

**ESCUELA POLITECNICA DEL EJERCITO**

**FACULTAD DE INGENIERIA EN SISTEMAS E INFORMATICA**

**SISTEMA DE CONTROL DE TRANSACCIONES EN CAJAS Y  
CAJEROS AUTOMATICOS MEDIANTE LA CAPTURA DE  
IMAGENES**

**Previa a la obtención del Título de:**

**INGENIERO EN SISTEMAS E INFORMATICA**

**POR:**

**MARIO ERAZO LOPEZ  
RUBEN MONTENEGRO CASTILLO**

**SANGOLQUI, JUNIO DE 2005**

## CERTIFICACION

Certifico que el presente trabajo fue realizado en su totalidad por los Srs. Mario Erazo Lopez y Rubén Montenegro Castillo como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA

SANGOLQUI, JUNIO DEL 2005

Ing. Edison Andrés de la Torre Díaz

## DEDICATORIA

Al único amor de mi vida quien a pesar de mis tropezones siempre confía en mí, y me enseñó que los sueños están para alcanzarlos y que los alcanza con esfuerzo, perseverancia y amor.

Una vez más gracias Pauly.

Mario Vinicio Erazo López

## DEDICATORIA

Este trabajo está dedicado a mi madre Isabel Castillo que ha sido la persona que me ha ofrecido su ayuda incondicional apoyándome y dándome fuerza para culminar con esta carrera, a mi hija Sheyla Priscila que ha sido la motivación el aliento y la fuerza para seguir adelante, a mi hermana Cristina Montenegro, a mi padre Richard Montenegro, al único amor de mi vida Kathy Vera, a todos ellos por recibir su apoyo y que me han ayudado para realizarme como profesional.

Rubén Alexander Montenegro Castillo

## AGRADECIMIENTO

En primer lugar doy gracias a Dios ya que a pesar de todo siempre me regala un nuevo día, agradezco además a mis padres que toda la vida me dieron su apoyo para poder cumplir mis sueños, a mi querida abuelita que me brindó su ayuda y sobre todo se soportó todo este tiempo, a mis hermanos los cuales hacen que mi vida sea mucho más entretenida y siempre confían en mí, a mi compañero y amigo Rubén sin el cual el desarrollo de este proyecto no hubiera sido posible.

Mario Vinicio Erazo López

## AGRADECIMIENTO

Agradezco a Dios por estar presente conmigo todos los días, darme la sabiduría necesaria para saber elegir entre lo bueno y lo malo, la oportunidad de ser alguien y por darme un día más de vida.

Agradezco a la Escuela Politécnica del Ejército en especial a los maestro por darme la oportunidad de adquirir los conocimientos necesarios, al director de tesis Ing. Andrés de la Torre y codirector de la tesis Ing. Miroslava Zapata que nos han ayudado con este trabajo, a mis amigos y familiares que siempre estuvieron conmigo en las buenas y en las malas.

Quiero agradecer a mi jefe y amigo Francisco Simbaña por darme una oportunidad y siempre confiar en mí, a mi querido amigo y compañero Mario Erazo por que sin su ayuda y su amistad no hubiera sido posible esto.

Rubén Alexander Montenegro Castillo

---

## INDICE DE CONTENIDO

### ***I. MARCO TEORICO***

1.1 Objetivos del Capítulo	2
1.2 Introducción	2
1.2.1 JUSTIFICACIÓN	3
1.2.2 ANTECEDENTES	4
1.2.3 OBJETIVOS	5
1.2.4 ALCANCE DEL PROYECTO	5
1.2.5 METODOLOGIA	6
1.3 Cajeros Automáticos.	7
1.3.1 Definición de Cajero Automático.	7
1.3.2 Tipos de cajeros.	7
1.3.3 Servicios que prestan estos tipos de cajeros:	8
1.3.4 Proceso que sigue una transacción	8
1.3.5 Switch Transaccional	11
1.4 Arquitectura de Aplicaciones	12
1.4.1 Tipos de Arquitectura	13
1.4.2 Uso de Sockets en el desarrollo de aplicaciones	24
1.5 Compresión de Imágenes	26
1.5.1 Compresión con perdida y si perdida	27
1.5.2 Cantidad de colores	27
1.5.3 Tipos de Ficheros	28
1.6 METODOLOGIAS DE DESARROLLO	32

---

1.6.1 Que es una metodología	32
1.6.2 Análisis de metodologías para el desarrollo del software	32
1.6.3 COMPARACION Y CONCLUSION	54
1.6.4 Metodología aplicada al proyecto	56
1.7 HERRAMIENTAS DE DISEÑO	56
1.7.1 Por qué debemos utilizar herramientas CASE en UML?	56
1.7.2 Rational Rose	57
1.7.3 Poseidon	60
1.7.4 Power designer	61
1.8 HERRAMIENTAS DE DESARROLLO.	63
1.8.1 Visual Basic	63
1.8.2 SQL SERVER 2000	66
1.8.3 MYSQL	69
 <b>II. ANALISIS DEL SISTEMA</b>	
2.1 Objetivos del Capítulo	72
2.2 Especificación de Requerimientos del Sistema.	72
2.1.1 Definición.	72
2.1.2 Especificación de Requerimientos del Sistema "MTG".	73
2.3 Análisis del Sistema	90
2.3.1 Desarrollo del Modelo Estático	90
2.3.2 Modelo Dinámico	103



---

### **III. DISEÑO Y DESARROLLO DEL SISTEMA**

3.1 Objetivos del Capítulo	105
3.2 Definición de Diseño de Sistemas	105
3.3 Diseño Detallado del Sistema	107
3.3.1 Modelo Dinámico	107
3.4 Diseño del Sistema	109
3.4.1 Estándares de Diseño para la Base de Datos	109
3.5 Diseño Arquitectónico del Sistema	111
3,5,1 Diagrama de Despliegue (Arquitectura Física del Sistema)	112
3,5,2 Arquitectura Lógica	115
3.6 Diseño Interfaz Gráfica	121
3.7 Herramientas utilizadas en el Diseño y Desarrollo del Sistema “MTG”	123
3.8 Procesos del Sistema “MTG”	123
3.8.1 Módulo de Captura de Imagen	123
3.8.2 Módulo de Actualización de la Base de Datos Central.	124
3.8.3 Módulo de búsqueda de transacciones.	125
3.8.4 Módulo de Administración de sucursales, Cajas y Cajeros.	126
3.8.5 Módulo de Administración de Usuarios.	126

### **IV. PRUEBAS Y RESULTADOS DEL SISTEMA**

4.1 Objetivos del Capítulo	127
4.2 Alcance de Pruebas	127
4.3 Ambiente de pruebas	127

4.4 Recursos utilizados para ejecutar las pruebas	128
4.5 Generación de Script de Pruebas	129
4.6 Verificación del proceso de pruebas	133
4.7 Documentación de las pruebas	134
4.8 Resultados del Sistema	134

## ***V. CONCLUSIONES Y RECOMENDACIONES***

6.1 Objetivos del Capítulo	138
6.2 Conclusiones	138
6.3 Recomendaciones	140

## **BIBLIOGRAFIA**

## **WEBLIOGRAFIA**

## **ANEXOS**

---

## LISTADO DE FIGURAS

### ***I. MARCO TEORICO***

Figura 1.1: Flujo de una Transacción	10
Figura 1.2: Aplicación Cliente / Servidor en dos capas	20
Figura 1.3: Aplicación Cliente / Servidor en dos capas	21
Figura 1.4: Capa para Reglas del Negocio	21
Figura 1.5: Peticiones al Servidor	22
Figura 1.6: Aplicación Cliente / Servidor en tres capas	23
Figura 1.7 Análisis y diseño orientado a objetos	34

### ***II. ANALISIS DEL SISTEMA***

Figura 2.1 Diagrama de Casos de Uso para Petición de Transacción	96
Figura 2.2 Diagrama de Casos de Uso para Toma de Foto	97
Figura 2.3 Diagrama de Casos de Uso para Almacenamiento de la Información	99
Figura 2.4 Diagrama de Casos de Uso para El Administrador	100
Figura 2.5 Diagrama de Casos de Uso para El Administrador	102

### ***III. DISEÑO Y DESARROLLO DEL SISTEMA***

Figura 3.1: Diagrama de Estados del Objeto "Transacción_Foto"	108
Figura 3.2: Diagrama de Despliegue del Sistema "MTG"	112
Figura 3.3: Arquitectura Lógica del Sistema "MTG"	116
Figura 3.4: Diagrama de Paquetes de Sistema "MTG"	117

Figura 3.5: Módulos del Sistema “MTG”	118
Figura 3.6: Módulo de Búsqueda de Transacciones y Administración de cajas, cajeros y sucursales del sistema “MTG Viewer”	119
Figura 3.7: Módulo de Administración de Usuarios del Sistema “MTG User”	120

#### ***IV. PRUEBAS Y RESULTADOS DEL SISTEMA***

Figura 4.1 Esquema del Proyecto Finalizado	134
--	-----

**LISTADO DE TABLAS*****II. ANALISIS DEL SISTEMA***

Tabla 2.1: Actores del Sistema "MTG"	94
--------------------------------------	----

***III. DISEÑO Y DESARROLLO DEL SISTEMA***

Tabla 3.1: Estados del Objeto "Transacción_Foto"	108
Tabla 3.2: Formato para Texto	121
Tabla 3.3: Formato para Objetos	122

**LISTADO DE ANEXOS**

ANEXO A: DIAGRAMA DE CLASES DEL SISTEMA "MTG"

ANEXO B: DIAGRAMA DE SECUENCIAS DEL SISTEMA "MTG"

ANEXO C: DIAGRAMA DE COLABORACION DEL SISTEMA "MTG"

ANEXO D: DIAGRAMA DE ACTIVIDAD DEL SISTEMA "MTG"

ANEXO E: RESULTADOS DE LAS PRUEBAS

ANEXO F: MANUAL DE USUARIO

ANEXO G: MANUAL TECNICO

## CAPITULO I

### MARCO TEORICO

#### *Objetivos del Capítulo*

- Explicar aspectos relacionados con el Tema de Tesis para mejorar la comprensión del tema a desarrollar.
- Definir los términos a utilizar en todo el desarrollo del documento para evitar el desconocimiento de los mismos.

#### *Introducción*

Este documento contiene la declaración del tema de Tesis "SISTEMA DE CONTROL DE TRANSACCIONES EN CAJAS Y CAJEROS AUTOMATICOS MEDIANTE LA CAPTURA DE IMAGENES", a desarrollarse previo a obtener el título de "Ingeniero de Sistemas e Informática".

A continuación se definen los antecedentes, Alcance y la Justificación. Seguidamente se establecen los objetivos y metodología.

Los bancos desde ya mucho tiempo se han esmerado por brindar confianza a sus clientes, cada banco siempre busca mejorar la calidad de su servicio; en las cajas las personas que atienden siempre están en constante riesgo a sufrir cualquier atentado de parte de personas que viven de forma fácil, la seguridad se la ha reforzado de múltiples maneras, ya sea con personal armado y cámaras de seguridad, pero de todas maneras es un tanto difícil asegurar del todo las transacciones bancarias que se realicen, y mas aun en los cajeros automáticos en los que los fraudes y las estafas son mas comunes ya que tan solo basta con saber la clave de una tarjeta de débito o crédito y el mal esta

hecho. Pero en la mayoría de los casos los autores de estos delitos son expertos conocedores de la tecnología. Los delitos cometidos a través de los cajeros automáticos generalmente son denunciados más por las instituciones financieras que por los particulares afectados.

Además un cajero automático trabaja las veinte y cuatro horas, los trescientos sesenta y cinco días del año, y su ubicación geográfica varía de un lugar a otro sin considerarse a veces seguridades físicas. Estos factores hacen que un cajero automático sea demasiado susceptible a cualquier tipo de ataque, que sin duda afectaría a más de una entidad o persona.

A pesar de esto sus servicios son un cien por ciento primordiales para los Bancos, entre los modos operando más comunes destaca la introducción de una lámina en la ranura receptora del cajero automático para retener la tarjeta y posteriormente realizar los retiros. pero no se puede determinar fácilmente la persona que cometió el ilícito.

Es por eso que se tomo en cuenta el desarrollo de un sistema capaz de llevar un control exacto para cada transacción realizada en una caja o cajero automático mediante la captura de imágenes en el preciso momento.

## **JUSTIFICACIÓN**

Hoy en día las instituciones se ven abocadas a desarrollarse junto con la tecnología informática ya que la información se ha convertido en uno de los recursos más importantes, constituyéndose en un factor clave para todas las organizaciones. En entidades de servicios financieros la información tiene un tratamiento mucho más delicado por tratarse de datos que se relacionan con el dinero. No obstante en este tipo de instituciones la seguridad se convierten en el factor ciento por ciento primordial, pero como en todas las organizaciones este factor es combinado con un servicio ciento por ciento eficiente, es decir con calidad. Bajo este preámbulo los Bancos siempre tratan de



mejorar sus servicios, con el fin de atraer miles de clientes, pero estos clientes lo que mas buscan de una institución financiera es que su dinero este ciento por ciento seguro, y como los fraudes y las estafas son formas comunes de actuar en nuestro país se hace necesario llevar un control de cada una de las transacciones bancarias que se llevan día a día, para esto es fundamental conocer que persona realizó una determinada transacción, a que hora del día, en que lugar del país, estos son motivos muy importantes para llevar a cabo el desarrollo de un sistema capaz de almacenar toda esta información, sin importar nada para hacerlo, cabe acotar que este proyecto no se trata de un mejoramiento o actualización de otro, éste se constituye como un sistema nuevo, con funciones totalmente innovadoras y con bases diferentes.

## **ANTECEDENTES**

El sistema de control de transacciones en cajas y cajeros automáticos nace en base a dos sistemas que se encuentran en producción en el país en diferentes Bancos y Cooperativas, propiedad de MULTISOFT Cia Ltda, como son TRANPRO y SHARING , estos sistemas se encargan de la administración, control y comunicaciones de las transacciones en cajas y cajeros automáticos. Entonces como complemento a estos sistemas surge la necesidad de un nuevo software que se acople a éstos para que se encargue del control de cada una de estas transacciones realizadas. En el país no existe un sistema capaz de llevar este control cada banco o institución financiera se encarga de manejar sus propias seguridades, pero siempre en este aspecto se hace necesario nuevas formas de monitoreo.

Mediante este proyecto es posible monitorear las transacciones acontecidas en las cajas y cajeros automáticos de cualquier institución financiera con detalle de cada transacción (cuenta a la que afectó, valor, identificación de la caja o cajero, foto de la persona, etc.)

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Desarrollar un Sistema que se encargue de llevar un control de cada una de las transacciones realizadas en cajas y cajeros automáticos, mediante la captura y almacenamiento de la información de las transacciones y la imagen o foto de la persona que la realiza.

### **OBJETIVOS ESPECIFICOS**

- Analizar el proceso mediante el cual se realizan las transacciones bancarias, tomando en cuenta los conocimientos básicos que se deben adquirir para conceptualizar el problema a resolver, de manera que se realice un correcto análisis.
- Analizar el proceso de captura de imágenes mediante software, por medio de peticiones de otros sistemas y evitando el uso de demasiado espacio para almacenamiento.
- Analizar el proceso de almacenamiento de imágenes como datos binarios, para su futura reconstrucción a un archivo de imagen.

### **ALCANCE DEL PROYECTO**

El Proyecto pretende realizar un sistema que se encargue del control de las transacciones bancarias realizadas tanto en cajas y cajeros automáticos de cualquier institución financiera, para lo cual se procederá a la captura de imágenes el momento mismo que se realiza la transacción, de manera que esta información sea almacenada en una base de datos central para su futuro uso, ya sea en auditorías o control de seguridades.

## METODOLOGIA

En esta parte del documento es importante señalar que el Modelamiento del Sistema es netamente Orientado a Objetos lo que proporciona una visión más exacta del mundo real.

El Paradigma de Ingeniería de Software más apropiado para este tipo de proyecto es el Modelo en Espiral ya que por medio de éste se puede construir el software en forma interactiva existiendo mayor control en las actividades a desarrollarse, además permite desarrollar tareas de comunicación entre el cliente y el desarrollador y sobre todo planificar tareas que permitirán las evaluaciones de riesgos técnicos que puedan surgir a futuro. Para el Análisis y Diseño de proyecto se utilizará la técnica OMT (Técnica de Modelamiento de Objetos) y UML (Lenguaje de Modelamiento Unificado) los cuales permiten complementar el proceso de desarrollo. La Metodología OMT pone énfasis en el análisis del sistema además por medio de tres modelos capturan los aspectos estáticos, dinámicos y funcionales que combinados proveen una descripción completa del software. UML es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprenderá el desarrollo de software. UML entrega una forma de modelar las cosas conceptuales como son los procesos del negocio y funciones del sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

Es importante recalcar que la etapa del Diseño del Sistema va a estar reforzada con los modelos que ofrece la técnica del OMT, con la finalidad de obtener diagramas que sirvan de apoyo en la construcción del Software.

### ***Cajeros Automáticos.***

### **Definición de Cajero Automático.**

Son equipos electrónicos que permiten al usuario bancario hacer una serie de transacciones, sin necesidad de valerse de un empleado bancario, se puede considerar como uno de los últimos avances en tecnología con que cuentan los bancos o entidades financieras.

Físicamente un cajero automático es una máquina que tiene comunicación directa con la central de un banco determinado realizando operaciones en tiempo real, dichas operaciones son transacciones que pueden afectar a cuentas de clientes del banco, cualquier movimiento realizado en una cuenta es registrado por el banco en el momento que se realiza; se pueden realizar consultas, retiros y otras operaciones.

Un cajero automático cuenta con una caja fuerte para el resguardo del dinero en efectivo que reciben y que se entrega, a la vez que está protegida con cerraduras de alta seguridad y llaves únicas. Los cajeros automáticos son sistemas desarrollados con tecnología de punta, que utilizan componentes de la más alta calidad para ofrecer, tanto al usuario como a la institución bancaria, la mayor seguridad, eficiencia y rapidez en el manejo de las diferentes operaciones que se llevan a cabo.

### **Tipos de cajeros.**

La mayoría de los bancos actualmente tienen instalados tres tipos de cajeros automáticos:

- ***Función completa de pared:*** Como su nombre lo indica se encuentra empotrado en la pared de una oficina o cerca de esta.
- ***Función completa de lobby:*** Tiene tipo de escritorio y es especial para ser ubicado en sitios cerrados, Oficinas, Centros Comerciales y Supermercados.

- **Función completa de auto:** Para que el cliente realice transacciones sin bajarse de su automóvil. Estos pueden estar localizados en zonas de parqueo o los costados de una zona especial.

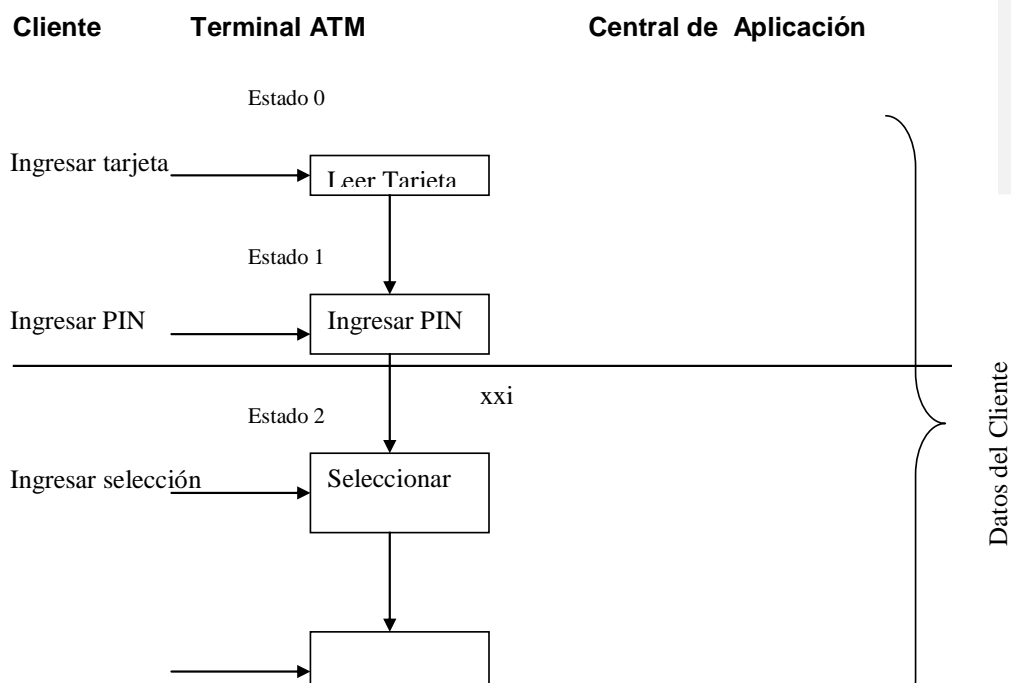
**Servicios que prestan estos tipos de cajeros:**

Estos tres tipos de cajeros prestan los siguientes servicios: retiros, consultas de saldos, depósitos, pagos, transferencias de fondos, pagos de servicios públicos, cajero automático en línea, límite de retiro diario y buzón.

Tanto el Cajero Dispensador de Lobby como el de Pared, Realizan las funciones de un cajero automático convencional a excepción de las transacciones de depósito o consignación.

**Proceso que sigue una transacción**

Una terminal se encuentra en servicio cuando las transacciones que se realizan en ellas siguen el flujo que se muestra en la Figura 1.1:





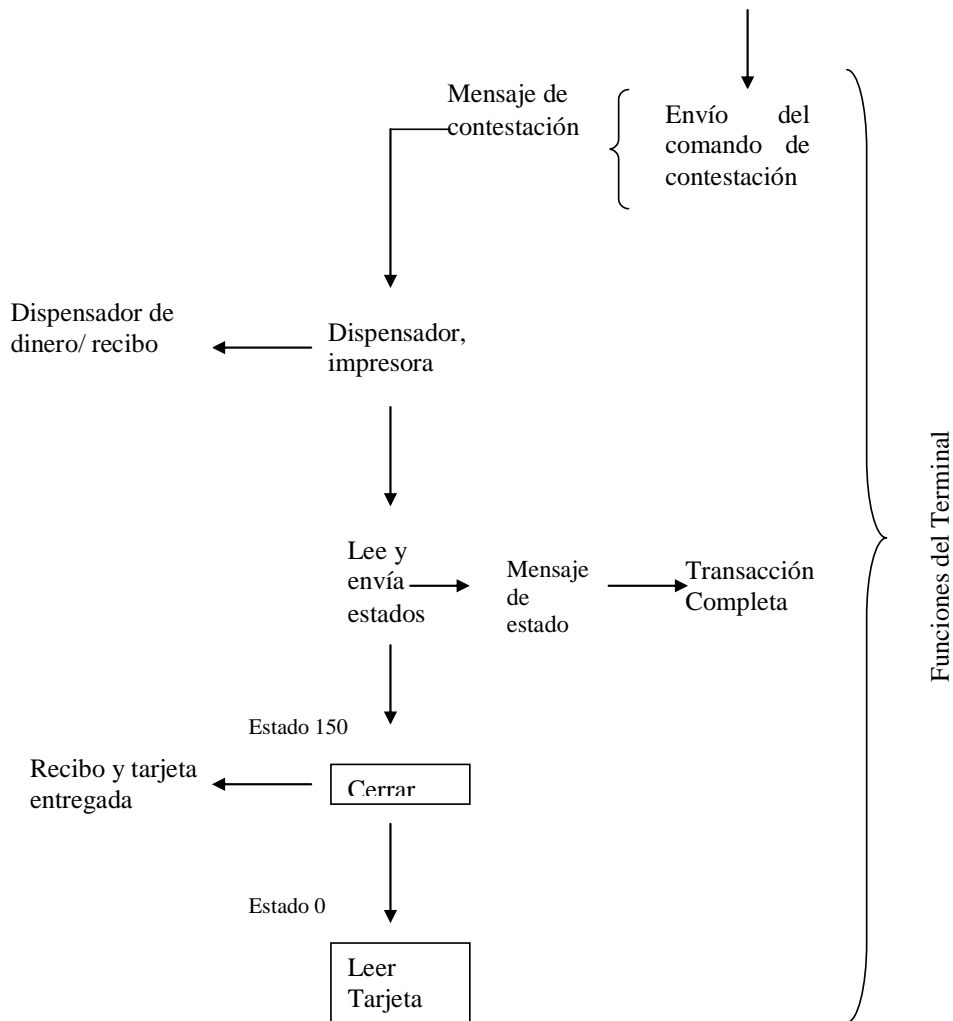


Figura 1.1: Flujo de una Transacción

La anterior figura ilustra los estados típicos por los que pasa cualquier transacción, a continuación se detalla dichos estados:

- Se lee la tarjeta ingresada por el usuario
- Ingreso del PIN
- Selección de una de las cuatro funciones que se despliegan en pantalla
- Solicitud de los estados de la función seleccionada

- Respuesta de los estados solicitados por parte de la Central, la central responde al Terminal mediante un comando de respuesta que contiene información sobre el estado de la función que seleccionó el usuario.
- Despliegue en pantalla de los datos seleccionados
- Entrega al usuario de dinero si fuera el caso
- Terminada la transacción se imprime el recibo de dicha transacción

Si la transacción fue correcta se envía a la Central un mensaje de CLOSE, de lo contrario se envía a la central un mensaje de ERROR en donde la Central realiza un reverso de dicha transacción.

### **Switch Transaccional**

Es un servidor autónomo de cajeros automáticos y un ruteador de transacciones hacia otros servidores o mainframes que proveen servicio de ATM's a sus clientes a través de Filtros de comunicaciones o Host Interface (programas que permiten la conexión entre diferentes productos, que pueden ser desde sistemas de arquitectura propietaria como hosts IBM u otros sistemas de arquitectura abierta a los cuales se desee conectar, para obtener información o procesar transacciones).

Su objetivo fundamental es proveer un servicio confiable y rápido de transacciones tanto para clientes del Banco como para clientes de otros bancos que utilizan sus ATM's en una Red de Cajeros Automáticos. Un Switch transaccional controla y administra la red de cajeros automáticos y provee un monitor en tiempo real y del estado de cada uno de los cajeros de la red y sus componentes individuales.

Un Switch transaccional procesa transacciones de diferentes tipos de cajeros y con protocolos diversos como son: X.25, TCP/IP, SDLC, ISO ASINCRONO, etc. Entre las



diferentes transacciones definidas para cajeros automáticos un switch transaccional permite realizar entre otras:

- Retiro de cuentas corrientes
- Retiro de cuentas de ahorros
- Consultas de saldos
- Transferencias entre cuentas
- Emisión de estados de cuenta

### ***Arquitectura de Aplicaciones***

La Arquitectura de una aplicación hace referencia a la forma cómo están diseñadas las aplicaciones tanto en forma física como lógica.

En el diseño físico se especifica donde se encontrarán las partes de la aplicación (hardware / comunicaciones), tales como discos, ejecutables, cable de red y computadoras.

En el diseño lógico o conceptual se especifica la estructura de la aplicación y sus componentes sin tomar en cuenta el lugar en donde se localizará el software, hardware e infraestructura.

Una de las ventajas de saber diseñar una aplicación es permitir la implantación de la misma en varias plataformas y configuraciones, esta característica de portabilidad es un punto deseable para permitir que una aplicación sea flexible y escalable.

## **Tipos de Arquitectura**

### **Arquitectura Centralizada**

Se caracteriza por tener un servidor principal y una serie de terminales tontas que no ejecutan ningún proceso. La terminal recibe los pedidos y los envía al servidor, éste procesa las solicitudes y devuelve los resultados a las terminales.

#### **Características funcionales**

- El servidor central es la única computadora de la organización.
- El servidor central contiene todos los datos y es el responsable de la consolidación de la información.
- Desde el servidor central se controla el acceso a múltiples terminales.
- Los terminales funcionan como "esclavos" del servidor central.
- Cada usuario tiene un número asignado, derechos y prioridades de ejecución de peticiones.

#### **Características físicas**

- Existe un único servidor corporativo preparado para soportar todos los procesos de la organización.
- El servidor central posee una gran base de datos donde residen todos los datos de la empresa.
- Existen impresoras y terminales conectados al servidor central.

#### **Características lógicas**

- Ejecución de todos los procesos en el servidor central.
- Si la empresa está dispersa geográficamente y dispone de comunicaciones, todos los puestos de trabajo están conectados al computador formando una "estrella".

### **Ventajas**

- Alto rendimiento transaccional.
- Alta disponibilidad.
- Entorno probado y personal experimentado.
- Control total del computador, al ser éste único centro de proceso de datos.
- Alto nivel de seguridad

### **Desventajas**

- Alto precio del servidor central, al requerir de características especiales y potentes para administrar un conjunto de terminales tontos.
- Alta dependencia de las comunicaciones. En caso de caída de una línea, todos los puestos de trabajo dependientes de dicha línea quedan inoperantes.
- Interfaces de usuario poco amigables con el usuario.

### **Arquitectura Distribuida**

En esta arquitectura existen varios tipos de servidores y terminales quienes son los que envían solicitudes de transacciones a los servidores para que los procese y envíe los resultados. Entre los tipos de servidores que pueden incluirse en este tipo de arquitectura están los Servidores de Base de Datos, Servidores de Componentes, Servidor Web, etc.

### **Características funcionales**

- Cada usuario trabaja con su computadora (inteligente), con lo que obtiene mejores tiempo de respuesta.
- Los recursos necesarios que no estén disponibles sobre la computadora local pueden tomarse del servidor central a través de la red de telecomunicaciones.

#### **Características físicas**

- Sistemas informáticos distribuidos en los que las computadoras están conectadas por medio de una red de comunicaciones.
- Las computadoras locales proporcionan acceso a otros elementos de la red o a otros servidores centrales.
- La red de comunicación de datos es de gran importancia.

#### **Características lógicas**

- Cada tarea individual puede ser analizada para determinar si puede distribuirse o no. En general, las tareas más complejas o de carácter estratégico para la organización se mantienen en el servidor central. Las tareas de complejidad media o específica para un determinado grupo de usuarios se distribuyen entre las máquinas locales de ese grupo.
- La plataforma física seleccionada puede ajustarse a las necesidades del grupo de usuarios, con lo que surgen los servidores especializados para determinados tipos de tareas.

#### **Ventajas**

- Funcionamiento autónomo de los sistemas locales, lo que origina un buen tiempo de respuesta.
- Los sistemas de información llegan a todos los departamentos de la empresa.

- Abre posibilidades de trabajo mucho más flexibles y potentes.

### **Desventajas**

- Requiere un intenso flujo de informaciones (muchas veces no útiles, como pantallas y datos incorrectos) dentro de la red, lo que puede elevar los Costos de comunicaciones.
- Mayor complejidad de diseño y administración.
- Inconsistencia de datos en el caso de bases de datos no integradas.

### **Arquitectura Distribuida Cliente / Servidor**

Cliente / Servidor es un tipo de arquitectura distribuida, en donde dos o más procesos separados cooperan entre sí para intercambiar información, servicios o recursos y así poder completar una tarea. El proceso cliente solicita al proceso servidor la ejecución de alguna acción en particular. Esta operación se conoce como Proceso Cooperativo, dado que dos procesos separados cooperan para completar una tarea en particular.

Los procesos pueden o no estar en una sola máquina física. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios y en el cliente contiene sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.
- Las principales funciones que los servidores realizan son:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en computadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales.

### **Características**

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### **Ventajas**

#### Aumento de la productividad

- Los usuarios pueden utilizar herramientas, como hojas de cálculo y herramientas de acceso a bases de datos.
- Mediante la integración de las aplicaciones cliente / servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes.
- Una interfaz gráfica de usuario consistente, reduce el tiempo de aprendizaje de las aplicaciones.

#### Menores costos de operación

- Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión.
- Proporcionan un mejor acceso a los datos. La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información.
- El movimiento de funciones desde un computador central hacia servidores o clientes locales origina el desplazamiento de los costos de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.

#### Mejora en el rendimiento de la red

- Las arquitecturas cliente / servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los computadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.
- Tanto el cliente como el servidor pueden escalarse para ajustarse a las necesidades de las aplicaciones. Las CPUs utilizadas en los respectivos equipos pueden dimensionarse a partir de las aplicaciones y el tiempo de respuesta que se requiera.
- La existencia de varias CPUs proporciona una red más fiable: un fallo en uno de los equipos no significa necesariamente que el sistema deje de funcionar.
- En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema.

- La arquitectura modular de los sistemas cliente / servidor permite el uso de computadores especializados (servidores de base de datos, servidores de ficheros, etc.).
- Permite centralizar el control de sistemas que estaban descentralizados, como por ejemplo la gestión de los computadores personales que antes estuvieran aislados.

### **Desventajas**

- Complejidad tecnológica al tener que integrar una gran variedad de productos.
- Requiere rediseño de todos los elementos involucrados en los sistemas de información (modelos de datos, procesos, interfaces, comunicaciones, almacenamiento de datos, etc.).
- Es más difícil asegurar un elevado grado de seguridad en una red de clientes y servidores distribuidos que en un sistema con un único computador centralizado.
- Los problemas de congestión de la red pueden degradar el rendimiento del sistema por debajo de lo que se obtendría con una única máquina (arquitectura centralizada). En ocasiones la interfaz gráfica de usuario puede relentizar el funcionamiento de la aplicación.

### **Aplicaciones Cliente / Servidor en dos capas**

Estas aplicaciones son divididas en las siguientes capas:

- Front-end (la interfaz del usuario). El proceso front-end se desarrolla en algún lenguaje de 4ª generación (4GL). Se llama front-end dado que es la capa en donde el usuario interactúa con su PC.
- Back-end (servidor de Bases de datos SQL, Sistema operativo multitareas, etc.). El proceso back-end es el servidor de bases de datos como SQL Server. Se llama así dado que típicamente reside en un servidor central en un entorno controlado.



En la Figura 1.2, se muestra una forma de organizar este tipo de aplicaciones, por un lado se encuentra la interfaz y por otro las reglas de negocio junto con el Acceso a Bases de Datos:

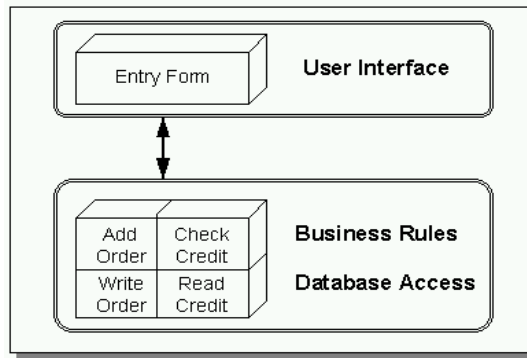


Figura 1.2: Aplicación Cliente / Servidor en dos capas

También se podría organizar de la siguiente manera: en el mismo lado la interfaz junto con las reglas de negocio y en otro la base de datos como se muestra en la siguiente Figura 1.3.

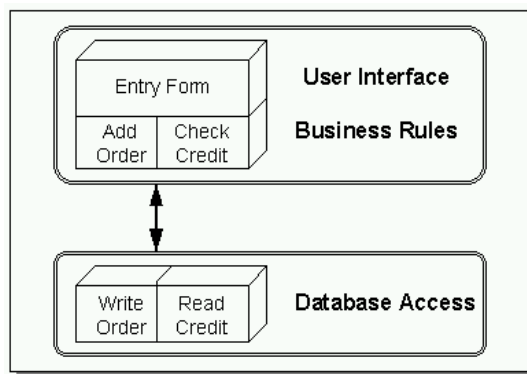


Figura 1.3: Aplicación Cliente / Servidor en dos capas

Los mejores gestores de base de datos relacionales proporcionan soporte para implementar en ellos muchas reglas de negocio, mediante el uso de claves primarias, integridad referencial, triggers, etc. Como en la Figura 1.4.

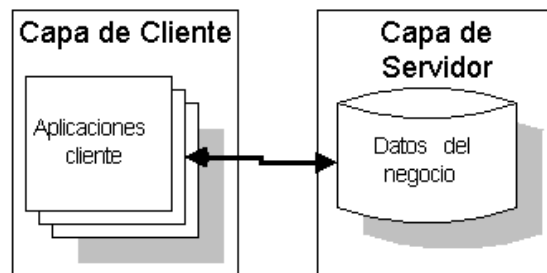


Figura 1.4: Capa para Reglas del Negocio

Un servidor es un computador remoto, en algún lugar de la red que proporciona información según petición. El servidor se lo utiliza para dar servicio a muchos clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

Un cliente funciona en su computador local, se comunica con el servidor remoto, y le pide información.

El servidor envía la información solicitada, la Figura 1.5 muestra el proceso antes mencionado.

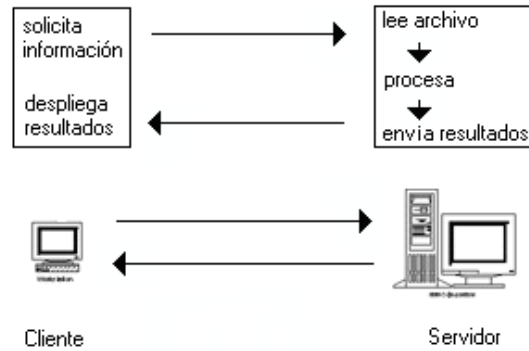


Figura 1.5: Peticiones al Servidor

### Aplicaciones Cliente / Servidor en tres capas

En este tipo de aplicaciones cada uno de los componentes se separa en una sola entidad. Esto permite implementar componentes de una manera más flexible, pero la arquitectura se hace más compleja como en la figura 1.6.

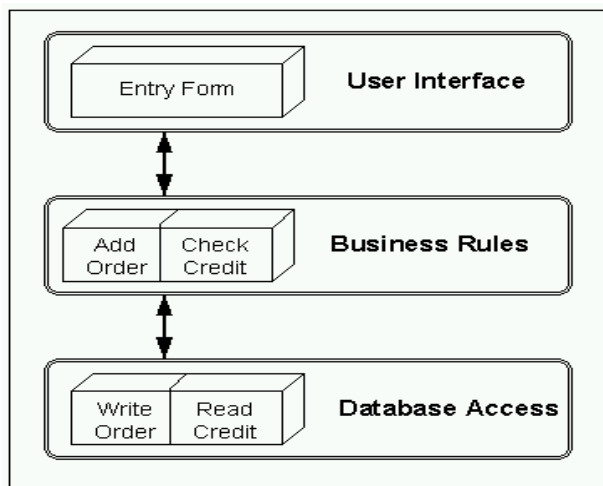


Figura 1.6: Aplicación Cliente / Servidor en tres capas

La capa adicional provee de una capa explícita para las reglas del negocio que se sitúa entre lo que se ha llamado front-end y back-end. Esta capa intermedia encapsula el modelo de negocios (o "reglas de negocios") asociado con el sistema, todas las peticiones de los clientes se controlan en esta capa.

La arquitectura en tres capas hace a las aplicaciones más portables para que trabajen en diferentes tipos de plataformas. La seguridad es fácil de implementar ya que el software de la aplicación está fuera del cliente.

Entre las principales características tenemos:

- El cliente no tiene que tener drivers ODBC ni la problemática de instalación de los drivers por tanto se reduce el costo de mantener las aplicaciones cliente
- El Cliente y el Gestor de Reglas de negocio tienen que hablar el mismo lenguaje
- El Gestor de Reglas de Negocio y el Servidor de Datos tienen que hablar el mismo lenguaje (ODBC, JDBC)

### **Uso de Sockets en el desarrollo de aplicaciones**

Un socket es un punto de comunicación por el cual un proceso puede emitir o recibir información., es un mecanismo de comunicación que permiten que dos procesos hablen entre sí, incluso estando en distintas máquinas.

Los sockets trabajan sobre la capa de transporte, que es la que establece una conexión entre dos hosts utilizando el protocolo TCP o UDP. Para establecer dicha conexión, se crea el socket para comunicar los procesos. Estos procesos pueden ser algún servicio en particular (TELNET, FTP, SMTP, Domain, etc.).

En el interior de un proceso, un socket se identifica por un identificador de la misma naturaleza que los que identifican los archivos.

La comunicación entre procesos por medio de sockets tiene los siguientes pasos:

- El proceso servidor crea un socket con nombre y espera la conexión.
- El proceso cliente crea un socket sin nombre.
- El proceso cliente realiza una petición de conexión al socket servidor.
- El cliente realiza la conexión a través de su socket mientras el proceso servidor mantiene el socket servidor original con nombre.

Es muy común en este tipo de comunicación lanzar un proceso hijo, una vez realizada la conexión, que se ocupe del intercambio de información con el proceso cliente mientras el proceso padre servidor sigue aceptando conexiones.

Todo socket viene definido por las siguientes características:

- El **tipo** del socket, que indica la naturaleza del mismo, el tipo de comunicación que puede generarse entre los sockets.
- El **dominio** del socket especifica el conjunto de sockets que pueden establecer una comunicación con el mismo.
- El **protocolo** se refiere al protocolo que se va a utilizar en la comunicación a través del socket.

## Tipos de SOCKETS

Define las propiedades de las comunicaciones con las que trabaja un socket. Las características que hay que tomar en cuenta para una buena comunicación son:

- Fiabilidad de transmisión.
- Mantenimiento del orden de los datos.
- No duplicación de los datos.
- El "Modo Conectado" en la comunicación.
- Envío de mensajes urgentes

Los tipos de sockets disponibles son:

**SOCK\_DGRAM**: sockets para comunicaciones en modo no conectado, con envío de datagramas de tamaño limitado (tipo telegrama). En dominios Internet el protocolo del nivel de transporte sobre el que se basa es el **UDP**.

**SOCK\_STREAM**: para comunicaciones fiables en modo conectado, de dos vías y con tamaño variable en los mensajes de datos. En dominios Internet subyace el protocolo **TCP**.

**SOCK\_RAW**: permite el acceso a protocolos de más bajo nivel como el **IP** (nivel de red)

**SOCK\_SEQPACKET**: tiene las características del **SOCK\_STREAM** y el tamaño del mensaje es fijo.

## *Compresión de Imágenes*

---

A medida que la Informática ha avanzado, las imágenes se han convertido en una pieza muy importante de ésta. Actualmente surgen cada día más entornos gráficos orientados a múltiples aplicaciones. En los comienzos sólo había texto en los ordenadores, el código ASCII, con lo que la compresión de éste era sencilla y con resultados buenos.

Las imágenes se están haciendo con la Informática, de ahí la necesidad de compactarlas, hay que reducir la cantidad de datos necesarios para representar una imagen digital. La compresión se basa en la eliminación de datos redundantes. Expresado esto matemáticamente, equivale a transformar una distribución bidimensional de píxeles en un conjunto de datos estadísticos sin correlacionar. Esta transformación (compresión) es aplicada a las imágenes antes de que sean almacenadas o antes de ser enviadas, por ejemplo vía red. La imagen una vez en su destino o almacenada es descomprimida para su visionado.

En un principio la compresión de imágenes se limitaba a comprimir el ancho de banda en las transmisiones de vídeo mediante métodos analógicos. Con la llegada de las computadoras digitales, los métodos de compresión analógicos fueron dejando paso a la compresión digital. Que en la actualidad se hayan adoptado diversos estándares internacionales, ha hecho este campo haya avanzado de manera considerable.

Hoy en día la compresión de imágenes es crucial, el crecimiento de la informática multimedia (las computadoras se utilizan para la video producción, difusión, etc.). Es muy relevante el papel que se desempeña en temas como la videoconferencia, imágenes médicas, envío de FAX, el control remoto de aplicaciones militares, etc.

También en lo referente a seguridad ya que en muchos casos se necesita la foto de una persona para identificarlo en el momento que haya cometido algún delito.

## Compresión con pérdida y si pérdida

Una compresión sin pérdidas devuelve la imagen descomprimida exactamente igual a la original. Por el contrario, la compresión con pérdidas acepta alguna degradación en la imagen de cara a una mayor compresión.

## Cantidad de colores

Las imágenes más simples contienen sólo dos colores: blanco y negro, y sólo se necesitan 1 bit para representar cada píxel. La mayoría de las tarjetas de video en los PC soportaban sólo 16 colores prefijados. Más adelante soportaban 256 y en la actualidad admiten 224 o 16 millones de colores.

## Tipos de Ficheros

### TIFF (Tagged Image File Format)

TIFF es, en principio, un formato muy flexible con o sin pérdida. Los detalles del algoritmo de almacenamiento de la imagen se incluyen como parte del fichero. En la práctica, TIFF se usa casi exclusivamente como formato de almacenamiento de imágenes sin pérdidas y sin ninguna compresión. Consecuentemente, los archivos en este formato suelen ser muy grandes. Algunas veces se usan un algoritmo de compresión sin pérdidas llamado **LZW**, pero no siempre.

### PNG

PNG es también un formato de almacenamiento sin pérdida. Al contrario que ocurre con el formato TIFF puede comprimir la imagen. Además tal compresión es totalmente reversible y por tanto la imagen que se recupera es exacta a la original.

### GIF (Graphics Interchange Format)

GIF crea una tabla de 256 colores a partir de una de 16 millones. Si la imagen tiene menos de 256 colores, GIF puede almacenar la imagen sin pérdidas. Cuando la



imagen contiene muchos colores, el software que crea el archivo GIF usa algún algoritmo para aproximar los colores de la imagen con una paleta limitada de 256 colores disponibles. Un buen algoritmo de este tipo, tratará de encontrar un conjunto óptimo de 256 colores. Algunas veces, GIF usa el color más cercano para representar cada píxel, y algunas veces usa un "error de difusión" para ajustar los colores de los píxeles vecinos y así corregir el error producido en cada píxel.

GIF produce compresión de dos formas. Primero, reduce el número de colores de la imagen a 256 y por tanto, reduce el número de bits necesario por píxel. Después, reemplaza áreas de color uniforme usando código de secuencias: en lugar de almacenar "blanco, blanco, blanco, blanco, blanco" almacena "5 blanco"

Por tanto, GIF es una compresión de imágenes sin pérdida sólo para imágenes de 256 colores o menos. Sin embargo, para una imagen de 16 millones de colores GIF puede "perder" el 99.998% de los colores.

## **JPG**

JPG es el método de compresión más adecuado para fotografías e imágenes de tonos continuos similares que contienen muchos colores. Permite obtener unos ratios de compresión muy altos manteniendo a su vez una calidad en la imagen muy elevada. JPG analiza las imágenes y elimina la información que no es apreciable. JPG almacena imágenes de 16 millones de colores. Otro aspecto importante es que el método JPG permite distintos niveles de compresión. En niveles de compresión de imágenes moderado, es muy difícil discernir las diferencias de la imagen original. Programas de tratamiento de imágenes avanzados como Paint Shop Pro o Photoshop permiten ver la calidad de la imagen y el tamaño del fichero como una función de nivel de compresión, de esa forma, se puede elegir convenientemente la calidad y el tamaño del fichero deseado.

**RAW, BMP, PSP, PSD, ...**

RAW es la imagen de salida que ofrece algunas cámaras digitales. Aunque es un método sin pérdida, ofrece un factor de tres o cuatro menor que el formato TIFF de la misma imagen. La desventaja es que el método RAW no está estandarizado y cada marca tiene su propia versión de dicho método, por tanto, se debe usar el software de la cámara para poder visualizar las imágenes.

BMP es un formato de almacenamiento sin compresión de imágenes propiedad de Microsoft.

PSP, PSD son formatos usados en distintos programas básicos (Paint Shop Pro, Photoshop).

En resumen, los métodos de compresión TIFF, PNG, GIF y JPG son públicos y por tanto se pueden implementar en cualquier programa gráfico. GIF y JPG son los más usados en las páginas Web.

Dado que PNG hace lo mismo que GIF e incluso mejor, se espera que PNG sustituirá GIF en el futuro. PNG no sustituirá JPG dado que JPG consigue una mayor compresión en imágenes fotográficas.



**JPEG**

Lo primero que tenemos que destacar de este formato son dos cosas: que ofrece los actuales e imprescindibles 16 millones de colores y que alcanza ratios de compresión superiores a 20:1. La limitación de este formato es que para obtener esos enormes ratios la imagen es modificada, descartando partes que el algoritmo de compresión sabe que el ojo humano no va a poder detectar. Cuando tratamos de comprimir imágenes sencillas

(vectoriales, con pocos colores, etc.) vemos como la compresión es mínima y las modificaciones hechas en la imagen son visibles tras la compresión

La abreviación JPEG viene de la iniciales "Joint Photographic Experts Group". Se trata de un grupo de expertos que definieron las bases de este formato. Debemos tomar la precaución, cuando trabajamos con formatos que modifican la imagen, de guardar la imagen original. Si por ejemplo, convertimos una GIF a JPEG y luego hacemos la conversión inversa, es posible que el fichero GIF ocupe el doble que el original. Esto ocurre debido a que el formato JPEG no funciona correctamente con imágenes que contengan zonas de un sólo color. Existe un formato gráfico que toma lo mejor de estos dos formatos, el HSI JPEG, que desactiva la compresión JPEG cuando encuentra zonas extensas del mismo color. El problema es que es incompatible tanto con el GIF como con el JPEG.

#### Comparando tamaño de ficheros

GRAFICO 1 (FOTO)	GRAFICO 2 (IMAGEN CUALQUIERA)
	

TIPO DE FICHERO	TAMAÑO GRAFICO 1	TAMAÑO GRAFICO 2
<b>BMP MONOCROMO</b>	88 KB	25 KB
<b>BMP DE 16 COLORES</b>	341 KB	97 KB
<b>BMP DE 256 COLORES</b>	680 KB	194 KB
<b>BMP DE 24 BITS</b>	481 KB	577 KB
<b>JPEG</b>	99 KB	14 KB

<b>GIF</b>	323 KB	21 KB
<b>TIFF</b>	1,81 MB	19 KB
<b>PNG</b>	1,058 MB	10 KB

En conclusión y de acuerdo a las necesidades que nosotros necesitamos vamos a utilizar el formato **jpeg** ya que produce una excelente calidad incluso con radio de compresión muy elevados, y como se va almacenar en una base de datos y lo que se trata es de no saturar la base de datos este formato es el ideal para nuestros requerimientos

## ***METODOLOGIAS DE DESARROLLO***

### **Que es una metodología**

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

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

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

- Técnicas para adaptar el método, y
- Técnicas definidas

### **Análisis de metodologías para el desarrollo del software**

Las metodologías de desarrollo de sistemas tratan las siguientes fases del ciclo de vida del desarrollo de sistemas: planeación, análisis, diseño, protección y transición.

*La planeación de la estrategia* produce modelos de alto nivel de un negocio y, con éstos, define un plan para desarrollar un conjunto de proyectos de sistemas interrelacionados.

*Al análisis del sistema* modela un área de sistemas basados en ideas y conceptos de los expertos de dominio proponiendo cualquier decisión relacionada con la instrumentación.

*El diseño del sistema* desarrolla un modelo de instrumentación o implementación basado en los modelos conceptuales desarrollados durante el análisis del sistema.

*La construcción del sistema* implica la elaboración y la prueba de programas, base de datos y redes de acuerdo con el que haya quedado definido durante el diseño del sistema.

La transición del sistema instala los sistemas construidos.

Cuando analizamos sistemas, creamos modelos del área de aplicación que nos interesa. Un modelo puede incorporar un sistema, centrarse en el área de la empresa o abarcar toda la empresa.

El modelo representa un aspecto de la realidad y se construye de modo que nos ayude a comprender a esta.

Con el análisis orientado a objetos, la forma de modelar la realidad difiere del análisis convencional. Modelamos el mundo en términos de tipos de objetos y lo que le ocurre a éstos. Los modelos que construimos en el análisis OO reflejan la realidad de modo más natural que los del análisis tradicional de sistemas. Mediante las técnicas OO, construimos software que modela más fielmente el mundo real. Cuando el mundo real cambia, nuestro software es más fácil de cambiar, lo que es una ventaja real.

FIGURA 1.7. Análisis y diseño orientado a objetos

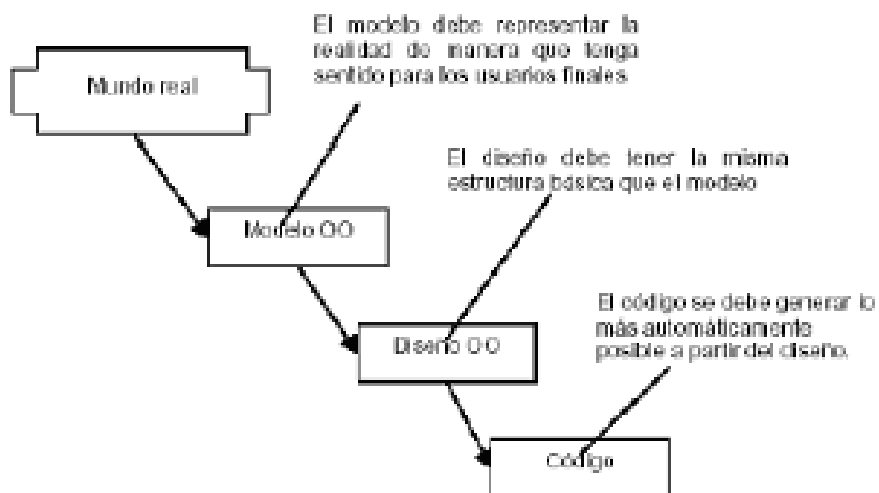


Figura 1.7 Análisis y diseño orientado a objetos

Para el presente análisis nos enfocamos al estudio y comprensión de las metodologías OMT (Object Modeling Technique) desarrollada por James Rumbaugh, la metodología propuesta por Booch (y que lleva su mismo nombre), la metodología OOSE desarrollada por Ivar Jacobson, y la métrica Versión 3.

## **METODOLOGÍA OMT (RUMBAUGH)**

La metodología OMT es una técnica de modelado de objetos, desarrollada por James Rumbaugh, que es uno de los precursores del Lenguaje Unificado de Modelado (UML).

El significado de las siglas de esta metodología es Técnica de Modelado en Objetos (Object Modeling Technique), la definen como una de las metodologías de la Ingeniería de Software aplicable al desarrollo orientado a objetos en las fases de análisis y diseño.

La fase de análisis comienza con la declaración del problema que incluye, una lista de de objetivos (metas) y conceptos claves definitivos definidos para el dominio del problema a resolver. La declaración del problema se “expande después en tres modelos “:

- Modelo de objetos
- Modelo dinámico
- Modelo funcional

El modelo de objetos representa los objetos del sistema. El modelo dinámico representa la interacción entre esos objetos representados como eventos, estados y transiciones. El modelo funcional representa los métodos del sistema desde la perspectiva de flujo de datos. La fase de análisis genera diagramas del modelo de objetos, diagramas de estado, diagramas de eventos de flujo y diagramas de flujos de datos. Es entonces cuando se tiene completa la fase de análisis.

Después de la fase de análisis, se sigue con la fase de diseño de sistema. Aquí se define la arquitectura completa del sistema. Primero el sistema se organiza en subsistemas que están asignados a ciertos procesos y tareas, tomando en cuenta la colaboración y concurrencia entre ellos. Luego, el almacenamiento persistente de datos

se establece por medio de una estrategia de manejo de información global compartida. Después, se examinan las situaciones límite para ayudar a establecer las prioridades de negociación.

La fase de diseño de objetos viene después de la fase de diseño del sistema. Aquí se establece el plan de implementación. Se definen las clases de objetos, así como sus algoritmos, poniendo especial atención con la optimización y persistencia de datos. Se definen cuestiones de herencia, asociaciones, agregaciones y valores por omisión.

La metodología OMT es secuencial en el sentido de que la primera fase es la de análisis, seguida por el diseño. En cada fase, se hacen aproximaciones iterativas entre los pequeños pasos a seguir. La metodología OMT es muy similar a la metodología Booch, cuyo principal criterio es hacer énfasis en las fases de diseño y análisis para una primera entrega del producto. Ambas OMT y Booch no hacen prioritarias las fases de implementación, evaluación u otras del ciclo de vida.

OMT pone especial atención en el modelo y uso de modelos para lograr una abstracción, en el cual el análisis está enfocado en el mundo real a nivel de diseño, también pone detalles particulares para modelado de recursos físicos. Esta tecnología es aplicable en varios aspectos de implementación incluyendo archivos, bases de datos relacionales y orientadas a objetos. OMT se construye alrededor de descripciones de estructuras de datos, constantes, sistemas de procesos de transacciones.

OMT hace énfasis en especificaciones declarativas de la información, captura de manera transparente los requerimientos, especificaciones imperativas para poder descender prematuramente en el diseño y declaraciones que permiten optimizar los estados.



Las fases de la de la metodología OMT son:

- Conceptualización
- Análisis
- Diseño del sistema
- Mantenimiento
- UML

### **METODOLOGÍA BOOCH**

La metodología Booch se enfoca principalmente al diseño de estado de un proyecto. Booch describe una serie de propiedades generales de los sistemas complejos bien estructurados. Los sistemas construidos con una metodología de análisis y diseño orientado a objetos deben satisfacer estas propiedades.

En el análisis y diseño orientado a objetos, el dominio del problema se modela a partir de dos perspectivas distintas. La estructura lógica del sistema y la estructura física. Para cada perspectiva (dinámica y estática) se modela la semántica.

La metodología Booch define diferentes modelos para la descripción de un sistema. El modelo lógico (dominio del problema) se representa en la estructura clase-objeto. En el diagrama de clase, se construye la arquitectura y el modelo estático. El diagrama de objeto, representa la interacción de clases entre sí, captura algunos momentos de la vida del sistema y ayuda en la descripción del modelo dinámico.

La arquitectura del modelo y del proceso describe la ubicación física de las clases en módulos y procesos.

## **METODOLOGÍA OOSE (IVAR JACOBSON)**

Este método proporciona un soporte para el **diseño creativo** de productos de software, las actividades consisten en la transformación de un conjunto de requerimientos en un plan estructurado de construcción y un plan de acción.

El diseño creativo tomando como referencia una base arquitectónica es seguir paso a paso los métodos y procesos con la asistencia de herramientas, para convertir los requerimientos dentro de una arquitectura viable para la construcción de un proyecto incluyendo la creación de prototipos.

A grandes rasgos el modelo consta de cinco pasos generales.

- Modelo de análisis
- Modelo de requerimientos
- Modelo de diseño de objetos
- Modelo de implementación
- Modelo de pruebas

## **METRICA VERSION 3**

La metodología MÉTRICA Versión 3 proporciona un conjunto de métodos y técnicas que guía a los distintos profesionales de Sistemas y Tecnologías de la Información y Comunicaciones (STIC) en la obtención de los diversos productos de los procesos del

ciclo de vida de un proyecto informático. Con el fin de mejorar la productividad de los distintos participantes y asegurar la calidad de los productos resultantes, la mayoría de las técnicas propuestas están soportadas por herramientas disponibles en el mercado que automatizan en mayor o menor grado su utilización.

En la elaboración de MÉTRICA Versión 3 se han tenido en cuenta los métodos de desarrollo más extendidos, así como los últimos estándares de ingeniería del software y calidad, además de referencias específicas en cuanto a seguridad y gestión de proyectos.

También se ha tenido en cuenta la experiencia de los usuarios de las versiones anteriores para solventar los problemas o deficiencias detectados.

En una única estructura la metodología MÉTRICA Versión 3 cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad.

La automatización de las actividades propuestas en la estructura de MÉTRICA Versión 3 es posible ya que sus técnicas están soportadas por una amplia variedad de herramientas de ayuda al desarrollo.

MÉTRICA Versión 3 ha sido concebida para abarcar el desarrollo completo de Sistemas de Información sea cual sea su complejidad y magnitud, por lo cual su estructura responde a desarrollos máximos y deberá adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto.

La metodología descompone cada uno de los procesos en actividades, y éstas a su vez en tareas. Para cada tarea se describe su contenido haciendo referencia a sus principales acciones, productos, técnicas, prácticas y participantes.

El orden asignado a las actividades no debe interpretarse como secuencia en su realización, ya que éstas pueden realizarse en orden diferente a su numeración o bien en paralelo. Sin embargo, no se dará por acabado un proceso hasta no haber finalizado todas las actividades del mismo determinadas al inicio del proyecto.

Así los procesos de la estructura principal de MÉTRICA Versión 3 son los siguientes:

- PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN.
- DESARROLLO DE SISTEMAS DE INFORMACIÓN.
- MANTENIMIENTO DE SISTEMAS DE INFORMACIÓN.

### **Planificación de Sistemas de Información (PSI)**

El objetivo de un Plan de Sistemas de Información es proporcionar un marco estratégico de referencia para los Sistemas de Información de un determinado ámbito de la Organización.

El resultado del Plan de Sistemas debe, por tanto, orientar las actuaciones en materia de desarrollo de Sistemas de Información con el objetivo básico de apoyar la estrategia corporativa, elaborando una arquitectura de información y un plan de proyectos informáticos para dar apoyo a los objetivos estratégicos.

Por este motivo es necesario un proceso como el de Planificación de Sistemas de Información, en el que participen, por un lado los responsables de los procesos de la organización con una visión estratégica y por otro, los profesionales de SI capaces de

enriquecer dicha visión con la aportación de ventajas competitivas por medio de los sistemas y tecnologías de la información y comunicaciones.

Como productos finales de este proceso se obtienen los siguientes, que podrán constituir la entrada para el siguiente proceso de Estudio de Viabilidad del Sistema:

- Catálogo de requisitos de PSI que surge del estudio de la situación actual en el caso de que sea significativo dicho estudio, del diagnóstico que se haya llevado a cabo y de las necesidades de información de los procesos de la organización afectados por el plan de sistemas.
  
- Arquitectura de información que se compone a su vez de los siguientes productos:
  - Modelo de información.
  - Modelo de sistemas de información.
  - Arquitectura tecnológica.
  - Plan de proyectos.
  - Plan de mantenimiento del PSI.

Un Plan de Sistemas de Información proporcionará un marco de referencia en materia de Sistemas de Información. En ocasiones podrá servir de palanca de cambio para los procesos de la Organización, pero su objetivo estará siempre diferenciado del de un análisis de dichos procesos por sí mismos. Dicho en otras palabras, no se debe confundir el resultado que se persigue con un Plan de Sistemas de Información, con el de una mejora o reingeniería de procesos, ya que los objetivos en ambos casos no son los mismos, aunque el medio para conseguirlos tenga puntos en común (estudio de los procesos y alineamiento con los objetivos estratégicos).

Este nuevo enfoque de alineamiento de los sistemas de información con la estrategia de la organización, la implicación directa de la alta dirección y la propuesta de solución presenta como ventajas:

- La implicación de la alta dirección facilita que se pueda desarrollar con los recursos necesarios y el calendario establecido.
- La perspectiva horizontal de los procesos dentro de la Organización facilita que se atienda a intereses globales y no particulares de unidades organizativas que puedan desvirtuar los objetivos del Plan. Para mantener la visión general que apoye los objetivos estratégicos, el enfoque de un Plan de Sistemas de Información debe orientarse al estudio por procesos.
- La prioridad del desarrollo de los sistemas de información de la organización por objetivos estratégicos.
- La propuesta de Arquitectura de Información que se hace en el plan es más estratégica que tecnológica. El modelo de sistemas de información de la propuesta no es teórico y se contemplan los sistemas de información actuales que se mantendrán.

### **Desarrollo de Sistemas de Información**

El proceso de Desarrollo de MÉTRICA Versión 3 contiene todas las actividades y tareas que se deben llevar a cabo para desarrollar un sistema, cubriendo desde el análisis de requisitos hasta la instalación del software. Además de las tareas relativas al análisis, incluye dos partes en el diseño de sistemas: arquitectónico y detallado. Este proceso es, sin duda, el más importante de los identificados en el ciclo de vida de un sistema y se relaciona con todos los demás.

Las actividades y tareas propuestas por la norma se encuentran más en la línea de un desarrollo clásico, separando datos y procesos, que en la de un enfoque orientado a objetos.

En MÉTRICA Versión 3 se han abordado los dos tipos de desarrollo: estructurado y orientado a objeto, por lo que ha sido necesario establecer actividades específicas a realizar en alguno de los procesos cuando se utiliza la tecnología de orientación a objetos. Para este último caso se ha analizado alguna de las propuestas de otras metodologías orientadas a objetos y se han tenido en cuenta la mayoría de las técnicas que contempla UML 1.2 (Unified Modeling Language).

El desarrollo en MÉTRICA Versión 3 lo constituyen los procesos:

- ESTUDIO DE VIABILIDAD DEL SISTEMA (EVS).
- ANÁLISIS DEL SISTEMA DE INFORMACIÓN (ASI).
- DISEÑO DEL SISTEMA DE INFORMACIÓN (DSI).
- CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI).
- IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA (IAS).

### **Estudio de Viabilidad del Sistema (EVS)**

El propósito de este proceso es analizar un conjunto concreto de necesidades, con la idea de proponer una solución a corto plazo. Los criterios con los que se hace esta propuesta no serán estratégicos sino tácticos y relacionados con aspectos económicos, técnicos, legales y operativos.

Los resultados del Estudio de Viabilidad del Sistema constituirán la base para tomar la decisión de seguir adelante o abandonar. Si se decide seguir adelante pueden

surgir uno o varios proyectos que afecten a uno o varios sistemas de información. Dichos sistemas se desarrollarán según el resultado obtenido en el estudio de viabilidad y teniendo en cuenta la cartera de proyectos para la estrategia de implantación del sistema global.

Se ha considerado que este proceso es obligatorio, aunque el nivel de profundidad con el que se lleve a cabo dependerá de cada caso. La conveniencia de la realización del estudio de la situación actual depende del valor añadido previsto para la especificación de requisitos y para el planteamiento de alternativas de solución. En las alternativas se considerarán soluciones "a medida", soluciones basadas en la adquisición de productos software del mercado o soluciones mixtas.

Para valorar las alternativas planteadas y determinar una única solución, se estudiará el impacto en la organización de cada una de ellas, la inversión y los riesgos asociados. El resultado final de este proceso son los productos relacionados con la solución que se propone para cubrir la necesidad concreta que se planteó en el proceso, y que depende de si la solución conlleva desarrollo a medida o no:

- Contexto del sistema (con la definición de las interfaces en función de la solución).
- Impacto en la organización.
- Costo/beneficio de la solución.
- Valoración de riesgos de la solución.
- Enfoque del plan de trabajo de la solución.
- Planificación de la solución.
- Solución propuesta:
  - Descripción de la solución.



- Modelo de descomposición en subsistemas.
- Matriz de procesos/localización geográfica.
- Matriz datos/localización geográfica. Entorno tecnológico y comunicaciones.
- Estrategia de implantación global del sistema.
- Descripción de los procesos manuales.

Si la alternativa incluye desarrollo:

- Modelo abstracto de datos/Modelo de procesos.
- Modelo de negocio/Modelo de dominio.

Si la alternativa incluye un producto software estándar de mercado:

- Descripción del producto.
- Evolución del producto.
- Costos ocasionados por el producto.
- Estándares del producto.
- Descripción de adaptación si es necesaria.

Si en la organización se ha realizado con anterioridad un Plan de Sistemas de Información que afecte al sistema objeto de este estudio, se dispondrá de un conjunto de productos que proporcionarán información a tener en cuenta en todo el proceso.

### **Análisis del Sistema de Información (ASI)**

El propósito de este proceso es conseguir la especificación detallada del sistema de información, a través de un catálogo de requisitos y una serie de modelos que cubran las necesidades de información de los usuarios para los que se desarrollará el sistema de información y que serán la entrada para el proceso de Diseño del Sistema de Información.

Como ya se ha dicho MÉTRICA Versión 3 cubre tanto desarrollos estructurados como orientados a objetos, y las actividades de ambas aproximaciones están integradas en una estructura común aunque presenta alguna actividad exclusiva para cada tipo de desarrollo.

En primer lugar se describe inicialmente el sistema de información, a partir de los productos generados en el proceso Estudio de Viabilidad del Sistema (EVS). Se delimita su alcance, se genera un catálogo de requisitos generales y se describe el sistema mediante unos modelos iniciales de alto nivel.

Se recogen de forma detallada los requisitos funcionales que el sistema de información debe cubrir, catalogándolos, lo que permite hacer la traza a lo largo de los procesos de desarrollo. Además, se identifican los requisitos no funcionales del sistema, es decir, las facilidades que ha de proporcionar el sistema, y las restricciones a que estará sometido, en cuanto a rendimiento, frecuencia de tratamiento, seguridad, etc.

Para facilitar el análisis del sistema se identifican los subsistemas de análisis, y se elaboran los modelos de Casos de Uso y de Clases, en desarrollos orientados a objetos, y de Datos y Procesos en desarrollos estructurados. Se ha incorporado una actividad específica para la definición de Interfaces de Usuario al tiempo que se van obteniendo y depurando los requisitos y los anteriores modelos. Se especificarán todas las interfaces entre el sistema y el usuario, como formatos de pantallas, diálogos, formatos de informes y formularios de entrada.

Finalizados los modelos, se realiza un análisis de consistencia, mediante una verificación y validación, lo que puede forzar la modificación de algunos de los modelos obtenidos.

Una vez realizado dicho análisis de consistencia se elabora el producto *Especificación de Requisitos Software*, que constituye un punto de referencia en el desarrollo del software y la línea base de referencia para las peticiones de cambio sobre los requisitos inicialmente especificados.

En este proceso se inicia también la especificación del Plan de Pruebas, que se completará en el proceso Diseño del Sistema de Información (DSI).

Los productos resultantes del Análisis del Sistema de Información, dependen del tipo de desarrollo de que se trate y se detallan a continuación especificando los que son distintos, según los dos tipos de desarrollo a los que da respuesta MÉTRICA Versión 3:

- Descripción general del entorno tecnológico.
- Glosario de términos.
- Catálogo de normas.
- Catálogo de requisitos.
- Especificación de interfaz de usuario.

Además, en **Análisis Estructurado**:

- Plan de migración y carga inicial de datos.
- Contexto del sistema.

- Matriz de procesos/localización geográfica.
- Descripción de interfaz con otros sistemas.
- Modelo de procesos.
- Modelo lógico de datos normalizado.

Además, en **Análisis Orientado a Objetos**:

- Descripción de subsistemas de análisis.
- Descripción de interfaces entre subsistemas.
- Modelo de clases de análisis.
- Comportamiento de clases de análisis.
- Análisis de la realización de los casos de uso.

En este proceso es muy importante la participación de los usuarios, a través de técnicas interactivas, como diseño de diálogos y prototipos, que permiten al usuario familiarizarse con el nuevo sistema y colaborar en la construcción y perfeccionamiento del mismo.

### **Diseño del Sistema de Información (DSI)**

El propósito del Diseño del Sistema de Información (DSI) es obtener la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información. A partir de dicha información, se generan todas las especificaciones de construcción relativas al propio sistema, así como la especificación técnica del plan de pruebas, la definición de los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial, éstos últimos cuando proceda.

El diseño de la arquitectura del sistema dependerá en gran medida de las características de la instalación, de modo que se ha de tener en cuenta una participación activa de los responsables de Sistemas y Explotación de las Organizaciones para las que se desarrolla el sistema de información.

Este proceso consta de un primer bloque de actividades, que se realizan en paralelo, y cuyo objetivo es obtener el diseño de detalle del sistema de información que comprende la partición física del sistema de información, independiente de un entorno tecnológico concreto, la organización en subsistemas de diseño, la especificación del entorno tecnológico sobre el que se despliegan dichos subsistemas y la definición de los requisitos de operación, administración del sistema, seguridad y control de acceso. En el caso de diseño orientado a objetos, conviene señalar que se ha contemplado que el diseño de la persistencia se lleva a cabo sobre bases de datos relacionales

De este primer bloque de actividades se obtienen los siguientes productos:

- Catálogo de requisitos (se completa).
- Catálogo de excepciones.
- Catálogo de normas para el diseño y construcción.
- Diseño de la arquitectura del sistema.
- Entorno tecnológico del sistema.
- Procedimientos de operación y administración del sistema.
- Procedimientos de seguridad y control de acceso.
- Diseño detallado de los subsistemas de soporte.
- Modelo físico de datos optimizado.
- Asignación de esquemas físicos de datos a nodos.

Además, en ***Diseño Estructurado***:

- Diseño de la arquitectura modular.
- Diseño de interfaz de usuario.

Además, en ***Diseño Orientado a Objetos***:

- Diseño de la realización de casos de uso.
- Modelo de clases de diseño.
- Comportamiento de clases de diseño.
- Diseño de interfaz de usuario.

Al igual que en el proceso de Análisis del Sistema de Información (ASI), antes de proceder a la especificación de los componentes, se realiza una verificación y validación, con el objeto de analizar la consistencia entre los distintos modelos y formalizar la aceptación del diseño de la arquitectura del sistema por parte de los usuarios de Explotación y Sistemas.

Un segundo bloque de actividades complementa el diseño del sistema de información, en el que se generan todas las especificaciones necesarias para la construcción del sistema de información:

- Las especificaciones de construcción de los componentes del sistema (módulos o clases, según el caso) y de las estructuras de datos.
- Los procedimientos de migración y sus componentes asociados.
- La definición y revisión del plan de pruebas, y el diseño de las verificaciones de los niveles de prueba establecidos.
- El catálogo de excepciones que permite establecer un conjunto de verificaciones relacionadas con el propio diseño o con la arquitectura del sistema.

- La especificación de los requisitos de implantación.

### **Construcción del Sistema de Información (CSI)**

La construcción del Sistema de Información (CSI) tiene como objetivo final la construcción y prueba de los distintos componentes del sistema de información, a partir del conjunto de especificaciones lógicas y físicas del mismo, obtenido en el Proceso de Diseño del Sistema de Información (DSI). Se desarrollan los procedimientos de operación y seguridad y se elaboran los manuales de usuario final y de explotación, estos últimos cuando proceda.

Para conseguir dicho objetivo, se recoge la información relativa al producto del diseño Especificaciones de construcción del sistema de información, se prepara el entorno de construcción, se genera el código de cada uno de los componentes del sistema de información y se van realizando, a medida que se vaya finalizando la construcción, las pruebas unitarias de cada uno de ellos y las de integración entre subsistemas.

Si fuera necesario realizar una migración de datos, es en este proceso donde se lleva a cabo la construcción de los componentes de migración y procedimientos de migración y carga inicial de datos.

Como resultado de dicho proceso se obtiene:

- Resultado de las pruebas unitarias.
- Evaluación del resultado de las pruebas de integración.
- Evaluación del resultado de las pruebas del sistema.

- Producto software:
  - Código fuente de los componentes.
  - Procedimientos de operación y administración del sistema.
  - Procedimientos de seguridad y control de acceso.
  - Manuales de usuario.
  - Especificación de la formación a usuarios finales.
  - Código fuente de los componentes de migración y carga inicial de datos.
  - Procedimientos de migración y carga inicial de datos.
  - Evaluación del resultado de las pruebas de migración y carga inicial de datos.

### **Implantación y Aceptación del Sistema (IAS)**

Este proceso tiene como objetivo principal, la entrega y aceptación del sistema en su totalidad, que puede comprender varios sistemas de información desarrollados de manera independiente, según se haya establecido en el proceso de Estudio de Viabilidad del Sistema (EVS), y un segundo objetivo que es llevar a cabo las actividades oportunas para el paso a producción del sistema.

Se establece el plan de implantación, una vez revisada la estrategia de implantación y se detalla el equipo que lo realizará.

Para el inicio de este proceso se toman como punto de partida los componentes del sistema probados de forma unitaria e integrados en el proceso Construcción del Sistema de Información (CSI), así como la documentación asociada. El Sistema se someterá a las Pruebas de Implantación con la participación del usuario de operación



cuya responsabilidad, entre otros aspectos, es comprobar el comportamiento del sistema bajo las condiciones más extremas.

También se someterá a las Pruebas de Aceptación cuya ejecución es responsabilidad del usuario final.

En este proceso se elabora el plan de mantenimiento del sistema de forma que el responsable del mantenimiento conozca el sistema antes de que éste pase a producción.

También se establece el acuerdo de nivel de servicio requerido una vez que se inicie la producción. El acuerdo de nivel de servicio hace referencia a servicios de gestión de operaciones, de soporte a usuarios y al nivel con el que se prestarán dichos servicios.

Como resultado de este proceso se obtienen los siguientes productos:

- Plan de implantación del sistema en su totalidad.
- Equipo de implantación que realizará la implantación.
- Plan de formación del equipo de implantación (esquema, materiales, recursos necesarios, planificación y especificación de la formación de usuarios finales).
- Evaluación de las pruebas de implantación del sistema por parte del usuario de operación.
- Evaluación de las pruebas de aceptación del sistema por parte del usuario final.
- Plan de mantenimiento previo al paso a producción.
- Acuerdo de nivel de servicio del sistema.
- Sistema en producción.

### **Mantenimiento de Sistemas de Información (MSI)**

El objetivo de este proceso es la obtención de una nueva versión de un sistema de información desarrollado con MÉTRICA, a partir de las peticiones de mantenimiento que los usuarios realizan con motivo de un problema detectado en el sistema o por la necesidad de una mejora del mismo.

Como consecuencia de esto, sólo se considerarán en MÉTRICA Versión 3 los tipos de Mantenimiento Correctivo y Evolutivo. Se excluyen los tipos de Mantenimiento Adaptativo y Perfectivo, que abarcan actividades tales como la migración y la retirada de software que precisarían el desarrollo de un tipo de metodología específica para resolver su cometido.

Ante una petición de cambio de un sistema de información ya en producción, se realiza un registro de las peticiones, se diagnostica el tipo de mantenimiento y se decide si se le da respuesta o no, en función del plan de mantenimiento asociado al sistema afectado por la petición, y se establece con qué prioridad. La definición de la solución al problema o necesidad planteada por el usuario que realiza el responsable de mantenimiento, incluye un estudio del impacto, la valoración del esfuerzo y costo, las actividades y tareas del proceso de desarrollo a realizar y el plan de pruebas de regresión.

Los productos que se obtienen en este proceso son los siguientes:

- Catálogo de peticiones de cambio.
- Resultado del estudio de la petición.
- Propuesta de solución.
- Análisis de impacto de los cambios.
- Plan de acción para la modificación.

- Plan de pruebas de regresión.
- Evaluación del cambio.
- Evaluación del resultado de las pruebas de regresión.

## COMPARACION Y CONCLUSION

Ivar Jacobson (OOSE) es el autor de los casos de uso y afirma que su metodología soporta el ciclo total de vida del software orientado-a-objetos. OMT se combina con OOSE en los casos de uso.

La técnica de modelado de objetos (OMT) es considerado ampliamente como uno de los sistemas de análisis orientados a objetos más completos que se han publicado hasta el momento. El OMT tiene la intención de ser un método tanto para el análisis como para el diseño, pero aún cuando contiene un método bastante completo para el análisis, solamente tiende a dar indicaciones prácticas para el diseño. El OMT abarca más temas que la mayoría de los demás métodos, pero sigue siendo incompleto en algunos aspectos y resulta muy complejo aprender y utilizar sus notaciones.

OMT está dividida en tres etapas: análisis, sistema de diseño, y diseño de objetos, además provee técnicas para describir el dominio del problema en tres perspectivas diferentes; la estructura estática de objetos, clases y el comportamiento dinámico de objetos, y la estructura funcional.

Booch describe un número de propiedades generales para una buena estructura de un sistema complejo. El problema es modelado en dos perspectivas diferentes, una estructura lógica del sistema y una estructura física del sistema. La metodología de diseño propuesta por Booch es una de las mejor desarrolladas y contiene una noción de estructura mucho más general.

El método original de Booch comienza por un análisis de flujo de datos, que se utiliza entonces como ayuda para identificar objetos, buscando tanto objetos concretos como objetos abstractos en el espacio del problema, que se encontraran a partir de las burbujas y almacenes de datos en el diagrama de flujo de datos (DFD). Booch utiliza una noción de estructura mucho más general.

La métrica versión 3 distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad y utiliza las normas ISO para su desarrollo completo.

### **Metodología aplicada al proyecto**

- De las metodologías que mencionamos todas nos ofrecen ventajas y desventajas, pero la metodología que mejor se adapta a nuestro proyecto es la metodología de la METRICA Versión 3, ya que nos ofrece una guía completa de lo que necesitamos para el desarrollo de nuestro sistema, además de que nuestro sistema es orientado a objetos y describe de forma detallada cada uno de los pasos que debe seguir nuestro sistema desde la recopilación de la información hasta la finalización del sistema

## ***HERRAMIENTAS DE DISEÑO***

### **Por qué debemos utilizar herramientas CASE en UML?**

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, administrador del proyecto, analistas,

arquitectos, desarrolladores y otros. Las herramientas CASE de modelado con UML nos permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE. Al usar las herramientas CASE:

- Los Analistas de Negocio/Sistemas pueden capturar los requisitos del negocio/sistema con un modelo de casos de uso
- Los Diseñadores/Arquitectos pueden producir el modelo de diseño para articular la interacción entre los objetos o los subsistemas de la misma o de diferentes capas (los diagramas UML típicos que se crean son los de clases y los de interacción)
- Los Desarrolladores pueden transformar rápidamente los modelos en una aplicación funcionando, y buscar un subconjunto de clases y métodos y asimilar el entendimiento de cómo lograr interfaces con ellos.

El modelo actúa como el plano y guiará finalmente la construcción del sistema. De manera semejante, la administración es capaz de ver, puntualmente y desde un alto nivel, una representación del diseño y comprender lo que está sucediendo.

Por estas razones, las herramientas CASE de UML acompañadas con metodologías, nos brindan una forma de representar sistemas demasiado complejos para comprenderlos a través de su código fuente subyacente y nos permiten desarrollar la solución de software correcta más rápido y más económicamente.

Sin embargo, las herramientas CASE varían con respecto a las capacidades de modelado con UML, el soporte del ciclo de vida del proyecto, las ingenierías directa y

reversa, el modelado de datos, la performance, el precio, el soporte, la facilidad de uso, etc. Aquí analizaremos algunas herramientas y veremos cual es la mejor para nuestro sistemas.

## Rational Rose

### Introducción

Rational Rose es una herramienta para “**modelado visual**”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software.

Rational Rose permite completar una gran parte de las **disciplinas** (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto:

- Modelado del negocio
- Captura de requisitos (parcial)
- Análisis y diseño (completo)
- Implementación (como ayuda)
- Control de cambios y gestión de configuración (parte)

### Características

- Admite como notaciones: UML, COM, OMT y Booch
- Realiza Chequeo semántico de los modelos
- Ingeniería “de ida y vuelta”: Rose permite generar código a partir de modelos y viceversa.
- Desarrollo multiusuario
- Integración con modelado de datos
- Generación de documentación

- Tiene un lenguaje de *script* para poder ampliar su funcionalidad
- Soporta OLE
- Disponible en múltiples plataformas

#### Versiones

- *Modeler*: no tiene soporte para lenguajes de programación
- *Professional*: incluye soporte sólo para un lenguaje
- *Enterprise*: múltiples lenguajes, incluyendo C++, Visual C++ (6.0), Visual Basic, Java, CORBA, etc.).

#### Tipos de diagramas que soporta

- de Clases
- de Casos de uso
- de Colaboración
- de Secuencia
- de Componentes
- de Estados
- de Actividad
- de Despliegue

En el árbol de todo modelo aparecen cuatro **vistas**, cada una de las cuales admite determinados tipos de diagramas:

- *Caso de uso*: esta vista define la interacción entre actores y casos de uso.  
Diagramas principales: casos de uso, colaboración, secuencia, actividad.

- *Lógica*: esta vista define fundamentalmente las clases del sistema y sus relaciones. Diagramas principales: clases, estados.
- *Componentes*: esta vista contiene información sobre ficheros, ejecutables y librerías del sistema. Diagrama de componentes.
- *Despliegue*: esta vista muestra la asignación de procesos al hardware. Diagrama de despliegue.

## **Poseidon**

### Introducción

Poseidon para UML es una gran herramienta, no simplemente para su uso como una herramienta de modelado UML cotidiano, sino porque tiene funciones extendidas tales como generador de código, y porque de esta importación/ exportación de interfases que permiten el usuario avanzado desempeñe procesamiento adicional con modelado UML.

### Características

Versiones de Poseidon más avanzadas permiten generar documentos HTML con la documentación y esquemas oportunos.

La versión comunitaria permite generar plantillas de código en Java .La Developer Edition permite además generar plantillas en varios entornos como HTML o C#.

### Versiones

- Estándar Edition
- Professional Edition
- Embedded Edition



- Enterprise Edition
- 

#### Tipos de diagramas que soporta

- Diagramas de casos de uso
- Diagramas de Clases
- Diagramas de Objetos
- Diagramas de Actividad
- Diagramas de Estado
- Diagramas de Secuencia
- Diagrama de Colaboración
- Diagramas de Componentes
- Diagramas de Despliegue

#### Requisitos

Se recomienda Java Runtime Environment o Java Developer Kit. JDK 1.4. Es obligatoria una versión posterior a la 1.3. Poseidon for UML Community Edition 2.0 no funciona con JDK 1.1.X o JRE 1.2.X., etc...

Como mínimo 128 MB de memoria.

Pentium III o equivalente.

No se requiere ningún sistema operativo específico, funciona en Windows 98, 2000 y NT y XP, también en Linux SuSe 6.X, 7.X y Red Hat y MacOS

### **Power designer**

#### Introducción

---

PowerDesigner es una herramienta de desarrollo de aplicaciones de fácil uso. Permite a los diseñadores y desarrolladores mejorar la productividad del ciclo de desarrollo desde el análisis al diseño generando un esquema de base de datos y de objetos de negocio

#### Características

La última versión, PowerDesigner 9.0, presenta nuevas características de modelamiento de procesos, modelamiento mejorado basado en UML, y soporta las técnicas de modelamiento tradicionales y emergentes dentro de un ambiente altamente gráfico. Esto recorta tiempo y complejidad a los proyectos de desarrollo que cubren múltiples plataformas y tipos de código. PowerDesigner también presenta un verdadero repositorio empresarial para almacenar y administrar toda la información de modelamiento y desarrollo de la empresa; esto minimiza inconsistencias y mejora dramáticamente la productividad del desarrollador.

PowerDesigner 9.0 es un entorno verdaderamente integrado para el análisis y diseño de aplicaciones empresariales, con completas capacidades para el modelamiento de negocios, datos y objetos:

- *Modelamiento de Procesos de Negocio:* PowerDesigner le brinda poder a los usuarios no técnicos para diseñar y modelar procesos de negocio en términos reales del negocio, usando un modelo simple, fácil de usar, altamente gráfico, y no técnico. Incluye soporte a la generación e ingeniería reversa de código XML.
- *Modelamiento de Datos:* PowerDesigner diseña y genera el esquema de la base de datos a través de un verdadero modelamiento de bases de datos relacionales de dos niveles (conceptual y físico) - basado en métodos probados. PowerDesigner también soporta técnicas específicas de modelamiento para data *warehouse*.

- *Modelamiento de Objetos:* PowerDesigner completa el análisis y el diseño usando técnicas UML estándar. A partir de un diagrama de clase, PowerDesigner automáticamente genera y realiza ingeniería reversa de ambientes populares como Java (incluyendo EJB 2.0), XML, Servicios Web, C++, PowerBuilder®, VisualBasic® y más, a través de un generador personalizable.
- *Repositorio Empresarial:* La versión Enterprise de PowerDesigner agrega el valor de un repositorio de clase empresarial. El repositorio permite fácilmente visualizar y compartir modelos y otra información entre todos los miembros del equipo de desarrollo. El repositorio es altamente escalable y soporta seguridad basada en roles, control de versiones, búsqueda y generación de reportes.

#### Versiones

Actualmente Sybase esta desarrollando la versión 10 de este software.

#### Tipos de diagramas que soporta

Power Designer soporta modelamiento de datos, de procesos y de objetos con soporte UML.

## **HERRAMIENTAS DE DESARROLLO.**

### **Visual Basic**

#### Introducción

El lenguaje de programación BASIC (Beginner's All purpose Symbolic Instruction Code ) nació en el año 1964 como una herramienta destinado a principiantes, buscando una forma sencilla de realizar programas, empleando un lenguaje casi igual al usado en la vida ordinaria (en inglés), y con instrucciones muy sencillas y escasas.

La evolución del BASIC por los años 70 fue escasa, dado el auge que tomaron en aquella época lenguajes de alto nivel como el FORTRAN y el COBOL. En 1978 se definió una norma para unificar los Basics existentes creándose la normativa BASIC STANDARD

Cabe señalar lo útil de sus prestaciones con otros lenguajes cuando deseamos llegar al fondo de la máquina y controlar uno a uno sus registros. En la mayor parte de las aplicaciones, las herramientas aportadas por VB son mas que suficiente para lograr un programa fácil de realizar y de altas prestaciones.

### Características

Visual-Basic es una herramienta de diseño de aplicaciones para Windows, en la que estas se desarrollan en una gran parte a partir del diseño de una interface gráfica. En una aplicación Visual - Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interface gráfica.

Es por tanto un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos. Combina ambas tendencias. Ya que no podemos decir que VB pertenezca por completo a uno de esos dos tipos de programación, debemos inventar una palabra que la defina: PROGRAMACION VISUAL.

La creación de un programa bajo Visual Basic lleva los siguientes pasos:

- Creación de un interface de usuario. Este interface será la principal vía de comunicación hombre máquina, tanto para salida de datos como para entrada. Será necesario partir de una ventana - Formulario - a la que le iremos añadiendo los controles necesarios.

- Definición de las propiedades de los controles - Objetos - que hayamos colocado en ese formulario. Estas propiedades determinarán la forma estática de los controles, es decir, como son los controles y para qué sirven.
- Generación del código asociado a los eventos que ocurran a estos objetos. A la respuesta a estos eventos (click, doble click, una tecla pulsada, etc.) le llamamos Procedimiento, y deberá generarse de acuerdo a las necesidades del programa.
- Generación del código del programa. Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. Sin embargo, VB ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en unos bloques llamados Módulos, en otros bloques llamados Funciones, y otros llamados Procedimientos. Estos Procedimientos no responden a un evento acaecido a un objeto, sino que responden a un evento producido durante la ejecución del programa.

#### Requisitos para la Instalación.

- Microprocesador Pentium® 90MHz o superior.
- Pantalla VGA de 640x480 o de resolución superior compatible con Microsoft Windows.
- 24 MB de RAM para Windows 95, 32 MB para Windows NT.
- Microsoft Windows NT 3.51 o posterior, o Microsoft Windows 95 o posterior.
- Microsoft Internet Explorer versión 4.01 o posterior (versión 4.01 Service Pack 1 o posterior para programadores de aplicaciones DHTML y 4.x para los usuarios finales de dichas aplicaciones).
- Requisitos de espacio en disco duro:
  - Edición Estándar: instalación típica 48MB, instalación completa 80MB.
  - Edición Profesional: instalación típica 48MB, instalación completa 80MB.

- Edición Empresarial: instalación típica 128MB, instalación completa 147MB.
- Componentes adicionales (si son necesarios): MSDN (para documentación): 67MB, Internet Explorer 4.x: aproximadamente 66MB.
- CD-ROM (no se supone soporte para MS-DOS).

## **SQL SERVER 2000**

### Introducción.

SQL Server 2000 es una herramienta completa de base de datos y análisis. Tanto por la capacidad para consultar la base de datos mediante un explorador como por la compatibilidad con el Lenguaje de marcado extensible (XML, *Extensible Markup Language*), SQL Server 2000 es la base de datos totalmente habilitada para Web. Además, SQL Server 2000 ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de una base de datos empresarial. Tanto si lo que se mide es la velocidad en el desarrollo de aplicaciones como la velocidad del procesamiento de transacciones, SQL Server 2000 es la base de datos más rápida, lo que la convierte en la opción principal para la empresa que busca agilidad en sus operaciones.

### Características.

- Habilitado por completo a la web.
  - Compatibilidad con XML
  - Análisis habilitado para Web
  - Acceso Web a los datos
  - Alojamiento de aplicaciones
  - Seguridad
  - Búsqueda de texto

- Escalable y Confiable.
  - Alta disponibilidad
  - Seguridad
  - Vistas indizadas
  - Duplicación
  
- Menor plazo de Comercialización.
  - Administración de bases de datos simplificada
  - Servicios de transformación de datos
  - Alojamiento Web y de aplicaciones
  - Servicios de análisis (OLAP)

#### Requisitos para la instalación

Procesador	Procesador Intel Pentium o compatible a 166 megahercios (MHz) o superior.
Sistema operativo	Las ediciones Enterprise y Standard de SQL Server 2000 se ejecutan en los sistemas operativos Microsoft Windows NT® Server versión 4.0 Service Pack 5 (SP5) o posterior, Microsoft Windows NT Server 4.0 Enterprise Edition con SP5 o posterior, Microsoft Windows® 2000 Server, Microsoft Windows 2000 Advanced Server y Microsoft Windows 2000 Datacenter Server. Las ediciones Evaluation y Developer de SQL Server 2000 se ejecutan en los sistemas operativos enumerados anteriormente para las ediciones Enterprise y Standard, así como en Microsoft Windows 2000 Professional y Microsoft Windows NT Workstation 4.0 con Service Pack 5 o posterior.

SQL Server 2000 Personal Edition y Desktop Engine se ejecutan en los sistemas operativos indicados anteriormente para las ediciones Enterprise y Standard, así como en Microsoft Windows 98, Microsoft Windows Millennium Edition (Windows Me), Windows 2000 Professional y Windows NT Workstation 4.0 con Service Pack 5 o posterior.

**Memoria**

Enterprise Edition: 64 megabytes (MB) de memoria RAM; se recomienda 128 MB.

Standard Edition: 64 MB.

Evaluation Edition: 64 MB; se recomienda 128 MB.

Developer Edition: 64 MB.

Personal Edition: 64 MB para Windows 2000; 32 MB para otros sistemas operativos.

Desktop Engine: 64 MB para Windows 2000; 32 MB para otros sistemas operativos.

**Disco duro**

Las ediciones Enterprise, Standard, Evaluation, Developer y Personal requieren:

95-270 MB de espacio libre en el disco duro para el servidor; 250 MB para la instalación típica.

50 MB de espacio libre en el disco duro para la instalación mínima de Analysis Services; 130 MB para la instalación típica.

80 MB de espacio libre en el disco duro para Microsoft English Query (admitido en el sistema operativo Windows 2000 pero sin logotipo certificado).

Desktop Engine requiere 44 MB de espacio disponible en el disco duro.

**Unidad**

Unidad de CD-ROM



- Pantalla** Monitor VGA o de resolución superior.
- Otros dispositivos** Microsoft Internet Explorer, versión 5.0 o posterior.
- Microsoft Windows 95, Windows 98, Windows Me, Windows NT 4.0 y Windows 2000 disponen de software de red integrado. Si utiliza Banyan VINES o AppleTalk ADSP será necesario software de red adicional. El protocolo NWLink de redes basadas en Windows proporciona compatibilidad con el cliente IPX/SPX de Novell NetWare.
- Clientes compatibles**
- Se admite Windows 95, Windows 98, Windows Me, Windows NT Workstation 4.0 y Windows 2000 Professional.
- UNIX, Apple Macintosh y OS/2 requieren software de cliente de Conectividad abierta de base de datos (ODBC) de otro fabricante.

## **MYSQL**

### Introducción

La versión 4 de MySQL ha estado en desarrollo desde el 2001. Para cuando te encuentres leyendo este artículo, MySQL 4.0 debe ser una versión estable (o al menos estar en la última versión beta - no terminada aún, pero compatible para el trabajo de desarrollo que se suele requerir).

La mayoría del trabajo de desarrollo en 4.0 se ha enfocado en tres áreas: mejorar las características y la eficiencia existentes, agregar nuevas características, y cambiar la arquitectura del software de MySQL para estar preparados con el futuro crecimiento. Simplemente no tenemos el suficiente espacio aquí para discutir todos los cambios en MySQL 4.0, así que nos enfocaremos en señalar las mejoras más importantes y las nuevas características que acompañan a esta versión.

Antes de continuar, vale la pena mencionar que virtualmente todos los cambios en MySQL 4.0 son completamente transparentes. En la mayoría de los casos se podrán realizar actualizaciones sin cambiar el código de las aplicaciones (aún las primeras versiones de MySQL 4.0 pasaron una prueba extensiva de compatibilidad sin problemas).

### Características

- El optimizador de consultas de MySQL es ahora más inteligente en el uso de índices para resolver las consultas. Algunas consultas que requieren un ordenamiento extra son ahora significativamente más rápidas.
- En MySQL 3.23, se necesitaba recompilar MySQL para ajustar las opciones de índices de texto completo, tales como la longitud mínima de una palabra. En 4.0, las opciones de índices de texto completo han sido movidas al archivo de configuración de MySQL, así que solamente se tienen que hacer las adecuaciones necesarias y reiniciar MySQL para que los cambios tengan efecto. Muchos fallos en las búsquedas de texto completo han sido corregidos también.
- Cambios al código de la caché de claves han producido un significativo aumento en el tiempo de ejecución durante algunas consultas basadas en índices. Esto es especialmente útil en servidores que tienen demasiada carga.
- Si alguna vez has querido eliminar registros relacionados de múltiples tablas al mismo tiempo, ahora MySQL 4.0 dispondrá de borrados multi-tablas. Al especificar múltiples tablas y la cláusula WHERE correcta, MySQL hará sin problemas lo que esperas. También se pueden agregar opciones ORDER BY y LIMIT a las consultas DELETE, para obtener un mejor control sobre cuántos registros son eliminados y el orden en el que son eliminados dichos registros.
- El sistema de replicación de MySQL ha sido mejorado notablemente. Muchos de los cambios fueron hechos en anticipación del sistema de replicación *fail-safe*. En

4.0, el proceso mismo de replicación es multi-hilo en los servidores esclavos. Si el servidor principal llega a fallar, es ahora mucho más probable que cada esclavo tenga los datos necesarios para hacer por sí mismo una recuperación de los datos y trabajar como si fuera el servidor maestro. Los registros de replicación ahora contienen los marcadores de transacción necesarios para asegurarse que las transacciones son replicadas apropiadamente.

- El número de variables de estado en MySQL casi se ha duplicado. Ahora se puede tener un panorama mucho más claro y amplio de lo que está pasando dentro de MySQL. Muchas de las herramientas de administración de MySQL creadas por terceros han sido actualizadas para utilizar estas nuevas variables.
- Cache de consultas: MySQL implementaron una caché de consultas para acelerar las consultas que son ejecutadas comúnmente

#### *Requisitos*

- En cuanto a hardware: Pentium II y 128 Mb de RAM.
- En cuanto a software podemos instalarlo en cualquier versión de Windows de 32 bits, pero obviamente funcionará mucho mejor en un sistema operativo preparado para dar servicios como puede ser Windows 2000 y 2003 Server.
- Necesitamos que esté instalado la pila de protocolos TCP/IP

<b>MARCO TEORICO.....</b>	<b>XV</b>
1.1	Objetivos del Capítulo..... xv
1.2	Introducción ..... xv
1.2.1	JUSTIFICACIÓN ..... xvi
1.2.2	ANTECEDENTES ..... xvii
1.2.3	OBJETIVOS ..... xviii
1.2.3.1	OBJETIVO GENERAL ..... xviii
1.2.3.2	OBJETIVOS ESPECIFICOS ..... xviii
1.2.4	ALCANCE DEL PROYECTO ..... xviii
1.2.5	METODOLOGIA ..... xix
1.3	Cajeros Automáticos. .... xix
1.3.1	Definición de Cajero Automático. .... xx
1.3.2	Tipos de cajeros. .... xx
1.3.3	Servicios que prestan estos tipos de cajeros: ..... xxi
1.3.4	Proceso que sigue una transacción ..... xxi
	Cliente      Terminal ATM                      Central de Aplicación..... xxi
1.3.5	Switch Transaccional ..... xxiv
1.4	Arquitectura de Aplicaciones..... xxv
1.4.1	Tipos de Arquitectura ..... xxvi
1.4.1.1	Arquitectura Centralizada..... xxvi
1.4.1.2	Arquitectura Distribuida ..... xxvii
1.4.1.3	Arquitectura Distribuida Cliente / Servidor ..... xxix
1.4.1.4	Aplicaciones Cliente / Servidor en dos capas ..... xxxii
1.4.1.5	Aplicaciones Cliente / Servidor en tres capas ..... xxxv
1.4.2	Uso de Sockets en el desarrollo de aplicaciones ..... xxxvi
1.4.2.1	Tipos de SOCKETS ..... xxxviii
1.5	Compresión de Imágenes ..... xxxviii
1.5.1	Compresión con perdida y si perdida..... xl
1.5.2	Cantidad de colores ..... xl
1.5.3	Tipos de Ficheros ..... xl
1.5.3.1	TIFF (Tagged Image File Format) ..... xl
1.5.3.2	PNG..... xl
1.5.3.3	GIF (Graphics Interchange Format)..... xl
1.5.3.4	JPG..... xli
1.5.3.5	RAW, BMP, PSP, PSD, ... ..... xlii
1.5.3.6	JPEG ..... xlii
1.6	METODOLOGIAS DE DESARROLLO ..... xliv
1.6.1	Que es una metodología..... xliv
1.6.2	Análisis de metodologías para el desarrollo del software ..... xlv
1.6.2.1	METODOLOGÍA OMT (RUMBAUGH)..... xlvii
1.6.2.2	METODOLOGÍA BOOCH ..... xlix
1.6.2.3	METODOLOGÍA OOSE (IVAR JACOBSON)..... l
1.6.2.4	METRICA VERSION 3..... l
1.6.2.4.1	Planificación de Sistemas de Información (PSI) ..... lii
1.6.2.4.2	Desarrollo de Sistemas de Información..... liv
1.6.2.4.2.1	Estudio de Viabilidad del Sistema (EVS) ..... lv
1.6.2.4.2.2	Análisis del Sistema de Información (ASI) ..... lvii

---

1.6.2.4.2.3	Diseño del Sistema de Información (DSI) .....	lx
1.6.2.4.2.4	Construcción del Sistema de Información (CSI) .....	lxiii
1.6.2.4.2.5	Implantación y Aceptación del Sistema (IAS) .....	lxiv
1.6.2.4.3	Mantenimiento de Sistemas de Información (MSI) .....	lxvi
1.6.3	COMPARACION Y CONCLUSION .....	lxvii
1.6.4	Metodología aplicada al proyecto .....	lxviii
1.7	HERRAMIENTAS DE DISEÑO .....	lxviii
1.7.1	Por qué debemos utilizar herramientas CASE en UML? .....	lxviii
1.7.2	Rational Rose .....	lxx
1.7.3	Poseidon .....	lxxii
•	Diagramas de casos de uso .....	lxxiii
•	Diagramas de Clases .....	lxxiii
•	Diagramas de Objetos .....	lxxiii
•	Diagramas de Actividad .....	lxxiii
•	Diagramas de Estado .....	lxxiii
•	Diagramas de Secuencia .....	lxxiii
•	Diagrama de Colaboración .....	lxxiii
•	Diagramas de Componentes .....	lxxiii
•	Diagramas de Despliegue .....	lxxiii
1.7.4	Power designer .....	lxxiii
	Introducción .....	lxxiii
	Características .....	lxxiv
1.8	HERRAMIENTAS DE DESARROLLO .....	lxxv
1.8.1	Visual Basic .....	lxxv
1.8.2	SQL SERVER 2000 .....	lxxviii
1.8.3	MYSQL .....	lxxxi
	Requisitos .....	lxxxiii

## CAPITULO II

### ANALISIS DEL SISTEMA

#### *Objetivos del Capítulo*

- Conocer el ámbito del problema en el cual se va a involucrar el grupo de desarrolladores del software.
- Verificar y analizar la viabilidad técnica y de conocimiento que se requiere para la elaboración del sistema.
- Analizar la viabilidad económica del sistema con referencia a los beneficios que pueda producir.
- Realizar un documento que sirva de base para el acuerdo entre el cliente y los desarrolladores referentes a las funciones y restricciones detalladas del producto a desarrollar.

#### *Especificación de Requerimientos del Sistema.*

##### **Definición.**

Es un documento que describe la función y el rendimiento del sistema de información, y las restricciones de desarrollo. Se describe la información de entrada y de salida al sistema con el fin de poder tener una guía de verificación y poder comprobar que los requisitos del cliente se encuentran funcionando en el sistema desarrollado.

## **Especificación de Requerimientos del Sistema “MTG”.**

### **Introducción**

Este documento será la base para un acuerdo entre la empresa auspiciante MULTISOFT CIA. LTDA. y los desarrolladores del sistema MTG, con respecto a todo lo que el sistema deberá manejar para llevar un control de las transacciones realizadas dentro de una caja o cajero automático.

#### **a. Objetivos**

- Detallar todos los requerimientos que la empresa auspiciante necesita para el correcto funcionamiento del Sistema MTG
- Dar a conocer en forma general el funcionamiento, especificaciones y tipos de datos que va a manejar el sistema.
- Registrar el detalle específico de cada uno de los componentes que conforman el Sistema MTG

#### **b. Alcance.**

El sistema que vamos a desarrollar se encargará de llevar un control detallado de cada una de las transacciones bancarias realizadas tanto en cajas y cajeros automáticos de cualquier institución financiera almacenando en una base de datos información de dicha transacción además de la foto del cliente que utiliza el servicio, para lo cual se va a proceder a conectar una web cam al computador tanto de la caja como del cajero, y mediante un software a desarrollar se procederá a escuchar peticiones de toma de fotos de parte de un Switch Transaccional (elemento de software que utilizan todos los bancos para el manejo de las transacciones bancarias), la imagen y la información de esta transacción se almacenará localmente en el computador de cada caja y cajero.

Además durante intervalos de tiempo se correrá un proceso que se encargará de almacenar la información de todas las computadoras de las cajas y cajeros de las sucursales que tengan instalado el software de captura de imágenes en un servidor central para un futuro uso ya sea para auditorias o control de seguridades para poder realizar este almacenamiento de imágenes hay que tomar en cuenta algoritmos de compresión de imágenes para su posterior almacenamiento en una base de datos.

El proceso de migración y centralización, es otra parte de un conjunto de aplicaciones que se va ha desarrollar, este tiene como objetivo migrar las bases de datos locales de cada caja y cajero, de cada agencia y sucursal, enviándola a una base de datos central.

Luego se procederá a desarrollar otro software el cual se encargará de la administración de sucursales, cajas, cajeros además de realizar búsquedas de una transacción o grupo transacciones específicas. Este software nos permitirá realizar consultas de transacciones realizadas en cualquier sucursal que cuente con el software de captura de imágenes ya sea por fecha, por número de cuenta, por sucursal, por cajero o caja o por tarjeta, además el software brindará la posibilidad de enviar detalles de transacciones específicas mediante correo electrónico a personas que soliciten esta información, ya sea por carácter de seguridades, auditorias o circunstancias que lo ameriten.

Cabe comentar que este conjunto de sistemas a desarrollar se presenta como una necesidad de las instituciones financieras del país, por lo que surge como una idea planteada entre los bancos y MultiSoft Cia. Ltda., es por este motivo que nosotros nos comprometimos a desarrollar el sistema en un plazo máximo de nueve meses, para lo cual recibiremos la información necesaria para su desarrollo.



El conjunto de aplicaciones que se desarrollarán se la conocerá con el nombre de suite **MTG (Multisoft Transaction Guard)**, nombre que tomará el sistema en el resto de la documentación presentada.

## Descripción General

### **e.a. Perspectiva del Producto**

Este sistema debe ser totalmente independiente, debe ser instalado en cada computador y todas las transacciones deben ser almacenadas localmente. Debe existir una conexión mediante ODBC para conectarse al servidor central en donde se almacenarán todas las transacciones ya sea de las cajas como de los cajeros automáticos

El hardware que se requiere para el uso del software es el siguiente:

- Un servidor Windows XP o superior
- Estaciones de Trabajo para los usuarios que lo requieran
- Web Cam instaladas en las cajas y los cajeros automáticos

### **a.b. Funciones Generales del Producto**

Todos los requerimientos mencionados han sido tomados sobre la base de las necesidades que tienen las empresas financieras como primeros clientes y posibles principales consumidores del Sistema MTG, entre los requerimientos generales que la empresa auspiciante ha dado para el desarrollo del producto se encuentran los siguientes:

- Modular, Flexible y con facilidad de reutilización de código.<sup>1</sup>

---

<sup>1</sup> Reutilización de Código, se desea realizar un producto totalmente modular de tal manera que ciertas Funciones o Clases se puedan utilizar a futuro en otro producto o en una versión superior con nuevos

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

- El Sistema deberá contar con una Interfaz Gráfica de Usuario (GUI) de fácil utilización y muy amigable con el usuario.
- Todo el proceso de Análisis y Diseño del software debe ser correctamente documentado, en los respectivos Manuales que se construirán a lo largo de todo el proceso de Análisis y Diseño del Software.

Nota: Los manuales a entregarse se detallarán más adelante

- Los documentos deben ser claros y sencillos de entender.
- El sistema de captura de imágenes debe ejecutarse de manera invisible para el usuario, evitando de esta manera que su trabajo diario en las cajas no se vea interrumpido, de igual forma para los cajeros automáticos, la persona que realiza una transacción bancaria no conocerá de la existencia de un programa que esta tomándole una foto al instante mismo de realizar un transacción.
- La información de la transacción que se almacenará en la base de datos del banco destinada para este propósito será la siguiente:

- Tipo de Terminal (Caja ó Cajero)
- Número de la cámara
- Fecha de la transacción
- Hora de la transacción
- Número de Referencia
- Número de la tarjeta
- Tipo de transacción
- Número de cuenta
- Valor de la transacción
- Código de error
- Código del host desde donde se realiza la transacción
- Descripción de la transacción

- El sistema será capaz de tomar fotos del cliente sin importar el intervalo de tiempo entre una transacción y otra, para lo cual será una especie de servidor que siempre estará escuchando peticiones de toma de fotos de un switch transaccional, estas fotos se tomarán al instante mismo de realizar dicha petición.
- Las fotos que se tomarán de todas las sucursales del banco serán almacenadas en una única base de datos central, para su posterior uso.
- La congestión que se genera en la red por el almacenamiento de las fotos en la base de datos central debe ser mínima o casi nula, para evitar problemas con otras que se realizan diariamente en el banco.
- Los parámetros de acceso a la base de datos deberán estar encriptados, para que ninguna persona sin autorización pueda manipular la información de las transacciones.
- Las búsquedas de las transacciones que se almacenan en la base de datos central deberán tomar en cuenta los siguientes parámetros o criterios de búsqueda:
  - Por fecha
  - Por Sucursal
  - Por Terminal
  - Por tipo de transacción
  - Por referencia
  - Por Tarjeta
  - Por Cuenta
- Una vez realizada la búsqueda de una determinada transacción, el sistema le permitirá imprimir todos los datos de la misma además de la foto del cliente que realizó en ese momento la transacción bancaria.
- Además el sistema le brindará la posibilidad de enviar los datos de una transacción bancaria específica con su respectiva foto vía email.

- El sistema que maneja todo lo que se refiere a búsquedas de transacciones no será de uso de cualquier persona; solamente personas autorizadas podrán consultar transacciones de la base de datos central.
- Debe existir un sistema en donde pueda describir todas las sucursales, cajas y cajeros en donde se encuentre instalado el software de captura de imágenes para facilitar los criterios de búsqueda de transacciones.

Una vez analizados todos los requerimientos por parte del cliente, analizaremos una serie de módulos que ayudaran a satisfacer estas necesidades de la mejor manera.

Hay que tomar en cuenta que la mejor manera de desarrollar el sistema es dividiéndolo en subsistemas, es por eso que se tratará al software MTG como una suite de programas que tienen funciones específicas dentro del gran sistema de “Control de Transacciones en cajas y cajeros automáticos mediante la captura de imágenes”.

Para una mejor comprensión del sistema a desarrollar lo hemos dividido en varios programas los cuales tienen nombres específicos dentro del gran sistema MTG, cabe aclarar que estos nombres se encuentran relacionados con la tarea principal que hace cada uno de ellos siendo los siguientes:

## **Software MTG SnapShot.**

### **Módulo de Captura de Imágenes.**

Este módulo es el principal responsable de mantener una comunicación permanente con el switch transaccional para atender peticiones de captura de fotos y de transacciones. Trabaja de la siguiente manera: una vez se reenvía una orden de captura, inmediatamente interactúa con la web cam instalada en el equipo y procede a tomar una foto, para luego comprimirla y guardarla en una base de datos local con la demás información de la transacción.

Cabe acotar que todo este proceso de captura de la imagen y los datos, además del almacenamiento en la base de datos local es un proceso invisible para el usuario, por lo que no se verá interrumpido en sus tareas diarias normales.

## **Software MTG Up DataBase.**

### **Módulo de Actualización de la Base de Datos Central.**

Este módulo se encargara de resolver el problema de envío de la información desde las bases de datos locales de todas las sucursales hacia la base de datos central donde convergerá toda la información diaria, además mediante este proceso se resolverá el problema de congestión en la red, ya que este envío no se lo realizará por cada transacción sino mas bien por un conjunto de transacciones, este envío se lo realizará por intervalos de tiempo que serán configurables por el usuario del sistema MTG.

Este software también será invisible para el usuario y será un proceso que se ejecutará permanentemente en la máquina del cliente.

## **Software MTG Viewer.**

---

**Módulo de búsqueda de transacciones.**

Este módulo se encargará específicamente de interactuar con la base de datos central, realizando búsquedas tomando en cuenta los siguientes parámetros: fecha, Sucursal, Terminal, tipo de transacción, referencia, Tarjeta, Cuenta.

Este módulo lo manejan exclusivamente usuarios con privilegios es decir con un login y un password.

Este módulo además se encargará de imprimir transacciones específicas o resultados de alguna búsqueda en particular, además existirá la posibilidad de enviar vía email resultados de búsquedas para personas que necesiten dicha información, tal es el caso problemas con retiros de dinero, estafas o cualquier problema que se pueda suscitar.

**Módulo de Administración de sucursales, Cajas y Cajeros.**

Este módulo es el encargado de administrar las sucursales de un banco en particular, permitiéndonos observar en que lugares se encuentra instalado el software para captura de imágenes, de esta se facilitará las búsquedas y nos permitirá administrar de mejor manera el traspaso de datos desde los clientes hacia la base de datos central.

**Software MTG User.****Módulo de Administración de Usuarios.**

Este módulo será el encargado de proporcionar acceso a usuarios para que manejen el software MTG Viewer, ya que no cualquier persona podrá manejar dicho sistema por cuestiones de seguridad y confidencialidad de la información contenida en la base de datos central.

Es decir asignará a un cliente un *nombre de usuario* y *clave de acceso* para que pueda ingresar o hacer uso del sistema MTG Viewer.

### **b.c. Características del Usuario**

Los usuarios finales deberían estar en capacidad de manejar cualquier versión del Sistema Operativo Windows de Microsoft ya que el sistema va a estar desarrollado bajo este ambiente.

Para la correcta manipulación se requerirá de un pequeño curso de capacitación el cual no tendrá un mayor grado de complejidad porque la aplicación será desarrollada con la mayor claridad posible y considerando el mayor beneficio hacia el usuario final.

### **e.d. Restricciones Generales**

#### **1. Limitaciones en Hardware**

No existe éste tipo de limitaciones porque el Sistema se desarrollará en las instalaciones de la Empresa Auspiciante "MULTISOFT CIA LTDA" la cual proporcionará al equipo desarrollador todo lo necesario para que el sistema cumpla con todos los requisitos planteados.

#### **2. Políticas Reguladoras**

MULTISOFT CIA LTDA por ser una empresa certificada con la norma de calidad ISO 9001:2000 posee ciertas políticas que deberán ser tomadas en cuenta para el desarrollo y la documentación del Sistema.

Las políticas reguladoras que se usarán para el cumplimiento total de los objetivos planteados son:

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

- Cumplir los estándares para el Diseño del Documento de Especificación de Requerimientos del Sistema.
- Cumplir estándares para el nombramiento de variables en la Construcción del Sistema
- En caso de que el Sistema necesite cambios o nuevos requerimientos el personal encargado deberá llenar los registros correspondientes para dichas necesidades.
- Realizar pruebas de calidad del producto cada vez que se termine un Módulo.

### **3. Interfaces a otras Aplicaciones**

Esta aplicación posee una interfaz externa la misma que provee las tramas de datos que deberá ser analizada por el Sistema y así poder presentar en pantalla los estados de los Cajeros Automáticos o de las cajas.

En MULTISOFT CIA LTDA esta interfaz se llama Switch Transaccional que es el software base para la comunicación entre Central de Comunicaciones y una red de Cajeros Automáticos.

### **4. Requerimientos de un lenguaje de más alto orden**

- *Front End*

La Herramienta que se usará es Visual Basic.

- *Back End*

El sistema se desarrollará utilizando SQLServer versión 2000, mysql.

### **Requerimientos Específicos**

#### **a. Requerimientos de Interfaces Externas**



## 1. Interfaces de Usuario

El usuario Normal en muchos casos no sabrá que existe el sistema, el Usuario Administrador interactuará con el producto en un ambiente visual mediante una interfaz gráfica donde habrá botones, menús, etc., que ayudarán a acceder a la información que se requiera de manera organizada y rápida.

## 2. Interfases de Hardware

Los requerimientos básicos del hardware para un correcto desempeño del producto son:

### Características básicas de los Servidores:

Procesador:	Min PENTIUM III
Memoria:	Min 64 Mb
Disco:	Min 10 Gb
Monitor:	14" .28 SVGA Color
Bus de datos:	100Mbps

### Características básicas de las Estaciones de Trabajo y Cajeros Automáticos:

Procesador:	Min PENTIUM II
Memoria:	Min 64 Mb
Disco:	Min 6 Gb
Monitor:	14"
Bus de datos:	100Mbps
Puerto USB	
Webcam USB	

### 3. Interfases de Software

Para que el producto se pueda ejecutar correctamente se necesitará del siguiente software:

- PC clientes:
  - En los clientes que reciben las transacciones.
    - Drivers de Webcam instalada
    - Paquete de distribución Microsoft Data Access Component (MDAC)
  - En los clientes que envían las transacciones
    - Simulador de envío de transacciones
    - Paquete de distribución Microsoft Data Access Component (MDAC)
  
- Servidor de Base de datos
  - Paquete de distribución Microsoft Data Access Component (MDAC)
  - Microsoft SQL Server 2000
  
- Sistemas Operativos
  - Windows 2000 Server para los servidores.
  - Windows 2000 Professional o Win98 para las estaciones de trabajo.
  
- Herramientas CASE
  - Rational Rose para el graficar los diagramas correspondientes al Modelo Estático, Modelo Funcional y Modelo Dinámico.
  - PowerDesigner para graficar el Modelo Lógico y Físico.
  
- Lenguajes de Programación

- Microsoft Visual Basic 6.0

#### 4. Interfaces de Comunicaciones

- Equipos conectados en red usando el protocolo de comunicaciones TCP / IP.
- Puerto que reciba las conexiones hacia el servidor.
- IP privada

#### 4.1. Requerimientos Funcionales

Con formato: Numeración y viñetas

codigo 1	CAPTURA DE TRANSACCION
<b>DESCRIPCION:</b>	Este proceso es el responsable de mantener una comunicación permanente con el SWICHT TRANSACCIONAL, para atender peticiones de captura de transacciones
<b>ENTRADA:</b>	Trama de datos de la transacción
<b>PROCESO:</b>	- Validar longitud de trama de datos
	- Armar la trama de datos identificando cada campo de la transacción
<b>SALIDA:</b>	Almacenamiento en variables cada uno de los datos de la transacción

codigo 2	CAPTURA DE IMAGEN
<b>DESCRIPCION:</b>	Este proceso se encarga de capturar la imagen o foto del cliente, para almacenarla en un archivo en formato

	jpeg
<b>ENTRADA:</b>	Petición de toma de foto a la web cam o cámara digital
<b>PROCESO:</b>	- Captura de drivers de la cámara
	- Toma de foto
	- Compresión a un archivo tipo jpeg
<b>SALIDA:</b>	imagen almacenada en un archivo jpeg

<b>codigo 3</b>	<b>ALMACENAMIENTO EN BASE LOCAL</b>
<b>DESCRIPCION:</b>	Este proceso se encarga de almacenar en una base de datos local la información de la transacción realizada conjuntamente con la imagen del cliente
<b>ENTRADA:</b>	Variables con la información de la trama de datos desglosada
	archivo de imagen jpeg
<b>PROCESO:</b>	Codificación de la imagen en una variable que entienda la base de datos
	Almacenamiento de los datos de la transacción en la base de datos local
<b>SALIDA:</b>	Base de datos local actualizada

<b>codigo 4</b>	<b>ENVIO DE DATOS A BASE DE DATOS CENTRAL</b>
<b>DESCRIPCION:</b>	Este proceso es el encargado de actualizar la base de datos central con los registros de la base de datos local a excepción de la imagen o foto
<b>ENTRADA:</b>	Registros de la base de datos local

<b>PROCESO:</b>	Almacenamiento de cada uno de los campos del registro en variables temporales
	Envío de las variables temporales a la base de datos central
<b>SALIDA:</b>	Base de datos local y central actualizadas

<b>codigo 5</b>	<b>ENVIO DE LA IMAGEN A LA BASE DE DATOS CENTRAL</b>
<b>DESCRIPCION:</b>	Este proceso es el encargado de armar la imagen de la base de datos local para almacenarla en la base de datos central
<b>ENTRADA:</b>	Registro de la imagen de la base de datos local
<b>PROCESO:</b>	Decodificación del registro de la base de datos a un archivo de formato jpeg
<b>SALIDA:</b>	Base de datos local y central actualizadas

<b>codigo 6</b>	<b>ENCRIPCIÓN DE PARÁMETROS DE ACCESO A LA BASE DE DATOS CENTRAL</b>
<b>DESCRIPCION:</b>	Este proceso se encarga de encriptar los parámetros de acceso a la base de datos tales como son el nombre usuario y contraseña
<b>ENTRADA:</b>	Datos de conexión a la base de datos

<b>PROCESO:</b>	- Captura de datos de conexión a la base de datos central
	- Encriptación de los datos
<b>SALIDA:</b>	Archivo con datos de conexión encriptados

<b>codigo 7</b>	<b>BUSQUEDA DE TRANSACCIONES</b>
<b>DESCRIPCION:</b>	Este proceso se encargará específicamente de interactuar con la base de datos central, realizando búsquedas de transacciones específicas
<b>ENTRADA:</b>	Dependiendo del tipo de búsqueda a realizar por:
	Fecha
	Sucursal
	Terminal
	Tipo de transacción
	Tarjeta
	Referencia
	Cuenta
<b>PROCESO:</b>	Captura de los parámetros de búsqueda
	Se envía una consulta a la base central con los parámetros seleccionados
<b>SALIDA:</b>	Datos y foto de la transacción buscada

<b>codigo 8</b>	<b>IMPRESIÓN DE TRANSACCIÓN BUSCADA</b>
<b>DESCRIPCION:</b>	Este proceso se encargará de imprimir los resultados de

	la búsqueda de una transacción específica incluyendo la foto
<b>ENTRADA:</b>	Descripción de la transacción buscada
<b>PROCESO:</b>	Se detecta la impresora instalada en el equipo
	Se envía la orden de impresión
<b>SALIDA:</b>	Hoja impresa con los datos de la transacción buscada

<b>codigo 9</b>	<b>ENVIO DE MAIL</b>
<b>DESCRIPCION:</b>	Este proceso es el encargado de enviar los resultados de una búsqueda vía correo electrónico a personas que necesiten dicha información
<b>ENTRADA:</b>	Descripción de la transacción buscada
<b>PROCESO:</b>	Se detecta el servidor de correo electrónico
	Se captura la dirección del destinatario
	Se efectúa la orden de envío
<b>SALIDA:</b>	Mensaje de envío satisfactorio o error

<b>codigo 10</b>	<b>REGISTRO DE USUARIOS</b>
<b>DESCRIPCION:</b>	Será el encargado de proporcionar acceso a usuarios para que manejen el software MTG Viewer
<b>ENTRADA:</b>	- Datos personales del usuario
	- Datos de acceso al sistema

---

<b>PROCESO:</b>	Se captura los datos de usuario y de acceso, para posteriormente almacenarlos en la base de datos central
<b>SALIDA:</b>	base de datos central actualizada

### **Análisis del Sistema**

Análisis de Sistemas es un proceso secuencial que conlleva a verificar la viabilidad técnica de un Sistema así como los Costos y Beneficios que producirá el Sistema.

El análisis de requisitos es un proceso de refinamiento, en donde se modelan todos los requisitos que se encuentran descritos en el documento de Especificación de requerimientos del Sistema.

### **Desarrollo del Modelo Estático**

Se encarga de presentar de manera comprensible la estructura estática del sistema a desarrollar, es decir expresa el contexto para el desarrollo del software. Se lo representa mediante un *Diagrama de Clases*.

#### **a. Diagrama de Clases del Sistema “MTG”**

El diagrama de Clases es el diagrama principal para el análisis y diseño estático. Un diagrama de clases presenta las clases y objetos del sistema con sus relaciones estructurales y de herencia. La definición de clase u objeto incluye definiciones para atributos y métodos.



El diagrama de Clases debería permitir la especificación en un ámbito de detalle igual o mayor al utilizado para especificar los tres diagramas anteriores. El diagrama de Clases es la especificación de requisitos en su aspecto estático.

Para nuestro sistema en particular se han identificado las siguientes clases:

Usuarios: esta clase contiene a las personas que realizan transacciones bancarias, esta es la persona a la cual se le tomara la foto, es decir en el caso de un cajero automático el usuario es la persona que realiza una transacción bancaria, dicho en otras palabras es la persona que utiliza el cajero automático; en el caso de una caja de un banco o cooperativa el usuario es la persona que se acerca a ventanilla a realizar su deposito, retiro, cobro, etc.

Agencia: es la sucursal de un banco o cooperativa, en lo que se refiere al sistema MTG, Agencia es el lugar en donde se encuentran un grupo de terminales (cajas o ventanillas y cajeros automáticos).

Terminal: es el equipo electrónico (computador de una ventanilla o cajero automático) que cuenta con el sistema MTG, es decir a estos equipos electrónicos se encuentra conectado una cámara digital o webcam para la captura de imágenes de los usuarios.

Tipo Transacción: esta clase representa los tipos de transacciones bancarias que puede realizar la organización financiera por ejemplo, depósitos, retiros, pagos, cobros, etc.

Tipo Usuario: esta clase representa todas las categorías de usuarios, dependiendo de las operaciones que pueda realizar cada unos de ellos en el sistema de mantenimiento, es decir creación, eliminación y búsqueda.

Ciudad: esta clase representa a cada una de las ciudades en las cuales la organización cuenta con una sucursal.

Provincia: esta clase nos indica el grupo de ciudades en las cuales exista una sucursal o agencia del banco o cooperativa.

Transacción Foto: esta clase representa toda la información que pueda generar una transacción bancaria además de la foto o imagen del usuario que realizó una transacción determinada, esta clase se encuentra tanto localmente como centralizada.

Todas las clases mencionadas son representados en el siguiente DIAGRAMA DE CLASES del sistema MTG.

El Diagrama de Clases de Sistema “MTG” se lo puede ver en el ANEXO A.

## **DIAGRAMAS DE CASOS DE USO PARA EL SISTEMA “MTG”**

### Definición:

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja actualmente, o de cómo se desea que trabaje. No es realmente un enfoque orientado a objeto, más bien es un enfoque de construcción de escenarios en los cuales se modelan los procesos del sistema. Sin embargo, constituye un buen modo de llevar a cabo la fase de captura de requisitos del sistema al comienzo del análisis orientado a objeto.

A continuación se detallan cada uno de los casos de uso con los respectivos actores que interviene en el Sistema

### ACTORES DEL SISTEMA “MTG”

**Actor** es un agente externo al sistema, alguien o algo que solicita un servicio al sistema o actúa como catalizador para que ocurra algo. UML especifica que un actor es una clase de objeto, no una instancia particular de una clase. En la Tabla 2.1 se describe los actores que intervienen en el sistema con su respectiva función.

En el sistema “MTG” se van a identificar cinco tipos de actores:

1. Usuario, puede ser un cajero o un Cliente que va a utilizar el cajero automático
2. Administrador
3. Switch
4. Socket
5. Actualizador Base de Datos Central

A continuación se va a describir las funciones de cada uno de los actores

Actor	Función
<b>Administrador</b>	Se encarga de la seguridad, administra los usuarios, administras las contraseñas. Se encarga de Administrar las sucursales que tiene cada institución. Se encarga del control e instalación las cámaras en las terminales de cada sucursal. Se encarga de administrar las búsquedas

	de transacciones definiendo parámetros como tarjeta, referencia, caja o cajero, tipo de transacción y fecha, se encarga del envío de la información de la búsqueda realizada, ya se por vía mail o imprimir la búsqueda
<b>Switch</b>	Se encarga de recibir la información enviada por el usuario desde la caja o cajero automático para enviarla a la red correcta y procesar dicha transacción.
<b>Socket</b>	Se encarga de escuchar continuamente las peticiones de toma de foto, además de recibir la trama de datos con la información de la transacción.
<b>Actualizador Base de Datos Central</b>	Se encarga de almacenar la información en la Base de Datos local para almacenarla en la Base de datos Central.
<b>Usuario</b>	El usuario en mucho de los casos no va a saber que existe el sistema, este se ejecutara de forma oculta.  Se encarga de enviar la información, cada vez que haga una petición de alguna transacción esta información es enviada al Switch

Tabla 2.1: Actores del Sistema "MTG"

## Descripción de los Casos de Uso del Sistema “MTG”

En esta parte del Análisis del Sistema se describirán en forma general los casos de uso que se presentarán en el Sistema, es decir la interacción de los usuarios con sus acciones en cada uno de los procesos que realizan.

A continuación se describen los casos de uso:

### **a.b.** Petición de Transacción

**Caso de** Petición de Transacción

**Uso:**

**Actores:** Usuario/Switch

**Propósito:** Recibir la información enviada por el usuario y enviar la orden para la toma de la foto

**Resumen:** El usuario hace un pedido de una transacción (retiro, deposito, cobro, etc) por medio de una caja o cajero automático, esta información es recibida por el Switch la misma que se encarga de recibir la trama de datos, verificar la información, y de armar la trama de datos.

Como se puede apreciar este es el primer paso que cualquier persona que realice un transacción bancaria la hace indirectamente.

El gráfico correspondiente al Diagrama de Casos de Uso para Petición de Transacción se lo puede apreciar en la Figura 2.1

Con formato: Numeración y viñetas

– **CASO DE USO PARA PETICION DE TRANSACCION**

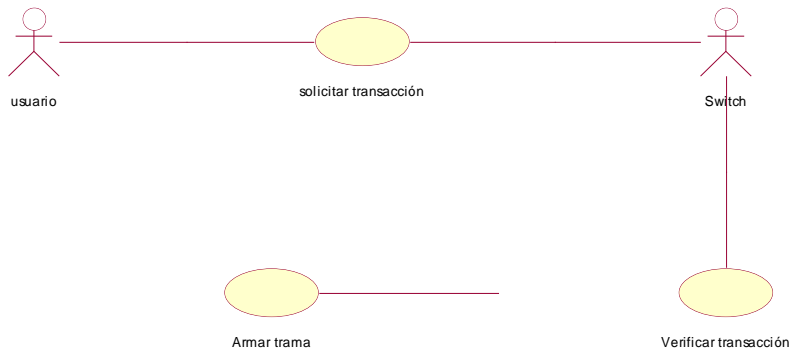


Figura 2.1 Diagrama de Casos de Uso para Petición de Transacción

**b-c. Toma de Foto**

**Caso de** Toma de Foto

**Uso:**

**Actores:** Switch/Socket

**Propósito:** Recibir la trama de la transacción y solicitar la toma de foto o captura de imagen del usuario.

**Resumen:** El Switch envía la trama de datos de la transacción, el socket que esta en escucha continua recibe la petición del Switch, una vez recibida la información, se verifica la trama de datos, verifica que exista una cámara instalada, busca los drivers de la cámara, manda la orden para la toma de la foto, y por último toma la foto.

El gráfico correspondiente al Diagrama de Casos de Uso para Toma de Foto se lo puede apreciar en la Figura 2.2

Con formato: Numeración y viñetas

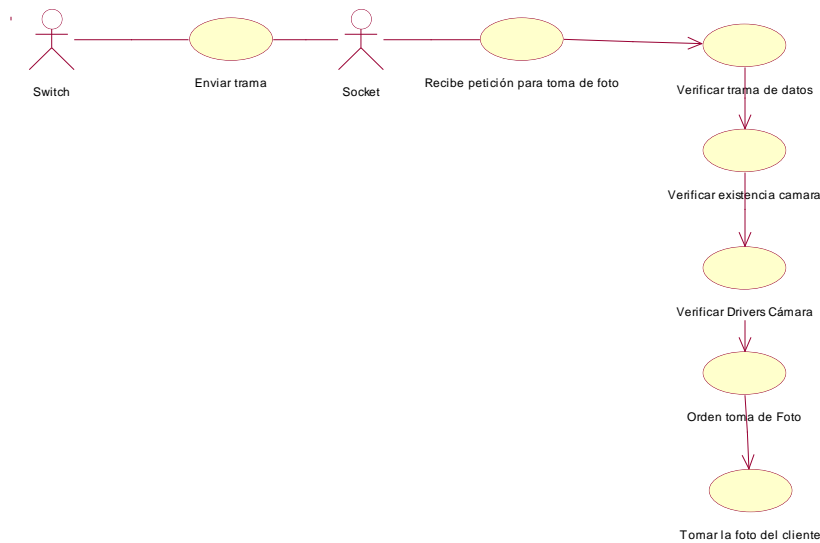


Figura 2.2 Diagrama de Casos de Uso para Toma de Foto

**e.d. Almacenamiento de la Información en Base de Datos Local y Base de Datos Central**

Con formato: Numeración y viñetas

- Caso de** Almacenamiento de la Información en Base de Datos Local y Base de Datos
- Uso:** Central
- Actores:** Socket/Actualizador Base Datos Central
- Propósito:** Almacenar la información de la transacción en Base de Datos Local y Base de Datos Central
- Resumen:** Una vez recibida la trama de datos enviada por el switch se procede a descomponer la misma para almacenarla temporalmente en variables locales las mismas que serán enviadas a la base de datos local, en seguida se lanza la orden para la toma de la foto, la imagen capturada se descompone desde un archivo .jpeg para poder almacenarla en la base de datos local, una vez

almacenada toda la información de cada caja o cajero automático se elimina la imagen capturada en el archivo jpeg.

Todos los datos de la transacción bancaria almacenados en la base de datos local, son almacenados en la base de datos central y posteriormente eliminados del origen, mediante un proceso de actualización, este proceso genera un subproceso de armado de la foto en un archivo .jpeg para en seguida desarmarlo otra vez y almacenarlo en otro formato en la base de datos central.

El gráfico correspondiente al Diagrama de Casos de Uso para Almacenamiento de la Información en Base de Datos Local y Base de Datos Central se lo puede apreciar en la Figura 2.3



**CASO DE USO PARA ALMACENAMIENTO DE INFORMACION EN BASE DE DATOS LOCAL Y CENTRAL**

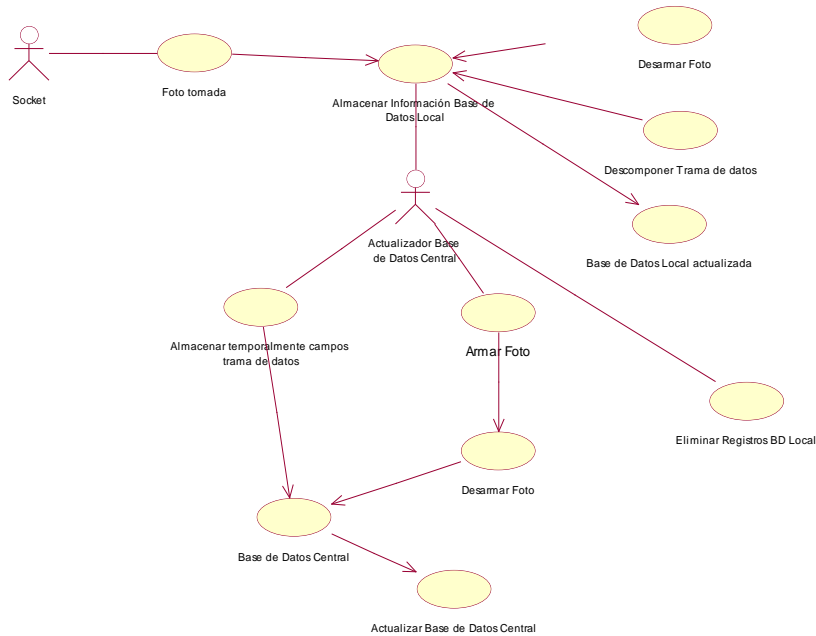


Figura 2.3 Diagrama de Casos de Uso para Almacenamiento de la Información

**d.e. El Administrador**

Con formato: Numeración y viñetas

**Caso de** El Administrador

**Uso:**

**Actores:** Administrador

**Propósito:** Identificar las opciones que tiene el Administrador en el sistema

**Resumen:** El Administrador es el encargado de llevar un control del sistema en lo que se refiere a los siguientes aspectos:

- Administrar las cámaras que se encuentren en una sucursal
- Administrar las sucursales que tiene cada institución financiera

- Administrar las búsquedas de transacciones por cada uno de los siguientes parámetros:
  - o Por tarjeta
  - o Por referencia
  - o Por caja o cajero
  - o Por tipo de transacción
  - o Por fecha
- Se va a encargar del envío de los resultados de la búsqueda vía mail o de la impresión de la búsqueda

El gráfico correspondiente al Diagrama de Casos de Uso para El Administrador se lo puede apreciar en la Figura 2.4



Figura 2.4 Diagrama de Casos de Uso para El Administrador

**e.f. Creación de Usuarios**

**Caso de** Creación de Usuarios

**Uso:**

**Actores:** Administrador

**Propósito:** Registrar los usuarios que podrán utilizar el sistema

**Resumen:** El Administrador es el encargado de registrar todos los usuarios que podrán acceder al sistema

- Crear, eliminar, buscar, modificar nombres de usuario y contraseñas para el ingreso al sistema.

El gráfico correspondiente al Diagrama de Casos de Uso para El Administrador se lo puede apreciar en la Figura 2.5

Con formato: Numeración y viñetas

## CASO DE USO PARA LA CREACIÓN DE USUARIOS

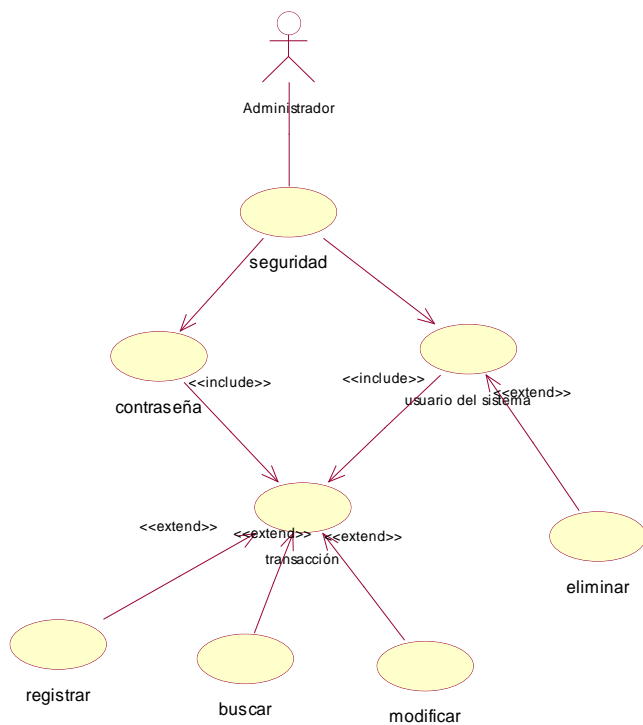


Figura 2.5 Diagrama de Casos de Uso para El Administrador

## Modelo Dinámico

### a. Diagramas de Interacción del Sistema “MTG”

Los diagramas de Interacción son usados para establecer mayor detalle de un escenario del sistema, determinando los objetos y mensajes involucrados.

El diagrama de Secuencia muestra los objetos involucrados en el escenario mediante líneas verticales y punteadas, y los mensajes entre objetos como flechas horizontales conectando líneas de pares de objetos. Los mensajes son dibujados cronológicamente desde arriba hacia abajo.

El diagrama de Colaboración es usado para modelar la interacción entre los objetos de un Caso de Uso. Los objetos están conectados por enlaces (*links*) en los cuales se representan los mensajes enviados acompañados de una flecha que indica su dirección.

El diagrama de Colaboración ofrece una mejor visión del escenario cuando el analista está intentando comprender la participación de un objeto en el sistema.

El sistema MTG cuenta con varios diagramas de secuencia y colaboración para los procesos identificados en los casos de uso descritos anteriormente, los mismos que son detallados a continuación

#### **f.g. Diagramas de Secuencias**

Los diagramas de secuencia han sido realizados y clasificados de acuerdo a los módulos que posee el Sistema, se los puede visualizar en el ANEXO B.

#### **g.h. Diagramas de Colaboración**

De igual manera que los Diagramas de Secuencia estos diagramas han sido clasificados de acuerdo a los módulos del Sistema, los Diagramas de Colaboración del Sistema "MTG" se lo puede ver en el ANEXO C.

<b>CAPITULO II</b> .....	<b>lxxxvi</b>
<b>ANALISIS DEL SISTEMA</b> .....	<b>lxxxvi</b>
2.1 Objetivos del Capítulo .....	lxxxvi
2.2 Especificación de Requerimientos del Sistema.....	lxxxvi
2.1.1 Definición.....	lxxxvi
2.1.2 Especificación de Requerimientos del Sistema “MTG”.....	lxxxvii
2.1.2.1 Introducción.....	lxxxvii
a. Objetivos.....	lxxxvii
b. Alcance.....	lxxxvii
2.1.2.2 Descripción General.....	lxxxix
a. Perspectiva del Producto.....	lxxxix
b. Funciones Generales del Producto.....	lxxxix
c. Características del Usuario.....	xcv
d. Restricciones Generales.....	xcv
1. Limitaciones en Hardware.....	xcv
2. Políticas Reguladoras.....	xcv
3. Interfaces a otras Aplicaciones.....	xcvi
4. Requerimientos de un lenguaje de más alto orden.....	xcvi
2.1.2.3 Requerimientos Específicos.....	xcvi
2.1.2.4 Requerimientos Funcionales.....	xcix
2.3 Análisis del Sistema.....	civ
2.3.1 Desarrollo del Modelo Estático.....	civ
a. Diagrama de Clases del Sistema “MTG”.....	civ
2.3.1.1 DIAGRAMAS DE CASOS DE USO PARA EL SISTEMA “MTG”.....	cvi
2.3.1.2 ACTORES DEL SISTEMA “MTG”.....	cvii
2.3.1.3 Descripción de los Casos de Uso del Sistema “MTG”.....	cix
b. Petición de Transacción.....	cix
c. Toma de Foto.....	cx
d. Almacenamiento de la Información en Base de Datos Local y Base de Datos Central.....	cx
e. El Administrador.....	cxiii
f. Creación de Usuarios.....	cxv
2.3.2 Modelo Dinámico.....	cxvii
a. Diagramas de Interacción del Sistema “MTG”.....	cxvii
g. Diagramas de Secuencias.....	cxviii
h. Diagramas de Colaboración.....	cxviii

## CAPITULO III

### DISEÑO Y DESARROLLO DEL SISTEMA

#### **3.1 Objetivos del Capítulo**

- Moldear los requerimientos del negocio, con la tecnología disponible y diseñar los planos a partir de los cuales se construirá el sistema
- Incluir nuevas clases para facilitar o mejorar la implementación del sistema
- Determinar los accesos y permisos que van a tener los diferentes tipos de usuarios registrados en el Sistema.
- Clasificar los macro procesos que se desarrollan en el sistema para el correcto funcionamiento del mismo.
- Diseñar algoritmos eficientes para la implementación de procesos claves del sistema.

#### **3.2 Definición de Diseño de Sistemas**

El Diseño de Sistemas es el proceso de aplicar ciertas técnicas y principios con el propósito de definir detalladamente un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física, a partir de los detalles y el cumplimiento de la metodología adecuada se puede obtener un Sistema de Calidad.

Para evaluar la calidad del diseño, se deben establecer criterios técnicos como son:

- Debe ser modular, es decir, debe clasificar funciones y subfunciones específicas del sistema, módulos que presenten características de funcionamiento independiente.



- Interfaces sencillas que simplifiquen las conexiones entre los módulos y el entorno exterior.

Para un buen Diseño del Sistema es importante cumplir los siguientes tipos de diseños que son partes de diseño general de un sistema.

- **El Diseño Detallado del Sistema**

Se lo representa mediante del Modelo Dinámico

- **El Diseño del Sistema**

Trasforma el modelo de dominio de la información, creado durante el análisis, en estructuras de datos necesarias para implementar el Software.

- **El Diseño Arquitectónico**

Define la relación entre cada uno de los elementos estructurales del programa.

- **El Diseño de la Interfaz**

Describe como se comunica el sistema consigo mismo, con los sistemas que operan junto con él y con los usuarios que lo emplean.

En el Diseño del Software se fomenta la calidad del Proyecto, aquí es la única manera de materializar con exactitud los requerimientos del cliente.

### **3.3 *Diseño Detallado del Sistema***

#### **3.3.1 Modelo Dinámico**

##### **a. Diagramas de Comportamiento del Sistema “MTG”**

###### **1. Diagrama de Estados**

El diagrama de estados representa el comportamiento del sistema a través del tiempo. Típicamente se elabora un diagrama de Estados para cada clase que tenga un comportamiento significativo. El comportamiento es modelado en términos del estado en el cual se encuentra el objeto, qué acciones se ejecutan en cada estado y cuál es el estado al que transita después de un determinado evento.

Los estados representan condiciones que son válidas en el objeto en un cierto instante. Los eventos representan la causa que origina el cambio desde un estado a otro y están asociados a transiciones, líneas que unen el estado inicial y el final. Las acciones ocurren cuando un objeto llega a un estado.

A continuación se presenta el diagrama de estados para la clase Transacción\_Foto, la más representativa de las clases.

Objeto: Transacción\_Foto

La Tabla 3.1 contiene los estados del objeto “Transacción\_Foto” con sus respectivas descripciones.

La Figura 3.1 muestra los estados que se describen en la Tabla 3.1.

ESTADO	DESCRIPCIÓN
Escucha	El momento de iniciar el sistema este se pone en escucha, hasta recibir una trama correcta del Switch transaccional.
Recibido	Recibe la trama de datos, si la trama es correcta descompone la trama de datos.
Captura Imagen	Si la trama es correcta se procede a la toma de la foto y se almacena en una variable.
Almacenado	Se almacena la trama de datos mas la foto

Tabla 3.1: Estados del Objeto "Transacción\_Foto"

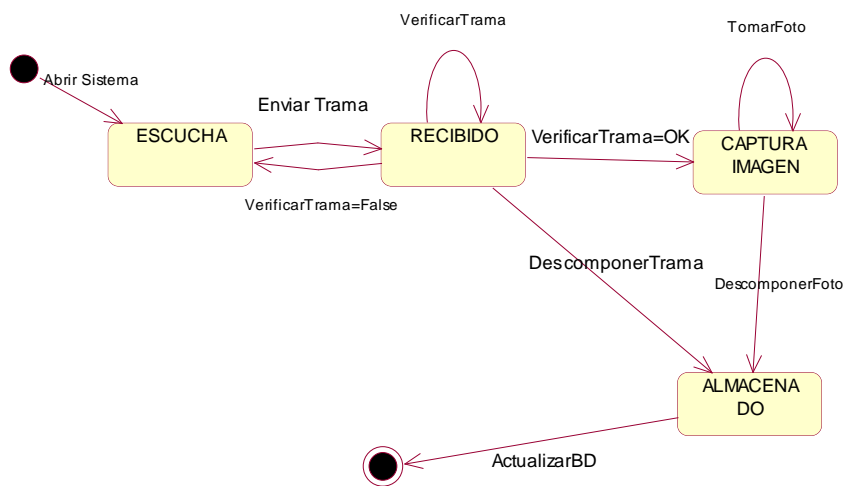


Figura 3.1: Diagrama de Estados del Objeto "Transacción\_Foto"

## 2. Diagramas de Actividad

Este diagrama sirve fundamentalmente para modelar el flujo de control entre actividades. La idea es generar una especie de diagrama Pert, en el que se puede ver el flujo de actividades que tienen lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez. El diagrama de actividades sirve para representar el sistema desde otra perspectiva.

Los diagramas de Actividad de los puede visualizar en el Anexo D.

## 3.4 Diseño del Sistema

### 3.4.1 Estándares de Diseño para la Base de Datos

El nombre de los objetos y de sus atributos va a depender del módulo en donde se hagan referencia.

Los módulos existentes son:

<b>MODULOS</b>	<b>IDENTIFICADOR</b>
Captura de Imágenes.	C
Actualización de la Base de Datos Central.	A
Parámetros de Acceso a la base de datos.	P
Búsqueda de transacciones.	B
Administración de sucursales, Cajas y Cajeros.	S
Administración de Usuarios.	U

### Formato para el Nombre de Tablas

- La primera letra del nombre de la clase debe ser la misma letra que identifica el módulo al cual pertenece.
- La siguiente letra es E indicando que es una entidad
- Siguiendo deben ser tres letras de un nombre corto del objeto al que se hace referencia
- Luego es el nombre largo del objeto, el máximo de caracteres es 25 pero por comodidad en la construcción del Sistema se utilizarán hasta 12 caracteres.

Ejemplo:

#### Nombre de la Tabla AGENCIA

Pertenece al Módulo de Búsqueda de Transacciones B

Los tres caracteres que la identifican AGE

Nombre largo del objeto AGENCIA

Por lo tanto el nombre es BEAGE\_AGENCIA

### Formato para el Nombre de Atributos

El nombre de los atributos depende mucho de la tabla a la que pertenece, así el nombre del atributo CODIGO de la tabla AGENCIA es el siguiente: **BEAGE\_Codigo**

En donde:

- Las primeras cinco letras son de la tabla a la cual pertenece MEAGE
- Las siguientes de preferencia no más de ocho caracteres son el nombre del atributo CODIGO.

### **3.5 *Diseño Arquitectónico del Sistema***

### 3.5.1 Diagrama de Despliegue (Arquitectura Física del Sistema)

La Figura 3.2 indica el esquema en el cual el sistema trabajará, es decir como interactúa con los equipos y dispositivos de comunicaciones.

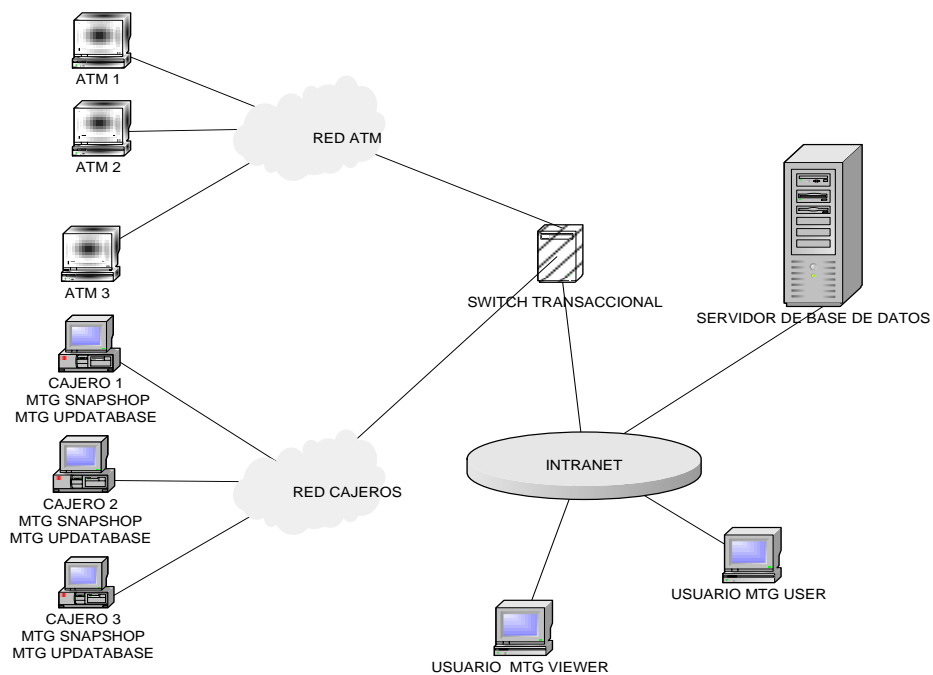


Figura 3.2: Diagrama de Despliegue del Sistema "MTG"

**a. Restricciones Técnicas**

- El sistema construido para recibir e interpretar tramas provenientes del Switch Transaccional.
- El sistema recibe las tramas de los estados de los cajeros y cajas a través de un solo canal o puerto.
- Para el envío de e-mail's, el sistema podrá enviar a cualquier servidor de e-mail's.
- El sistema actúa solo bajo plataformas Windows, por las características de las herramientas utilizadas en la construcción del mismo.

**b. Lenguajes de Programación**

*El sistema es un conjunto de aplicaciones construidas en el lenguaje Visual Basic 6.0*

**c. Sistema de Administración de Datos**

El Motor de Base de Datos es MICROSOFT SQL SERVER 2000

**d. Ambientes de Red**

- Equipos conectados en red usando el protocolo de comunicaciones TCP / IP.
- Puerto que reciba las conexiones hacia el servidor.
- IP privada

**e. Sistemas Operativos**

- El Servidor pueden correr bajo plataformas WINDOWS NT y 2000 SERVER, ya que se tratan de servidores que atienden a uno o varios clientes.



- El cliente MTG corre sobre plataformas WINDOWS a partir de la versión 98.

#### **f. Administrador de transacciones a utilizar**

Para la administración de los datos en Mantenimientos internos se utiliza Microsoft SQL SERVER 2000

Para atender peticiones de uno o vario clientes, pudiendo realizar consultas a través de ODBC hacia SQL SERVER.

#### **g. Requerimientos de Hardware para su implementación**

Los requerimientos básicos del hardware para un correcto desempeño del producto son:

##### **Características básicas de los Servidores:**

Procesador:	Min PENTIUM III
Memoria:	Min 64 Mb
Disco:	Min 10 Gb
Monitor:	14" .28 SVGA Color
Bus de datos:	100Mbps

##### **Características básicas de las Estaciones de Trabajo:**

Procesador:	Min PENTIUM II
Memoria:	Min 64 Mb
Disco:	Min 6 Gb
Monitor:	14"

Bus de datos: 100Mbps

Puerto USB

Webcam USB

#### **h. Requerimientos de Software para su implementación**

Para que el producto se pueda ejecutar correctamente se necesitará del siguiente software:

- **PC clientes:**

- En los clientes que reciben las transacciones.

- Drivers de Webcam instalada

- Paquete de distribución Microsoft Data Access Component (MDAC)

- En los clientes que envían las transacciones

- Simulador de envío de transacciones

- Paquete de distribución Microsoft Data Access Component (MDAC)

- **Servidor de Base de datos**

- Paquete de distribución Microsoft Data Access Component (MDAC)

- Microsoft SQL Server 2000

- **Sistemas Operativos**

- Windows 2000 Server para los servidores.
  - Windows 2000 Professional o Win98 para las estaciones de trabajo.

### 3.5.2 Arquitectura Lógica

Mediante la Figura 3.3 se pretende explicar la distribución de los diferentes componentes del Sistema con las respectivas herramientas con las que van a ser construidos.

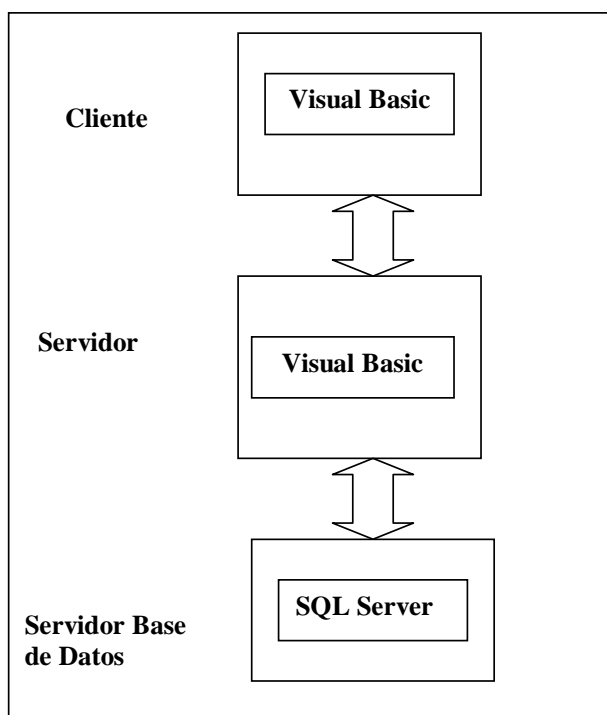


Figura 3.3: Arquitectura Lógica del Sistema "MTG"

### a. Diseño Arquitectónico de Módulos y Submódulos del Sistema

La arquitectura de sistema viene dada en forma de paquetes y las relaciones de dependencia entre ellos. La Figura 3.4 muestra los módulos del sistema y las relaciones que existe entre ellos.

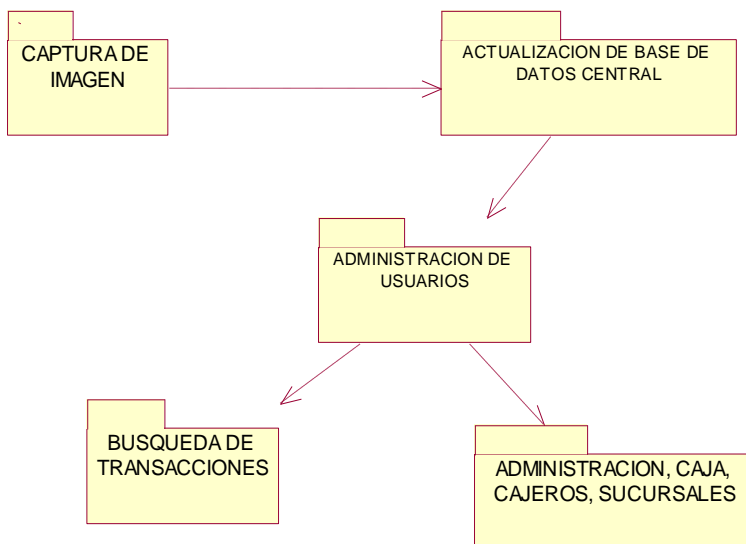
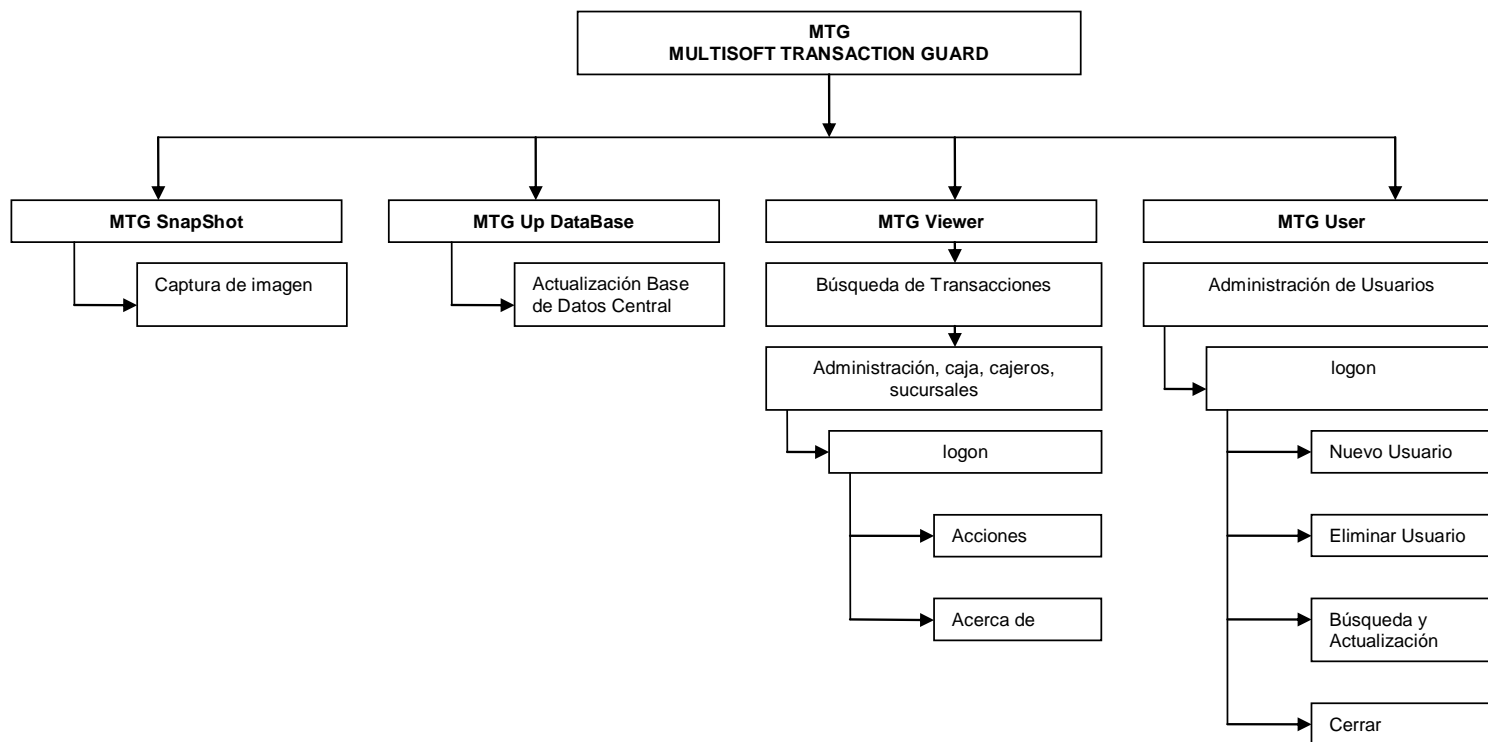


Figura 3.4: Diagrama de Paquetes de Sistema "MTG"

**b. Diseño Secuencial de Pantallas**

El diseño de pantallas que se muestra en la Figura 3.5 corresponde a los sistemas y los módulos de cada sistema con sus respectivos

Menú Principal



*Figura 3.5: Módulos del Sistema "MTG"*

La Figura 3.6 muestra las opciones del Menú Principal del Módulo de Búsqueda de Transacciones y Administración de cajas, cajeros y sucursales del sistema "MTG Viewer"

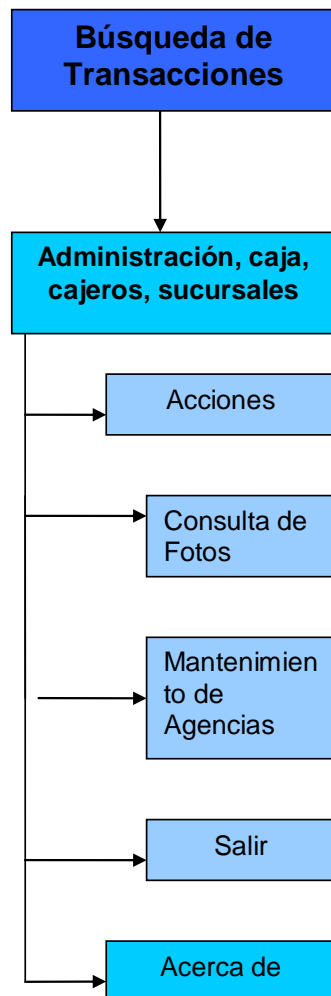


Figura 3.6: Módulo de Búsqueda de Transacciones y Administración de cajas, cajeros y sucursales del sistema "MTG Viewer"

La Figura 3.7 muestra las opciones del Menú Principal del Módulo de Administración de Usuarios del Sistema “MTG User”

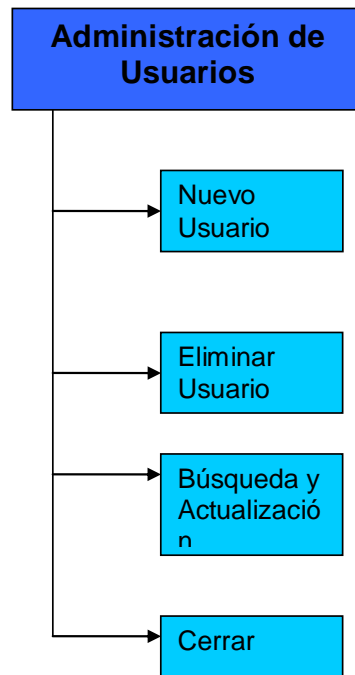


Figura 3.7: Módulo de Administración de Usuarios del Sistema “MTG User”



### 3.6 *Diseño Interfaz Gráfica*

#### a. Ambiente Normal (Visual Basic)

##### Formato de Texto

Descripción	Tamaño	Estilo	Color	Tipo
Etiqueta de campos	10	Negrilla	Negro	Tahoma
Textos campos	10	Normal	Negro	Arial
Mensajes de información	10	Normal	Negro	Times New Roman
Mensajes de error	10	Normal	Negro	Times New Roman
Títulos de Menús	10	Negrilla	Negro	Tahoma
Opciones de Menús	10	Normal	Negro	Tahoma

Tabla 3.2: Formato para Texto

**Objetos utilizados**

Descripción	Ancho	Alto	Color Fondo
Cajas de Texto	Variable de acuerdo a la necesidad	Por defecto	Blanco
Combo Box	Variable de acuerdo a la necesidad	Por defecto	Blanco
Ckeck Box	Por defecto	Por defecto	Blanco
Botones	Variable de acuerdo a la necesidad	35	Plomo

Tabla 3.3: Formato para Objetos

<b>DISEÑO Y DESARROLLO DEL SISTEMA .....</b>	<b>CXX</b>
3.1 Objetivos del Capítulo .....	CXX
3.2 Definición de Diseño de Sistemas .....	CXX
• <b>El Diseño Detallado del Sistema</b> .....	CXXi
Se lo representa mediante del Modelo Dinámico .....	CXXi
• <b>El Diseño del Sistema</b> .....	CXXi
Trasforma el modelo de dominio de la información, creado durante el análisis, en estructuras de datos necesarias para implementar el Software. ....	CXXi
• <b>El Diseño Arquitectónico</b> .....	CXXi
Define la relación entre cada uno de los elementos estructurales del programa .....	CXXi
• <b>El Diseño de la Interfaz</b> .....	CXXi
Describe como se comunica el sistema consigo mismo, con los sistemas que operan junto con él y con los usuarios que lo emplean. ....	CXXi
3.3 Diseño Detallado del Sistema .....	CXXii
3.3.1 Modelo Dinámico .....	CXXii
a. Diagramas de Comportamiento del Sistema “MTG” .....	CXXii
1. Diagrama de Estados .....	CXXii
Objeto: Transacción_Foto .....	CXXii
Tabla 3.1: Estados del Objeto “Transacción_Foto” .....	Xiii
Figura 3.1: Diagrama de Estados del Objeto “Transacción_Foto” ..	xi
2. Diagramas de Actividad .....	CXXiv
3.4 Diseño del Sistema .....	CXXiv
3.4.1 Estándares de Diseño para la Base de Datos .....	CXXiv
Formato para el Nombre de Tablas .....	CXXv
Formato para el Nombre de Atributos .....	CXXv
3.5 Diseño Arquitectónico del Sistema .....	CXXvi
3.5.1 Diagrama de Despliegue (Arquitectura Física del Sistema) .....	127
a. Restricciones Técnicas .....	113
b. Lenguajes de Programación .....	113
El sistema es un conjunto de aplicaciones construidas en el lenguaje Visual Basic 6.0 .....	113
c. Sistema de Administración de Datos .....	113
d. Ambientes de Red .....	113
e. Sistemas Operativos .....	113
f. Administrador de transacciones a utilizar .....	114
g. Requerimientos de Hardware para su implementación .....	114
h. Requerimientos de Software para su implementación .....	115
3.5.2 Arquitectura Lógica .....	116
Figura 3.3: Arquitectura Lógica del Sistema “MTG” .....	xi
a. Diseño Arquitectónico de Módulos y Submódulos del Sistema .....	117
b. Diseño Secuencial de Pantallas .....	118
Figura 3.5: Módulos del Sistema “MTG” .....	119
3.6 Diseño Interfaz Gráfica .....	122
a. Ambiente Normal (Visual Basic) .....	122
Formato de Texto .....	122
Objetos utilizados .....	123

## CAPITULO III

### DISEÑO Y DESARROLLO DEL SISTEMA

#### ***3.1 Objetivos del Capítulo***

- Moldear los requerimientos del negocio, con la tecnología disponible y diseñar los planos a partir de los cuales se construirá el sistema
- Incluir nuevas clases para facilitar o mejorar la implementación del sistema
- Determinar los accesos y permisos que van a tener los diferentes tipos de usuarios registrados en el Sistema.
- Clasificar los macro procesos que se desarrollan en el sistema para el correcto funcionamiento del mismo.
- Diseñar algoritmos eficientes para la implementación de procesos claves del sistema.

#### ***3.2 Definición de Diseño de Sistemas***

El Diseño de Sistemas es el proceso de aplicar ciertas técnicas y principios con el propósito de definir detalladamente un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física, a partir de los detalles y el cumplimiento de la metodología adecuada se puede obtener un Sistema de Calidad.

Para evaluar la calidad del diseño, se deben establecer criterios técnicos como son:

- Debe ser modular, es decir, debe clasificar funciones y subfunciones específicas del sistema, módulos que presenten características de funcionamiento independiente.
- Interfaces sencillas que simplifiquen las conexiones entre los módulos y el entorno exterior.

Para un buen Diseño del Sistema es importante cumplir los siguientes tipos de diseños que son partes de diseño general de un sistema.

- **El Diseño Detallado del Sistema**

Se lo representa mediante del Modelo Dinámico

- **El Diseño del Sistema**

Trasforma el modelo de dominio de la información, creado durante el análisis, en estructuras de datos necesarias para implementar el Software.

- **El Diseño Arquitectónico**

Define la relación entre cada uno de los elementos estructurales del programa.

- **El Diseño de la Interfaz**

Describe como se comunica el sistema consigo mismo, con los sistemas que operan junto con él y con los usuarios que lo emplean.

En el Diseño del Software se fomenta la calidad del Proyecto, aquí es la única manera de materializar con exactitud los requerimientos del cliente.

## **3.7 Diseño Detallado del Sistema**

### **3.7.1 Modelo Dinámico**

#### **a. Diagramas de Comportamiento del Sistema “MTG”**

#### **3. Diagrama de Estados**

El diagrama de estados representa el comportamiento del sistema a través del tiempo. Típicamente se elabora un diagrama de Estados para cada clase que tenga un comportamiento significativo. El comportamiento es modelado en términos del estado en el cual se encuentra el objeto, qué acciones se ejecutan en cada estado y cuál es el estado al que transita después de un determinado evento.

Los estados representan condiciones que son válidas en el objeto en un cierto instante. Los eventos representan la causa que origina el cambio desde un estado a otro y están asociados a transiciones, líneas que unen el estado inicial y el final. Las acciones ocurren cuando un objeto llega a un estado.

A continuación se presenta el diagrama de estados para la clase Transacción\_Foto, la más representativa de las clases.

Objeto: Transacción\_Foto

La Tabla 3.1 contiene los estados del objeto “Transacción\_Foto” con sus respectivas descripciones.

La Figura 3.1 muestra los estados que se describen en la Tabla 3.1.

ESTADO	DESCRIPCIÓN
Escucha	El momento de iniciar el sistema este se pone en escucha, hasta recibir una trama correcta del Switch transaccional.
Recibido	Recibe la trama de datos, si la trama es correcta descompone la trama de datos.
Captura Imagen	Si la trama es correcta se procede a la toma de la foto y se almacena en una variable.
Almacenado	Se almacena la trama de datos mas la foto

Tabla 3.1: Estados del Objeto "Transacción\_Foto"

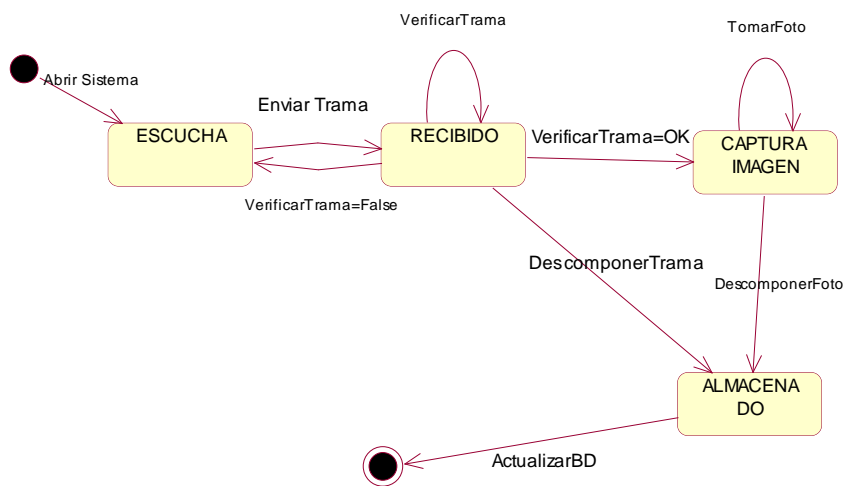


Figura 3.1: Diagrama de Estados del Objeto "Transacción\_Foto"



## 4. Diagramas de Actividad

Este diagrama sirve fundamentalmente para modelar el flujo de control entre actividades. La idea es generar una especie de diagrama Pert, en el que se puede ver el flujo de actividades que tienen lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez. El diagrama de actividades sirve para representar el sistema desde otra perspectiva.

Los diagramas de Actividad de los puede visualizar en el Anexo D.

## 3.8 Diseño del Sistema

### 3.8.1 Estándares de Diseño para la Base de Datos

El nombre de los objetos y de sus atributos va a depender del módulo en donde se hagan referencia.

Los módulos existentes son:

<b>MODULOS</b>	<b>IDENTIFICADOR</b>
Captura de Imágenes.	C
Actualización de la Base de Datos Central.	A
Parámetros de Acceso a la base de datos.	P
Búsqueda de transacciones.	B
Administración de sucursales, Cajas y Cajeros.	S
Administración de Usuarios.	U

### Formato para el Nombre de Tablas

- La primera letra del nombre de la clase debe ser la misma letra que identifica el módulo al cual pertenece.
- La siguiente letra es E indicando que es una entidad
- Siguiendo deben ser tres letras de un nombre corto del objeto al que se hace referencia
- Luego es el nombre largo del objeto, el máximo de caracteres es 25 pero por comodidad en la construcción del Sistema se utilizarán hasta 12 caracteres.

Ejemplo:

#### Nombre de la Tabla AGENCIA

Pertenece al Módulo de Búsqueda de Transacciones B

Los tres caracteres que la identifican AGE

Nombre largo del objeto AGENCIA

Por lo tanto el nombre es BEAGE\_AGENCIA

### Formato para el Nombre de Atributos

El nombre de los atributos depende mucho de la tabla a la que pertenece, así el nombre del atributo CODIGO de la tabla AGENCIA es el siguiente: **BEAGE\_Codigo**

En donde:

- Las primeras cinco letras son de la tabla a la cual pertenece MEAGE
- Las siguientes de preferencia no más de ocho caracteres son el nombre del atributo CODIGO.

### **3.9 *Diseño Arquitectónico del Sistema***

### 3.5.1 Diagrama de Despliegue (Arquitectura Física del Sistema)

La Figura 3.2 indica el esquema en el cual el sistema trabajará, es decir como interactúa con los equipos y dispositivos de comunicaciones.

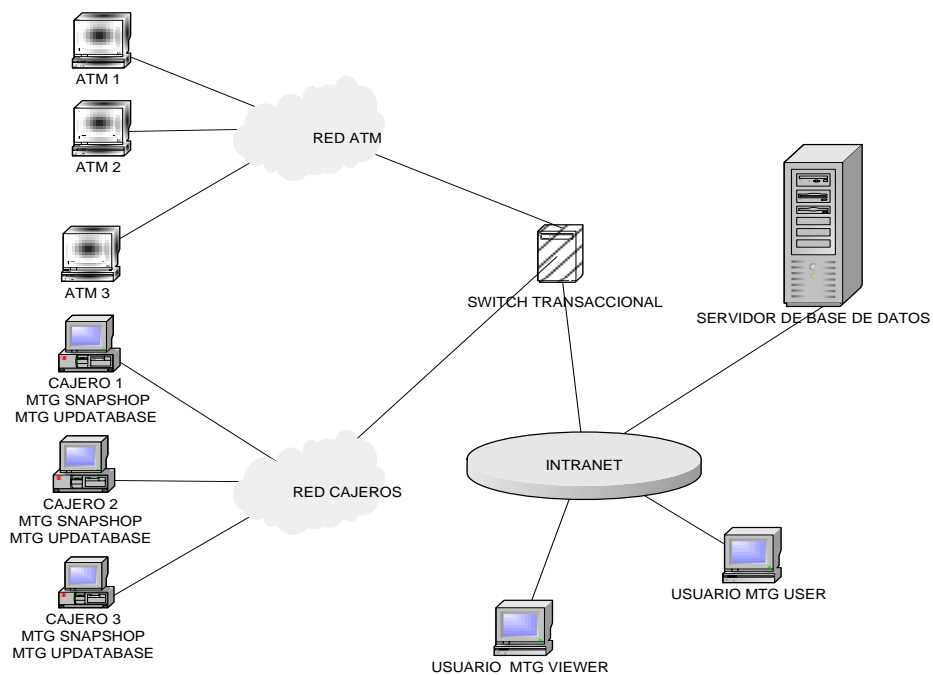


Figura 3.2: Diagrama de Despliegue del Sistema "MTG"

**i. Restricciones Técnicas**

- El sistema construido para recibir e interpretar tramas provenientes del Switch Transaccional.
- El sistema recibe las tramas de los estados de los cajeros y cajas a través de un solo canal o puerto.
- Para el envío de e-mail's, el sistema podrá enviar a cualquier servidor de e-mail's.
- El sistema actúa solo bajo plataformas Windows, por las características de las herramientas utilizadas en la construcción del mismo.

**j. Lenguajes de Programación**

*El sistema es un conjunto de aplicaciones construidas en el lenguaje Visual Basic 6.0*

**k. Sistema de Administración de Datos**

El Motor de Base de Datos es MICROSOFT SQL SERVER 2000

**l. Ambientes de Red**

- Equipos conectados en red usando el protocolo de comunicaciones TCP / IP.
- Puerto que reciba las conexiones hacia el servidor.
- IP privada

**m. Sistemas Operativos**

- El Servidor pueden correr bajo plataformas WINDOWS NT y 2000 SERVER, ya que se tratan de servidores que atienden a uno o varios clientes.

- El cliente MTG corre sobre plataformas WINDOWS a partir de la versión 98.

#### **n. Administrador de transacciones a utilizar**

Para la administración de los datos en Mantenimientos internos se utiliza Microsoft SQL SERVER 2000

Para atender peticiones de uno o vario clientes, pudiendo realizar consultas a través de ODBC hacia SQL SERVER.

#### **o. Requerimientos de Hardware para su implementación**

Los requerimientos básicos del hardware para un correcto desempeño del producto son:

##### **Características básicas de los Servidores:**

Procesador:	Min PENTIUM III
Memoria:	Min 64 Mb
Disco:	Min 10 Gb
Monitor:	14" .28 SVGA Color
Bus de datos:	100Mbps

##### **Características básicas de las Estaciones de Trabajo:**

Procesador:	Min PENTIUM II
Memoria:	Min 64 Mb
Disco:	Min 6 Gb
Monitor:	14"

Bus de datos: 100Mbps

Puerto USB

Webcam USB

#### **p. Requerimientos de Software para su implementación**

Para que el producto se pueda ejecutar correctamente se necesitará del siguiente software:

- **PC clientes:**

- En los clientes que reciben las transacciones.

- Drivers de Webcam instalada

- Paquete de distribución Microsoft Data Access Component (MDAC)

- En los clientes que envían las transacciones

- Simulador de envío de transacciones

- Paquete de distribución Microsoft Data Access Component (MDAC)

- **Servidor de Base de datos**

- Paquete de distribución Microsoft Data Access Component (MDAC)

- Microsoft SQL Server 2000

- **Sistemas Operativos**

- Windows 2000 Server para los servidores.
  - Windows 2000 Professional o Win98 para las estaciones de trabajo.

### 3.5.2 Arquitectura Lógica

Mediante la Figura 3.3 se pretende explicar la distribución de los diferentes componentes del Sistema con las respectivas herramientas con las que van a ser construidos.

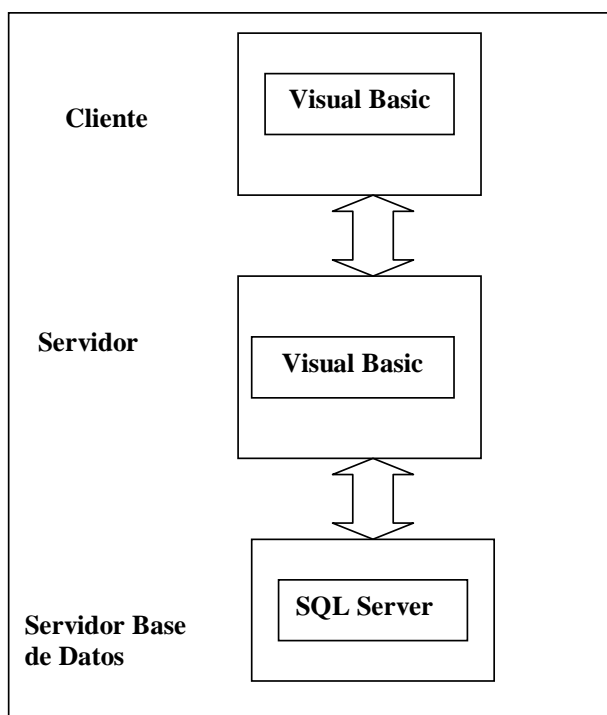


Figura 3.3: Arquitectura Lógica del Sistema "MTG"



### c. Diseño Arquitectónico de Módulos y Submódulos del Sistema

La arquitectura de sistema viene dada en forma de paquetes y las relaciones de dependencia entre ellos. La Figura 3.4 muestra los módulos del sistema y las relaciones que existe entre ellos.

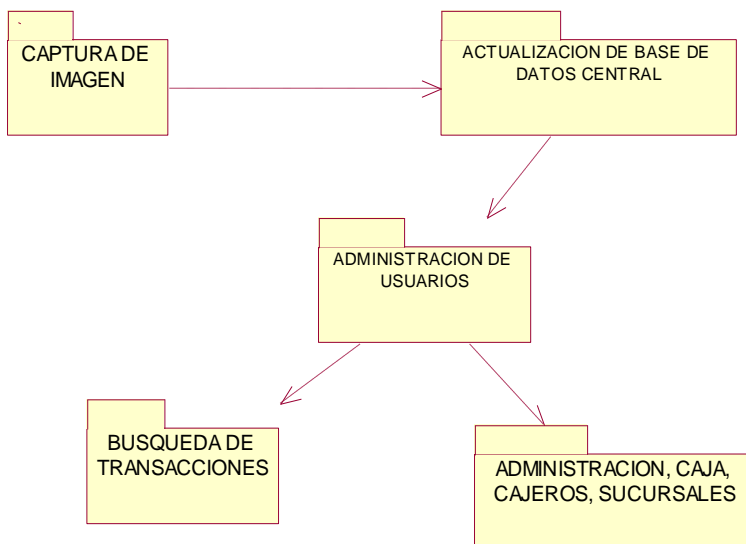
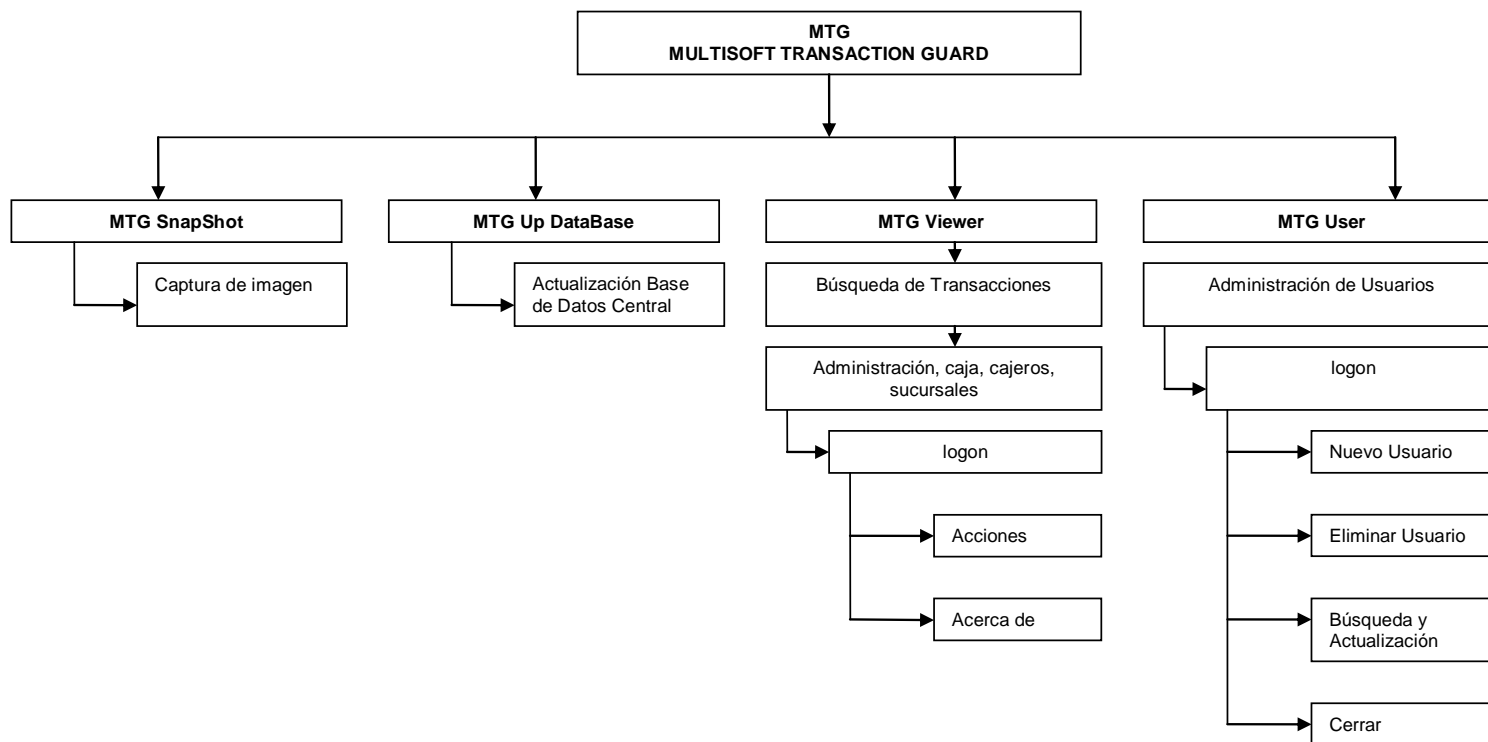


Figura 3.4: Diagrama de Paquetes de Sistema "MTG"

**d. Diseño Secuencial de Pantallas**

El diseño de pantallas que se muestra en la Figura 3.5 corresponde a los sistemas y los módulos de cada sistema con sus respectivos

Menú Principal



*Figura 3.5: Módulos del Sistema "MTG"*

La Figura 3.6 muestra las opciones del Menú Principal del Módulo de Búsqueda de Transacciones y Administración de cajas, cajeros y sucursales del sistema "MTG Viewer"

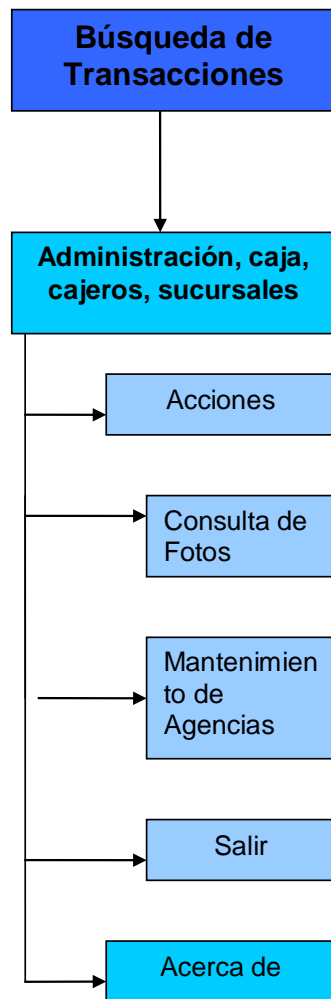


Figura 3.6: Módulo de Búsqueda de Transacciones y Administración de cajas, cajeros y sucursales del sistema "MTG Viewer"

La Figura 3.7 muestra las opciones del Menú Principal del Módulo de Administración de Usuarios del Sistema “MTG User”

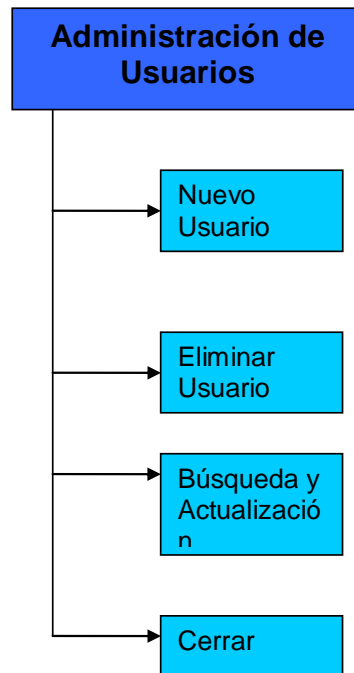


Figura 3.7: Módulo de Administración de Usuarios del Sistema “MTG User”

### 3.10 Diseño Interfaz Gráfica

#### a. Ambiente Normal (Visual Basic)

##### Formato de Texto

Descripción	Tamaño	Estilo	Color	Tipo
Etiqueta de campos	10	Negrilla	Negro	Tahoma
Textos campos	10	Normal	Negro	Arial
Mensajes de información	10	Normal	Negro	Times New Roman
Mensajes de error	10	Normal	Negro	Times New Roman
Títulos de Menús	10	Negrilla	Negro	Tahoma
Opciones de Menús	10	Normal	Negro	Tahoma

Tabla 3.2: Formato para Texto

**Objetos utilizados**

Descripción	Ancho	Alto	Color Fondo
Cajas de Texto	Variable de acuerdo a la necesidad	Por defecto	Blanco
Combo Box	Variable de acuerdo a la necesidad	Por defecto	Blanco
Ckeck Box	Por defecto	Por defecto	Blanco
Botones	Variable de acuerdo a la necesidad	35	Plomo

Tabla 3.3: Formato para Objetos



<b>DISEÑO Y DESARROLLO DEL SISTEMA .....</b>	<b>CXX</b>
3.1 Objetivos del Capítulo .....	CXX
3.2 Definición de Diseño de Sistemas .....	CXX
• <b>El Diseño Detallado del Sistema</b> .....	CXXi
Se lo representa mediante del Modelo Dinámico .....	CXXi
• <b>El Diseño del Sistema</b> .....	CXXi
Trasforma el modelo de dominio de la información, creado durante el análisis, en estructuras de datos necesarias para implementar el Software. ....	CXXi
• <b>El Diseño Arquitectónico</b> .....	CXXi
Define la relación entre cada uno de los elementos estructurales del programa .....	CXXi
• <b>El Diseño de la Interfaz</b> .....	CXXi
Describe como se comunica el sistema consigo mismo, con los sistemas que operan junto con él y con los usuarios que lo emplean. ....	CXXi
3.3 Diseño Detallado del Sistema .....	CXXii
3.3.1 Modelo Dinámico .....	CXXii
a. Diagramas de Comportamiento del Sistema “MTG” .....	CXXii
1. Diagrama de Estados .....	CXXii
Objeto: Transacción_Foto .....	CXXii
Tabla 3.1: Estados del Objeto “Transacción_Foto” .....	Xiii
Figura 3.1: Diagrama de Estados del Objeto “Transacción_Foto” ..	xi
2. Diagramas de Actividad .....	CXXiv
3.4 Diseño del Sistema .....	CXXiv
3.4.1 Estándares de Diseño para la Base de Datos .....	CXXiv
Formato para el Nombre de Tablas .....	CXXv
Formato para el Nombre de Atributos .....	CXXv
3.5 Diseño Arquitectónico del Sistema .....	CXXvi
3.5.1 Diagrama de Despliegue (Arquitectura Física del Sistema) .....	127
a. Restricciones Técnicas .....	113
b. Lenguajes de Programación .....	113
El sistema es un conjunto de aplicaciones construidas en el lenguaje Visual Basic 6.0 .....	113
c. Sistema de Administración de Datos .....	113
d. Ambientes de Red .....	113
e. Sistemas Operativos .....	113
f. Administrador de transacciones a utilizar .....	114
g. Requerimientos de Hardware para su implementación .....	114
h. Requerimientos de Software para su implementación .....	115
3.5.2 Arquitectura Lógica .....	116
Figura 3.3: Arquitectura Lógica del Sistema “MTG” .....	xi
a. Diseño Arquitectónico de Módulos y Submódulos del Sistema .....	117
b. Diseño Secuencial de Pantallas .....	118
Figura 3.5: Módulos del Sistema “MTG” .....	119
3.6 Diseño Interfaz Gráfica .....	122
a. Ambiente Normal (Visual Basic) .....	122
Formato de Texto .....	122
Objetos utilizados .....	123

### **3.7 Herramientas utilizadas en el Diseño y Desarrollo del Sistema “MTG”**

La aplicación a desarrollar tiene como principal característica ser en dos capas, por esto las herramientas a utilizar serán:

- *Front End*

Microsoft Visual Basic 6.0

- *Back End*

El sistema se desarrollará utilizando SQLServer versión 2000, mysql.

### **3.8 Procesos del Sistema “MTG”**

Los procesos que se detallan a continuación se encuentran clasificados por módulos que son los que integran al Sistema.

#### **3.8.1 Módulo de Captura de Imagen**

Este módulo es el principal responsable de mantener una comunicación permanente con el switch transaccional para atender peticiones de captura de fotos y de transacciones. Trabaja de la siguiente manera: una vez se reenvía una orden de captura, inmediatamente interactúa con la web cam instalada en el equipo y procede a verificar los drivers de la cámara y que esté conectada la cámara una vez comprobado esto procede a tomar una foto, para luego comprimirla y guardarla en una base de datos local con la

demás información de la transacción esta información es descompuesta y almacenada en variables locales para posteriormente ser almacenada conjuntamente con la foto.

El sistema captura la dirección IP de la máquina, el puerto por el que recibe la información, también existe una bitácora de registra, donde está si la transacción es válida o errónea.

Cabe acotar que todo este proceso de captura de la imagen y los datos, además del almacenamiento en la base de datos local es un proceso invisible para el usuario, por lo que no se verá interrumpido en sus tareas diarias normales.

### **3.8.2 Módulo de Actualización de la Base de Datos Central.**

Este módulo se encargara de resolver el problema de envío de la información desde las bases de datos locales de todas las sucursales hacia la base de datos central donde convergerá toda la información diaria, además mediante este proceso se resolverá el problema de congestión en la red, ya que este envío no se lo realizará por cada transacción sino mas bien por un conjunto de transacciones, este envío se lo realizará por intervalos de tiempo que serán configurables por el usuario del sistema MTG.

La primera vez que se ejecuta el programa se debe indicar donde se encuentra instalado el programa MTG Snapshot. Una vez indicado el directorio se crea un archivo con extensión **.mer** donde se almacena la ubicación del sistema MTG SnapShot.

Este software también será invisible para el usuario y será un proceso que se ejecutará permanentemente en la máquina del cliente.

### **3.8.3 Módulo de búsqueda de transacciones.**

Este módulo se encargará específicamente de interactuar con la base de datos central, realizando búsquedas tomando en cuenta los siguientes parámetros: fecha, Sucursal, Terminal, tipo de transacción, referencia, Tarjeta, Cuenta.

Este módulo lo manejan exclusivamente usuarios con privilegios es decir con un login y un password.

Este módulo además se encargará de imprimir transacciones específicas o resultados de alguna búsqueda en particular, además existirá la posibilidad de enviar vía email resultados de búsquedas para personas que necesiten dicha información, tal es el caso problemas con retiros de dinero, estafas o cualquier problema que se pueda suscitar.

Todas las consultas se las realiza indicando una fecha de inicio y una fecha de final los tipos de consultas que se pueden realizar son:

- Por Sucursal
- Por Referencia
- Por Tipo\_Transacción
- Por Terminal
- Por Tarjeta o Cuenta.

### **3.8.4 Módulo de Administración de sucursales, Cajas y Cajeros.**

Este módulo es el encargado de administrar las sucursales de un banco en particular, permitiéndonos observar en que lugares se encuentra instalado el software

para captura de imágenes, de esta se facilitará las búsquedas y nos permitirá administrar de mejor manera el traspaso de datos desde los clientes hacia la base de datos central.

Se pueden crear agencias y terminales en cada terminal se debe especificar si es una caja o un cajero automático la dirección IP de la terminal y el puerto por el cual se va a enviar la información.

### **3.8.5 Módulo de Administración de Usuarios.**

Este módulo será el encargado de proporcionar acceso a usuarios para que manejen el software MTG Viewer, ya que no cualquier persona podrá manejar dicho sistema por cuestiones de seguridad y confidencialidad de la información contenida en la base de datos central.

En este módulo se van a poder crear, eliminar, modificar y buscar a los usuarios del sistema.

Es decir asignará a un cliente un *nombre de usuario* y *clave de acceso* para que pueda ingresar o hacer uso del sistema MTG Viewer

---

3.7	Herramientas utilizadas en el Diseño y Desarrollo del Sistema “MTG” .....	iii
3.8	Procesos del Sistema “MTG” .....	iii
3.8.1	Módulo de Captura de Imagen .....	iii
3.8.2	Módulo de Actualización de la Base de Datos Central. ....	iv
3.8.3	Módulo de búsqueda de transacciones.....	v
3.8.4	Módulo de Administración de sucursales, Cajas y Cajeros.....	v
3.8.5	Módulo de Administración de Usuarios. ....	vi

## **CAPITULO IV**

### **PRUEBAS Y RESULTADOS DEL SISTEMA**

#### ***4.1 Objetivos del Capítulo***

- Clasificar los macro procesos que se desarrollan en el sistema para el correcto funcionamiento del mismo.
- Diseñar algoritmos eficientes para la implementación de procesos claves del sistema.
- Definir los resultados obtenidos en la finalización del proyecto.

#### ***4.2 Alcance de Pruebas***

Los aspectos aplicados en la planificación y ejecución de pruebas son las siguientes:

- Utilizar diferentes tipos de pruebas con el fin de poder comprobar la funcionalidad total del sistema.
- Identificar los procesos críticos del sistema que son más propensos a fallas.
- Verificar la aceptación de interfaz gráfica global usada en el sistema
- Probar una o más aplicaciones a la vez.
- Verificar el diseño de integridad relacional que soporta el sistema.

#### ***4.3 Ambiente de pruebas***

Para ejecutar las pruebas se creó un ambiente especial de "Pruebas" con el objetivo de poder obtener resultados que comprueben la calidad del sistema.

El ambiente de pruebas lo conformaron una pequeña red en donde se conectaron 5 computadoras, en donde 4 de ellas actuaron como clientes es decir: dos de ellas actuaron como hosts desde donde se envían las transacciones y las otras dos como host que almacenan las transacciones y las fotos localmente, el quinto equipo vino a funcionar como servidor de Base de Datos central a donde convergen todas las transacciones recibidas por los host.

Además de lo descrito anteriormente fue necesario preparar con el software necesario a todas las computadoras que intervendrían en el proceso de pruebas. Los requerimientos en hardware, software y comunicaciones serán descritos en “Recursos utilizados para ejecutar las pruebas”.

#### ***4.4 Recursos utilizados para ejecutar las pruebas***

A continuación se enumera los recursos utilizados en las pruebas

##### **Recursos Hardware:**

5 computadoras

- **Requerimientos básicos para Pc clientes**

Procesador:	Min PENTIUM II
Memoria:	Min 64 Mb
Disco:	Min 6 Gb
Monitor:	14”
Puerto USB	
Webcam USB	(en los dos clientes)

- **Requerimientos básicos para Servidor**

Procesador:	Min PENTIUM III
Memoria:	Min 128 Mb
Disco:	Min 10 Gb

##### **Requerimientos Software:**

- PC clientes:



En los clientes que reciben las transacciones.

Drivers de Webcam instalada

Paquete de distribución Microsoft Data Access Component (MDAC)

MTG SNAPSHOT

MTG UPDATABASE

En los clientes que envían las transacciones

Simulador de envío de transacciones

Paquete de distribución Microsoft Data Access Component (MDAC)

- Servidor de Base de datos

MTG User

MTG Viewer

Paquete de distribución Microsoft Data Access Component (MDAC)

Microsoft SQL Server 2000

#### **Requerimientos Comunicaciones:**

- Equipos conectados en red usando el protocolo de comunicaciones TCP / IP.
- Puerto que reciba las conexiones hacia el servidor.
- IP privada

#### ***4.5 Generación de Script de Pruebas***

El Script de pruebas es generado bajo la responsabilidad del Departamento de Calidad de la empresa auspiciante Multisoft Cia. Ltda.

Todas las pruebas son ejecutadas bajo la supervisión del departamento de calidad de Multisoft ya que de esto depende la calidad el sistema a distribuirse.

#### 4.5.1 Pruebas de Estándares

Estas pruebas consisten en verificar que se cumplan los estándares tanto en el diseño como en el desarrollo el sistema.

Estándares de Diseño:

- Descripción de tablas de datos
- Descripción de atributos de tablas de datos.

Estándares de Desarrollo:

- Formato de etiquetas de datos
- Formato de ingreso de datos
- Formato de formas (pantallas)
- Formato de ayudas
- Formato de mensajes

Todos los estándares antes mencionados fueron proporcionados por la empresa auspiciante y su verificación en el cumplimiento de las mismas fueron minuciosamente comprobados.

#### 4.5.2 Pruebas Unitarias

Las pruebas unitarias fueron realizadas con previa planificación ya que del correcto funcionamiento de cada una de ellas depende el funcionamiento integral del sistema.

A continuación se detallan cada uno de los aspectos que fueron tomados en cuenta:

**Todos los Módulos:**

- Conexión al servidor de base de datos
- Integridad de datos
- Acceso concurrente a las tablas
- Correcto funcionamiento de las operaciones de inserción, modificación, eliminación y de búsquedas.
- Mensajes de error

- Mensajes de información
- Validación de datos

**Módulo Captura de Imágenes:**

- Verificación del estado del socket cada que se recibe una transacción.
- Correcta comunicación con la cámara instalada.
- Verificación de la calidad de la imagen capturada.
- Correcta validación de las transacciones recibidas.
- Correcto almacenamiento de las transacciones validadas.
- Almacenamiento integro y total de foto.

**Módulo de creación de Usuarios:**

- Correcto almacenamiento de datos de tipos de usuarios
- Correcto almacenamiento de datos de usuarios
- Correcta encriptación de clave de usuario
- Correcta asignación de permisos

**Módulo de Búsquedas y Administración:**

- Correcto almacenamiento de Sucursales, Agencias, Terminales, Tipo de Terminales, Cámaras.
- Correcto despliegue de resultados para los distintos tipos de búsquedas.
- Correcta impresión del resultado de búsqueda en el formato establecido y acordado.
- Correcto envío de mail de los resultados generados por las búsquedas de transacciones.

**Módulo Actualización de Base de Datos:**

- Subida de datos en tiempos no demasiado largos.
- Completa e íntegra subida de la información desde los clientes hacia el servidor.
- Correcta transformación de las fotos a datos que entienda la base de datos del servidor.

#### 4.5.3 Pruebas de Validación de Datos

La validación de datos es otra de las pruebas que a sido tomado en cuenta para que MTG cumpla con todos sus requerimientos.

- Ingreso de datos tipo fecha
- Validación de fechas válidas
- Ingreso de datos tipo caracteres
- Ingreso de datos tipo numéricos
- Ingreso de caracteres en blanco

El resultado de las pruebas de validaciones se las puede verificar en los ANEXO.

#### 4.5.4 Pruebas de Instalación

Las pruebas de instalación sirvieron para verificar que todos los componentes del prototipo están listos para que entren a producción.

En éstas pruebas se tomaron en cuenta los siguientes aspectos:

- Ejecución de Script de Base de Datos
- Ejecución del sistema con Base de Datos en blanco (sin datos)
- Instalación de los programas ejecutables en diferentes versiones de sistemas operativos Windows.
- Pruebas del sistema con distintos tipos de webcam`s.

#### 4.5.5 Pruebas de Seguridad

En las pruebas de seguridad se tomaron en cuenta:

- Acceso de usuarios no registrados en el sistema o usuarios que intentan ingresar con claves incorrectas.
- Comprobación de la encriptación de claves en el ingreso, así como en el registro de usuarios.
- Acceso de usuarios que ya se encuentran inicializados (logueados) en el sistema.
- Duplicidad de nombres y claves de usuarios.
- Restricciones de acceso (permisos) a los usuarios que se encuentran registrados en el sistema.

#### ***4.6 Verificación del proceso de pruebas***

Las pruebas fueron ejecutadas por el jefe del departamento técnico quien es el que se encarga de verificar el cumplimiento de todos los requerimientos que se especificaron para el desarrollo del proyecto y una persona del departamento de calidad quien es la que se encarga de dar veracidad en la ejecución de las pruebas, además que es la que da la aprobación del correcto funcionamiento del sistema.

Todos los resultados que se producen en las pruebas son revisados y en base a los resultados que se produzcan se generan nuevamente otros datos o registros de entrada para evaluar completamente el proyecto, más adelante se detallarán los procesos evaluados con sus respectivos datos de entrada y salida.

Todos los procesos que se evalúan son documentados para que exista veracidad en la ejecución y comprobación de calidad del sistema

#### ***4.7 Documentación de las pruebas***

Toda la información recogida durante este proceso se almacena por escrito para respaldar cada una de las etapas del proceso de pruebas.

## 4.8 Resultados del Sistema

### 4.8.1 Proyecto Obtenido

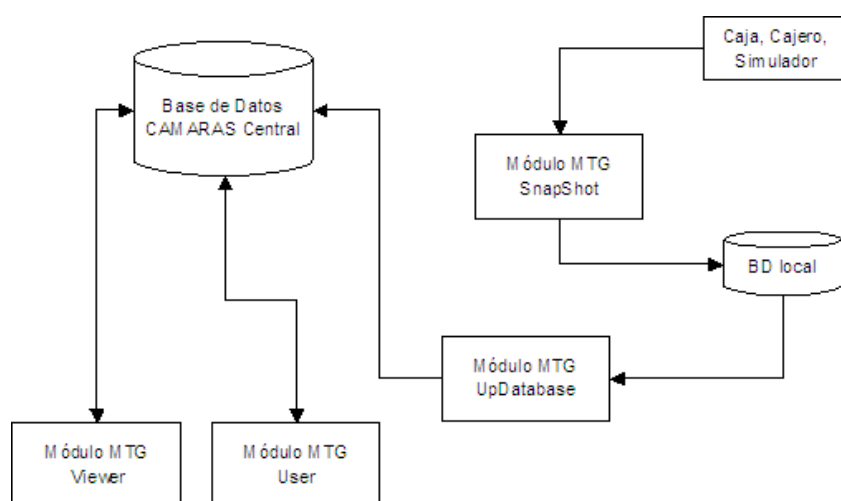


Figura 4.1 Esquema del Proyecto Finalizado

Al finalizar el proyecto se han cubierto los siguientes módulos como se muestra en la Figura 4.1

## **MODULO DE MTG USER**

Este módulo de mantenimientos permite llevar la administración y registro de:

### ***Usuarios***

Se trata de los usuarios del sistema MTG VIEWER.

### ***Tipos de Usuarios***

Se trata del cargo que puede tomar un determinado usuario dentro del sistema. Ejemplo: administrador, operador, etc.

### ***Contraseñas***

Son las contraseñas asignadas a cada usuario del sistema VIEWER.

## **MODULO DE MTG VIEWER**

### ***Agencia***

Se trata de las sucursales que tiene cada institución, en las cuales puede estar ubicado un determinado terminal.

### ***Terminales***

Se trata de los terminales a los cuales se les instalará el MTG SNAPSHOT, puede ser una caja o ventanilla y/o un cajero automático.

### ***Tipo Terminal***

Se trata del tipo de terminal existente en cada una de las sucursales a las cuales se les instaló el MTG SNAPSHOT, es decir un tipo de terminal es una caja o ventanilla y otro tipo de terminal es un cajero automático.

### ***Búsquedas***

Se trata todos los tipos de búsquedas que nos permite realizar el MTG VIEWER (Fecha, Referencia, Tarjeta o Cuenta, Terminal, Tipo Transacción) de las transacciones almacenadas en la base de datos central "CAMARAS", es decir el detalle de las transacciones y la foto del cliente.

### ***Impresión***

Se trata de la impresión del resultado de la búsqueda obtenida, detalle de la transacción y foto en el formato especificado.

### ***Mails***

Se trata del envío de mails a distintas personas de los resultados de la búsqueda obtenida, es decir detalle de la transacción y foto del cliente.

## **MODULO MTG SNAPSHOT**

Es el programa que atiende las peticiones del Switch Transaccional una vez que el cliente de una institución financiera realiza una determinada transacción bancaria, es decir se recibe una trama de datos con la información de la transacción y el MTG SNAPSHOT se encarga de capturar la imagen del cliente, mediante el uso de una webcam o cámara digital previamente configurada e instalada en una ventanilla o cajero automático.

El MTG SNAPSHOT es el responsable también del almacenamiento de los datos obtenidos en cada una de las bases de datos locales, para posteriormente mediante otro módulo subirlas a la base de datos central.

El MTG SNAPSHOT lleva además un control de cada una de las transacciones recibidas en una ventana tipo bitácora, con hora y fecha; en caso de que sucediera algún problema se almacenan los mensajes de error en archivos tipo .log (bitácoras).



## **MODULO UPDATABASE**

Este módulo nos permite actualizar la base de datos central (Servidor), con los datos recogidos de cada uno de los clientes donde se encuentre instalado el MTG SNAPSHOT, este proceso se lo realiza cada vez que el administrador de la red y la base de datos soliciten su ejecución, es decir es un proceso batch de actualización de la base central.

Este proceso no detiene la recepción de transacciones por parte del MTG SNAPSHOT, es independiente y se lo ejecuta en cualquier momento, los datos que se suben a la base de datos central son eliminados de la local.

---

<b>PRUEBAS Y RESULTADOS DEL SISTEMA.....</b>	<b>viii</b>
4.1 Objetivos del Capítulo.....	viii
4.2 Alcance de Pruebas.....	viii
4.3 Ambiente de pruebas.....	viii
4.4 Recursos utilizados para ejecutar las pruebas.....	ix
4.5 Generación de Script de Pruebas.....	x
4.5.1 Pruebas de Estándares.....	xi
4.5.2 Pruebas Unitarias.....	xi
4.5.3 Pruebas de Validación de Datos.....	xiii
4.5.4 Pruebas de Instalación.....	xiii
4.5.5 Pruebas de Seguridad.....	xiii
4.6 Verificación del proceso de pruebas.....	xiv
4.7 Documentación de las pruebas.....	xiv
4.8 Resultados del Sistema.....	xv
4.8.1 Proyecto Obtenido.....	xv
MODULO DE MTG USER.....	xvi
MODULO DE MTG VIEWER.....	xvi
MODULO MTG SNAPSHOT.....	xvii
MODULO UPDATABASE.....	xviii

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 *Objetivos del Capítulo*

- Aportar con las conclusiones que el grupo de desarrollo a adquirido durante el desarrollo del sistema.
- Documentar inconvenientes que el grupo de desarrollo ha tenido en el transcurso del desarrollo del sistema.

#### 6.2 *Conclusiones*

- Una vez analizadas tres herramientas de diseño y modelamiento de datos, hemos concluido que la mejor opción para este propósito es Rational Rose, ya que nos brinda toda la compatibilidad necesaria con el lenguaje UML, además de que es el que mejor maneja el diseño de los diagramas orientados a objetos, a mas de que Multisoft cuenta con las licencias de esta herramienta.
- Todas las herramientas analizadas en el Capítulo I son un estándar utilizado por la empresa Multisoft en lo que se refiere a aplicaciones que interactúan con el Switch Transaccional que maneja la empresa; en lo que respecta a la base de datos MultiSoft cuenta con las licencias de SQL SERVER 2000, además la posibilidad de utilizar MYSQL por su robustez y por la posibilidad de probar otra base de datos con fines académicos y de investigación.
- OMT y UML permitieron una correcta abstracción del problema para plasmar los procesos en papel y permitir un desarrollo eficiente de los mismos, así como el desarrollo de los diagramas requeridos por la metodología para una documentación estándar.

- El modelamiento de datos Orientado a Objetos permiten visualizar de forma más real los procesos o sistema a desarrollar por los que resulta más fácil la obtención de los objetivos planteados.
- Es importante que el documento posea temas relacionados con el contenido del proyecto desarrollado para que la o las personas que deseen estudiar el proyecto puedan hacerlo sin problemas.
- Es preciso tener conocimiento del problema a solucionar para obtener un desarrollo correcto del análisis y un correcto diseño de los diagramas necesarios para la implementación del sistema.
- MTG por ser un producto prospecto para comercializar por Multisoft Cia. Ltda. obtuvo todo el apoyo técnico y económico para su desarrollo por la empresa auspiciante.

### 6.3 Recomendaciones

- Para el manejo de las imágenes es recomendable utilizar el formato **jpeg** ya que produce una excelente calidad incluso con ratio de compresión muy elevados, y como se va almacenar en una base de datos y lo que se trata es de no saturar la base de datos este formato es el ideal para nuestros requerimientos
- Para las personas que no tienen mucha experiencia en el análisis y diseño de sistemas utilizando una metodología orientada a objetos se les recomienda primero realizar una aproximación del sistema utilizando una metodología estructurada para evitar confusiones de integración de datos.
- Realizar un buen análisis del problema a solucionar ya que de esto depende el logro del cumplimiento de todos los requerimientos que el sistema a desarrollar requiere.
- Documentar correctamente el sistema por medio de sus diferentes manuales ya que son el soporte para un correcto futuro mantenimiento del sistema.
- Durante la construcción del sistema es conveniente desarrollar un documento en donde se anoten todos los errores que el sistema a producido con sus respectivas soluciones para que en un futuro, si es el caso, existe el mismo error sea fácil de corregirlo.

**CONCLUSIONES Y RECOMENDACIONES ..... XX**

6.1 Objetivos del Capítulo ..... XX

6.2 Conclusiones ..... XX

6.3 Recomendaciones ..... XXii

### ***Bibliografía***

- Pressman Roger, (1990). Ingeniería de Software. Segunda Edición. México, McGraw Hill.
- James Rumbaugh, (1998). The Unified Modeling Language Reference Manual. Hardcover.
- Albert Delgado, MICROSOFT SQL SERVER 2000
- William R. Vaughn, VISUAL BASIC & SQL SERVER
- Microsoft, MICROSOFT VISUAL BASIC LENGUAJE REFERENCE

### ***Webliografía***

- [http://www.lafacu.com/apuntes/informatica/inge\\_soft/isw5/default.htm](http://www.lafacu.com/apuntes/informatica/inge_soft/isw5/default.htm)  
Análisis orientado a objetos.
- <http://www.ultrasist.com.mx/ProducServ/cursoadoo.htm>  
Análisis y Diseño Orientado a Objetos
- <http://www.geocities.com/SiliconValley/Bit/1703/tareas/administracion/herramientascomputacionales.doc>
- <http://sistemas.ing.ula.ve/sistemas/pd20/>
- <http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%F1o%20orientado%20a%20objetos/rumbaugh.pdf>  
Técnica de modelado de objetos (OMT) (James Rumbaugh).
- <http://di002.edv.uniovi.es/~cernuda/pfc/aoo.pdf>  
Análisis Orientado a Objetos
- <http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%F1o%20orientado%20a%20objetos/cap1.pdf>

### El Desarrollo del Software

- <http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%F1o%20orientado%20a%20objetos/cap2.pdf>  
Métodos de Software
- <http://www.cps.unizar.es/~jmerse/UML.pdf>  
Análisis y Diseño Orientado a Objetos usando la notación UML
- <http://158.251.11.99/pers/gbustos/PDF/Clasifica.PDF>  
Técnicas de Análisis Orientado a Objetos
- [www.inf.utfsm.cl/~liuba/taller/ad\\_oo1.pdf](http://www.inf.utfsm.cl/~liuba/taller/ad_oo1.pdf)  
Programación Orientada a Objetos
- <http://www.monografias.com/trabajos/anaydiseisis/anaydiseisis.shtml>  
Análisis y Diseño de Sistemas
- [http://grulla.hispalinux.es/enunciados/analisis\\_strmas.pdf](http://grulla.hispalinux.es/enunciados/analisis_strmas.pdf)  
Análisis de sistemas
- <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>  
Tutorial de UML
- [http://petroecuador.com.ec/uidt/docs/art\\_tecn.pdf](http://petroecuador.com.ec/uidt/docs/art_tecn.pdf)  
"Estándares para el desarrollo de software orientado a objetos"
- <http://www.monografias.com/trabajos/anaydiseisis/anaydiseisis.shtml>  
Análisis y Diseño de Sistemas
- [http://mx.geocities.com/hasta\\_la\\_victoria\\_siempre\\_cgh/](http://mx.geocities.com/hasta_la_victoria_siempre_cgh/)  
Que es UML?
- <http://delta.cs.cinvestav.mx/~pmejia/softeng/Cap17.ppt>  
Diseño de interfaz de usuario
- <http://www.intec.edu.do/~yberges/gui.doc>



### Interfaz gráfica de usuario (GUI)

- <http://usuarios.lycos.es/froufe/parteJ1/capa-1.html>  
Instalando Tomcat
- <http://www.uacam.mx/dgsc.nsf/pages/cadmtools>  
Herramientas de desarrollo