Esta fase permite obtener un plan para la utilización de sistema de Información en una empresa que esté acorde con las necesidades operativa como de dirección de la misma. Dicha fase permite obtener una visión general de la empresa, sus funciones, objetivos, datos y necesidades de Información.

Esta actividad consta de tres partes:

1. **Aclaración de solicitud:** la solicitud del proyecto debe examinarse para determinar con precisión lo que el solicitante desea, es decir, la solicitud del proyecto debe estar claramente planteada.
2. **Estudio de factibilidad:** es el análisis de un problema para determinar si puede ser resuelto efectivamente; a través de una investigación preliminar la cual consta de tres aspectos:
 - o Factibilidad técnica.
 - o Factibilidad económica.
 - o Factibilidad operacional.
3. **Aprobación de solicitud:** no todos los proyectos solicitados son deseables o factibles; sin embargo, aquellos que son deseables deben incorporarse en los planes de los analistas.



Análisis de los requerimientos del sistema.

Esta fase consiste en realizar el estudio de un problema antes de realizar una acción.

El aspecto fundamental es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. Para realizar este estudio se trabaja con los empleados de la empresa para dar respuesta a las siguientes preguntas claves:

1. ¿Qué es lo que se hace?
2. ¿Cómo se hace?
3. ¿Con que frecuencia se presenta?
4. ¿Qué tan grande es el volumen de transacciones o de decisiones?
5. ¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
6. ¿Existe algún problema?
7. Si existe algún problema, ¿Qué tan serio es?
8. Si existe un problema, ¿Cuál es la causa que lo origina?

Conforme se reúnen los detalles, se estudian los datos sobre los requerimientos con la finalidad de identificar las características que debe tener el sistema. El producto que se obtiene es primeramente una especificación de requisitos de software (**ERS**), luego el análisis de flujo de datos, que abarca la construcción de los diagramas de flujos (**DFD**) y el diccionario de datos; finalizando con la elaboración del modelo conceptual de datos (**MCD**).

- 1) **Especificación de requisitos de software (ERS):** es el establecimiento conciso de un conjunto de requisitos que deben ser satisfechos por un producto o proceso, indicando siempre que sea adecuado, el procedimiento mediante el cual se puede determinar si se han logrado satisfacer los requisitos.



ESTRUCTURA DE UN ERS

1. Introducción
 - 1.1. Propósito
 - 1.2. Alcance
 - 1.3. Definiciones, acrónimos y abreviaturas
 - 1.4. Referencias
 - 1.5. Visión General
2. Descripción General
 - 2.1. Relaciones del producto
 - 2.2. Funciones del producto
 - 2.3. Características del usuario
 - 2.4. Restricciones generales
 - 2.5. Suposiciones y dependencias
3. Requisitos Específicos
 - 3.1. Requisitos funcionales
 - 3.1.1. Nombre de función
 - 3.1.1.1. Especificación
 - 3.1.1.1.1. Introducción
 - 3.1.1.1.2. Entradas
 - 3.1.1.1.3. Proceso
 - 3.1.1.1.4. Salidas
 - 3.2. Interfaces externas
 - 3.2.1. Interfaces de usuario
 - 3.2.2. Interfaces hardware
 - 3.2.3. Interfaces software
 - 3.2.4. Interfaces de comunicación
 - 3.3. Requisitos de funcionamiento
 - 3.3.1. Requisitos estáticos



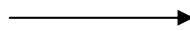
- 3.3.2. Requisitos dinámicos
- 3.4. Restricciones de diseño
- 3.5. Atributos
 - 3.5.1. Seguridad
 - 3.5.2. Mantenimiento
 - 3.5.3. Ayuda
- 3.6. Otros requisitos
 - 3.6.1. Bases de datos
 - 3.6.2. Operaciones

2) **Análisis de flujo de datos:** examina el empleo de los datos para llevar a cabo procesos específicos de la empresa, el cual esta compuesto de:

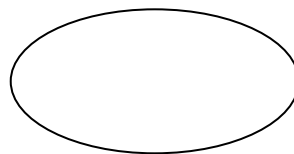
a. Diagrama de flujo de datos (DFD): es la descripción grafica del sistema por el método de análisis estructurado; se sigue un proceso descendente (top-down). Cada proceso puede desglosarse en diagramas de flujo de datos cada vez mas detallados, hasta obtener suficientes detalles que permitan comprender en su totalidad la parte del sistema que se encuentra bajo investigación.

Símbolos utilizados en los DFD:

Flujo de datos



Procesos



Almacén de datos





Entidad Externa

b. Diccionario de datos: en el se encuentran definidos de forma detallada todos los elementos del sistema (flujos de datos, procesos, y almacén de datos), además guarda los detalles y descripciones de todos los elementos. El diccionario se desarrolla durante el análisis del flujo d datos.

Importancia:

- Para manejar los detalles en sistemas grandes.
- Para comunicar un significado común para todos los elementos del sistema.
- Para documentar las características del sistema.
- Para facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar donde efectuar cambios en el sistema.
- Localizar errores y omisiones en el sistema.

Pasos para construir un diccionario de datos:

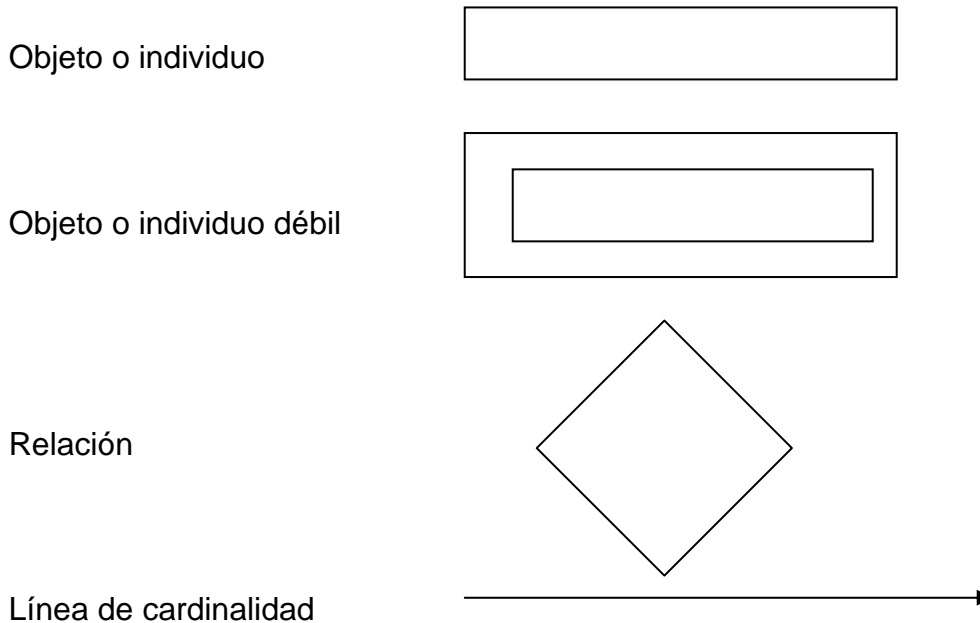
- Incorporar el proceso.
- Catalogar los flujos básicos de datos.
- Describir la estructura de datos.
- Desglosar la estructura de datos.

3) **Modelo conceptual de datos (MCD):** este modelo permite representar los datos con independencia de decisiones técnicas, encargándose de las descripciones de los mismos. Este modelo esta formado por un nivel de gran importancia como lo es el **Nivel conceptual**, existiendo diferentes modelos de datos como: **Modelo Entidad Relación (E-R)** siendo éste muy utilizado, el **Modelo Orientado a Objeto**, el **Modelo Infológico**, el **Modelo Binario**, etc.



Nivel conceptual: describe qué datos son realmente almacenados en la base de datos y las relaciones que existen entre ellos.

El MCD se hace con la ayuda de 4 símbolos:



- **Diseño del sistema**

Esta fase produce los detalles que establecen la forma en la que el sistema cumplirá con los requisitos identificados durante la fase de análisis. Es llamado diseño lógico y consiste en:

- El diseño de los reportes y demás salidas que debe producir el sistema.
- La identificación de los datos de entrada, aquellos que serán calculados y los que deben de ser almacenados. Así mismo, se escriben con detalle los procedimientos de calculo y los individuales; selección de estructuras de los archivos y dispositivos de almacenamiento.



Las especificaciones de diseño se representan a través de diagramas, tablas y símbolos especiales, todo esto para comenzar la fase de desarrollo de software.

La fase de diseño implica trabajar con cuatro tipos de diseños como son:

- Diseño de datos.
 - Diseño de los Diagramas de Dialogo.
 - Diseño de interfaces.
 - Diseño procedimental.
-
- **Desarrollo de software**

Esta fase consiste en:

- Codificación y prueba de programas en base a las especificaciones del diseño.
- Documentación para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

Para construir el sistema diseñado se pueden utilizar lenguajes de programación de alto nivel, lenguajes de cuarta generación y gestores de bases de datos, que utilizan generadores de informes, menús, pantallas, etc.; los más utilizados son los gestores de bases de datos como: Access, FoxPro, Oracle, DataBase, Visual FoxPro, SQL Server, etc.

En este documento comentaremos sobre MYSQL Server, Visual Basic y las demás herramientas utilizada para el desarrollo de nuestra aplicación.



- **Prueba del sistema**

Durante esta fase se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga.

- **Implantación y Evaluación**

La implantación es el proceso de verificar e instalar el nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarlo.

La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes; ocurre a lo largo de las siguientes dimensiones:

- Evaluación operacional.
- Impacto organizacional.
- Opinión de los administradores.
- Desempeño del desarrollo.



RESULTADOS

Análisis

Especificación de requisitos de software

4. Introducción

4.1 Propósito

Definición del conjunto de especificaciones de requisitos software que debe cumplir la aplicación de **“Sistema automatizado de cuentas por cobrar”**, consiste en la Automatización de los Trámites Contables referentes al Módulo de Cuentas por Cobrar (CC) existente en cualquier empresa de nuestro país.

Este documento se dirige a la Empresa **“Comercial Eléctrica”** y a todos sus usuarios finales que deberán estudiarlo para su aprobación o desacuerdo antes de abordar la fase de análisis.

4.2 Alcance.

El nombre con el que se conocerá a esta aplicación será:

“Sistema automatizado de cuentas por cobrar”
“SA - CC”

El producto realizará las siguientes funciones:

1. Registrar cuenta.
2. Modificar datos de la cuenta.
3. Borrar cuenta.



4. Imprimir cuenta.
5. Registrar datos del cliente.
6. Modificar datos del cliente.
7. Borrar cliente.
8. Imprimir cliente.
9. Registrar factura del cliente.
10. Modificar factura del cliente.
11. Eliminar factura del cliente.
12. Imprimir y/o visualizar factura.
13. Imprimir y/o visualizar calendario de pago.
14. Registrar recibo del cliente.
15. Imprimir ó Visualizar recibo.
16. Registrar nota de debito.
17. Imprimir y/o visualizar nota de debito.
18. Registrar nota de crédito.
19. Imprimir y/o visualizar nota de crédito
20. Consultar estado de cuentas.
21. Consultar estado general de cuentas.
22. Consultar antigüedad de saldos.
23. Consultar saldo y consolidado de los saldos.
24. Reporte de estado de cuentas.
25. Reporte de estado general de cuentas.
26. Reporte de saldo y consolidado de saldos.
27. Reporte de antigüedad de saldos.
28. Crear usuario.
29. Modificar permiso.
30. Respalidar.
31. Restaurar.
32. Enviar registro al historial.
33. Consultar historial.



34. Cambiar contraseña de usuario.

4.3 Definición acrónimos y abreviaturas

1. **Registrar Cuenta:** Acción de añadir una cuenta en la entidad “CUENTA” de aquí en adelante “Registrar_Cta”. Esta se clasifica en:
 - a. **Cliente:** Este se le asigna a las personas que hacen uso de este servicio y que no son personas miembros de la empresa. De aquí en adelante “**CLIENTE**” (CLTE).
 - b. **Impuesto por cobrar:** Es el impuesto que la persona dueña del negocio paga por la compra de un determinado producto para que luego en la realización de su venta se cobre. De aquí en adelante “**IMPUESTO POR COBRAR**” (IMPC)
 - c. **Préstamo a funcionarios y empleados:** Personas que forman parte del negocio y que solicitan préstamo en la empresa en la cual trabajan. De aquí en adelante “**PRESTAMO A FUNCIONARIOS Y EMPLEADOS**” (PFE).
 - d. **Otros:** otras cuentas que se estimen convenientes pertenecientes a cuentas por cobrar. De aquí en adelante OTROS (OTROS)
2. **Modificar datos de la cuenta:** Acción de modificar datos de una determinada cuenta registrada en la entidad “cuenta”. De aquí en adelante “MODIFICAR_CUENTA”.
3. **Borrar cuenta:** Acción de Borrar una cuenta correspondiente a una determinada “cuenta”, de aquí en adelante “BORRAR _ CUENTA”. Esto se hará siempre y cuando no existan clientes pertenecientes a esta cuenta.
4. **Imprimir cuenta:** Acción de imprimir las cuentas que se encuentran registradas en la entidad “cuenta” de aquí en adelante “IMPRIMIR _ CUENTA”.



- 5. Registrar datos del cliente:** Acción de añadir los datos correspondientes a un determinado cliente, registrados en la entidad “cliente”, de aquí en adelante “Datos_Clientes”.
- 6. Modificar datos del cliente:** Acción de modificar datos de un cliente registrado en la entidad “cliente”, de aquí en adelante “MODIFICAR_CLIENTE”
- 7. Borrar cliente:** Acción de eliminar un cliente registrado en la entidad “cliente”, de aquí en adelante “BORRAR_CLIENTE”.
- 8. Imprimir cliente:** Acción de imprimir todos los clientes ó un cliente registrado en la entidad “cliente” de aquí en adelante “IMPRIMIR _CLIENTE”.
- 9. Registrar factura del cliente:** Acción de añadir una factura a un cliente. Documento que se le entrega al cliente en el cual se especifica el importe correspondiente al crédito suministrado y la fecha en la que tendría que dar abonos por dicho crédito, perteneciente a la entidad “factcredito”, de aquí en adelante “Factura_Credito”.
- 10. Modificar factura del cliente:** Acción de modificar la factura de un cliente registrado en la entidad “factcredito”, de aquí en adelante “MODIFICAR_FACTURA”.
- 11. Eliminar factura del cliente:** Acción de eliminar la factura de un cliente registrado en la entidad “factcredito”, de aquí en adelante “ELIMINAR _FACTURA”. Cabe mencionar que solo se podrán eliminar aquellas facturas que se hallan emitido el mismo día
- 12. Imprimir y/o Visualizar factura:** Acción de imprimir y/o visualizar las facturas, registradas en la entidad “factcredito” de aquí en adelante “IMPRIMIR Y/O VISUALIZAR FACTURA”.
- 13. Imprimir y/o Visualizar Calendario de pago:** Acción de imprimir y/o visualizar las fechas de pagos correspondientes, registrada en la entidad “factcredito” de aquí en adelante “IMPRIMIR Y/O VISUALIZAR, CALENDARIO DE PAGO”.



- 14.Registrar recibo del cliente:** Acción de añadir un abono o pago del cliente, registrado en la entidad “recibo” de aquí en adelante ”Realizar_Abono”.
- 15.Imprimir o Visualizar recibo:** Acción de imprimir o visualizar un abono del cliente registrado en la entidad “recibo”, de aquí en adelante “IMPRIMIR O VISUALIZAR _ RECIBO”.
- 16.Registrar nota de débito:** Acción de añadir una nota de debito registrado en la entidad “notadebito”. Documento que se carga en contra del cliente (a favor de la empresa) por motivos de un mal cálculo, etc. De aquí en adelante “NOTA _ DÉBITO”.
- 17.Imprimir y/o visualizar nota de debito:** Acción de imprimir y/o visualizar una nota de debito. De aquí en adelante IMPRIMIR Y/O VISUALIZAR NOTA_DEBITO
- 18.Registrar nota de crédito:** Acción de añadir una nota de crédito registrado en la entidad notacredito Documento que se carga a favor del cliente cuando se le cobra de menos (en contra de la empresa). De aquí en adelante “NOTA_CRÉDITO”.
- 19.Imprimir y/o visualizar nota de crédito:** Acción de imprimir y/o visualizar nota de crédito. Registrado en la entidad notacredito.
- 20.Consultar estado de cuenta:** Acción de consultar el estado de cuentas por cliente, registrado en las entidades “factura” y “recibo”, donde verifica las facturas y recibos emitidos del cliente. de aquí en adelante “Consulta_Estado_Cta_Cliente”
- 21.Consultar estado de cuenta general:** Acción de consultar todos los clientes de las diferentes cuentas que existen en el sistema, este depende de las entidades “cuenta, cliente, factcredito y recibo”, donde verifica las facturas y recibos emitidos por los diferentes clientes. De aquí en adelante “Consulta_Estado_Cta_General”
- 22.Consultar antigüedad de saldos:** realiza consulta de la antigüedad de saldo de los clientes (saldos pendientes) por cuenta, en un rango de



fecha determinado, dependiendo este de las entidades “cuenta, factcredito, recibo (pagos o abonos)”. De aquí en adelante se llamara “Consultar Antigüedad Saldo”.

23. Consultar saldo y consolidado saldos: realiza una consulta del consolidado de los saldos totales de un cliente y de todos los clientes pertenecientes de los diferentes tipos de cuentas en un rango de fecha determinado. De aquí en adelante “Consulta_Saldo_Consolidado_Saldo”

24. Reporte: Acción de mostrar un documento donde se vean reflejados los contenidos de interés de determinada entidad .En nuestra tesis tendrá las misma características que del menú “**Consultar**”, con la única diferencia que este se imprimirá, de aquí en adelante “**REPORTE_X**”

Donde x puede ser:

24.1. Reporte de estado de cuentas→Reporte_Est_Cta_Cliente

25. Reporte de estado de cuenta general→Reporte_Est_Cta_General

26. Reporte de saldos de clientes. →Reporte_Saldo

27. Reporte de antigüedad de saldos→Reporte_Antigüedad Saldo

28. Reporte de consolidados de saldo→Reporte_Consolidado Saldo.

29. Crear usuarios: Acción de crear un usuario, De aquí en adelante Crear_Usuario

30. Modificar permiso: Acción de modificar los permisos de usuario. De aquí en adelante Modificar _ permisos

31. Respalda: acción de respaldar o guardar la base de datos , de aquí en adelante Respalda

32. Restaurar: Acción de restaurar o reestablecer la base de datos respaldada. Restaurar

33. Enviar registro al historial: acción de enviar los registros al historial, estas son todas las facturas canceladas con sus abonos correspondientes. De aquí en adelante Enviar registros al historial



34. Consultar historial: acción de consultar historial, estos son todos los registros que anteriormente se hallan enviado al historial.

35. Cambiar contraseña de usuario. : Acción de modificar la contraseña de usuario. De aquí en adelante Cambiar contraseña de usuario..

4.4 Referencias.

Documento donde se tomo información base para el análisis y la realización del software:

- 1) Direcciones de Internet
 - a) http://www.MySQL_Manual
 - b) <http://www.mysql.com/documentation/index.html>
 - c) www.monografias.com
 - d) www.dubina.com/internet/tutoriales.asp

- 2) Bibliografía:
 - a) Monografía Proyecto de software contable de la Empresa ASOGAL.
 - b) Monografía. Automatización del Modulo Contable Cuentas por Pagar.
 - c) Programación avanzada con Visual Basic.
 - d) Enciclopedia de Visual Basic.



4.5 Visión general.

Primero se realizara una Descripción General del producto que se desea desarrollar para pasar posteriormente a estudiar cada uno de los Requisitos Específicos individualmente.

5 Descripción general.

5.1. Relaciones del producto.

La aplicación interactúa con el “sistema automatizado de Cuentas por Cobrar”.

El equipo en el que se desarrollara el sistema será una computadora con las siguientes características:

Modelo: HP Vectra 128 Mb de RAM .20 Gb de Disco Duro, con unidad lectora de CD-ROM.

Cabe destacar lo siguiente:

- La captura de los datos se realizara de forma interactiva por pantalla.
- El software interactúa con la base de datos “cuentacobrar” Se podrá utilizar cualquier terminal conectado a un servidor de Bases de Datos, los cuales deberán estar conectados a una red de área local dentro de la empresa – usuario del sistema. Nuestro sistema podrá trabajar dentro de un solo computador donde deberá de estar la base de datos y el sistema.



- El sistema podrá ser implementado en una red de área local, por lo tanto se hará uso de terminales que tengan instaladas tarjetas de red.

5.2. Funciones del producto.

El producto software debe contener todas las tareas que realizan manualmente las empresas de forma diaria, referentes al sistema de cuentas por cobrar. Estas son:

- 1) Cuando se necesite dar de alta a un nuevo registro referente a una cuenta, el usuario deberá dar en el menú "**Mantenimiento**", a continuación en la opción "**Cuentas**", luego en la ventana principal de cuenta "**Nueva**"; introduciendo todos los datos correspondientes de dicha cuenta, y damos a continuación agregar. Cabe mencionar que una vez estando en el formulario principal de "cuentas", se presentan las opciones principales tales como: Nueva, modificar, Borrar, Imprimir, Cerrar.
- 2) Cuando se requiera modificar un registro existente de la entidad "**CUENTA**", el usuario debe dar en el menú "**Mantenimiento**", a continuación en la opción "**Cuentas**", luego en "**Modificar**", se presenta una ventana con los siguientes botones "Modificar","Guardar","Cancelar","Cerrar" ,damos enter en "Modificar" y hacemos las modificaciones y luego en el botón "Guardar"
- 3) Cuando se necesite borrar una cuenta de la entidad "**CUENTA**" el usuario debe dar en el menú "**Mantenimiento**", a continuación en la opción "**Cuentas**", luego en "**Borrar**".
- 4) Cuando se necesite imprimir las cuentas existentes en la entidad "**Cuenta**" el usuario debe dar en el menú "**Mantenimiento**", a continuación en la opción "**Cuentas**" luego en "**Imprimir**".
- 5) Cuando se requiera dar de alta a un nuevo registro referente a un cliente, el usuario debe en el menú "**Mantenimiento**", a continuación en la opción



“**Cientes**”, luego en “**Nuevo**” donde se agrega los datos correspondientes a dicho cliente y damos a continuación Agregar. Cabe mencionar que una vez estando en el formulario principal de Cliente se presentan las opciones principales tales como: Nuevo, Modificar, Borrar, Imprimir, Cerrar.

- 6) Cuando se necesite modificar un registro correspondiente a la entidad “**CLIENTE**”, el usuario debe dar en el menú “**Mantenimiento**”, a continuación en la opción “**Cientes**”, luego en “**Modificar**”. se presenta una ventana con los siguientes botones “Modificar”, “Guardar”, “Cancelar”, “Cerrar” , damos clic en “Modificar” y hacemos las modificaciones y luego en el botón “Guardar”
- 7) Cuando el usuario necesite eliminar un registro correspondiente a la entidad “**CLIENTE**”, debe dar en el menú “**Mantenimiento**”, a continuación en la opción “**Cliente**”, luego en “**Borrar**”.
- 8) Cuando se necesite imprimir el cliente existente en la entidad “**Cliente**” el usuario debe dar en el menú “**Mantenimiento**”, a continuación en la opción “**Cliente**” luego en “**Imprimir**” presentándonos una ventana con dos opciones :Registro actual ó Todos los registros y los botones Aceptar y Cancelar.
- 9) Cuando el usuario necesite registrar un crédito (Factura de crédito), el usuario debe dar en el menú “**Mantenimiento**”, a continuación en la opción “**Crédito**”, luego en el submenú “**Nuevo Crédito**”. Aparece una nueva ventana para poder registrar la nueva factura. Cabe mencionar que una vez estando en el formulario principal de “Nuevo Crédito”, se presentan las opciones principales tales como: Nuevo, Guardar, Cancelar, Imprimir, Cerrar.
- 10) Cuando el usuario necesite modificar un crédito (Factura de crédito), el usuario deberá dar en menú “**Mantenimiento**”, a continuación en la opción “**Crédito**”, luego en el submenú “**Modificar Crédito**”. Aparecerá una nueva ventana para poder registrar las modificaciones. Cabe mencionar que una



vez estando en el formulario principal de “Modificar Crédito”, se presentara las opciones principales tales como: Modificar, Anular, Imprimir, Cerrar. Luego damos clic en modificar, de nuevo se presenta una ventana con las siguientes opciones Modificar, Guardar, Cancelar, Imprimir, Cerrar, se hacen las respectivas modificaciones y damos clic en guardar.

- 11) Cuando el usuario necesite anular un crédito (Factura de crédito), el usuario deberá dar en menú “**Mantenimiento**”, a continuación en la opción “**Crédito**”, luego en el submenú “**Modificar Crédito**”. Aparecerá una nueva ventana para poder anular. Cabe mencionar que una vez estando en el formulario principal de “Modificar Crédito”, se presentara las opciones principales tales como: Modificar, Anular, Imprimir, Cerrar. Luego damos clic en anular, de nuevo se presenta una ventana preguntando si estamos seguros de anular esa factura con dos opciones SI ó NO, y daremos si.
- 12) Cuando el usuario necesite imprimir un crédito (Factura de crédito), el usuario deberá dar en menú “**Mantenimiento**”, a continuación en la opción “**Crédito**”, luego en el submenú “**Nuevo Crédito**”. Aparecerá una nueva ventana. Cabe mencionar que una vez estando en el formulario principal de “Nuevo Crédito”, se presentará las opciones principales tales como: Modificar, Anular, Imprimir, Cerrar. Elegimos si lo que queremos es imprimir ó Visualizar por factura, calendario de pagos o ambos.
- 13) Cuando el usuario necesite registrar abonos referentes a una cuenta por cobrar que corresponda a un determinado “**CLIENTE**” el usuario deberá dar en el menú “**Mantenimiento**”, a continuación en la opción “**Recibo**”. Cabe mencionar que una vez estando en el formulario principal de “Crédito”, se presentara las opciones principales tales como: Nuevo, Guardar, Imprimir, Cancelar.
- 14) Cuando el usuario necesite registrar una nota de debito deberá dar en el menú “Mantenimiento “, a continuación en la opción **Nota Debito**. Cabe mencionar que una vez estando en el formulario principal de “Nota Debito” ,



- se presentaran las opciones principales tales como : Nuevo, Guardar , Cancelar, Imprimir y Cerrar
- 15) Cuando el usuario necesite registrar una nota de crédito deberá dar en el menú **Mantenimiento**”, a continuación en al opción **Nota Crédito** .Cabe mencionar que una vez estando en el formulario principal de “Nota Crédito”, se presentaran las opciones principales tales como: Nuevo, Guardar, Cancelar, Imprimir, Cerrar.
- 16) Cuando el usuario necesite consultar facturas de crédito (que aun no este cancelada), deberá dar en el menú **“Mantenimiento”**, a continuación en la opción **“Créditos”**, luego en la submenú **Modificar Crédito**.
- 17) Cuando el usuario necesita consultar el estado de cuenta referente a cuentas por cobrar de un cliente deberá dar en el menú **“Consultar”**, a continuación en la opción **“Estado de Cuenta”**.a continuación seleccionamos “Por Cliente” buscando el cliente que queremos visualizar.
- 18) Cuando el usuario necesite consultar el estado de cuenta general de los clientes que pertenecen a los diferentes tipos de cuenta en la que se clasifica **“cuenta por cobrar**, Este debe dar en el menú **“Consultar”**, a continuación en la opción **“Estado de Cuenta”** , luego en la opción **“General”**
- 19) Cuando el usuario necesite consultar los saldos de los clientes de las diferentes tipos de cuentas en las que se clasifica **“cuenta por cobrar”**, deberá dar en el menú **“Consultar”**, a continuación en el la opción **“Saldos y consolidado de saldos”**.
- 20) Cuando el usuario necesite consultar el consolidado de saldos de todas las cuentas que pertenecen a cuentas por cobrar (con sus clientes respectivos), debe dar en el menú **“Consultar”**, a continuación en la opción **“Saldos y consolidado de saldos.”**
- 21) Cuando el usuario necesite consultar la antigüedad de saldos por cuentas de los clientes debe dar en el menú **“Consultar”**, a continuación en la opción **“Antigüedad de Saldo”**.



- 22) Cuando el usuario necesite imprimir un reporte acerca del estado de cuenta de un cliente, debe dar en el menú "**Reporte**", a continuación en la opción "**Estado de cuenta**". Seleccionamos "Por Cliente", buscamos el nombre del cliente a imprimir
- 23) Cuando el usuario necesite imprimir un reporte referente al estado de cuentas general debe dar en el menú "**Reporte**", a continuación en la opción "**Estado de Cuenta**"., seleccionando "General"
- 24) Cuando el usuario necesite imprimir un reporte de los saldos por cuentas de los clientes debe dar en el menú "**Reporte**", a continuación en "**Saldos y consolidados de saldos**".
- 25) Cuando el usuario necesite imprimir un reporte del consolidado de los saldos de todas las cuentas con sus respectivos clientes, debe dar en el menú "**Reporte**", a continuación en la opción "**Saldos y consolidados de saldos**".
- 26) Cuando el usuario necesite imprimir un reporte de la antigüedad de saldo de los clientes por cuentas, debe dar en el menú "**Reporte**", a continuación en la opción "**Antigüedad de saldo**".
- 27) Cuando el usuario necesite ingresar al sistema deberá escribir el **nombre del servidor**, su **nombre de usuario** y su respectiva **contraseña**; que sea válida para el sistema. Luego se dará en la orden "**ACEPTAR**".
- 28) Cuando el usuario necesite dar de alta a un identificador de usuario y su contraseña, deberá dar en el menú "**Herramientas**", a continuación en la opción "**Usuarios**", luego en el submenú "**Crear usuario**". Cabe mencionar que una vez estando en el formulario principal de "**Crear usuario**", se presentará las opciones principales tales como: Agregar, Seleccionar, Deshacer, Borrar y Cerrar.
- 29) Cuando se necesite modificar los permisos de un determinado usuario, se deberá dar en el menú "**Herramientas**", a continuación en la opción "**Usuario**", seleccionando "**Modificar permisos**", se deberá dar doble clic al usuario que se le pretende modificar los permisos, realizadas las



modificaciones damos clic en el botón permisos, aceptando o no los cambios efectuados.

30) Cuando se necesite respaldar la información se deberá dar en la opción **Herramientas** luego en **Respaldar**, seleccionando la unidad y la carpeta en la cual queremos respaldar.

31) Cuando se necesite restaurar la información respaldada se deberá dar en la opción **Herramientas** luego en **Restaurar**.

32) Cuando el usuario necesite enviar registros al historial (las facturas canceladas con sus respectivos recibos, notas de créditos y nota de debitos asociados) el usuario deberá dar en la opción **Herramientas** a continuación en **Enviar registros al historial**, seleccionando la fecha de inicio y la fecha de fin de los registros que se desean enviar al historial.

33) Cuando el usuario necesite consultar una factura de crédito (ya cancelada), con sus respectivos recibos de abono deberá dar en el menú "**Herramientas**", a continuación en la opción "**Consultar Historial**", seleccionamos el rango de fecha en el cual se pueden encontrar las facturas; si se quiere buscar de forma más rápida se seleccionaría en el campo buscar por, seleccionamos "**Numero de Facturas**", escribiendo a continuación el numero de factura a buscar, apareciendo la factura buscada con todos sus recibos asociados, o por el **Nombre del cliente**.

34) Cuando el usuario necesite cambiar su contraseña deberá seleccionar el menú **Herramienta**, en la opción **Cambiar contraseña de usuario**, introduciendo su **contraseña actual**, a continuación su **nueva contraseña** y **confirmación de ésta**.

35) Cuando el usuario necesite realizar un respaldo, primero deberá seleccionar la ruta en donde se encuentra el comando a ejecutar (MysqIDump), Esto se hará seleccionando en el menú **Herramientas** la opción **Configuración de parámetros**.

5.3 Características del Usuario.



El usuario de nuestro sistema deberá tener conocimientos acerca del manejo básico de un sistema computarizado y conocimientos de contabilidad de empresas, haciendo énfasis en lo que es Cuentas por Cobrar.

5.4 Restricciones Generales.

La empresa comercial que utilizará nuestro sistema deberá tener como requisitos mínimos:

- Computadoras de 800 Mhz de procesador, Disco Duro de 10 Gb Y 128 Mb de Memoria RAM.
- Sistema Operativo Windows 2000 o Xp.

- MySQL Server 5.0.18.
- Driver Mysql ODBC 3.5.1
- Mysql Administrator 1.1.7
- Mysql Query – Browser 1.1.19
- Opcional a una red Área Local.
- Controles Ocx (List Pro 3).
- Controles Active Reports Pro 2.
- Setup Factory 7.0.
- Botones KewlButtonz.ocx.
- MZTools3.dll.
- Manifest.exe.
- Recursos (iconos).
- VB6IDEMouse.dll

6. Requisitos Específicos.

6. 1.Requisitos Funcionales



6. 1. 1 Nombre de la función: Registrar_Cta.

6.1.1.1. Especificación:

6.1.1.1.1. **Introducción:** esta función permitirá al usuario adicionar en el sistema, los datos correspondientes a una nueva cuenta, haciendo uso de la entidad “cuenta”.

6.1.1.1.2. Entradas:

Por pantalla:

- **Código Cuenta:** es un dato obligatorio, debe de existir, de lo contrario se presentará un mensaje de que la cuenta no existe, este campo representa el código a través del cual se le conocerá a una cuenta de manera única en el sistema. Este dato lo proporcionará el usuario del sistema (llámese administrador, etc.) y tendrá el formato siguiente: donde los cuatros primeros caracteres serán letras a continuación de dos números.
- **Nombre de la cuenta:** es un dato obligatorio, debe de existir, de lo contrario se presentará un mensaje “Falta el Nombre de la Cuenta”. Representa el Nombre de la cuenta

Por el sistema:

Esta función no tendrá entradas por el sistema

6.1.1.1.3. **Proceso:** se mostrará la interfaz a través de la cual se podrá ingresar los datos correspondientes a una nueva cuenta, esta se añadirá en la entidad “cuenta”.



6.1.1.1.4. **Salida:** se añadirá un nuevo registro en la entidad “cuenta”

6.1.2. Nombre de la función: MODIFICAR_CUENTA.

6.1.2.1. Especificación:

6.1.2.1.1. **Introducción:** esta función permitirá al usuario modificar un registro existente en la entidad “cuenta”, referente a una determinada cuenta. Los campos que almacenaría un registro a modificar serian: código de la cuenta y nombre de la cuenta.

6.1.2.1.2. Entradas:

Por pantalla:

- **Código Cuenta.**
- **Nombre de la Cuenta.**

Por el sistema:

Esta función no tendrá entradas por el sistema

6.1.2.1.3. **Proceso:** se mostrará la interfaz a través de la, cual se podrá modificar o editar los datos correspondientes a una determinada cuenta, a continuación se nos mostrará los datos correspondiente a esa cuenta, datos que después de la modificación serán actualizados en la entidad “cuenta”.

6.1.2.1.4. **Salida:** se guardará el registro modificado en la entidad “cuenta”.

6.1.3. Nombre de la función: BORRAR _ CUENTA



6.1.3.1. **Especificación:**

6.1.3.1.1. **Introducción:** esta función permitirá al usuario eliminar del sistema una cuenta almacenada en la entidad “cuenta”

6.1.3.1.2. **Entradas:**

Por pantalla:

- **Código cuenta.**
- **Nombre Cuenta:**

Por el sistema:

Esta función no tendrá entradas por el sistema

6.1.3.1.3 **Proceso:** se visualiza una interfaz mostrando la información de la cuenta que deseamos borrar.

6.1.3.1.4 **Salida:** se elimina la cuenta almacenada en la entidad “cuenta”

6.1. 4. **Nombre de la función: IMPRIMIR _ CUENTAS.**

6.1.4.1. **Especificación:**

6.1.4.1.1. **Introducción:** Esta función permite que el usuario imprima todas las cuentas que existen en el sistema.

6.1.4.1.2. **Entradas:**

Por pantalla:

Esta función mostrará todas las cuentas.



Por el sistema:

Esta función no tendrá entradas por el sistema

6.1.4.1.3. **Proceso:** se muestran las cuentas a imprimir que están registradas en el sistema

6.1.4.1.4. **Salida:** el usuario puede visualizar en la hoja impresa las diferentes cuentas que están registradas en la entidad "cuenta".

6.1. 5. Nombre de la función: ADICIONAR _ CLIENTE.

6.1.5. 1. Especificación:

6.1.5.1.1. **Introducción:** esta función permite al usuario registrar en el sistema los datos correspondientes a un nuevo cliente, haciendo uso de la entidad "cliente".

6.1.5.1.2. Entradas:

Por pantalla:

- **Código cliente:** es un dato obligatorio, debe existir, de lo contrario no se activará el botón de guardar el cliente .Representa la identificación única del cliente al cual se le esta realizando el crédito.
- **Nombres:** es un dato obligatorio, debe de existir, de lo contrario se presentara un mensaje "Falta nombre del cliente", este campo nos permite saber cual es el nombre del cliente al que se le esta asignando determinado código.
- **Apellidos:** Es un dato obligatorio, debe existir, de lo contrario se presenta un mensaje de que "Falta apellidos. Este campo nos permitirá saber los apellidos del cliente.



- **Dirección:** Es un dato obligatorio , debe de existir de los contrario se presenta un mensaje “Falta Dirección”
- **Cédula:** es un dato obligatorio, debe de existir, de lo contrario se presenta un mensaje “Falta cédula del cliente”.
- **Ciudad:** es un dato obligatorio, debe de existir, de lo contrario se presenta un mensaje “Falta ciudad o municipio del cliente.
- **Celular:** no es un dato obligatorio, puede o no existir.
- **Teléfono Casa:** no es un dato obligatorio, puede o no existir.
- **Límite del Crédito:** es un dato obligatorio, debe existir, de lo contrario se presenta un mensaje “Falta límite de crédito.

Por el sistema:

- **Fecha:** Este campo representa la fecha en que el cliente se está registrando en la comercial.

6.1.5.1.3. **Proceso:** se muestra la interfaz a través de la cual se podrá ingresar los datos correspondientes a un nuevo cliente, este se añadirá en la entidad “cliente”.

6.1.5.1.4. **Salida:** se añadirá un nuevo registro en la entidad “cliente”.

6.1. 6. Nombre de la función: MODIFICAR_CLIENTE

6.1.6.1. Especificación:

6.1.6.1.1. **Introducción:** esta función permite al usuario modificar un registro existente en la entidad “cliente”.

6.1.6.1.2. Entradas:

Por pantalla:



- **Código cliente.**
- **Nombre cliente.**
- **Apellido.**
- **Dirección.**
- **Cédula.**
- **Ciudad.**
- **Celular.**
- **Teléfono Casa.**
- **Limite del Crédito.**

Por el sistema:

- **Fecha.**

6.1.6.1.3. **Proceso:** se muestra la interfaz a través de la cual se podrá modificar los datos correspondientes a un cliente determinado, datos que serán actualizados en la entidad “cliente”.

6.1.6.1.4. **Salida:** se guarda el registro modificado en la entidad “cliente”.

6.1.7. Nombre de la función: Borrar _ cliente.

6.1.7.1. Especificación:

6.1.7.1.1. **Introducción:** esta función permitirá al usuario eliminar un registro correspondiente a la entidad “cliente”, Cabe mencionar que si este tiene créditos pendientes no se podrá borrar

6.1.7.1.2. Entradas:



Por pantalla:

- **Código cliente.**
- **Nombre cliente.**
- **Apellido.**
- **Dirección.**
- **Cédula.**
- **Teléfono Casa.**
- **Limite del Crédito.**

Por el sistema:

- **Fecha.**

6.1.7.1.3. **Proceso:** se muestra la interfaz a través de la cual se podrá eliminar los datos correspondientes a la entidad “cliente”, únicamente se podrán borrar los clientes que no tenga créditos pendientes.

6.1.7.1.4. **Salida:** se elimina el registro correspondiente de la entidad “cliente”

6.1. 8. Nombre de la función: IMPRIMIR _ CLIENTE.

6.1.8.1. Especificación

6.1.8.1.1. **Introducción:** esta función permite que el usuario imprima un registro ó todos los registros de los clientes existen en la entidad “cliente”.

6.1.8.1.2. Entradas:

Por pantalla:

Esta función muestra uno o todos los clientes que se impriman.



Por el sistema:

Esta función no tendrá entradas por el sistema.

6.1.8.1.3. **Proceso:** se muestra la interfaz a través de la cual el usuario podrá imprimir los datos de los clientes pertenecientes a cuenta por cobrar.

6.1.8.1.4. **Salida:** el usuario podrá visualizar en una(s) hoja uno ó todos los clientes que están registrados en la entidad “cliente”.

6.1. 9. Nombre de la función: **Factura_Credito**

6.1.9.1. **Especificación:**

6.1.9.1.1. **Introducción:** esta función permite al usuario registrar en el sistema una nueva factura dentro de la base de datos en concepto de un nuevo crédito , perteneciendo esta a la entidad “factcredito”, cabe mencionar que dicha función actualiza el valor del monto de dicho cliente si en un caso tuviese saldo pendiente de créditos anteriores.

6.1.9.1.2. **Entradas:**

Por pantalla:

- **Numero Factura:** es un dato obligatorio, debe de existir, de lo contrario se presenta un mensaje “Falta número de factura”. Representa la identificación única de la factura.
- **Monto del crédito:** es un dato obligatorio, debe de existir, de lo contrario se presenta un mensaje “Falta monto de crédito”. Este campo representa la cantidad por la que se esta efectuando el crédito.



- **Descripción:** No es un dato obligatorio, puede o no especificarse. Este campo representa una descripción breve del por que se dio el crédito.

Por el sistema:

- **Nombre de la Cuenta:** es un dato obligatorio, nos muestra los tipos de cuentas en que se clasifica en la entidad “cuenta”,
- **Nombre del Cliente:** es un dato obligatorio, nos muestra todos los clientes que están registrado en la entidad “cliente”.
- **Fecha:** es un dato obligatorio. Este campo representa la fecha en la que se está emitiendo la factura de crédito.
- **Plazo Crédito:** es un dato obligatorio. La comercial establece que el menor es 15 días y el mayor periodo de crédito es para 60 días.

- **Formas Pago:** es un dato obligatorio. Es la forma de los abonos de manera “SEMANAL, QUINCENAL Y MENSUAL”.
- **Primera Fecha Pago:** es un dato obligatorio. Es la primera fecha de abono del crédito solicitado.
- **Fecha Vencimiento:** es un dato obligatorio. Este campo representa la fecha máxima en la que debe cancelar el crédito.
- **Cantidad a Abonar:** es un dato obligatorio. Este campo le indica la cantidad del abono.
- **Limite del Crédito:** es un dato obligatorio. Este campo indica el límite de crédito con el que cuenta un cliente.

6.1.9.1.3.**Proceso:** se muestra la interfaz a través de la cual se podrá introducir los datos referentes a una nueva factura de crédito, efectuada a un determinado cliente , así como su correspondiente soporte en las diferentes entidades utilizadas.



6.1.9.1.4. **Salida:** se añadirá un nuevo registro en la entidad “factcredito”.

6.1. 10. Nombre de la función: MODIFICAR_FACTURA.

6.1. 10.1.Especificación:

6.1.10.1.1. **Introducción:** esta función permite al usuario modificar un registro existente en la entidad “factcredito”, referente a una factura emitida en concepto de crédito. Cabe mencionar que se podrán modificar únicamente los campos “Nombre de la Cuenta”, “Descripción”, “Plazo Crédito”.

6.1.10. 1. 2. Entradas:

Por pantalla:

- **Nombre de la cuenta.**
- **Descripción.**
- **Plazo Crédito.**

Por el sistema:

- **Numero Factura.**
- **Nombre del Cliente.**
- **Fecha.**
- **Monto Crédito.**
- **Formas pago.**
- **Primera Fecha Pago.**
- **Fecha vencimiento.**
- **Cantidad a Abonar.**
- **Limite del Crédito.**



6.1.10.1.3. **Proceso:** se muestra la interfaz a través de la cual se podrá modificar los datos referentes a una factura ya existente almacenada en la entidad “factcredito”.

6.1.10.1.4. **Salida:** se modifica y actualiza un registro ya existente en la entidad “factcredito”.

6.1. 11. Nombre de la función: ELIMINAR _ FACTURA

6.1.11.1. Especificación:

6.1.11.1.1. **Introducción:** esta función permite al usuario eliminar una factura mal grabada ya existente almacenada en la entidad “factcredito”.

6.1.11.1.2. Entradas:

Por pantalla:

- **Numero Factura.**
- **Nombres.**
- **Apellidos.**
- **Cuenta.**
- **Descripción.**
- **Monto.**
- **Fecha.**
- **Abono.**

Por el sistema:

Esta función no tendrá entradas por el sistema.



6.1.11.1.3. **Proceso:** se muestra la interfaz a través de la cual se podrá anular los datos referentes a una factura ya existente almacenada en la entidad “factcredito”. Preguntándonos que si estamos seguro de anular el crédito.

6.1.11.1.4. **Salida:** se anula el registro correspondiente de la factura en la entidad “factcredito”.

6.1. 12. Nombre de la función: IMPRIMIR Y/O VISUALIZAR FACTURA

6.1.12.1. Especificación:

6.1.12.1.1. **Introducción:** esta función permite que el usuario visualice y/o imprima todos los registros de los clientes existen en la entidad “factcredito”.

6.1.12.1.2. Entradas:

Por pantalla:

Esta función muestra uno o todos los clientes que se imprimen.

Por el sistema:

Esta función no tendrá entradas por el sistema.

6.1.12.1.3. **Proceso:** se muestra la interfaz a través de la cual el usuario podrá imprimir los datos de las facturas pertenecientes a cuenta por cobrar.

6.1.12.1.4. **Salida:** el usuario podrá visualizar e imprimir una ó todas las facturas que están registrados en la entidad “factcredito”.(cabe mencionar que solo las que están pendientes de pagos).



6.1. 13. Nombre de la función: Realizar_abono

6.1.13.1. Especificación:

6.1.13.1.1. **Introducción:** esta función permite al usuario ingresar al sistema los datos correspondientes a un recibo de abono almacenándolo en la entidad “recibo” realizado por el cliente en concepto de crédito hecho a su favor (del cliente).

6.1.13.1.2. Entradas:

Por pantalla

- **Numero Recibo:** es un dato obligatorio, debe de existir, de lo contrario se presenta un mensaje “Falta el numero de recibo”. Representa la identificación única del recibo.
- **Interés por Mora:** es un dato no obligatorio, que depende de los “días de retraso”, aca la comercial dispone de la política para aplicar el interés por la mora. Cabe señalar que si no hay días de retraso no se le aplica interés.
- **Descripción:** No es un dato obligatorio, puede o no especificarse. Este campo representa una descripción breve del por que se esta dando el abono o pago.

Por el sistema:

- **Nombre Cliente:** es un dato obligatorio, es dado por el sistema, aquí se encuentran todos los clientes que tienen pendientes créditos.
- **Numero Factura:** es o son la(s) factura(s) que tienen pendiente un o unos determinado(s) cliente(s), a la cual se le realizan abonos.



- **Saldo Anterior:** es un dato obligatorio que el sistema presenta automáticamente, hace referencia al saldo que el cliente tiene en conceptos de los créditos o abonos efectuados anteriormente.
- **Monto:** es un dato obligatorio que el sistema presenta automáticamente. Este campo representa las cuotas que el cliente esta en su obligación de efectuar en concepto de crédito, que la empresa le ha otorgado.
- **Fecha de pago:** es un dato obligatorio, ésta lo genera automáticamente el sistema, este depende de la “Forma de Pago y Primera fecha de pago” que están en la entidad “factcredito”.
- **Fecha que pagó:** es generado por el sistema, pero está en dependencia del día que abono, que puede ser la fecha de pago o ya sea antes o después de la fecha limite del abono.
- **Días de retraso:** es generado por el sistema, pero está en dependencia de la “Fecha que Pagó y la fecha de pago.”
- **Mora:** es generado por el sistema, pero depende de dos campos que son “**Días de Retraso e Interés por Mora**”.
- **Saldo Actual:** es generado por el sistema automáticamente, donde este se obtiene del **Saldo Anterior** menos **Monto**.
- **Cantidad Pagar:** es generado por el sistema automáticamente, donde éste se obtiene del **Monto** más **Mora**. Cabe señalar que esto sólo se da si hay **Mora**, si no es únicamente el **Monto**.

6.1.13.1.3. **Proceso:** se muestra la interfaz a través de la cual se puede introducir solamente el interés que se aplica en caso de retrasos de pago, el sistema genera el resto de información correspondiente al recibo emitido.

6.1.13.1.4. **Salida:** se añadirá un nuevo registro en la entidad “recibo”.

6.1. 14. Nombre de la función: IMPRIMI O VISUALIZAR_ RECIBO.



6.1.14.1. **Especificación:**

6.1.14.1.1. **Introducción:** esta función permite que el usuario visualice o imprima el registros del cliente al cual le están emitiendo el recibo”.

6.1.14.1.2. **Entradas:**

Por pantalla:

Se muestran los campos correspondientes a la entidad “recibo” que se visualicen o impriman para un determinado cliente.

Por el sistema:

Esta función no tendrá entradas por el sistema.

6.1.14.1.3. **Proceso:** se muestra la interfaz a través de la cual el usuario podrá visualizar o imprimir los datos del recibo perteneciente a un cliente.

6.1.14.1.4. **Salida:** el usuario podrá visualizar en hoja su abono correspondiente a un determinado crédito.



6.1.15. Nombre de la función: **NOTA _ DÉBITO.**

6.1.15.1. **Especificación:**

6.1.15.1.1. **Introducción:** esta función permite al usuario registrar en el sistema una nota de debito dentro de la base de datos en concepto de un mal movimiento de una determinada factura (registrar más de lo que el cliente debía a abonar), perteneciendo esta a la entidad “notadebito”,

6.1.15.1.2. **Entradas:**

Por pantalla:

- **Numero nota debito:** representa el identificador único de una nota de debito.
- **Monto del crédito:** es un dato obligatorio, debe de *existir, de lo contrario se presenta un mensaje de que “se introduzca el monto”.Este campo representa la cantidad por la que se esta efectuando el crédito.
- **Descripción:** No es un dato obligatorio, puede o no especificarse. Este campo representa una descripción breve del por que se dio el crédito.

Por el sistema:

- **Nombre del Cliente:**, nos muestra todos los clientes que están registrado en la entidad “cliente” pendientes con facturas de créditos.
- **Numero Factura:**. Representa la identificación única de la factura a cual se le esta corrigiendo el mal movimiento
- **Fecha:** Este campo representa la fecha en la que se está corrigiendo el mal movimiento.



- **Saldo anterior:** Este campo le indicará el saldo anterior antes de haber corregido el mal movimiento
- **Monto:** Este campo indica el monto que se le asignara a la nota de debito
- **Saldo Actual:** es el saldo después de haber corregido el mal movimiento.

6.1.15.1.3. **Proceso:** se muestra la interfaz a través de la cual se podrá introducir los datos referentes a una nota de debito, efectuada a una determinada factura.

6.1.15.1.4. **Salida:** se añade un nuevo registro en la entidad “notadebito”.

6.1. 16. Nombre de la función: NOTA _ CRÉDITO.

6.1.16.1. Especificación:

6.1.16.1.1. **Introducción:** esta función permite al usuario ingresar al sistema los datos correspondientes a una nota de crédito en concepto de un mal movimiento de un determinado recibo (registrar menos de lo que el cliente debía a abonar), perteneciendo esta a la entidad “notacredito”,

6.1.16.1.2. Entradas:

Por pantalla

- **Monto del crédito:** es un dato obligatorio, debe de *existir, de lo contrario se presenta un mensaje de que “se introduzca el monto”.Este campo representa la cantidad por la que se esta efectuando la nota de crédito.



- **Numero nota de credito:** representa el identificador único de una nota de credito
- **Descripción:** No es un dato obligatorio, puede o no especificarse. Este campo representa una descripción breve del por que se dio la nota de crédito.

Por el sistema:

- **Nombre del Cliente:** nos muestra todos los clientes que están registrando en la entidad “cliente” pendientes con facturas de créditos.
- **Numero Factura:** Representa la identificación única de la factura a la cual se le esta corrigiendo el mal movimiento
- **Numero de recibo:** Representa el identificador único del recibo al cual se le esta corrigiendo el error.
- **Fecha:** Este campo representa la fecha en la que se está corrigiendo el mal movimiento.
- **Saldo anterior:** Este campo le indica el saldo anterior antes de haber corregido el mal movimiento
- **Monto:** Este campo indica el monto que se le asigna a la nota de crédito
- **Saldo Actual:** es el saldo actual después de haber corregido el mal movimiento.

6.1.16.1.3. **Proceso:** se muestra la interfaz a través de la cual se puede introducir los datos referentes a una nota de crédito, efectuado a un determinado recibo e indirectamente a una factura.

6.1.16.1.4. **Salida:** se añade un nuevo registro en la entidad “notacredito”.

6.1. 17. Nombre de la función: Consulta_Estado _ Cta_Cliente.



6.1.17.1. Especificación:

6.1.17.1.1. **Introducción:** la función muestra al usuario el estado de cuenta por cliente ó en general, esto se hace a través de una consulta SQL a la base de datos, esta función es llamada desde la opción Consulta_Estado_Cta_Cliente a través de una interfaz donde nos aparece la opción de consultar por:

- Cliente.
- En General.

6.1.17.1.2. Entradas:

Si la opción es por cliente nos muestra que cliente es el queremos examinar y daremos aceptar en la ventana

Por pantalla:

- **Código Cliente:**
- **Nombre del Cliente:**

Por el sistema:

El sistema hará uso de las siguientes consultas SQL:

```
"SELECT factcredito.numfactura, cuenta.codcta, cuenta" _  
& ".nombrecta, factcredito.descripcion, factcredito.fecha, factcr" _  
& "edito.montocredito, factcredito.saldo FROM cuenta, factcredito" _  
& "WHERE (cuenta.codcta = factcredito.codcta) AND (factcredito.c" _  
& "odclte="" & strCodigo & "")"
```

```
"SELECT recibo.numfactura, recibo.numrecibo As NoRecibo" _
```



&", recibo.fecha as Fecha, recibo.montabono As Abono, recibo.saldo" _
& ", recibo.interesxmora As InteresMora, recibo.mora, recibo.cantpa" _ &
"gar As Total FROM factcredito, recibo WHERE (factcredito.numfact" _
& "ura=recibo.numfactura) AND (factcredito.codclte=" & strCodigo _
& ")"

6.1.17.1.3.**Proceso:** se muestra la interfaz a través de la cual se puede visualizar la información referente al estado de cuenta de un determinado cliente, todo esto hace uso de una consulta SQL correspondiente, la cual toma como parámetro el nombre del cliente,

6.1.17.1.4.**Salida:** se muestra el resultado de la consulta efectuada a la base de datos, la cual proporciona el Código del cliente, Nombre del cliente que pertenece a la entidad "cliente", Número de factura, Nombre de la cuenta, Descripción, Fecha, Monto, Saldo que corresponde a la entidad "factcredito", salto total de los montos de las facturas. Numero de recibo, Fecha, Abono, saldo, Interés por mora, Mora y Total que vincula a la entidad recibo.

6.1.18. Nombre de la función: Consulta_Estado_Cta_General

6.1.18.1. Especificación:

6.1.18.1.1. **Introducción:** esta función muestra al usuario el estado de cuenta por cliente ó en general esto se hace a través de una consulta SQL a la base de datos, esta función es llamada desde la consulta Consulta_Estado_Cta_General a través de una ventana donde nos aparece la opción de consultar por:

- CLIENTE.
- EN GENERAL.

6.1.18.1.2. Entradas:



Si la opción es en general nos muestra todos los clientes a los que se le han generados facturas con sus respectivos recibos

Por pantalla:

Por el sistema:

El sistema hace uso de las siguientes consultas SQL:

```
"SELECT DISTINCT cliente.codclte As Codigo, " _  
& "cliente.nombreclte As Nombres, cliente.apellido As Ap" _  
& "ellidos FROM cliente, factcredito WHERE cliente.codcl" _  
& "te=factcredito.codclte AND factcredito.cancelado = 'N'"
```

```
"SELECT factcredito.codclte, factcredito.numfactura" _  
& ", cuenta.nombrecta, factcredito.fecha, factcredito.siguiete" _  
& "pago, factcredito.montocredito, factcredito.saldo FROM cuent" _  
& "a, factcredito WHERE (cuenta.codcta=factcredito.codcta) AND " _  
& "(factcredito.cancelado='N')"
```

```
"SELECT recibo.numfactura, recibo.numrecibo, recibo." _  
& "fecha, recibo.montabono, recibo.saldo, recibo.interesxmora, " _  
& "recibo.mora, recibo.cantpagar FROM recibo, factcredito WHERE " _ &  
"(factcredito.numfactura=recibo.numfactura) AND (factcredito." _  
& "cancelado='N') ORDER BY recibo.fecha ASC, recibo.saldo DESC"
```

6.1.18.1.3.**Proceso:** se muestra la interfaz a través de la cual se podrá visualizar la información referente al estado de cuenta de los clientes de todas las cuentas perteneciente a “cuenta por cobrar”, todo esto se hace por medio de una consulta SQL correspondiente.



6.1.18.1.4. **Salida:** se muestra el resultado de la consulta efectuada a la base de datos, la cual proporciona el Código , Nombre y Apellido de los clientes, que pertenece a la entidad “cliente”, Número de factura, Nombre de la cuenta, Fecha, Próximo pagó, Monto, Saldo que corresponde a la entidad “factcredito”. Numero de recibo, Fecha, Abono, saldo, Interés por mora, Mora, Total que vincula a la entidad “recibo”.

6.1.19. Nombre de la función: Consultar_Antigüedad_Saldo.

6.1.19.1. Especificación:

6.1.19.1.1. **Introducción:** Esta función muestra al usuario la antigüedad de saldo por cuentas, cuentas pertenecientes a “cuenta por cobrar” con sus respectivos clientes emitidos dentro de un rango de fecha, seleccionando el nombre de la cuenta a consultar, registrados en el sistema de la base de datos, haciéndose esto a través de una consulta SQL, esta función se llamada desde la función “Consultar_Antigüedad_Saldo”.

6.1.19.1.2. Entradas:

Por pantalla

- **Desde:** este campo representa el inicio del rango de fecha en el cual se han registrado las antigüedades de saldos.
- **Hasta:** este campo representa el fin del rango de fecha en la cual se han registrado las antigüedades de saldo saldos.
- **Nombre de la Cuenta:** son los tipos de cuentas que se encuentran en la entidad “cuenta”.

Por el sistema:



El sistema hace uso de las siguientes consultas SQL:

```
"SELECT * FROM viwantiguedad WHERE codcta=" & strCodCuenta _  
& " "& "AND fechavencimiento Between " & Format (dtpDesde.Value _  
, strFormat) & " " AND " & Format (dtpHasta.Value, strFormat) & "" _  
, cnn, adOpenForwardOnly, adLockOptimistic
```

Viwantiguedad

```
Select `cuentacobrar`.`cliente`.`codclte` AS `codclte`,  
concat (`cuentacobrar`.`cliente`.`apellido`,_latin1',  
`cuentacobrar`.`cliente`.`nombreclte`) AS `Nombre`,  
`cuentacobrar`.`factcredito`.`codcta` AS `codcta`,  
`cuentacobrar`.`factcredito`.`saldo` AS `saldo`,  
  
`cuentacobrar`.`factcredito`.`fechavencimiento`  
AS `fechavencimiento` from (`cliente` join `factcredito`)  
where ((`cuentacobrar`.`cliente`.`codclte` = `cuentacobrar`.  
`factcredito`.`codclte`) and (`cuentacobrar`.`factcredito`.`saldo` > 0));
```

6.1.19.1.3. **Proceso:** se muestra la interfaz a través de la cual se puede visualizar la información referente al consolidado o total de los saldos de cada una de las cuentas perteneciente a “cuenta por cobrar” con sus respectivos clientes, todo esto se realiza por medio de la consulta SQL correspondiente, la cual toma como parámetro: la fecha de inicio y la fecha de fin.

6.1.19.1.4. **Salida:** se muestra el resultado de la consulta efectuada a la base de datos , la cual proporciona al nombre de la cuenta, el saldo actual, los días que al cliente se le va registrando en concepto de morosidad de pago del saldo pendiente , este va estar dado mensualmente (Ej.: 1 a 30, 31 a 60, 61 a 90, 91 a mas) ubicándose debajo de cualquiera de estos periodos el saldo por el que el cliente



esta **moroso de pago**, esto significa que donde se ubique dicho saldo será el tiempo que el cliente tiene de morosidad y por ultimo está el total de saldo a cobrar de todos los clientes morosos.

6.1.20. Nombre de la función: Consultar_Saldos y Consolidado Saldo.

6.1.20.1. Especificación:

6.1.20.1.1. Introducción: esta función muestra al usuario los saldos de cada uno de los clientes correspondientes a una cuenta determinada de "cuenta por cobrar", registradas en el sistema de la base de datos, a la misma vez muestra el consolidado de saldos de cada una de las cuentas, haciéndose esto a través de una consulta SQL, esta función es llamada desde la función Consultar_Saldos y consolidado Saldos

6.1.20.1.2. Entradas:

Por pantalla:

- **Desde:** este campo representa el inicio del rango de fecha en el cual se han registrado los saldos y consolidado de saldos de los clientes y las cuentas.
- **Hasta:** este campo representa el fin del rango de fecha en la cual se han registrado los saldos y consolidados de saldos de los clientes y las cuentas.

Por el sistema:

El sistema hace uso de las siguientes consultas SQL:

```
"SELECT cliente.codclte, CONCAT (cliente.apellido, ' ', " _  
& "cliente.nombreclte) AS NCliente, cuenta.nombrecta, SUM (f" _
```



```
& "actcredito.saldo) As Saldo FROM cliente, cuenta, factcre" _  
& "dito WHERE (cliente.codclte=factcredito.codclte) AND (fa" _  
& "ctcredito.codcta=cuenta.codcta) AND (factcredito.fecha B" _  
& "etween "" & Format (dtpDesde.Value, strFormat) & "" AND "" _  
& Format (dtpHasta.Value, strFormat) & "") AND (factcredito." _  
& "saldo>0) GROUP BY cliente.codclte, NCliente, cuenta.nomb" _  
& "recta;", cnn, adOpenForwardOnly, adLockOptimistic
```

6.1.20.1.3. **Proceso:** se muestra la interfaz a través de la cual se puede visualizar la información referente a los saldos y consolidados de saldos de cada uno de los clientes de una determinada cuenta perteneciente a “cuenta por cobrar”, todo esto se realiza haciendo uso de la consulta SQL correspondiente, la cual toma como parámetro: la fecha de inicio y la fecha de fin.

6.1.20.1.4. **Salida:** se muestra el resultado de la consulta efectuada a la base de datos, la cual proporciona: código, el nombre del cliente, el saldo actual y el saldo total de todos los clientes pertenecientes a las diferentes cuentas., a si como también el consolidado de saldos de las cuentas

6.1.21. Nombre de la función: Reporte_Est_Cta_Cliente.

6.1.21.1. Especificación:

6.1.21.1.1. **Introducción:** esta función tiene como objetivo generar el reporte referente al estado de cuenta de un determinado cliente registrados en el sistema. Esto se hace mediante el uso de consultas SQL .A esta función le da soporte las entidades cliente, cuenta, factcredito, recibos.

6.1.21.1.2. Entradas:



Por pantalla:

- **Por cliente:** es una opción obligatoria si se desea saber el estado de cuenta de un cliente determinado.
- **Nombre Cliente:** Es un dato obligatorio, el cual representa el nombre del cliente que se le emitirá el reporte.

Por el sistema:

El sistema hará uso de las siguientes consultas SQL:

```
"SELECT * FROM viwestcxcxcliente WHERE " _  
& "codclte=" & tpCliente.strCodigo & " ORDER BY "& "numfactura, saldo  
DESC;"
```

viewestcxcxcliente:

```
Select `cuentacobrar`.`cliente`.`codclte` AS `codclte`,  
concat (`cuentacobrar`.`cliente`.`nombrecte`,_latin1',  
`cuentacobrar`.`cliente`.`apellido`) AS `Nombre`,`cuentacobrar`.`  
`cliente`.`ciudad` AS `ciudad`,`cuentacobrar`.`cliente`.`direccion`  
AS `direccion`,`cuentacobrar`.`cliente`.`cedula` AS `cedula`,`cuentacobrar`.`  
`cliente`.`telfcasa` AS `telfcasa`,`cuentacobrar`.`factcredito`.`numfactura`  
AS `numfactura`,`cuentacobrar`.`cuenta`.`nombrecta` AS  
`nombrecta`,`cuentacobrar`.`factcredito`.`fecha` AS  
`FechaCredito`,`cuentacobrar`.`factcredito`.`montocredito`  
AS `montocredito`,`cuentacobrar`.`factcredito`.`saldo` AS  
`saldo`,`cuentacobrar`.`recibo`.`numrecibo` AS  
`numrecibo`,`cuentacobrar`.`recibo`.`fecha` AS  
`fecha`,`cuentacobrar`.`recibo`.`montabono` AS
```



```
`montabono`,`cuentacobrar`.`recibo`.`saldo` AS  
`saldoActual`,`cuentacobrar`.`recibo`.`interesxmora` AS  
`interesxmora`,`cuentacobrar`.`recibo`.`mora` AS  
`mora`,`cuentacobrar`.`recibo`.`cantpagar` AS `cantpagar` from (((`cliente`  
join `cuenta`) join `factcredito`) join `recibo`) where  
((`cuentacobrar`.`cliente`.`codclte` = `cuentacobrar`.`factcredito`.`codclte`)  
and (`cuentacobrar`.`cuenta`.`codcta` = `cuentacobrar`.`factcredito`.`codcta`)  
and (`cuentacobrar`.`  
`factcredito`.`numfactura` = `cuentacobrar`.`recibo`.`numfactura`) and  
(`cuentacobrar`.`factcredito`.`cancelado` = _latin1'N'));
```

6.1.21.1.3. **Proceso:** se presenta el reporte conteniendo la información referente al estado de cuenta de un cliente .Esto se realiza haciendo uso de las correspondientes consultas SQL.

6.1.21.1.4. **Salida:** se presenta los siguientes datos en el reporte impreso del estado de cuenta de un cliente: nombre del cliente, ciudad , dirección , cedula, teléfono , numero de factura , nombre de la cuenta , fecha del crédito , monto , saldo , numero de recibo , fecha abono , abono , saldo actual , interés por mora , total pagar.

6.1.22. Nombre de la función: REPORTE_EST_CUENTAS_GENERAL

6.1.22.1. Especificación:

6.1.22.1.1. **Introducción:** esta función muestra un reporte del estado de cuentas de los clientes de una forma general (todos los clientes de las diferentes tipos de cuentas en las que se clasifica cuentas por cobrar) esto se hace mediante el uso de consultas SQL .A esta función le da soporte a las entidades cliente, cuenta, factcredito, abonos.



6.1.22.1.2. Entradas:

Por pantalla:

- **General (todos los clientes):** Es una opción obligatoria si se desea generar un reporte del estado de cuenta de todos los clientes en general

Por el sistema:

El sistema hace uso de las siguientes consultas SQL:

```
"SELECT * FROM viwEstCuentaGeneral ORDER BY codclte;"
```

viwEstCuentaGenera

```
Select `cuentacobrar`.`cliente`.`codclte` AS `codclte`,  
concat(`cuentacobrar`.`cliente`.`apellido`,_latin1' ',  
`cuentacobrar`.`cliente`.`nombreclte`) AS  
`Nombres`,`cuentacobrar`.`cliente`.`cedula` AS `cedula`,`cuentacobrar`.  
`cliente`.`telfcasa` AS `Telfcasa`,`cuentacobrar`.`cliente`.`ciudad` AS  
`ciudad`,`cuentacobrar`.`factcredito`.  
`numfactura` AS `factura`,`cuentacobrar`.`cuenta`.`nombrecta` AS  
`cuenta`,`cuentacobrar`.`factcredito`.`fecha`  
AS `fecha`,`cuentacobrar`.`factcredito`.`montocredito` AS  
`monto`,`cuentacobrar`.`factcredito`.`saldo` AS `saldo`  
from ((`cliente` join `cuenta` ) join `factcredito`) where  
((`cuentacobrar`.`cliente`.`codclte` = `cuentacobrar`.  
`factcredito`.`codclte`) and (`cuentacobrar`.`factcredito`.`codcta` =  
`cuentacobrar`.`cuenta`.`codcta`) and
```



`(`cuentacobrar`.`factcredito`.`cancelado` = _latin1'N'));`

6.1.22.1.3. **Proceso:** se presenta el reporte referente al estado de cuenta general de los clientes según la cuenta seleccionada por el usuario. Esto se hace mediante una consulta SQL.

6.1.22.1.4. **Salida:** se presentan los siguientes datos en el reporte impreso del estado de cuenta general: nombre del cliente, cedula, teléfono, ciudad, número de factura, nombre de la cuenta a la que pertenece, fecha que se emitió la factura, monto de ese credito, saldo , saldo total del cliente y el saldo total de todos los clientes.

6.1.23. Nombre de la función: Reporte_ Saldos Consolidado Saldo .

6.1.23. 1.Especificación:

6.1.23.1.1. **Introducción:** esta función muestra un reporte de los saldos de cada uno de los clientes y consolidado de saldos de las diferentes cuentas que pertenecen a cuenta por cobrar, registradas en el sistema de la base de datos, esto se hace mediante el uso de consultas SQL .A esta función le da soporte las entidades cliente, cuenta, factcredito, recibo.

6.1.23.1.2. Entradas:

Por pantalla:

- **Fecha de**”.Este campo representa el inicio del rango de fecha en el cual el usuario desea generar un reporte de saldos y consolidados de saldos de los clientes.



- **Fecha hasta:** Este campo representa el fin del rango de fecha en la cual el usuario desea generar un reporte de saldos y consolidados de saldos de los clientes.

Por el sistema:

El sistema hace uso de las siguientes consultas SQL:

```
"SELECT * FROM viwsaldosconsolidados WHERE fecha Between '" _  
& Format (dtDesde, strFormat) & "' AND '" & Format (dtHasta, strFormat) _  
& ORDER BY nombrecta"
```

viwsaldosconsolidados

```
select `cuentacobrar`.`cuenta`.`codcta`  
AS `codcta`, `cuentacobrar`.`cuenta`.`nombrecta` AS  
`nombrecta`, `cuentacobrar`.`cliente`.`codclte` AS  
`codclte`, concat(`cuentacobrar`.`cliente`.`apellido`, _latin1'  
' , `cuentacobrar`.`cliente`.`nombrecte`) AS `Nombres`,  
`cuentacobrar`.`factcredito`.`fecha`  
AS `fecha`, `cuentacobrar`.`factcredito`.`montocredito`  
AS `Monto`, `cuentacobrar`.`factcredito`.`saldo` AS `saldo` from ((`cliente` join  
`cuenta`) join `factcredito`) where ((`cuentacobrar`.`cliente`.`codclte` =  
`cuentacobrar`.`factcredito`.`codclte`) and  
(`cuentacobrar`.`factcredito`.`codcta` = `cuentacobrar`.`cuenta`.`codcta`) and  
(`cuentacobrar`.`factcredito`.`cancelado` = _latin1'N')) order by  
`cuentacobrar`.`cuenta`.`nombrecta`;
```

6.1.23.1.3. **Proceso:** se presenta el reporte conteniendo la información referente a los saldos y consolidados de saldos de cada una de las cuentas pertenecientes a "cuenta por cobrar" con sus respectivos clientes emitidos dentro de un rango de fecha a todo esto se hace haciendo uso de la consulta SQL.



6.1.23.1.4. **Salida:** se presentan los siguientes datos referente a los saldos de los clientes de cada una de las cuenta perteneciente a “cuenta por cobrar” codigo de la cuenta , codigo del cliente , nombre de la cuenta , nombre del cliente , monto , saldo actual del cliente , saldo total de cada cuenta y saldo total de todas las cuentas.

6.1.24. Nombre de la función: Reporte_ Antigüedad_Saldo

6.1.24.1. Especificación:

6.1.24.1.1. **Introducción:** esta función muestra al usuario la antigüedad de saldo de los clientes por cuentas, cuentas pertenecientes a “cuenta por cobrar” emitidos dentro de un rango de fecha e introduciendo el nombre de la cuenta a consultar, registrados en el sistema de la base de dato, esto se hace mediante el uso de consultas SQL .A esta función le dan soporte las entidades: clientes, cuenta, factcredito, recibo.

6.1.24.1.2. Entradas:

Por pantalla:

- **Fecha de:** Este campo representa el inicio del rango de fecha en el cual se desean emitir el reporte de las antigüedades de saldos.
- **Fecha hasta:** Este campo representa el fin del rango de fecha en el cual se desean emitir el reporte de las antigüedades de saldos
- **Cuenta:** es un dato obligatorio, debe de existir, de lo contrario se presentara un mensaje de que “Seleccione un nombre de cuenta”. Este campo nos indicará cuales de las cuentas pertenecientes a cuenta por cobrar el usuario desea generar el reporte.



Por el sistema:

El sistema hace uso de las siguientes consultas SQL:

```
select `cuentacobrar`.`cliente`.`codclte` AS `codclte`,  
concat(`cuentacobrar`.`cliente`.`apellido`,_latin1' ',`cuentacobrar`.  
`cliente`.`nombrecfte`) AS `Nombre`,`cuentacobrar`.`factcredito`.`codcta` AS  
`codcta`,`cuentacobrar`.`factcredito`.`saldo`  
AS `saldo`,`cuentacobrar`.`factcredito`.`fechavencimiento` AS  
`fechavencimiento` from (`cliente` join `factcredito`) where  
  
((`cuentacobrar`.`cliente`.`codclte` = `cuentacobrar`.`factcredito`.`codclte`)  
and (`cuentacobrar`.`factcredito`.`saldo` > 0));
```

6.1.24.1.3. **Proceso:** se presenta el reporte referente a la antigüedad de saldos de cada una de las cuentas según el nombre introducido. Todas estas pertenecientes a “cuenta por cobrar” con sus respectivos clientes emitidos dentro de un rango de fecha, esto se hace haciendo uso de la consulta SQL.

6.1.24.1.4. **Salida:** se presentan los siguientes datos referente al reporte impreso de la antigüedad de saldos por cuentas(según el código introducido de la cuenta) : el código de la cuenta y del cliente, el nombre del cliente y de las cuentas , el saldo actual, los días que al cliente se le va registrando en concepto de morosidad de pago del saldo pendiente , este va estar dado mensualmente (Ej: 1 a 30, 31 a 60, 61 a 90, 91 a mas) ubicándose debajo de cualquiera de estos periodos , el saldo por el que el cliente esta **moroso** de pago, esto significa que donde se ubique dicho saldo será el tiempo que el cliente tiene de morosidad ,el total de saldo de cada periodo de morosidad , el total de saldo a cobrar de todos los clientes morosos.



6.1.25. Nombre de la función: Ingresar _ Sistema.

6.1.25.1. Especificación:

6.1.25.1.1. Introducción: Esta función permite al usuario ingresar al sistema, siempre y cuando el usuario escriba un ID de usuario y la contraseña asociado a este, y que además estos se encuentren registrados en el Sistema (que sean validos).

6.1.25.1.2. Entradas:

Por pantalla:

- **Usuario:** representa el ID de cualquier usuario que sea valido en el sistema.
- **Contraseña:** representa la contraseña asociada al usuario para poder ingresar al sistema.

Por el sistema:

No se presenta ninguna entrada por parte del sistema.

6.1.25.1.3. Proceso: se presenta al usuario la interfaz para introducir el nombre de usuario y su contraseña , verificando si dichos datos son validos en el sistema y de ser así el usuario puede ingresar , de lo contrario no puede hacerlo.

6.1.25.1.4. Salida: no se presenta ninguna salida.

6.1.26. Nombre de la función: Crear _ Usuario.

6.1.26.1. Especificación:



6.1.26.1.1. **Introducción:** esta función permite al usuario con privilegios de administrador dar de alta en el sistema un nuevo ID de Usuario y su contraseña.

6.1.26.1.2. **Entradas:**

Por pantalla:

- **Usuario:** representa el ID de usuario a registrar.
- **Contraseña:** representa la contraseña asociada al Usuario que se esta intentando dar de alta.
- **Confirmar:** representa la confirmación de la contraseña.

Por el sistema:

No se presenta ninguna entrada por parte del sistema.

6.1.26.1.3. **Proceso:** se presenta al usuario la interfaz de creación de nombre del usuario y su contraseña (del usuario que se esta intentando dar de alta), además le asigna sus correspondientes permisos que dicho usuario tiene sobre el sistema. El sistema verifica que dicho ID de Usuario no exista de lo contrario dicho proceso no puede ser completado hasta que se proponga un nuevo ID de Usuario que no exista.

6.1.26.1.4. **Salida:** se añade una nueva cuenta de usuario en el sistema.

6.1.27. Nombre de la función: Modificar _ Contraseña:

6.1.27.1. **Especificación:**

6.1.27.1.1. **Introducción:** esta función permite al usuario modificar su contraseña.

6.1.27.1.2. **Entradas:**



Por pantalla:

- **Contraseña actual:** representa la contraseña actual del usuario.
- **Nueva Contraseña:** representa la nueva contraseña del usuario.
- **Confirmar Contraseña:** permite confirmar la nueva contraseña.

Por el sistema:

No se presenta ninguna entrada por el sistema.

6.1.27.1.3. **Proceso:** se presenta al usuario la interfaz de modificación de la contraseña del usuario que está actualmente utilizando el sistema.

6.1.27.1.4. **Salida:** se registra la modificación de la su contraseña

6.1.28. Nombre de la función: Modificar Permisos.

6.1.28.1. Especificación:

6.1.28.1.1. **Introducción:** esta función permite al usuario modificar sus permisos.

6.1.28.1.2. **Entradas:**

Por pantalla:

Se presentan todos los permisos que un usuario puede tener y los permisos que el usuario actual tiene.

Por el sistema:

No se presenta ninguna entrada por el sistema.

6.1.28.1.3. **Proceso:** se presenta al usuario la interfaz de modificación de sus permisos y los demás usuarios. Cabe mencionar que si el usuario tiene todos los permisos, este puede cambiar los permisos de los demás usuarios.



6.1.28.1.4. **Salida:** se presenta la misma interfaz con los permisos establecidos.

6.1.29. Nombre de la función: Eliminar _ Usuario

6.1.29.1. Especificación:

6.1.29.1.1. **Introducción:** esta función forma parte de la opción **Crear usuario**, permite al usuario con privilegios de administrador eliminar una determinada cuenta de usuario, siempre y cuando esta exista. Cabe mencionar que para poder borrar usuarios al menos tiene que quedar registrado en el sistema un usuario con privilegios de administrador.

6.1.29.1.2. Entradas:

Por pantalla:

Seleccionar el usuario a borrar, dar click en el botón borrar

Por el sistema:

No se presenta ninguna entrada por parte del sistema.

6.1.29.1.3. **Proceso:** se presenta al usuario la interfaz **crear cuentas de usuario**, a través de la cual el usuario podrá dar de baja a una determinada cuenta existente en el sistema, seleccionando la opción borrar.

6.1.29.1.4. **Salida:** se elimina una cuenta de usuario del sistema.

6.1.30. Nombre de la función: Enviar registros al historial.



6.1.30.1. **Especificación:**

6.1.30.1.1. **Introducción:** esta función permite al usuario enviar registros al historial esto con el fin de tenerlos disponible para una consulta determinada.

6.1.30.1.2. **Entradas:**

Por pantalla:

- **Desde:** Inicio del rango de fecha en el cual se desea enviar los registros al historial
- **Hasta:** Fin del rango de fecha en el cual se desea enviar los registros al historial

Por el sistema:

No se presenta ninguna entrada por parte del sistema.

6.1.30.1.3. **Proceso:** se presenta al usuario la interfaz de envío de registros al historial.

6.1.30.1.4. **Salida:** Se almacenan los registros correspondientes en el historial

6.1.31. **Nombre de la función: Consultar historial.**

6.1.31.1. **Especificación:**

6.1.31.1.1. **Introducción:** esta función permite al usuario consultar en el historial las facturas canceladas con sus respectivos recibos.

6.1.31.1.2. **Entradas:**



Por pantalla:

- **Desde:** inicio del rango de fecha en el cual se desea buscar la factura.
- **Hasta:** fin del rango de fecha en el cual se desea buscar la factura.
- **Buscar por:** campo por el cual desea buscar la factura ya sea: por cliente o por factura.

Por el sistema:

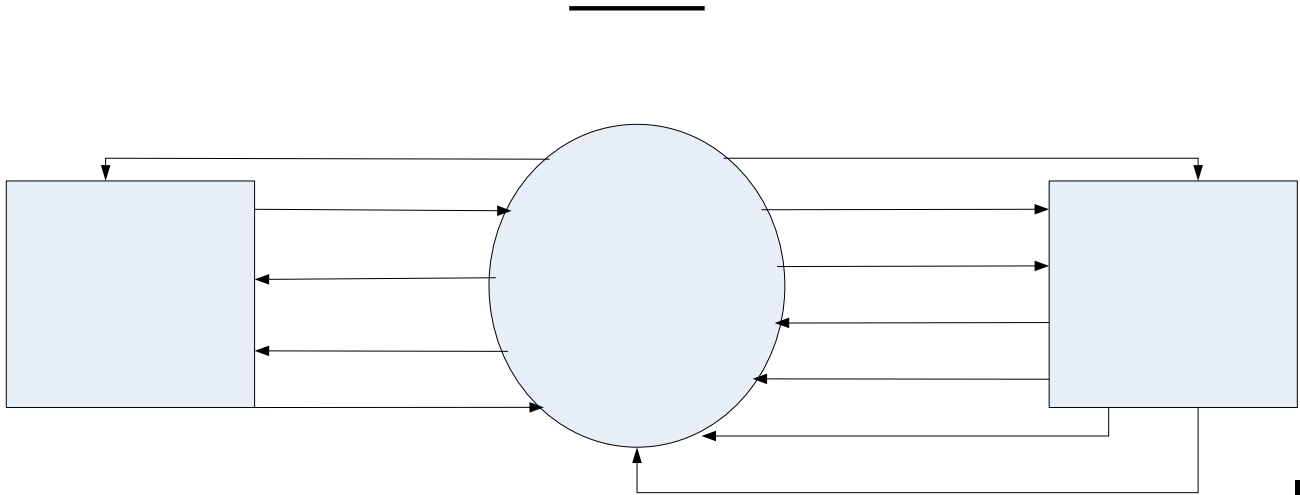
No se presenta ninguna entrada por parte del sistema.

6.1.31.1.3. **Proceso:** se presenta al usuario la interfaz en la cual se puede observar la factura buscada con sus respectivos recibos.

6.1.31.1.4. **Salida:** Se muestra la factura buscada con sus datos y recibos correspondientes.



Diagrama DFD



Nivel

Cliente

Recibo_Abono

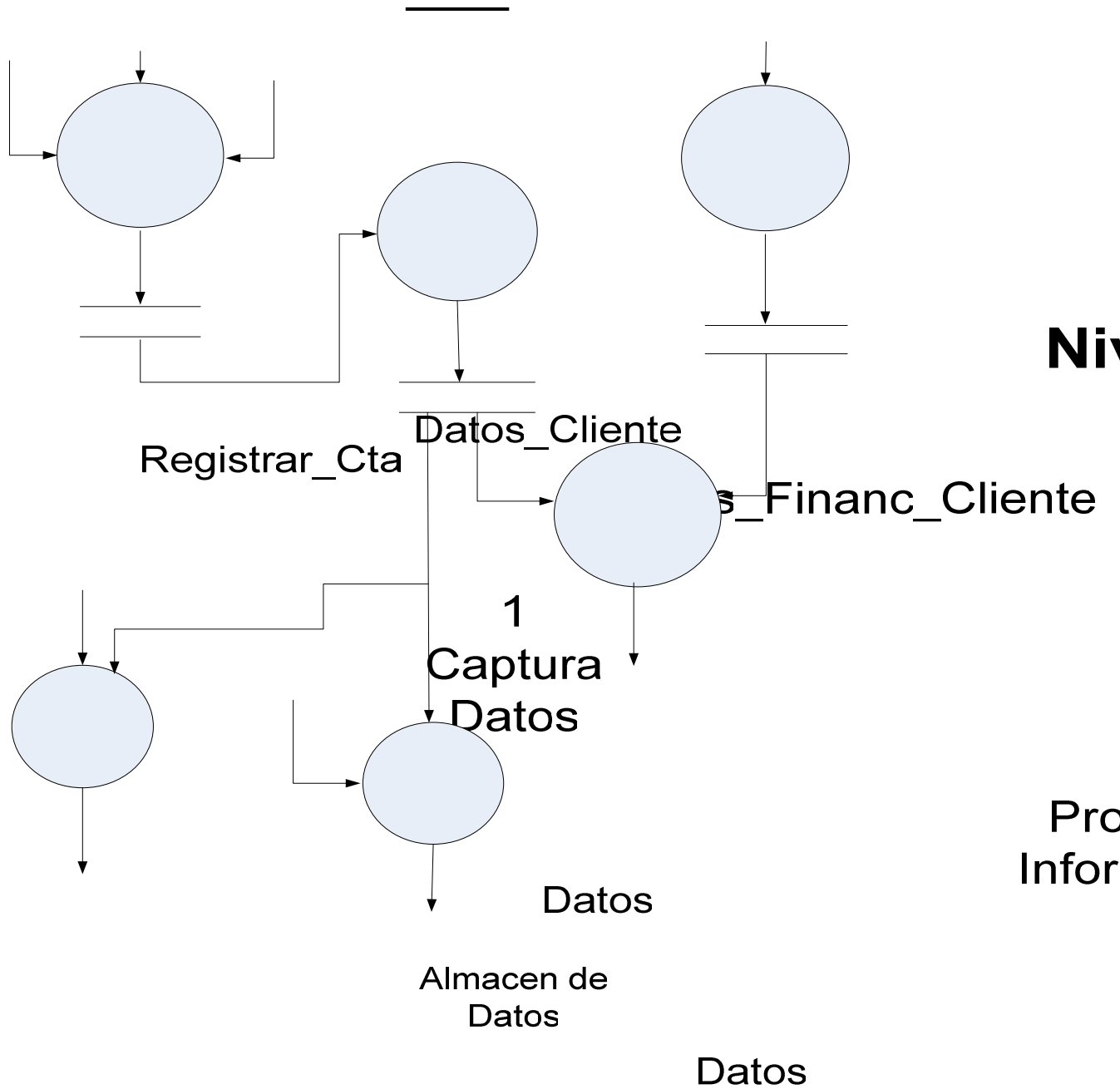
Datos_Cliente

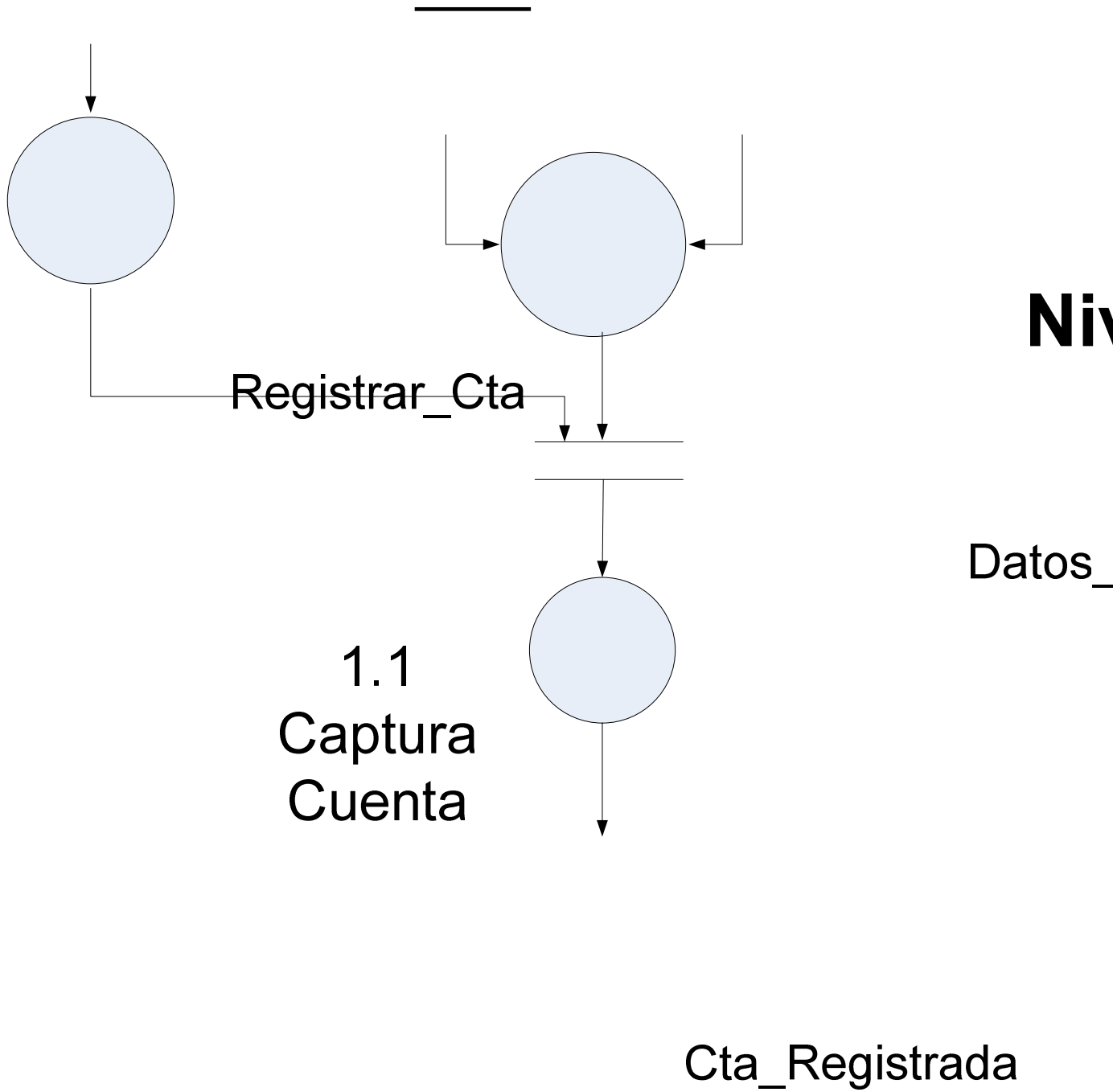
Estado_Cta_Cliente

Factura_Credito

Realizar_Abono

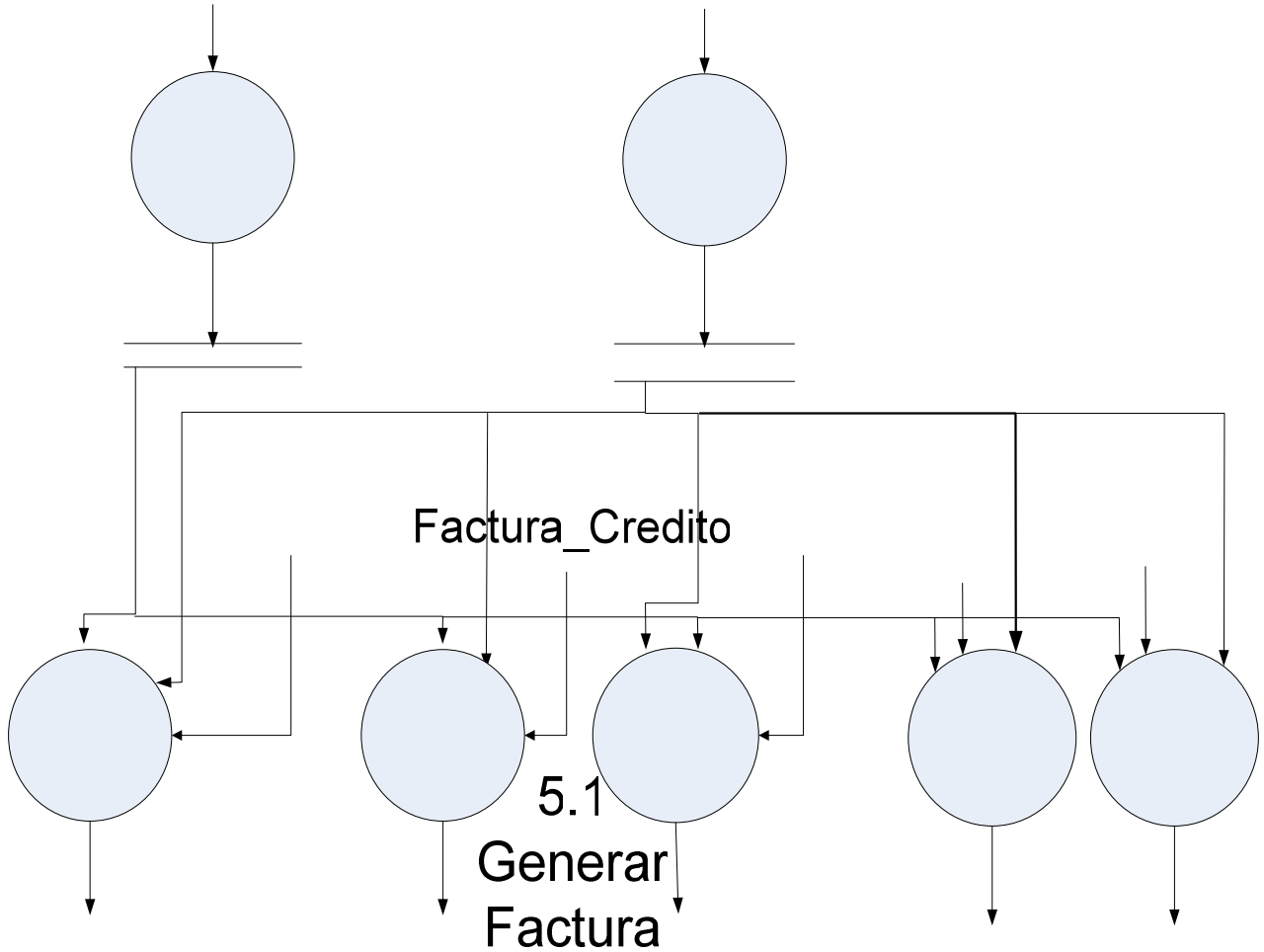
0
Gestio
De
CxC







Nivel 3



Estado

Factura

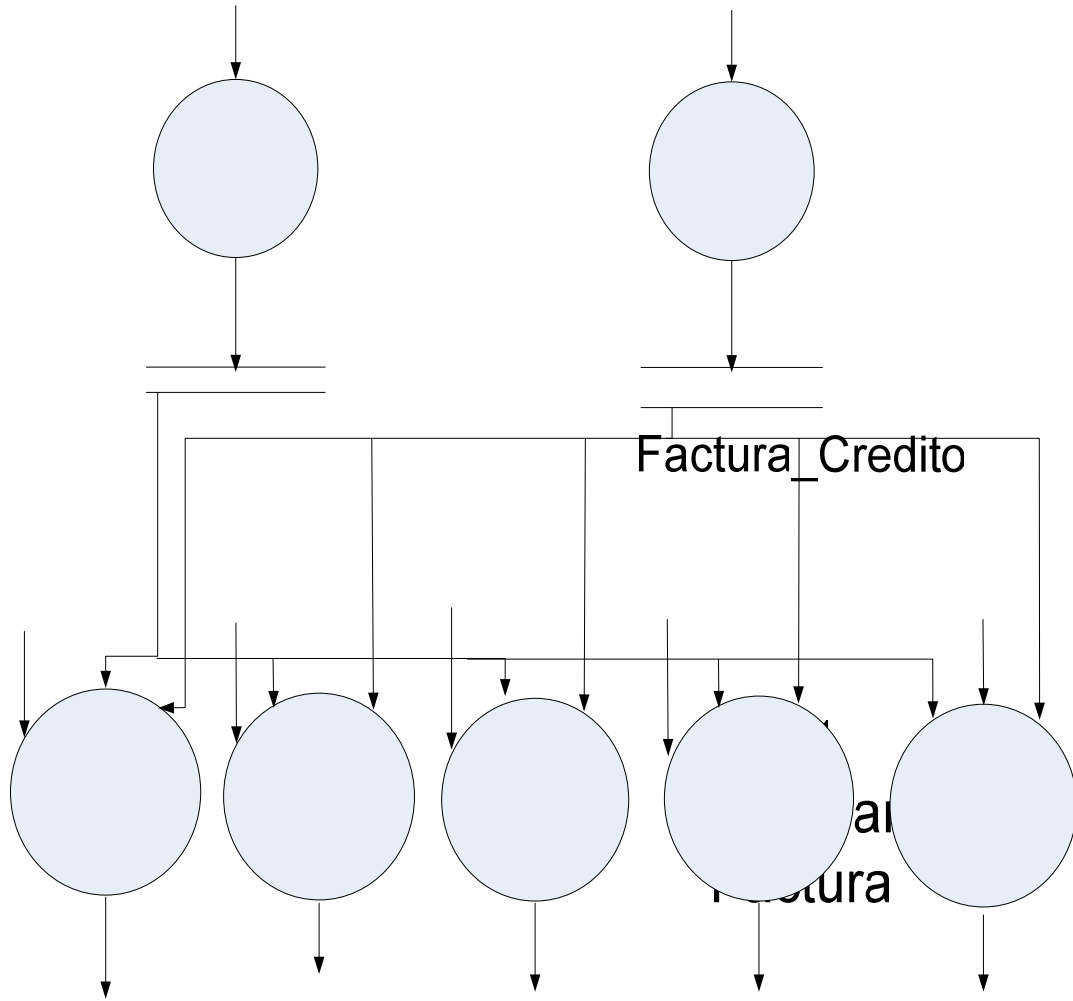
Almacen Factura

Estado_Cta_Clte

Alm
C



Nivel 4



Factura

Almacen Factura

Estado_

Factura



Diccionario de Datos.

Cada uno de los flujos de datos está desarrollado de la siguiente manera:

Cliente: ENTIDAD EXTERNA.

Administrador Sistema: Del sistema: ENTIDAD EXTERNA.

SOFTWARE CONTABLE “**CUENTAS POR COBRAR**”. : PROCESO DE DATOS.

Registrar_Cta: nombrecta + codcta. Se usa como entrada en el proceso Captura Cuenta.

Cta_Registrada: nombrecta + codcta. Se usa como salida en el proceso Captura Cuenta y como entrada en Almacén Datos Financieros y como entrada en el proceso Captura de datos

Datos _ Cliente, Datos_financ_Cliente: codclte + nombrecte + apellido + ciudad + cedula + telefcasa, celular, montinicial. Se usa como entrada en el proceso Captura Datos: codclte + cedula + montinicial + fecha + numfactura + limitcred + fechvencim + prestvencido. Se usa como estrada en el proceso Captura Datos.

Datos: Datos_Cliente + Dato_ Financ_Cliente + Registrar_Cta. Se usan como entrada en el proceso Procesar Información.

Factura_credito: numfactura + nombrecta + nombrecte + fecha + montocredito descripción + plazocredito + cantidadabonar + primera fechapago+fechavencimiento + cancelado, anulada, siguientepago + formaspagos + saldo. Se usa como salida en el proceso Procesar Información, y como entrada en los procesos: Generar Estado Cuenta, Generar Reporte y Generar Consulta.



Realizar_abono: numrecibo + nombrecite + numfactura + fechabono + descripción + montabono + interesxmora, mora, cantpagar .Se usa como entrada en el proceso Captura Abono.

Recibo_Abono: numrecibo + nombrecite + numfactura + fechabono + descripción + montabono + interesxmora, mora, cantpagar saldo.Se usa como salida en el proceso Captura Abono y como entrada en el proceso Generar Estado Cuentas.

Estado_cta_cliente: Factura_Credito + Recibo_Abono, saldo Total. Se usa como salida en el proceso Generar Estado Cuenta y como entrada en el proceso Generar Reporte y Generar Consulta.

Solicitud_Consulta: Se usa como entrada en el proceso Generar consulta; la cual se descompone en los siguientes: Solicitud Consulta Estado Cuenta Cliente, Solicitud Consulta Estado Cuenta General, Solicitud Consulta Saldo Cliente, Solicitud Consulta Consolidado Saldo, Solicitud Consulta Antigüedad Saldo. Se usa como salida)

Consulta Efectuada: Se usa como salida en el proceso Generar consulta.

Solicitud Consulta Estado Cuenta Cliente: No Factura + codclte + Nombre Cuenta + Descripción + Fecha + Monto + Saldo, No Recibo + Fecha + Abono + Saldo + InteresXMora + Mora + Total. Se usa como entrada en el proceso Generar Consulta Estado Cuenta Cliente.

Consulta_Estado_Cta Cliente: No Factura + codigo Clte + Nombre Cuenta + Descripción + Fecha + Monto + Saldo, No Recibo + Fecha + Abono + Saldo + InteresXMora + Mora + Total. Se usa como salida en el proceso Generar Consulta Estado Cuenta Cliente.



Solicitud Consulta Saldo Cliente: No Factura + código Clte + Nombre Cuenta + Descripción + Fecha + Monto + Saldo, No Recibo + Fecha + Abono + Saldo + InteresxMora + Mora + Total. Se usa como entrada en el proceso Generar Consulta Saldo Cliente.

Consulta_Saldo_Cliente: No Factura + código Clte + Nombre Cuenta + Descripción + Fecha + Monto + Saldo, No Recibo + Fecha + Abono + Saldo + InteresxMora + Mora + Total. Se usa como entrada en el proceso Generar Consulta Saldo Cliente.

Solicitud Consulta Estado Cuenta General: codclte + nombrecyte + Apellidos + No Factura + Nombre Cuenta + Fecha + Prox.Pago + Monto + Saldo No Recibo + Fecha + Abono + Saldo + InteresxMora + Mora + Total Se usa como entrada en el proceso Generar Consulta Estado Cuenta General.

Consulta_Estado_Cta_General , Solicitud Consulta Saldo Cliente: codclte + nombrecyte + Apellidos + No Factura + Nombre Cuenta + Fecha + Prox.Pago + Monto + Saldo No Recibo + Fecha + Abono + Saldo + InteresxMora + Mora + Total Se usa como salida en el proceso Generar Consulta Estado Cuenta General

Solicitud Consulta_Consolid_Saldo: Fecha desde + Fecha hasta + código cliente + Nombres Cliente + Nombres Cuentas + Saldo cada cliente + Total Saldo de Cada Cuenta + Total general. Se usa como entrada en el proceso Generar Consulta Consolidado Saldo.



Consultar Consolidado Saldo: Fecha desde + Fecha hasta + código cliente + Nombres Cliente + Nombres Cuentas + Saldo cada cliente + Total Saldo de Cada Cuenta + Total general. Se usa como salida en el Generar Consulta Consolidado Saldo.

Consulta Antigüedad Saldo: fecha desde + fecha hasta + Nombre cuenta + + código Cliente + nombre Cliente + saldo actual + 1-30 + 31-60 + 61-91 + 91 y más. Se usa como entrada en el proceso Generar Consulta Antigüedad Saldo.

Consultar Antigüedad Saldo fecha desde + fecha hasta + Nombre cuenta + + código Cliente + nombre Cliente + saldo actual + 1-30 + 31-60 + 61-91 + 91 y más. Se usa como salida en el proceso Generar Consulta Antigüedad Saldo y como entrada en proceso Generar Consulta.

Solicitud Reporte Est_Cta_Cliente: No Factura + código Clte + Nombre Cuenta + Descripción + Fecha + Monto + Saldo, No Recibo + Fecha + Abono + Saldo + InteresxMora + Mora + Total. Se usa como entrada en el proceso Generar Reporte Est_Cta_Clte

Reporte est_cta_cliente: No Factura + código Clte + Nombre Cuenta + Descripción + Fecha + Monto + Saldo, No Recibo + Fecha + Abono + Saldo + InteresxMora + Mora + Total. Se usa como salida en el proceso Generar Reporte Est_Cta_Clte y como entrada en el proceso Captura Reportes.

Solicitud Reporte Saldo: fecha desde + fecha hasta + codcta + alfabético o sucesivo + codclte + nombrecfte + saldo. Se usa como entrada en proceso Generar Rpte_saldo



Reporte_saldo: fecha desde + fecha hasta + codcta + alfabético o sucesivo + codclte + nombrecte + saldo. Se usa como salida en proceso Generar Rpte_Saldo y como entrada en el proceso Captura Reportes

Reporte Antig_Saldo: fecha desde + fecha hasta + Nombre cuenta + + codigo Cliente + nombre Cliente + saldo actual + 1-30 + 31-60 + 61-91 + 91 y mas Se usa como entrada en el proceso Generar Rpte_ Antig_Saldo.

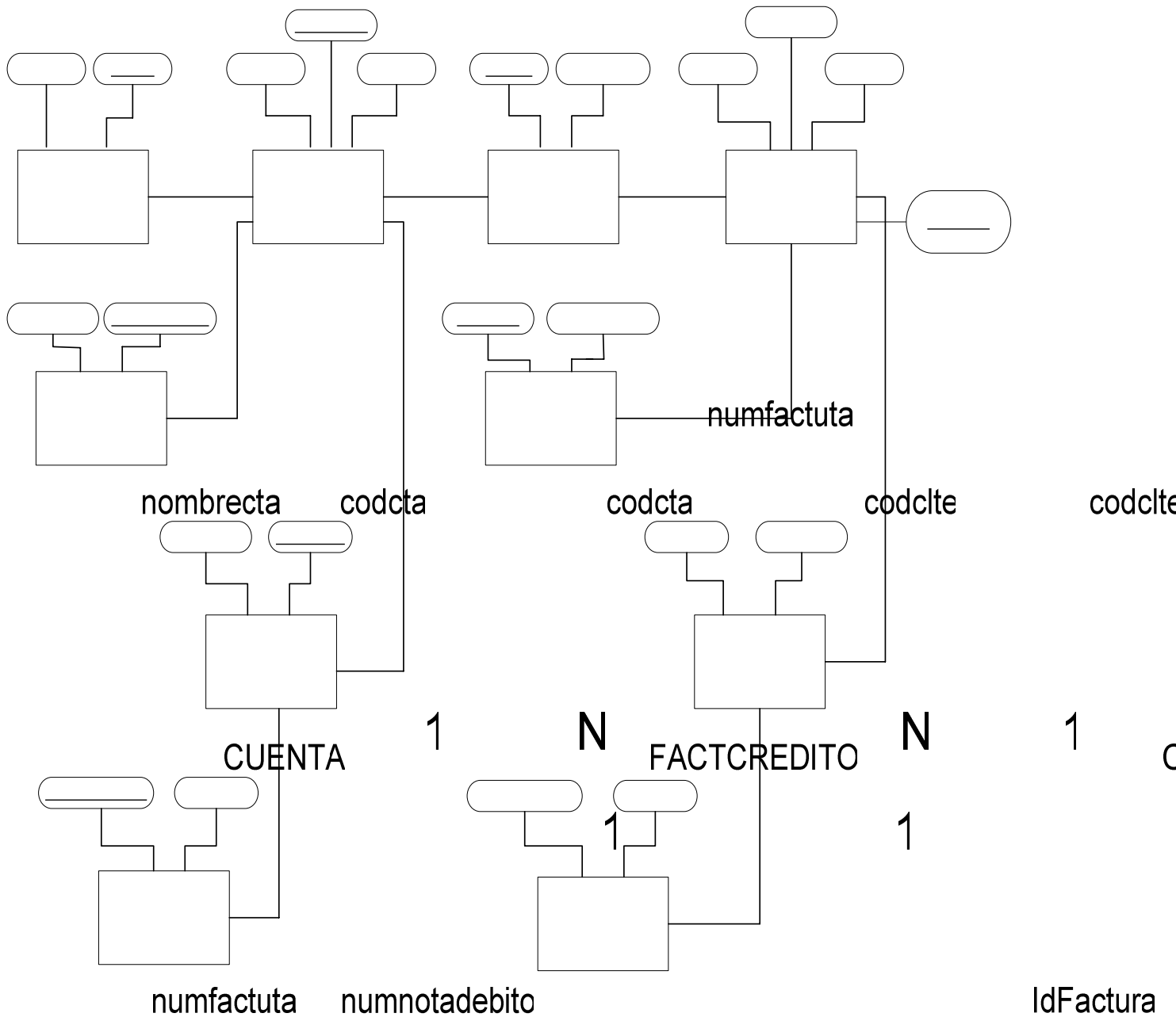
Reporte_Antig_saldo fecha desde + fecha hasta + Nombre cuenta + + codigo Cliente + nombre Cliente + saldo actual + 1-30 + 31-60 + 61-91 + 91 y mas. Se usa como salida en el proceso Generar Rpte_ Antig_Saldo y como entrada en el proceso Captura Reportes.

Reporte Consolidado_Saldo: Fecha desde + Fecha hasta + codigo cliente + Nombres Cliente + Nombres Cuentas +Saldo cada cliente + Total Saldo de Cada Cuenta + Total general. Se usa como entrada en el proceso Generar Rpte_Consolid_Saldo.

Reporte Consolidado saldo: Fecha desde + Fecha hasta + codigo cliente + Nombres Cliente + Nombres Cuentas +Saldo cada cliente + Total Saldo de Cada Cuenta + Total general .Se usa como salida del proceso Generar Rpte_Consolid_Saldo y como entrada en el proceso Captura Reportes.



DIAGRAMA ENTIDAD RELACION.





Diseño de datos.

1. Tabla Cuenta.

Nombre del Campo	Tipo de datos	Longitud	Descripción
nombrecta	varchar	50	Nombre de la cuenta contable
<u>Codcta</u>	char	9	Código de la cuenta



2. Tabla Cliente.

Nombre del Campo	Tipo de datos	Longitud	Descripción
<u>codclte</u>	char	9	Código del Cliente
nombrecite	Varchar	50	Nombre del cliente
apellido	Varchar	50	Apellido del cliente
cedula	char	30	Cedula del cliente
ciudad	Varchar	20	Ciudad del cliente
dirección	Varchar	80	Dirección del cliente
celular	char	9	Celular del cliente
telfcasa	Char	9	Teléfono del cliente
fecha	Date		Fecha de inicio del crédito
limitcredito	Float		Limite del crédito



3. Tabla factcredito

Nombre del Campo	Tipo de datos	Longitud	Descripción
codcta	char	9	Código del Cliente
Codclte	char	9	Código del cliente
<u>numfactura</u>	char	9	Numero de factura del crédito concedido
Fecha	date		Fecha del crédito
montocredito	float		El monto con el que se concede el crédito
descripción	char	70	Descripción en concepto de que se concede el crédito
cantidadabonar	float		Cantidad que abona
plazocredito	char	15	El plazo del crédito
formapagos	char	15	Forma de pagos
primerafechapago	date		Primera fecha de pago
fechavencimiento	date		Fecha de vencimiento del crédito
cancelado	char	1	Crédito cancelado
Anulada	char	1	Crédito anulado
siguientepago	date		Fecha del siguiente pago
Saldo	float		Saldo



4. Tabla factcreditohistory

Nombre del Campo	Tipo de datos	Longitud	Descripción
codcta	char	9	Código del Cliente
Codclte	char	9	Código del cliente
numfactura	char	9	Numero de factura del crédito concedido
Fecha	date		Fecha del crédito
montocredito	float		El monto con el que se concede el crédito
descripción	char	70	Descripción en concepto de que se concede el crédito
cantidadabonar	float		Cantidad que abona
plazocredito	char	15	El plazo del crédito
formapagos	char	15	Forma de pagos
primerafechapago	date		Primera fecha de pago
fechavencimiento	date		Fecha de vencimiento del crédito
cancelado	char	1	Crédito cancelado
Anulada	char	1	Crédito anulado
siguientepago	date		Fecha del siguiente pago
Saldo	float		Saldo
IdFactura	bigint	20	Identificar de factura.



5. Tabla recibo

Nombre del Campo	Tipo de datos	Longitud	Descripción
<u>numrecibo</u>	char	9	Numero de recibo
fecha	date		Fecha emisión del recibo
montabono	float		Monto del abono
descripción	varchar	70	Descripción
numfactura	char	9	Numero de factura del crédito concedido
saldo	float		Saldo
interesxmora	float		Interés por mora
mora	float		Mora aplicado al crédito
cantpagar	float		Cantidad a pagar

6. Tabla recibohistory

Nombre del Campo	Tipo de datos	Longitud	Descripción
numrecibo	Char	9	Numero de recibo
fecha	Date		Fecha emisión del recibo
montabono	Float		Monto del abono
descripción	Varchar	70	Descripción
IdFactura	Bigint	20	Identificar de factura
saldo	Float		Saldo
interesxmora	Float		Interés por mora
mora	Float		Mora aplicado al crédito
cantpagar	Float		Cantidad a pagar



7. Tabla notacredito

Nombre del Campo	Tipo de datos	Longitud	Descripción
<u>numnotacredito</u>	char	9	No. De nota de crédito
monto	float		Monto
descripción	varchar	70	Descripción
numrecibo	char	9	No. De recibo
fecha	date		Fecha de la nota de crédito

8. Tabla notacreditohistory

Nombre del Campo	Tipo de datos	Longitud	Descripción
numnotacredito	char	9	No. De nota de crédito
monto	float		Monto de la nota crédito
descripción	varchar	70	Descripción
numrecibo	char	9	No. De recibo
fecha	date		Fecha de la nota de crédito

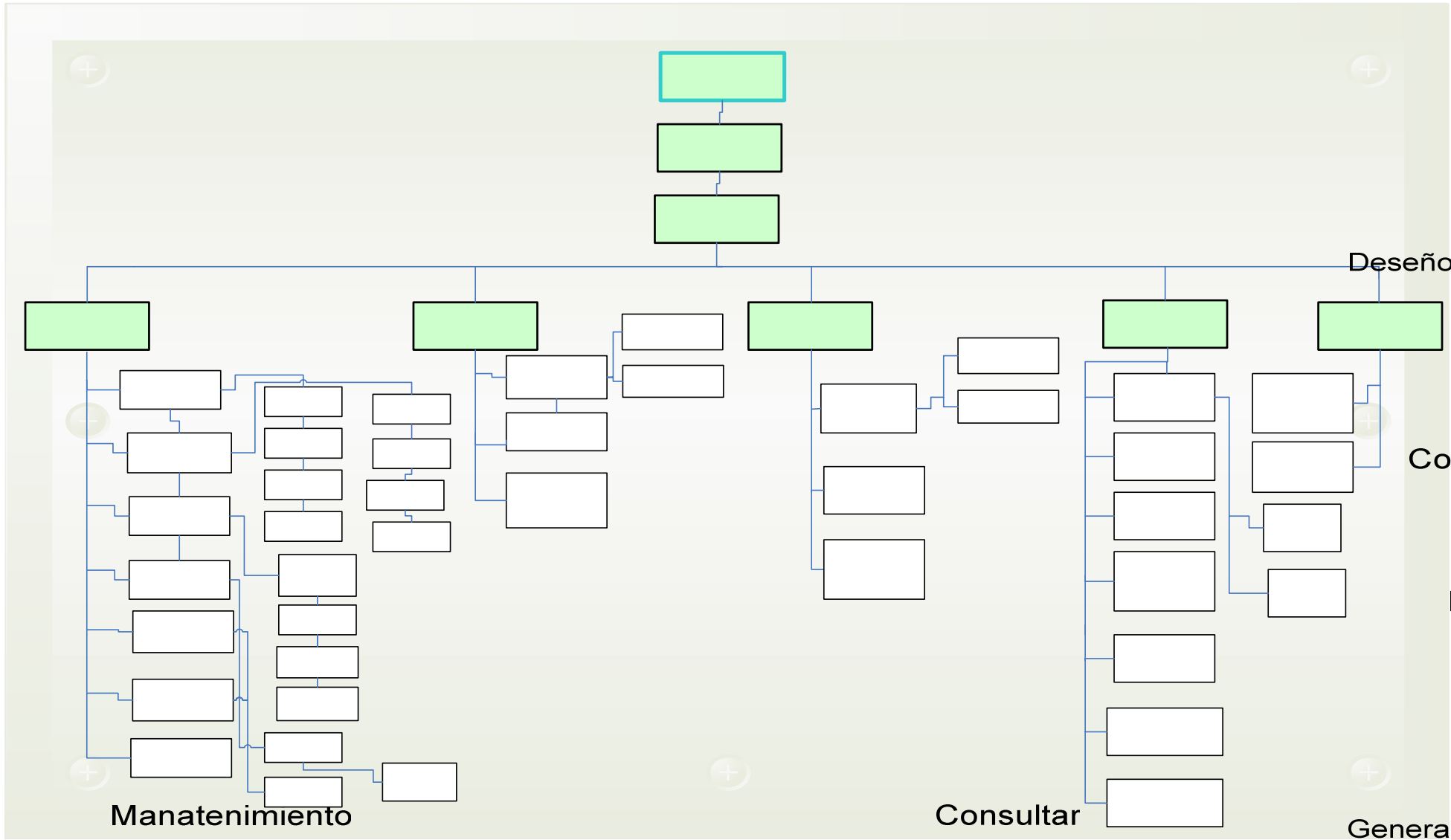


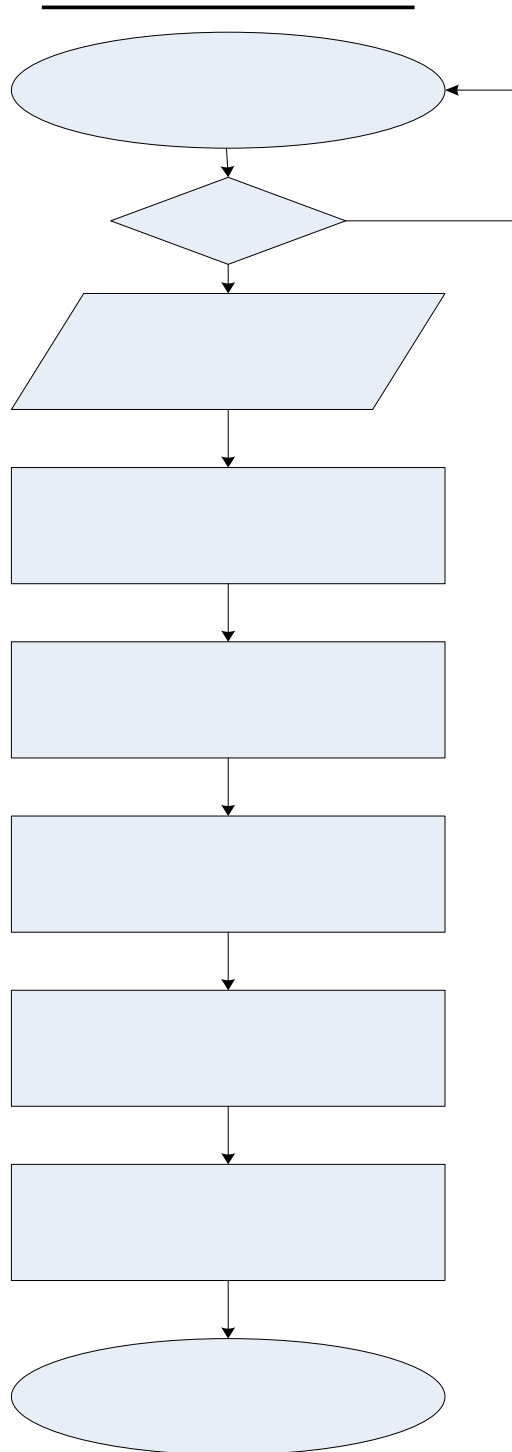
9. Tabla notadebito

Nombre del Campo	Tipo de datos	Longitud	Descripción
<u>numnotadebito</u>	char	9	No. nota de debito
monto	float		Monto
descripción	varchar	70	Descripción
numfactura	char	9	No. De factura
fecha	date		Fecha de la nota de debito

10. Tabla notadebitohistory

Nombre del Campo	Tipo de datos	Longitud	Descripción
numnotadebito	char	9	No. nota de debito
monto	float		Monto
descripción	varchar	70	Descripción
IdFactura	bigint	20	Identificar de factura
fecha	date		Fecha de la nota de debito





Diseño Preced

INICIO

Contrase

SI

**Entrar
visualizar
sistema C**

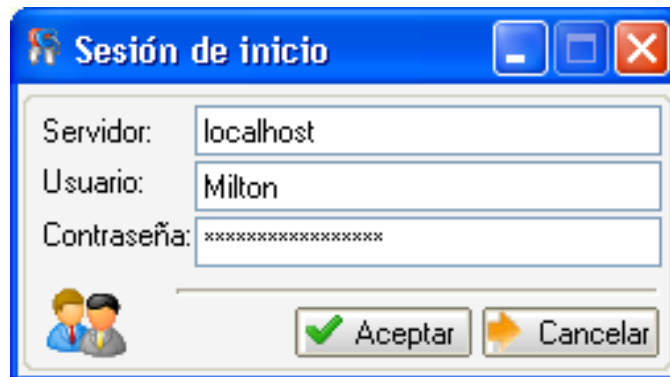
Abrir el menu p

Mostrar Opc

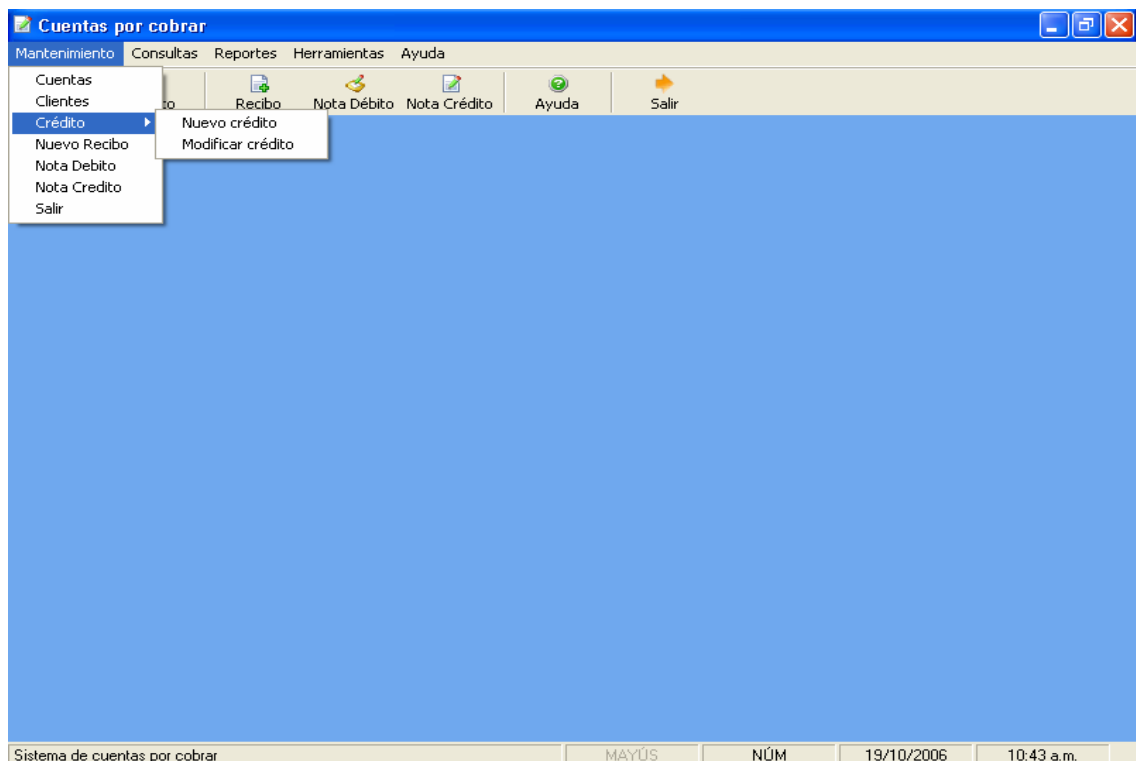


Diseño de la interfaz

A continuación se muestran algunas de las interfaces (formularios) del software. Se muestra la ventana de inicio de sesión de nuestro software contable “Sistema Automatizado de Cuentas Por Cobrar”.



Ventana principal de nuestro software, la cual cuenta con un menú en la parte superior de la misma donde el usuario tendrá opción de elegir una alternativa a utilizar por el:





Venta de captura de ingreso de catalogo de cuentas en las que se clasifica
“Cuentas Por Cobrar”

Agregar cuenta

Código de cuenta:

Nombre de la cuenta:

Ventana de captura de ingreso de clientes.

Agregar Clientes

Codigo del Cliente:

Nombres :

Apellidos :

Dirreccion:

Cedula :

Ciudad :

Celular :

Telefono Casa :

Fecha : ▼

Limite de Credito :



Ventana de captura de ingreso de factura de crédito. Esta ventana es una de las más importantes ya que registra el credito que se le concede a los clientes.

Agregar Factura

Numero Factura :	<input type="text"/>	Límite de crédito:	<input type="text" value="0.00"/>
Nombre de la Cuenta:	<input type="text"/>		
Nombre del cliente:	<input type="text"/>		
Fecha :	<input type="text" value="19/10/2006"/>		
Monto Credito :	<input type="text" value="0.00"/>		
Descripcion :	<input type="text"/>		
Plazo Credito :	<input type="text" value="15 días"/>	Imprimir	
Formas Pago:	<input type="text" value="Semanal"/>	<input checked="" type="radio"/> Factura	
Primera Fecha Pago :	<input type="text" value="19/10/2006"/>	<input type="radio"/> Calendario de pago	
Fecha Vencimiento :	<input type="text" value="19/10/2006"/>	<input type="radio"/> Ambos	
Cantidad a abonar:	<input type="text" value="0.00"/>	<input type="radio"/> Imprimir	
		<input checked="" type="radio"/> Visualizar	

Nuevo Guardar Cancelar Imprimir Cerrar



Ventana de captura de ingreso de recibos (abonos).

Agregar Recibo

Numero Recibo: Visualizar recibo Imprimir recibo

Nombre Cliente:

Nº. Factura:

Saldo anterior: Monto:

Descripcion:

Fecha de pago: Fecha que pagó:

Días de retraso: Mora:

Interes por Mora: en % Saldo actual:

Cantidad Pagar:

Ventana de captura de ingreso de Nota de Debito. Esta se emite cuando el usuario del sistema registra menos de la cantidad a abonar que el cliente depositó a su cuenta.

Agregar Nota Debito

No Nota Debito : Visualizar Imprimir

Nombre Cliente:

Nº. Factura:

Saldo anterior:

Monto:

Descripcion:


Fecha : Saldo actual:





Ventana de ingreso de Nota de Crédito. Esta se emite cuando el usuario del sistema registra más de la cantidad a abonar que el cliente depósito a su cuenta.

Agregar Nota Credito

No Nota Credito: Visualizar Imprimir

Nombre Cliente: 


No. Factura: 

No. Recibo: 

Saldo anterior:

Monto:

Descripcion:

Fecha :  Saldo actual:



Conclusiones.

Con nuestro trabajo hemos constatado que la contabilidad es un área en la cual hay que tener mucha precaución y exigencia para elaborar software; ya que se debe tener un control minucioso con las cuentas que se realizan.

Con el desarrollo de nuestro sistema consideramos que hemos cumplido satisfactoriamente con todos los objetivos planteados y las etapas consideradas tales como: el ciclo de vida, Elaboración de requisito del software, diagramas, diseño, etc; logrando con todo esto que el sistema haya abarcado con las necesidades que tiene la empresa y el contador.



Recomendaciones.

La persona encargada de manejar el software tiene que tener conocimiento en informática y conocimiento de contabilidad, para que el software opere correctamente y obtener los resultados esperados.

Debido que en contabilidad se lleva un manejo importante de cuentas se recomienda hacer un respaldo de la base de datos de forma mensual.

Para el buen funcionamiento del software debe cumplir con los requerimientos aportados en restricciones generales.

Debido a la carencia de información con respecto a la forma de hacer respaldo en mysql se recomienda que si hay otro método o versiones de mysql que implemente copia de seguridad mejor de la que estamos utilizando, aplicar dicha herramienta.



ANEXOS



CODIGO PARA EL INGRESO DE NUEVA FACTURA

Option Explicit

Private blInit As Boolean, curCredito As Currency

Private Sub cbxCuentas_Click()

'se filtra por el nombre de la cuenta para obtener el codigo

'de la cuenta que es lo que realmente guardaremos en la tabla

'factcredito

adcCuentas.Recordset.Filter = "nombrecta=" & cbxCuentas.Text & ""

End Sub

Private Sub cbxFormaPago_Click()

Call sObtenerAbono

End Sub

Private Sub cbxPlazo_Click()

Call sObtenerAbono

Call sSetFechaVence

End Sub

Private Sub sSetFechaVence()

Dim intDias As Integer

intDias = Val(cbxPlazo.Text)

dtpFechaVencimiento.Value = dtpFecha.Value + intDias

End Sub

'procedimiento para buscar un cliente en la lista

Private Sub cmdBuscar_Click()

With frmSubLstClientes

.blSqlEjecutar = True 'Muestra todos los cliente

.Show vbModal

End With

With tpCliente



```
lbLimitCredito.Caption = FormatNumber (.curLimitCredito, 2)
txtCliente.Text = ""
txtCliente.Text = .strNombres
End With
End Sub
Private Sub cmdCancel_Click()
If (Trim(txtFactura.Text) <> "" Or _
Trim (txtCliente.Text) <> "") Then
If MsgBox ("Desea registrar este crédito?", _
vbYesNo + vbQuestion, strTitulo) = vbYes Then
If fxVerificar() Then Exit Sub
If fxGuardar () Then
Call sEnabled (False)
cmdPrint.Enabled = True
Else
MsgBox "Se presentó un error al guardar el " _
& "crédito", vbInformation, strTitulo
End If
Exit Sub
End If
End If
Call sClearsFields
Call sEnabled (False)
End Sub
Private Sub cmdClose_Click ()
Unload Me
End Sub
Private Sub cmdGuardar_Click ()
If MsgBox ("Esta seguro que desea registrar este " _
& "crédito", vbYesNo + vbQuestion, strTitulo) = vbYes Then
```



```
If fxVerificar () Then Exit Sub
If fxGuardar () Then
    Call sEnabled (False)
    cmdPrint.Enabled = True
Else
    MsgBox "Se presentó un error al guardar el " _
        & "crédito", vbInformation, strTitulo
End If
End If
End Sub
Private Sub cmdNew_Click ()
    Call sClearsFields
    Call sEnabled (True)
    cmdPrint.Enabled = False
    txtFactura.SetFocus
End Sub
Private Sub cmdPrint_Click ()
    If (optPrint (0).Value) Then
        With rptFactura
            .strCodCuenta = adcCuentas.Recordset!codcta
            .strFecha = Me.dtpFecha.Value
            .strFormaPago = Me.cbxFormaPago.Text
            .strPrimerPago = Me.dtpPrimeraFechaPago.Value
            .strVence = Me.dtpFechaVencimiento.Value
            Set .SourceForm = Me 'Referencia del objeto frmNewFactura
            If optPrintReport (0).Value Then
                .PrintReport True
            Else
                .Show vbModal
            End If
        End With
    End If
End Sub
```



```
End With
Elseif optPrint (1).Value Then
  With rptCalendario
    Select Case cbxFormaPago.Text
      Case "Semanal":
        .intFrecuencia = 7
      Case "Quincenal":
        .intFrecuencia = 15
      Case Else
        .intFrecuencia = 30
    End Select
    .strPrimerPago = Me.dtpPrimeraFechaPago.Value
    Set .SourceForm = Me
    If optPrintReport (0).Value Then
      .PrintReport False
    Else
      .Show vbModal
    End If
  End With
End If
End Sub

Private Sub dtpFecha_KeyDown (KeyCode As Integer, Shift As Integer)
  Call sControlCursor (KeyCode)
End Sub

Private Sub dtpFecha_LostFocus ()
  Call sSetFechaVence
End Sub

Private Sub Form_Load ()
  Call sSetBackColorCtrl
  blInit = True 'controla el plazo con la forma de pago
```



Call sLoadCuentas

Call sLoadListas

dtpFecha.Value = Date 'fecha del sistema

dtpPrimeraFechaPago.Value = Date

dtpFechaVencimiento.Value = Date

bllnit = False

End Sub

Private Sub sLoadCuentas()

Dim cnn As ADODB.Connection

On Error GoTo Err_Load

Set cnn = New ADODB.Connection

cnn.ConnectionString = fxGetConnectString ("cuentacobrar")

cnn.Open

adcCuentas.ConnectionString = cnn

adcCuentas.RecordSource = "SELECT codcta, nombrecta FROM cuenta"

adcCuentas.Refresh

Set adcCuentas.Recordset.ActiveConnection = Nothing

cnn.Close: Set cnn = Nothing

With adcCuentas.Recordset

Do Until .EOF

 cbxCuentas.AddItem! nombrecta

 .MoveNext

Loop

End With

Exit Sub



Err_Load:

If Not (cnn Is Nothing) Then

 If (cnn.State = adStateOpen) Then cnn.Close

End If

Set cnn = Nothing

Err.Clear

MsgBox "Error al cargar el catálogo de cuentas", vbInformation, strTitulo

End Sub

Private Sub Form_Unload (Cancel As Integer)

 If Not (frmSubLstClientes Is Nothing) Then

 Unload frmSubLstClientes

 End If

 Set frmNewFactura = Nothing

End Sub

Private Sub txtCliente_Change ()

 Dim curLimite As Currency

 If (txtCliente.Text = "") Then

 curCredito = 0

 Else

 Call GetCreditos

 curLimite = lbLimitCredito.Caption

 lbLimitCredito.Caption = FormatNumber (curLimite - curCredito, 2)

 End If

End Sub

'Obtiene el total de los creditos que se le han dado a un cliente

Private Sub GetCreditos ()

 Dim cnn As ADODB.Connection, rs As ADODB.Recordset

 On Error GoTo Err_LoadCredito

 Set cnn = New ADODB.Connection



```
cnn.ConnectionString = fxGetConnectString ("cuentacobrar")
```

```
cnn.Open
```

```
Set rs = New ADODB.Recordset
```

```
rs.Open "SELECT SUM (montocredito) As Monto FROM factcredito " _
```

```
& "WHERE codclte=" & tpCliente.strCodigo & " AND cancelado=" _
```

```
& "lado='N'", cnn, adOpenForwardOnly, adLockOptimistic
```

```
curCredito = 0
```

```
If Not rs.EOF Then
```

```
    If Not IsNull (rs! monto) Then
```

```
        curCredito = rs!monto
```

```
    End If
```

```
End If
```

```
rs.Close: Set rs = Nothing
```

```
cnn.Close: Set cnn = Nothing
```

```
Exit Sub
```

```
Err_LoadCredito:
```

```
    If Not (rs Is Nothing) Then
```

```
        If (rs.State = adStateOpen) Then rs.Close
```

```
    End If
```

```
    If Not (cnn Is Nothing) Then
```

```
        If (cnn.State = adStateOpen) Then cnn.Close
```

```
    End If
```

```
    Set rs = Nothing: Set cnn = Nothing
```

```
    curCredito = 0
```

```
    Err.Clear
```

```
    MsgBox "Error al consultar estado de cuenta", vbInformation, strTitulo
```

```
End Sub
```

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




```
KeyAscii = Asc (UCase (Chr (KeyAscii)))
End Sub
Private Sub txtDescripcion_GotFocus ()
    With txtDescripcion
        .SelStart = 0
        .SelLength = Len(.Text)
    End With
End Sub
Private Sub txtDescripcion_KeyDown (KeyCode As Integer, Shift As Integer)
    Call sControlCursor(KeyCode)
End Sub
Private Sub txtDescripcion_KeyPress(KeyAscii As Integer)
    If (KeyAscii = vbKeyReturn) Then KeyAscii = 0
End Sub
Private Sub txtFactura_GotFocus()
    With txtFactura
        .SelStart = 0
        .SelLength = Len (.Text)
    End With
End Sub
Private Sub sControlCursor (KeyCode As Integer)
    Select Case KeyCode
        Case vbKeyReturn:
            SendKeys "{TAB}"
        Case vbKeyUp:
            SendKeys "+{TAB}"
        Case vbKeyDown:
            SendKeys "{TAB}"
    End Select
End Sub
```



```
Private Sub txtFactura_KeyDown (KeyCode As Integer, Shift As Integer)
    Call sControlCursor(KeyCode)
End Sub

Private Sub txtFactura_KeyPress (KeyAscii As Integer)
    KeyAscii = Asc (UCase (Chr (KeyAscii)))
End Sub

Private Sub txtMonto_Change ()
    Dim curMonto As Currency, curLimite As Currency

    curLimite = lbLimitCredito.Caption
    curMonto = Val (txtMonto.Text)
    If (curMonto > curLimite) Then
        MsgBox "Se ha excedido de su límite de crédito", vbInformation, strTitulo
        With txtMonto
            .Text = FormatNumber (0, 2)
            .SelStart = 0
            .SelLength = Len (.Text)
        End With
    End If
End Sub

Private Sub txtMonto_GotFocus ()
    With txtMonto
        .SelStart = 0
        .SelLength = Len (.Text)
    End With
End Sub

Private Sub txtMonto_KeyDown (KeyCode As Integer, Shift As Integer)
    Call sControlCursor (KeyCode)
End Sub

Private Sub txtMonto_KeyPress (KeyAscii As Integer)
```



```
If (IsNumeric (Chr(KeyAscii)) Or Chr(KeyAscii) = "." Or KeyAscii = vbKeyBack)
Then
    Exit Sub
Else
    KeyAscii = 0
End If
End Sub

Private Sub txtMonto_LostFocus ()
    If (Val (txtMonto.Text) = 0) Then
        txtMonto.Text = FormatNumber(0, 2)
    End If
    Call sObtenerAbono
End Sub

Private Sub sLoadListas ()
    With cbxPlazo
        .AddItem "15 días"
        .AddItem "30 días"
        .AddItem "45 días"
        .AddItem "60 días"
        .ListIndex = 0
    End With
    With cbxFormaPago
        .AddItem "Semanal"
        .AddItem "Quincenal"
        .AddItem "Mensual"
        .ListIndex = 0
    End With
End Sub

Private Function fxVerificar () As Boolean
    If (Trim(txtFactura.Text) = "") Then
```



```
MsgBox "Falta número de la factura", vbInformation, strTitulo
txtFactura.SetFocus
fxVerificar = True
Elseif (cbxCuentas.ListIndex = -1) Then
MsgBox "Falta nombre de la cuenta", vbInformation, strTitulo
cbxCuentas.SetFocus
fxVerificar = True
Elseif (Trim (txtCliente.Text) = "") Then
MsgBox "Falta el nombre del cliente", vbInformation, strTitulo
cmdBuscar.SetFocus
fxVerificar = True
Elseif (Val (txtMonto.Text) = 0) Then
MsgBox "Falta especificar el monto del crédito", vbInformation, strTitulo
txtMonto.SetFocus
fxVerificar = True
Else
fxVerificar = False
End If
End Function
'Guarda el crédito del cliente
Private Function fxGuardar () As Boolean
Dim cnn As ADODB.Connection, rs As ADODB.Recordset
On Error GoTo Err_AddNew

Set cnn = New ADODB.Connection
cnn.ConnectionString = fxGetConnectString ("cuentacobrar")
cnn.Open

Set rs = New ADODB.Recordset
rs.Open "factcredito", cnn, adOpenDynamic, adLockOptimistic
```



```
rs.AddNew
rs!codcta = adcCuentas.Recordset!codcta
rs!codclte = tpCliente.strCodigo
rs!numfactura = Trim (txtFactura.Text)
rs!fecha = dtpFecha.Value
rs!montocredito = txtMonto.Text
If (Trim(txtDescripcion.Text) <> "") Then
    rs!descripcion = txtDescripcion.Text
End If

rs!cantidadabonar = txtAbono.Text
rs!plazocredito = cbxPlazo.Text
rs!formapagos = cbxFormaPago.Text
rs!primerafechapago = dtpPrimeraFechaPago.Value
rs!fechavencimiento = dtpFechaVencimiento.Value
rs!cancelado = "N"
rs!siguientepago = dtpPrimeraFechaPago.Value
rs!Saldo = txtMonto.Text
rs.Update
rs.Close: Set rs = Nothing
cnn.Close: Set cnn = Nothing
fxGuardar = True
Exit Function
```

Err_AddNew:

```
If Not (rs Is Nothing) Then
    If (rs.State = adStateOpen) Then
        If (rs.EditMode = adEditAdd) Then rs.CancelUpdate
        rs.Close
```



```
End If
End If
If Not (cnn Is Nothing) Then
    If (cnn.State = adStateOpen) Then cnn.Close
End If
Set rs = Nothing: Set cnn = Nothing
MsgBox Err.Description, vbInformation, strTitulo
Err.Clear
fxGuardar = False
End Function
Private Sub sClearsFields()
    txtFactura.Text = ""
    txtCliente.Text = ""
    txtMonto.Text = FormatNumber (0, 2)
    txtAbono.Text = FormatNumber (0, 2)
    txtDescripcion.Text = ""
    lbLimitCredito.Caption = "0.00"
    cbxCuentas.ListIndex = -1
    dtpPrimeraFechaPago.Value = Date
    dtpFechaVencimiento.Value = Date
End Sub
Private Sub sEnabled(bIValue As Boolean)
    Dim n As Integer
    For n = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(n) Is TextBox Or _
            TypeOf Me.Controls(n) Is DTPicker Or _
            TypeOf Me.Controls(n) Is ComboBox Then
            Me.Controls(n).Enabled = bIValue
        End If
    Next n
End Sub
```



```
For n = 0 To optPrint.UBound
    optPrint (n).Enabled = Not bIValue
Next n
txtCliente.Enabled = False
txtAbono.Enabled = False
OptPrintReport(0).Enabled = Not bIValue
optPrintReport (1).Enabled = Not bIValue
cmdBuscar.Enabled = bIValue
cmdGuardar.Enabled = bIValue
cmdCancel.Enabled = bIValue
cmdNew.Enabled = Not bIValue
cmdClose.Enabled = Not bIValue
End Sub
Private Sub sObtenerAbono ()
    Dim intDias As Integer, intFormaPago As Integer, curMonto As Currency
    On Error Resume Next
    If blInit Then Exit Sub
    intDias = Val (cbxPlazo.Text)
    curMonto = If (Val(txtMonto.Text) = 0, 0, txtMonto.Text)
    Select Case cbxFormaPago.ListIndex
        Case 0:
            intFormaPago = 7
        Case 1:
            intFormaPago = 15
        Case Else
            intFormaPago = 30
    End Select
    If (intFormaPago > intDias) Then
        MsgBox "Cambie la forma de pago porque no " & vbCrLf _
            & "se corresponde con el préstamo", vbInformation, strTitulo
```



```
With cbxFormaPago
    .ListIndex = 0
    .SetFocus
End With
Exit Sub

End If

intDias = intDias / intFormaPago
curMonto = FormatNumber((curMonto / intDias), 2)
txtAbono.Text = curMonto

End Sub

Private Sub sSetBackColorCtrl()
    Dim i As Integer
    Me.BackColor = RGB(244, 244, 244)
    For i = 0 To Mrc.UBound
        Mrc(i).BackColor = Me.BackColor
        pic(i).BackColor = Me.BackColor
        optPrint(i).BackColor = Me.BackColor
    Next i
    optPrintReport(0).BackColor = Me.BackColor
    optPrintReport(1).BackColor = Me.BackColor
End Sub
```




Agregar Factura

Numero Factura :	<input type="text"/>	Límite de crédito:	<input type="text" value="0.00"/>
Nombre de la Cuenta:	<input type="text"/>		
Nombre del cliente:	<input type="text"/>		
Fecha :	<input type="text" value="19/10/2006"/>		
Monto Credito :	<input type="text" value="0.00"/>		
Descripcion :	<input type="text"/>		
Plazo Credito :	<input type="text" value="15 días"/>	Imprimir	
Formas Pago:	<input type="text" value="Semanal"/>	<input checked="" type="radio"/> Factura	
Primera Fecha Pago :	<input type="text" value="19/10/2006"/>	<input type="radio"/> Calendario de pago	
Fecha Vencimiento :	<input type="text" value="19/10/2006"/>	<input type="radio"/> Ambos	
Cantidad a abonar:	<input type="text" value="0.00"/>	<input type="radio"/> Imprimir	
		<input checked="" type="radio"/> Visualizar	

Nuevo Guardar Cancelar Imprimir Cerrar



Bibliografía

- MySQL_Manual] *Manual de MySQL*,
<http://www.mysql.com/documentation/index.html> .
- [PostgreSQL_Manual] *Manual de PostgreSQL*,
<http://www.mysql.com/documentation/index.html> .
- [Article_MySQL-PostgreSQL] *Artículo comparativo*,
<http://www.phpbuilder.com/columns/tim20000705.php3?page=1>.
- <http://www.elguruprogramador.com.ar/zonas/ver.asp?cod=115>
- <http://www.lawebdelprogramador.com>
- <http://www.elguruprogramador.com.ar/zonas/ver.asp?cod=115>
- <http://www.lawebdelprogramador.com>