

29



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

Tecnología Orientada a Objetos y Bases de
Datos Distribuidas en una Aplicación del
Grupo GEO

T E S I S

Que para obtener el título de
INGENIERIA EN COMPUTACION

Presentan:

Gerardo Alejandro Díaz Cisneros

Miguel Angel Carreón Adán

Victor Manuel Martínez López

Director de Tesis:

Dra. Ana María Vázquez Vargas

282346



Ciudad Universitaria, D.F.

2000



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Tabla de contenido

Agradecimientos

Introducción	1
Antecedentes	1
Área constructiva	1
Área comercial	2
Área administrativa.....	3
Definición del problema	4
Objetivos y alcances	6
Ventas	7
Cobranza.....	7
Titulación.....	7
Infraestructura tecnológica	7
Situación anterior.....	7
Situación actual - Propuesta de solución.....	9
Marco Teórico	12
Ingeniería de software	12
Fundamentos de ingeniería de software	12
La evolución del software.....	12
La información como un recurso.....	13
Tipos de sistemas de información.....	14
Tecnología Cliente/Servidor	15
Componentes de software para sistemas C/S.....	16
Distribución de componentes de software.....	17
Enlazado de componentes de software C/S.....	18
Enfoque clásico de un ciclo de vida de un sistema	18
Enfoque orientado a objetos.....	20
Tecnología orientada a objetos	21
Introducción	21
Características de la tecnología orientada a objetos.....	21
Temas de orientación a objetos.....	24
Metodología orientada a objetos	26
Desarrollo de un modelo orientado a objetos.....	27
Diferencias con respecto a la metodología funcional.....	28
Técnica de modelado de objetos OMT(Object Modeling Technique).....	29
Modelo de objetos	30
Objetos y clases	30
Ligas y asociaciones	34

Tesis

Tecnología orientada a objetos y bases de datos distribuidas en una aplicación del Grupo GEO
Universidad Nacional Autónoma de México

Facultad de Ingeniería
Tabla de contenido I

Ingeniería en computación

Generalización y herencia.....	40
Modelo dinámico.....	42
Eventos y estados.....	42
Diagramas de estado anidadas.....	49
Modelo funcional.....	61
Modelos funcionales.....	61
Diagramas de flujo de datos.....	61
Especificando operaciones.....	69
Relación de los modelos de objetos, dinámico y funcional.....	72
Análisis y diseño de bases de datos.....	74
Bases de datos relacionales.....	74
Bases de datos orientada a objetos.....	76
Bases de datos distribuidas.....	78
Introducción a los conceptos de DDBMS distribuidos.....	78
Arquitectura Cliente – Servidor para DBMS.....	80
Fragmentación, replicación y reparto de los datos para el diseño de bases de datos distribuidas.....	83
Fragmentación de datos.....	84
Replicación y reparto de los datos.....	85
Tipos de sistemas de bases de datos distribuidas.....	86
Costo por transferencia de datos en consultas distribuidas.....	87
Control de concurrencia y recuperación en base de datos distribuida.....	87
Análisis del sistema.....	89
Requerimientos.....	89
Ventas.....	93
Cobranza.....	94
Titulación.....	95
Análisis general.....	96
Ámbito empresarial operativo.....	97
Ámbito empresarial ejecutivo, empresarial directivo, delegacional directivo, corporativo directivo.....	125
Diseño del sistema.....	132
Diseño de los procesos y flujos de información.....	132
Diseño de módulos.....	134
Ámbito empresarial operativo.....	134
Ámbito ejecutivo – directivo.....	138
Diseño de bases de datos.....	142
Arquitectura de triple esquema para Modelo de objetos.....	142
Diseño de interfaz.....	152
Captación.....	152
Configuración de red corporativo.....	161
Configuración de red.....	163

Tests

Tecnología orientada a objetos y bases de datos distribuidas en una aplicación del Grupo GEO

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Ingeniería en computación

Tabla de contenido II

Contrataciones especiales	166
Conclusiones.....	169
Beneficios.....	169
Implementación.....	169
Comparación de metodologías.....	170
Comparación con OMT.	170
Futuro del sistema.....	171
Bibliografía	173
Apéndice	174
Metodología OMT – Notación	174
Código Fuente	177

Agradecimientos

A través del tiempo, conocemos diferentes tipos de personas, muchas de ellas pasan de largo continuando su camino, mientras que otras cambian fundamentalmente la forma y contenido de la vida. De estas últimas se alimenta el ser, tomando lo esencial de cada una de ellas para formar el carácter y personalidad que adquirimos. Sin embargo, es fascinante reflexionar que las primeras personas de mi vida son las que más admiración, fortaleza y comprensión me han brindado. A mis padres, por su paciencia, ejemplo, cariño y apoyo, dedico esta tesis.

También debo de incluir a la persona que me ha dado a conocer la otra mitad de mi vida. Por ayudarme a ser mejor persona cada día, amarme y enseñarme a amar la vida cada día con intensidad.

A mi esposa Noemi, con todo el amor. Y a mis hijos por enseñarme algo fundamental de la vida: que lo importante no es hacer las cosas, sino que lo importante es como haces las cosas, por que la calidad de la vida se encuentra en cada detalle.

- Gerardo Alejandro Díaz Cisneros

La presente tesis no hubiera sido posible sin la ayuda mis seres queridos que me orientaron para poder culminarla, a quienes se las dedico con todo cariño y afecto, pues no sólo lo material es necesario para terminar una carrera sino también la dedicación hacia uno y el apoyo incondicional en los momentos de indecisión lo cual hacen a uno madurar.

El llegar a esta meta, siendo una etapa mas de la vida, representa la satisfacción de ver culminados los esfuerzos conjuntos de una trayectoria de tropiezos y triunfos no solo por la escuela sino lo que se va aprendiendo en el camino donde la formación que brinda la escuela se ve complementada con las bases que se tienen desde la infancia para lograr ser un buen profesionista, por esto es que en agradecimiento a quienes me apoyaron no los defraudaré y verán en mí reflejado sus esfuerzos para ser un excelente profesionista productivo de forma ética y honrada sin pasar por alto los valores humanos en donde encontraran siempre una persona en quien confiar, gracias.

- Miguel Angel Carreón Adán

Una vez más y siempre, quiero agradecer a mis padres Juan Manuel y María Luisa (Amalia) por el apoyo incondicional que me brindaron, me brindan y me brindarán por la eternidad, ya que sin ellos no hubiera podido obtener mis logros personales por que son logros de ellos también, me enseñaron a luchar por todo lo que un hombre pudiera obtener en la vida y que más puedo decir, que no tendré las palabras para agradecer todo lo que soy. ¡Este trabajo es de ustedes!

A mi esposa Marina por ser paciente y aceptarme tal como soy, como esposo, como padre y como hombre además de darme la mejor hija del mundo. A mi hija Ana Monserrat, ya que ella abrió nuevamente la luz de bondad y de ternura que había perdido. A mi hermano Serafin que estuvo siempre ahí, cuando lo necesite.

Y quiero agradecer a todos aquellos que tuvieron que ver con preparación de esta obra, entre ellos, mis amigos, mi Abuela Camila, mi profesora de tesis Ana María Vázquez, etc.

- Víctor Manuel Martínez López

Introducción

Antecedentes

Corporación GEO, S.A. de C.V. es un corporativo mexicano establecido en el ramo de la construcción, que a través de 25 años en el mercado se ha especializado en la construcción de vivienda de interés social. El compromiso de la compañía es el proveer de vivienda de calidad para las familias mexicanas. Su mercado se establece en la promoción y trámites para créditos tipo INFONAVIT (Gubernamentales) y tipo FOVI (Privados), ofreciendo al cliente tramitar su crédito ante las instituciones otorgadoras de crédito como son INFONAVIT, FIVIDESU, Inmobiliarias privadas e Instituciones bancarias, junto al manejo de clientes que desean adquirir su vivienda de contado.

Actualmente GEO se encuentra formado por 14 compañías filiales dentro del territorio nacional, junto a nuevas compañías que se han ido integrando en Santiago de Chile, y Houston, Texas, en los Estados Unidos. Se tienen también interés de integrar a otros países de Sur y Centroamérica. Estas compañías se encuentran organizadas en delegaciones, la cual físicamente se encuentra generalmente en las oficinas de alguna de las compañías, con un director delegacional. Así mismo estas delegaciones se agrupan en lo que se denomina corporativo.

Todas las compañías GEO se organizan de manera similar, contando con las áreas de Ventas, Cobranza, Titulación, Recursos humanos, Sistemas, Promoción, Diseño, Construcción, Administración y Dirección. Dentro de cada compañía se realiza el proceso completo del negocio, desde la de elección de terrenos, hasta la adjudicación de la vivienda. Las responsabilidades de cada área se detallan a continuación

Área constructiva

Promoción

El área de promoción se dedica a la evaluación y adquisición de terrenos, proceso en el cual se consideran los factores de factibilidad comerciales y de construcción básicamente, junto a llevar a cabo todos los trámites legales y otros que dependen de diferentes instituciones gubernamentales y no gubernamentales.

Diseño

El área de diseño junto con promoción también evalúa el terreno desde el punto de vista de construcción, para poder planear el desarrollo a realizar en el predio. Realiza los estudios topográficos y de mecánica de suelos, para conocer la naturaleza del terreno, realiza el anteproyecto de desarrollo, preparando lo que es denominado la carpeta técnica para que la dirección y el corporativo la analicen para la aprobación del desarrollo. Así mismo realiza el proyecto del desarrollo una vez aprobado, y entrega planos de trazo para los diferentes frentes de construcción como son edificación, equipamiento, urbanización, etc.

Construcción

El área de construcción se encarga de llevar a cabo la realización del proyecto que Diseño ya ha entregado, teniendo como tareas la administración de los recursos y materiales de los diferentes frentes, planeando la estrategia de construcción, en comunicación con las áreas de ventas, diseño, promoción y dirección.

Área comercial

Ventas

El área de ventas tiene la responsabilidad del telemarketing, mercadotecnia y publicidad de los desarrollos en construcción y terminados a fin de captar clientes que deseen adquirir una vivienda. Así mismo lleva el control y administración de diferentes promotores, los cuales hacen promoción de GEO en las empresas con el perfil de empleados requeridos.

Cobranza

El área de cobranza lleva el control y seguimiento del pago de enganche de la vivienda por parte del cliente. Así mismo se encarga de llevar el control y seguimiento de las comisiones a las cuales se hacen acreedores los vendedores y promotores cuando una venta termina con la adjudicación de la vivienda.

Titulación

El área de titulación da el seguimiento a los trámites que tiene que llevar una persona para el otorgamiento de un crédito, ya sea tipo INFONAVIT o tipo FOVI; controla los requerimientos necesarios por cada intermediario financieros para otorgar crédito, así como verificar la factibilidad de

que un cliente pueda acceder al crédito. El trámite lo realiza en su totalidad la compañía GEO, siendo el cliente atendido durante este proceso, que dura entre 1 semana y varios meses, según la velocidad de respuesta de los intermediarios financieros y la disponibilidad de vivienda terminada. Al final se escritura y adjudica la vivienda. Así mismo tiene a su cargo la recuperación de recursos, proceso que se encarga de la cobranza a las instituciones financieras de los créditos otorgados los clientes, después de la adjudicación de la vivienda.

Área administrativa

Administración

El área de administración se encarga del control de los recursos contables y financieros de la empresa. Realiza el reflejo contable y fiscal de la empresa, y emite los estados financieros de la empresa a la dirección.

Recursos humanos

El área de recursos humanos se encarga de la evaluación y elección de personal para la empresa, teniendo como responsabilidad proveer de personal a las diferentes áreas de la empresa.

Sistemas

El área de sistemas es la encargada de proveer los recursos de cómputo e informática a las otras áreas para el desempeño de sus tareas. Esta área se encarga del control de adquisición y administración del equipo de cómputo, junto a la contratación de los consumibles de cómputo. Evalúa el software y hardware en el mercado para conocer sus ventajas y aplicaciones dentro de la dinámica de la empresa. Realiza los desarrollos de programas necesarios para cubrir necesidades específicas de las áreas.

Dirección

El área de dirección es la coordinadora de la empresa, que se encarga de dirigir las acciones de la empresa. Requiere constantemente de información de las diferentes áreas de la empresa, con la intención de crear las estrategias y tácticas necesarias para lograr la permanencia y crecimiento de la empresa dentro del ramo.

El corporativo ha formado diversas áreas de carácter corporativo con el fin de apoyar a la operación en las empresas desde el corporativo. Por ejemplo, se ha creado Ventas Corporativas, la cual tiene como objetivo apoyar a las áreas de ventas; Telemarketing Corporativo, el cual tiene como

objetivo apoyar a las áreas de Telemarketing. Estas nuevas áreas junto al apoyo que ofrecen a sus similares en las empresas, requieren constantemente de información para su función. Esto ha impuesto nuevas tareas para las empresas, puesto que es necesario crear información a ésta áreas corporativas junto a las ya existentes.

Definición del problema

Actualmente Corporación GEO, S.A. de C.V. cuenta con una operación de oficina muy grande, pues tiene que mantener en funcionamiento las áreas comercial, administrativa, dirección, etc., para poder dar vida a la parte sustantiva de su ser, que es el construir y ofrecer viviendas de interés social.

La operación se lleva actualmente con mecanismos manuales, predominando el papeleo de trámites tanto con áreas internas como con entidades externas.

En el área de promoción, el control de evaluación y adquisición de terrenos es controlado mediante cuadernos y carpetas, donde se anota el seguimiento que tiene el terreno, así como la serie de pasos en trámites gubernamentales que son necesarios para convertir de un terreno en una área lotificada para vivienda de interés social. Cuando es necesario tener informes de la situación actual del terreno, es necesario realizar una serie de tareas para recoger de los cuadernos, carpetas y datos específicos de quién o quienes lleven este seguimiento. El recopilar esta información toma de uno a varios días, utilizando a una o varias personas, lo que implica personal ocupado en informar a la parte directiva de la situación actual; así mismo, el tiempo de espera que es necesario para lograr tener la información resulta en una pérdida de oportunidad en muchos casos.

En el área de diseño, el tiempo típico de creación de un proyecto es de 4 meses; junto a esto existe la creación de diferentes prototipos de casas, y observar la aceptación que ha tenido cada prototipo en los proyectos anteriores; La distribución de viviendas, áreas comunes, vialidades, servicios, etc., también forman parte del tiempo de diseño; La serie de estudios del terreno toman un tiempo considerable, así como los resultados ya obtenidos para la toma de decisiones. En la toma de decisiones sobre un terreno y su proyecto arquitectónico resulta en la creación de la carpeta técnica, la cual es otro producto para la dirección que representa tener personal para informarla, siendo mucho de este trabajo realizado en forma manual.

El área de construcción tiene la función de materializar los diseños arquitectónicos del proyecto aprobado, enfrentándose a los problemas de llevar a cabo lo planeado en un plano del proyecto, y

necesitando informar de los avances de obra de los diferentes frentes, tanto en avance de obra como en gastos de materiales, estado actual de los almacenes, maquinaria. Aparte necesita informar y estar informado de las áreas de ventas, compras, recursos humanos, etc., para cumplir con sus funciones. Esta información la realizan mediante aplicaciones de automatización de oficina, los cuales se detallan posteriormente.

Ventas tiene la responsabilidad de la promoción de los desarrollos, ser el primer contacto del cliente con el grupo GEO, proporcionarles la información necesaria para buscar la compra de alguna de las casas. Así mismo, la promoción de los desarrollos provoca la contratación de personal por comisión, por lo que hace muy importante tener el control de quién atendió al cliente, quién visitó que empresas, quién integra su expediente para su tramitación de crédito, etc., pues estas acciones se traducen en dinero para los empleados, así como su ascenso dentro de la estructura administrativa de la misma. Los vendedores llevan sus propias carteras de clientes y seguimientos, siendo que esta información es parte de la empresa; así mismo, nuevamente cuando necesitan información de ventas, se requiere de una recopilación de información.

El área de cobranza continúa una de las líneas de seguimiento del cliente, el cual se encarga del control de la cobranza del enganche y otros conceptos. Este seguimiento se realiza con diferentes paquetes de cuentas por cobrar. En estos sistemas se controla el seguimiento de la cobranza y se introduce la información básica del cliente. Al principio se utilizaba el RFC del cliente para identificarlo, pero se encontró con el problema cuando un cliente quería más de una casa. Entonces la forma de identificarlo no fue inmediata. Aparte, muchos de los datos necesarios para GEO, no se encontraba forma de ingresarlos, comportándose las cuentas por cobrar como un componente aislado, al cual había que utilizar como fuente de datos para poder crear reportes personalizados para el área, como para otras. Cuando era necesario realizar informes para la dirección, el procedimiento se reducía a una serie de trabajos manuales con la información.

El área de titulación se dedica a continuar otra línea de seguimiento, encargada de realizar los trámites necesarios para obtener un crédito ante las instituciones bancarias y inmobiliarias. Esto requiere tener un registro de los diferentes documentos entregados del cliente, documentos faltantes, documentos vencidos a reponer, etc. Así mismo del avance por el expediente en los trámites ante la institución financiera, los cuales son diferentes para cada institución financiera. Además, una vez

terminado el trámite de otorgación de crédito, se encarga de la recuperación de recursos, que es el pago de las instituciones financieras al Grupo GEO por la vivienda vendida.

El área de administración y dirección continuamente requieren de información actualizada para poder tomar decisiones respecto al movimiento, dirección y estrategia a seguir por parte del Grupo GEO en cada una de las empresas, delegaciones y el corporativo. Por esta necesidad de información oportuna, se tiene un conjunto de gentes especializadas en la obtención, manejo y análisis de ésta, con el fin de tener mejor materia de decisión. Esta estructura de información necesita ser actualizada con herramientas que les ayuden a mejorar los tiempos de entrega, así como la calidad de información ofrecida.

Las expectativas de crecimiento del Grupo GEO hacen necesaria la creación de un sistema de información que ayude a la operación propia de las empresas, así como a la necesidad de datos e información analizada y oportuna. De no hacerse así, el volumen operativo manual sería tal que habría que incrementar el personal para que se mantuviera las compañías trabajando. Esto sería en un costo mayor para la compañía del costo de introducir tecnología a su actual personal.

Objetivos y alcances

El sistema de comercialización tiene como fin cubrir el área comercial, en los procesos de ventas, cobranza, titulación, comisiones, atención al cliente y posventa. Para lograr estas metas, se ha llevado a cabo una serie de entrevistas con los responsables de cada área, observando, analizando su trabajo diario, expectativas, problemas y soluciones. Este trabajo se ha llevado en todas las compañías, inclinándose en las compañías más participativas.

Con la implementación del sistema de comercialización, se espera aumentar la consistencia de la información en un 80% de los niveles actuales, y mejorando su oportunidad en un 200%, lo que viene a representar en el ámbito corporativo y delegacional una información de un día de atraso, junto a una información diaria en el ámbito empresarial. La reducción de esfuerzo manual de información se reducirá en un 80% respecto a los niveles actuales. Esto significa que mucha de la fuerza de trabajo que actualmente se ocupa en estas tareas, podrá realizar otras en beneficio de la compañía.

La tesis objeto de este trabajo se ha decidido enfocar a las áreas de ventas, cobranza y titulación, en los procesos generales de Ventas, Cobranza, Titulación; con el fin de hacer finito el trabajo realizado

para esta, incluyendo en el análisis, diseño e implementación del sistema la metodología OMT mostrada posteriormente, en sus tres modelos durante el ciclo de vida del sistema.

De acuerdo con el análisis hecho, se revisarán a detalle los siguientes procesos del área:

Ventas

- ✓ Desarrollar los planes de ventas
- ✓ Realizar la captación de cliente
- ✓ Realizar la labor de ventas
- ✓ Formalizar la venta
- ✓ Seguir el avance de la vivienda vendida
- ✓ Realizar el plan de comisiones
- ✓ Realizar el seguimiento de comisiones

Cobranza

- ✓ Realizar cobro de documentos
- ✓ Realizar la negociación con clientes
- ✓ Dar seguimiento a la cartera

Titulación

- ✓ Captación de prospectos
- ✓ Dictaminar a los prospectos
- ✓ Formalizar expediente
- ✓ Tramitar expediente en Institución
- ✓ Escriturar expediente
- ✓ Recuperación del crédito

Infraestructura tecnológica

Situación anterior

La infraestructura que existía en el Grupo GEO puede evaluarse desde dos puntos de vista principalmente: la infraestructura de tecnología en comunicaciones y la infraestructura de tecnología en sistemas de información.

Redes y comunicaciones

Dentro de la infraestructura de tecnología en comunicaciones, el estado de las comunicaciones consistía en redes de computadoras PC y redes de computadoras Macintosh. Las redes de computadoras PC consistían de un servidor Novell como servidor de archivos y de impresión. En el sistema de telefonía y telecomunicaciones, se mantenía comunicación telefónica local conmutada, teniendo que salir a las líneas públicas para la comunicación entre las diferentes compañías del Grupo GEO y otras instalaciones propias de la empresa, como son los puntos de venta y desarrollos constructivos.

El soporte técnico estaba fundamentado en un área con tres personas que su vez tenían la responsabilidad de desarrollo de sistemas, y monitoreo de la red.

Sistemas de información

Por su parte, la infraestructura de tecnología en sistemas de información, existía el área de desarrollo de sistemas, en el cual se habían desarrollado sistemas aislados sobre el área comercial, y diversos sistemas para otras, como el área administrativa y contable. Estos sistemas se fundamentaban sobre Clipper 5.x y Visual Fox Pro 2.5. El sistema comercial era el sistema más grande creado por la administración anterior; Estaba encaminado a mejorar la operación de las áreas de ventas, cobranza y titulación, llevando inicialmente el proceso de venta, y posteriormente el de titulación; para la parte de cobranza, este intercambiaba información con un paquete de cuentas por cobrar comercial llamado ContPaq* el cual es un sistema para seguimiento de cuentas por cobrar.

Por informes de los usuarios, se observó que el sistema comercial adolecía de múltiples deficiencias de diseño y análisis; el sistema era inestable y no ofrecía información fidedigna y consistente, lo que provocó el regreso al trabajo manual dentro de la operación diaria. Diversas compañías dejan de utilizar el sistema comercial, implementando informes por medio de herramientas de automatización de oficinas, junto a la implementación de sistemas de administración de empresas, principalmente uno llamado SAE, que sin embargo dadas las características de operación de las compañías GEO, no es suficiente, además de provocar heterogeneidad entre las aplicaciones usadas por el Grupo GEO.

De esta manera, se nota dentro del área de informática, una falta de dirección sobre los lineamientos corporativos a seguir con respecto a los servicios de red y telecomunicaciones, así como

en la paquetería que se utiliza para la operación y sistemas de información; esto provoca una heterogeneidad en el equipo de cómputo que se utiliza y los sistemas de información junto a los operativos, que lleva a diferencias en la forma de trabajar en cada compañía, con las desventajas de compatibilidad, y tareas de integración necesaria en el momento de tener información corporativa fiable.

Situación actual - Propuesta de solución

De acuerdo al análisis, la propuesta de solución y su actual implementación en el Grupo GEO es la creación de una infraestructura robusta y duradera para los servicios necesarios por las diferentes áreas de las empresas.

La dirección corporativa de informática (DCI) se le dan las siguientes características

- ✓ La estructura organizacional se divide en
 - Gerencia de redes y comunicaciones
 - Gerencia de sistemas
- ✓ El corporativo ha decidido integrar a diferentes compañías de outsourcing como asociados de negocio, con el fin de cubrir sus servicios de soporte técnico y mantenimiento, desarrollo de software a la medida, proveedores de equipo de cómputo y consultores de informática; esto con el fin de invertir en un gasto sobre cada servicio otorgado, liberando al Grupo GEO de pasivos de gastos en personal.
- ✓ La DCI tiene la responsabilidad de proveer los lineamientos y apoyo directivo como técnico del equipo de cómputo como herramientas de cómputo necesarias para cubrir las necesidades operativas, ejecutivas y directivas de información.

Gerencia de redes y comunicaciones

Objetivos y lineamientos

- ✓ Facilitar las actividades operativas, administrativas y directivas de la corporación a través de una infraestructura de cómputo adecuada y de medios de comunicación eficientes que permitan el intercambio de información (audio, video y datos)
- ✓ Lograr la estabilización de los servicio de cómputo y comunicaciones localmente.
- ✓ Estabilización de la red Lotus Notes local y remotamente
- ✓ Dimensionamiento e implementación de la red de comunicaciones a nivel nacional
- ✓ Dimensionamiento e implementación de una infraestructura de cómputo robusta que soporte las aplicaciones de misión crítica
- ✓ Estandarización a nivel nacional del protocolos de comunicación
- ✓ Implementación de comunicación vía internet para comunicaciones con el extranjero.

- ✓ Contratación de una empresa para el servicio de mantenimiento preventivo y correctivo para el equipo de cómputo
 - ✓ Estandarización de equipo de cómputo
 - ✓ Elaboración de un plan de capacitación para personal del área de informática en base al perfil del puesto.
 - ✓ Implementación de una mesa de ayuda o HelpDesk que permita darle seguimiento a los problemas de los usuarios.

Seguridad

- ✓ Respaldos diarios incrementales, semanales totales y mensuales totales e históricos
- ✓ Acceso restringido a áreas de cómputo, servidores y comunicaciones
- ✓ Difusión de información para claves de red y correo únicas, confidenciales y difíciles de deducir.

Gerencia de sistemas

Objetivos y lineamientos

Proporcionar al conjunto de personas participantes en los procesos de las áreas de las compañías, el mejoramiento de procesos y las herramientas de cómputo que ayuden y perfeccionen sus tareas dentro del Grupo GEO, haciendo más eficiente en tiempo, esfuerzo y/o recursos necesarios

- ✓ Estandarizar los sistemas de operación utilizados en el Grupo y las herramientas de cómputo que soportan la operación
 - Sistemas desarrollados a la medida, Sistema de comercialización
 - Sistemas administrativos
 - Sistemas de diseño
 - Sistemas de información financiera
- ✓ Proveer de la infraestructura informática necesaria para contar con información fiable y oportuna para la toma de decisiones
 - Información directiva
 - Información ejecutiva
 - Información administrativa
 - Información comercial
 - Información financiera
 - Información de construcción y diseño
- ✓ Integrar las diversas soluciones que por área existan o se implementen con el fin de facilitar la automatización de tareas
 - Integración de los sistemas de operación en línea con los sistemas de información ejecutiva y directiva
 - Integración de los sistemas de información a los sistemas de correo y mensajes por biper para mensajes automáticos generados por los sistemas a los responsables de cada área.
- ✓ Proporcionar la ayuda técnica de desarrollos a la medida, así como contratar a los proveedores de tecnología y consultoría para la generación de las herramientas de cómputo

Así mismo, a nivel técnico se definen las siguiente políticas

Software

Sistemas operativos	<p>Sistema de red - Windows NT4 Sistema operativo de clientes - Windows 95 Sistema de monitoreo de RED - HP OpenView y NotesView para Lotus Notes Sistema de respaldo - ArcServer para NT Sistema de antivirus - Norton Antivirus Sistema operativo de clientes MAC - MAC OS8</p>	
Automatización de oficina	<p>Sistema de oficina - Microsoft Office (Word, Excel, PowerPoint) Sistema de correo - Lotus Notes Correo Sistema de procedimientos y publicaciones - Lotus Notes</p>	
Sistemas desarrollados a la medida	<p>Manejador de base de dato - SYBASE SQL Server XI, SQL AnyWhere 5.53 MiddleWare - Estandart ODBC para conexión a datos Lenguaje de desarrollo. Microsoft Visual Basic 4.0, Microsoft Visual C++ 4.0 Metodología de desarrollo IDEF IDEF0 - INTEGRATION DEFINITION FOR FUNCTION MODELING IDEFIX - INTEGRATION DEFINITION FOR INFORMATION MODELING</p>	
Sistemas de apoyo a operación	<p>Sistema de control de obra y precios unitarios - ECOSOFT OPUS, versión especial para GEO. Sistema administrativo - ANтар Mizar, información básica, ANтар Mizar / SIF, información financiera, ANтар Mizar / ADA Información administrativa, ANтар NRH Nómina y recursos humanos; versiones especiales para GEO</p>	
Sistemas de apoyo a diseño arquitectónico	<p>Sistema de CAD - AutoCad para diseños arquitectónicos actuales Sistema de CAD especializado en arquitectura - ArchiCad, cuantificaciones y estudios especiales Intercambio de información entre ArchiCad y OPUS para cuantificación de materiales y evaluación de costos.</p>	

Marco Teórico.

Ingeniería de software

Los principios de ingeniería se encuentran orientados a obtener software económico, robusto, que sea fiable y funcione de manera eficiente sobre máquinas reales. La **ingeniería del software** que surge de la ingeniería de sistemas y de hardware, abarca un conjunto de tres elementos clave: métodos, herramientas y procedimientos.

Los **métodos**, indican como construir técnicamente el software, los cuales abarcan un amplio espectro de tareas que incluyen planeación de proyectos, análisis de requisitos del sistema y software, diseño de estructuras de datos, arquitectura de programas y procedimientos, algoritmos, codificación, prueba y mantenimiento. Las **herramientas** suministran un soporte automático y semiautomático para los métodos, es decir, cuando se integran las herramientas de forma que la información creada por una primera herramienta puede ser usada por otra segunda. Los **procedimientos** definen la secuencia en la que se aplican los métodos.

Fundamentos de ingeniería de software

La evolución del software

Durante las tres primeras décadas de la informática, el principal desafío fue el desarrollo del hardware de las computadoras; Actualmente el principal desafío es mejorar la calidad y reducir el costo de las soluciones basadas en computadoras implementando software; En la **primera era** de la informática, los sistemas se desarrollaban usando técnicas de administración orientadas al hardware, la programación se veía como un arte y existían pocos métodos formales, junto a que pocas personas los usaban; el programador aprendía normalmente su oficio mediante prueba y error, así como el desarrollo del software se realizaba sin ninguna planificación; en estos años lo normal era que el hardware fuera de propósito general y el software se diseñaba a medida para cada aplicación, teniendo una distribución relativamente pequeña. En la **segunda era** se extiende la multiprogramación; los sistemas multiusuarios introdujeron nuevos conceptos de interacción hombre - máquina, junto con técnicas interactivas que abrieron un nuevo mundo de aplicaciones y niveles de sofisticación en hardware y software; los sistemas de tiempo real podían recoger, analizar y transformar datos de múltiples fuentes, controlando así procesos y produciendo salidas en milisegundos; los avances en la tecnología condujeron a la primera generación de sistemas de administración de bases de datos; Esta era se

caracteriza también por el establecimiento del software como producto, pues los programas se distribuían para las computadoras grandes y minicomputadoras; conforme crecía el número de sistemas informáticos, comenzaron a extenderse las bibliotecas de software de computadora. La **tercera era** aparecieron los sistemas distribuidos, la conclusión de la tercera era se caracterizó por la llegada y amplio uso de microprocesadores, la cual produjo un extenso grupo de productos inteligentes. La **Cuarta era** se aleja de las computadoras mainframes, dirigiéndose al impacto colectivo de las computadoras personales. A medida que la cuarta generación progresa surgen nuevas tecnologías. En particular, la tecnología orientada a objetos están desplazando los enfoques de desarrollo de software más convencionales en muchas áreas de aplicaciones.

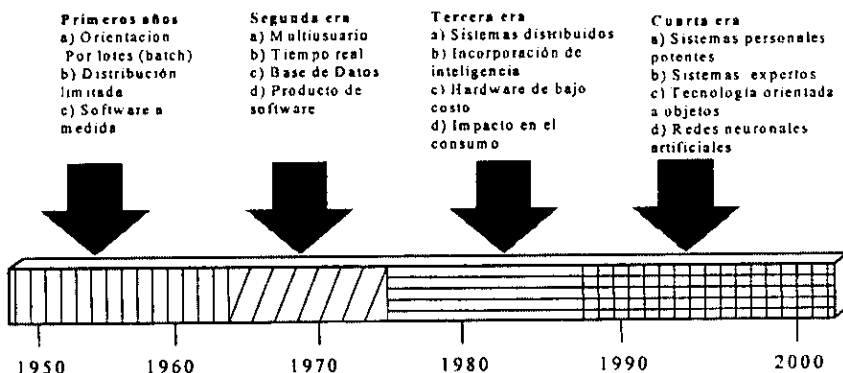


Figura 1 Evolución del software

La información como un recurso.

Las organizaciones han reconocido, desde hace mucho tiempo, la importancia de administrar los recursos principales tales como la mano de obra y las materias primas. Junto a esto, la información se ha colocado en un lugar adecuado como un recurso principal dentro de las organizaciones. Para maximizar la utilidad de la información, el negocio la debe manejar óptimamente tal cual maneja los demás recursos; se necesita comprender que hay costos asociados con la producción, distribución, seguridad, almacenamiento y recuperación de toda la información, ya que ésta no es gratis, y su uso es estratégico para posesionar competitivamente al negocio

El manejo de la información generada por computadora difiere en forma significativa de los datos producidos manualmente. En general los usuarios frecuentemente la tratan menos escépticamente que

la información obtenida por otras vías, pero es mayor la cantidad de información en computadora a administrar, junto a que el costo de mantenerla y organizarla puede crecer a tasas alarmantes.

Existen problemas asociados con el software que se han producido por su propio carácter y debido a que los trabajadores del software han tenido muy poco entrenamiento formal en las nuevas técnicas de desarrollo de software. En muchas organizaciones reina una suave forma de anarquía, cada individuo enfoca su tarea de "escribir programas", con la experiencia obtenida en trabajos anteriores, algunas personas desarrollan un método ordenado y eficiente de desarrollo del software mediante prueba y error, pero muchos otros desarrollan malos hábitos que dan como resultado una pobre calidad en la producción y mantenimiento del software.

Tipos de sistemas de información

Los sistemas de información son desarrollados con propósitos diferentes dependiendo de las necesidades del negocio

- ✓ **Sistemas de procesamiento de transacciones (TPS)** son sistemas de información computarizados desarrollados para procesar gran cantidad de datos para transacciones rutinarias de los negocios, tales como nómina e inventario.
- ✓ **Sistemas de automatización de oficina (OAS)** dan soporte a los trabajadores de la información, quienes por lo general, no crean un nuevo conocimiento sino que usan los datos para analizarlos y transformarlos en información o para modificarla en forma y luego compartirla formalmente por toda la organización y algunas veces más allá.
- ✓ **Sistemas de manejo de conocimientos (KWS)** dan soporte a los trabajadores profesionales, tales como científicos, ingenieros y doctores, les ayudan a crear un nuevo conocimiento.

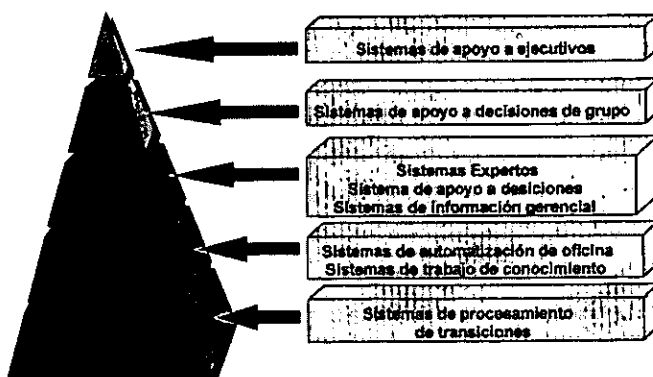


Figura 2 Diversos tipos de sistemas de información

✓ **Sistemas de información gerencial (MIS)** son sistemas de información que trabajan bajo la interacción de gentes, software y hardware. Para poder ligar la información, los usuarios de un sistema de información gerencial, comparten una base de datos en común, la cual guarda modelos que ayudan a los usuarios a interpretar y aplicar esos mismos datos. No reemplazan a los sistemas de procesamiento de transacciones, sino que a todos los MIS incluyen un sistema de procesamiento de transacciones.

✓ **Sistemas de apoyo a decisiones de grupo (GDSS)** son usados en cuartos especiales, equipadas en varias configuraciones diferentes, que permiten que los miembros del grupo interactúen con apoyo electrónico, frecuentemente en forma de software especializado y con una persona que da facilidades al grupo.

Tecnología Cliente/Servidor

La tecnología Cliente/Servidor se basa en dos elementos. El servidor, el cual generalmente es una mainframe, se encarga de realizar tareas de diferentes tipo a un grupo de usuarios. El cliente, es una máquina de capacidad mediana, la cual solicita que se realicen ciertas tareas por el servidor. Las tecnologías de hardware, software, base de datos y redes de computadoras contribuyen todas ellas a la arquitectura cliente/servidor distribuida y cooperativa. Generalmente el esquema se basa en un sistema raíz que esta conectado con varios servidores. Los servidores realizan el mantenimiento y creación de la información corporativa. Junto a esto, mantiene los sistemas departamentales y desempeña un papel de conexión al poner en red a los usuarios a través de redes locales.

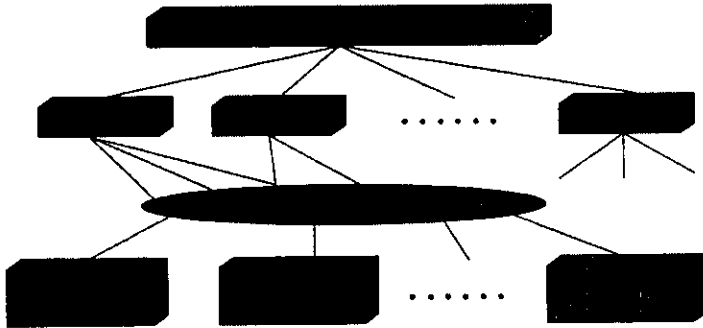


Figura 3 Arquitecturas de computadoras cooperativas y distribuida en un entorno corporativo.

En el esquema de operación, los clientes solicitan servicios y el servidor los proporciona; Sin embargo con diferentes arquitecturas se pueden otorgar diferentes tipos de servicios a los usuarios

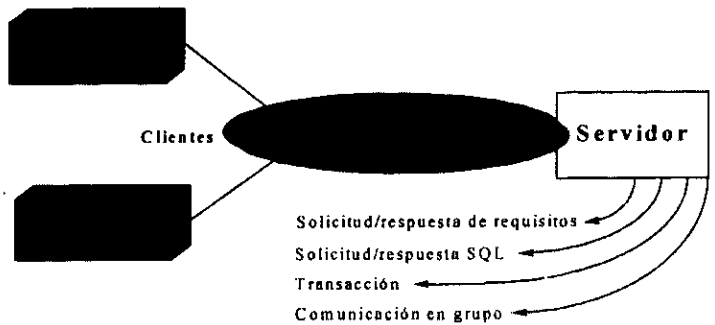


Figura 4 Opciones de la arquitectura cliente/servidor

- ✓ **Servidores de archivos.** El Cliente solicita creación, modificación y/o eliminación de archivos específicos. El servidor realiza estos servicios al cliente a través de la red.
- ✓ **Servidores de base de datos.** El cliente envía sentencias SQL al servidor. Éstas se transmiten como mensajes a través de la red. El servidor de base de datos procesa la solicitud SQL y realiza la inserción, modificación o eliminación de la información solicitada, pasando únicamente los resultados al cliente.
- ✓ **Servidores de transacciones.** El cliente envía una solicitud que invoca procedimientos remotos en el servidor. Los procedimientos remotos pueden ser un conjunto de sentencias SQL. Se produce una transacción cuando una solicitud da lugar a la ejecución de procedimientos remotos y a la transmisión del resultado devuelto al cliente.
- ✓ **Servidores de grupo de trabajo.** Cuando el servidor proporciona un conjunto de aplicaciones que hacen posible la comunicación entre clientes (y entre las personas que los usan) mediante el uso de texto, imágenes, boletines electrónicos, vídeo y otras representaciones, existe una arquitectura de grupo de trabajo.

Componentes de software para sistemas C/S.

La visión que se tiene del software cambia de ser una aplicación monolítica que deberá implementarse en una sola máquina, el software es adecuado para una arquitectura cliente/servidor, y posee varios componentes distintos que se pueden asociar al cliente o al servidor, o se pueden distribuir entre ambas máquinas:

- ✓ **Componente de interacción con el usuario y presentación.** Este componente implementa todas las funciones que típicamente se asocian a una interfaz gráfica de usuario.
- ✓ **Componente de aplicación.** Este componente implementa los requisitos definidos por la aplicación en el contexto del dominio en el cual funciona la aplicación. Por ejemplo, una aplicación de negocios podría producir toda una gama de informe impresos basados en entradas numéricas, cálculos, información de una base de datos, y otros aspectos.

- ✓ **Gestión de bases de datos.** Este componente lleva a cabo la manipulación y gestión de datos requerida por una aplicación. La manipulación y gestión de datos puede ser tan sencillas como la transferencia de un registro, o tan compleja como el procesamiento de sofisticadas transacciones SQL.

Además de estos componentes, existe otro bloque de construcción del software, que suele denominarse software intermedio de los sistemas C/S. El software intermedio consta de elementos de software que existen tanto en el cliente como en el servidor, e incluye elementos de sistemas operativos en red, así como un software de aplicación especializado que presta su apoyo a las aplicaciones específica de bases de datos, a estándares de distribución de solicitudes de objetos, a tecnologías de trabajo en grupo, a gestión de comunicaciones y otras características que facilitan la conexión cliente/servidor.

Distribución de componentes de software.

Después de analizar los requisitos del sistema, se debe decidir la forma de distribuir los componentes entre el cliente y el servidor. Cuando la mayor parte de la funcionalidad asociada a cada uno de los tres componentes se asocia al servidor, se ha creado un diseño de servidor principal. Éstos suelen diseñarse cuando se implementan sistemas de transacciones y de trabajo en grupo. Cuando el cliente implementa la mayor parte de los componentes de interacción/presentación con el usuario, de aplicación y de bases de datos, se tiene un diseño de cliente principal. Éstos suelen encontrarse cuando se implementan arquitecturas de servidor de archivos y de servidor de base de datos. En este caso, el servidor proporciona apoyo para la gestión de datos, pero todo el software de aplicación reside en el cliente. Junto a esto, un enfoque más granular para la asignación de componentes de software define cinco configuraciones diferentes.

- ✓ **Presentación centralizada.** En este enfoque cliente/servidor rudimentario, la lógica de la base de datos y la lógica de la aplicación permanecen en el servidor típicamente en una computadora central. El servidor contiene también la lógica para preparar información de pantalla, empleando un software. En general se utiliza un software especial basado en PC para transformar la información que se transmite desde el servidor, a una pantalla basado en caracteres.
- ✓ **Presentación distribuida.** En este enfoque, la lógica primaria de la base de datos y de la aplicación permanecen en el servidor, y los datos enviados por el servidor serán utilizados por el cliente para preparar la presentación del usuario.

- ✓ **Lógica distribuida.** Se asignan al cliente todas las tareas de presentación del usuario y también los procesos asociados a la introducción de datos tales como la validación de nivel de campo, la formulación de consultas del servidor, y las solicitudes de información de actualizaciones del servidor. Se asignan al servidor las tareas de gestión de las bases de datos, y los procesos para las consultas del cliente, para actualizaciones del servidor, para control de versión de clientes, y para aplicaciones de ámbito general de la empresa.
- ✓ **Gestión de datos remota.** Las aplicaciones del servidor crean una nueva fuente de datos dando formato a los datos que se han extraído de algún lugar (Por ejemplo, de una fuente de nivel corporativo). Las aplicaciones asignadas al cliente se utilizan para explotar los nuevos datos a los que se ha dado formato mediante el servidor. En esta categoría se incluyen los sistemas de apoyo a decisiones.
- ✓ **Bases de datos distribuidas.** Los datos de que consta la base de datos se distribuyen entre múltiples clientes y servidores. Consiguientemente, el cliente debe de admitir componentes de software de gestión de datos, así como componentes de aplicación.

Enlazado de componentes de software C/S

Se utiliza toda una gama de mecanismos distintos para enlazar los distintos componentes de la arquitectura cliente/servidor. Estos mecanismos están incluidos en la estructura de la red y del sistema operativo y resultan transparentes para el usuario final citado en el centro del cliente. Los tipos más comunes de mecanismo de enlazado son:

- ✓ **Tubos (pipes)** Se utilizan mucho en los sistemas basados en UNIX; los tubos permiten la mensajería entre distintas máquinas que funcionen con distintos sistemas operativos.
- ✓ **Llamadas a procedimientos remotos:** permiten que el proceso invoque la ejecución de otro proceso o módulo que resida en una máquina distinta
- ✓ **Interacción cliente/servidor SQL:** se utiliza para pasar solicitudes SQL y datos asociados de un componente (típicamente situado en el cliente) a otro componente.

Además, las implementaciones orientadas a objetos de componentes de software C/S dan lugar a una vinculación que haga uso de un distribuidor de solicitudes de objetos.

Los sistemas cliente/servidor se desarrollan a partir de un conjunto de requisitos de negocios generales y a medida que evoluciona el sistema, llegan a ser una colección de componentes de software ya validados que han sido implementados en máquinas cliente y servidor.

Enfoque clásico de un ciclo de vida de un sistema

El enfoque clásico del ciclo de vida de un sistema algunas veces es llamado "modelo en cascada", exige un enfoque sistemático y una secuencia de desarrollo de software que comienza en el nivel del

sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento, modelado a partir del ciclo convencional de la ingeniería; abarca las siguientes actividades:

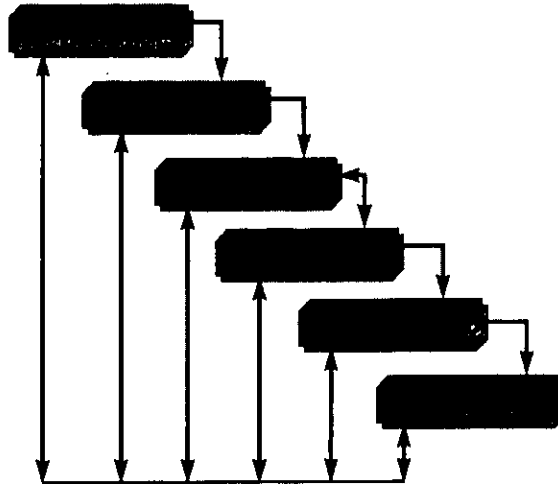


Figura 5 Ciclo clásico de vida de un sistema

- ✓ **Ingeniería y análisis del sistema.** Debido a que software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software, este planteamiento del sistema es esencial cuando el software debe interrelacionarse con otros elementos, tales como hardware, personas y bases de datos. Abarca los requisitos globales a nivel del sistema.
- ✓ **Análisis de los requisitos del software.** El proceso de recopilación de los requisitos se centra y se intensifica especialmente para el software. Para comprender la naturaleza de los programas que hay que construir, el ingeniero de software debe de comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.
- ✓ **Diseño.** El diseño del software es realmente un proceso multipaso que se enfoca sobre los cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz.
- ✓ **Codificación.** El diseño debe traducirse en una forma legible para la máquina, el paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
- ✓ **Prueba.** Una vez que se ha generado el código, comienza la prueba del programa, la prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado.

✓ **Mantenimiento.** El software indudablemente sufrirá cambios después de que se entregue al cliente. Los cambios ocurrirán debido a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo, o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

Enfoque orientado a objetos.

Un nuevo y muy diferente enfoque al análisis y diseño de sistemas es el orientado a objetos. Las técnicas orientadas a objetos, que están basadas en conceptos de objetos únicos, pueden ayudar a responder las demandas organizacionales para nuevos sistemas que requieren mantenimiento, adaptación y rediseño continuos. Básicamente, en la orientación a objetos son creados entes que incluyen no solamente código acerca de los datos sino también instrucciones acerca de las operaciones que se pueden realizar con ellos.

Tecnología orientada a objetos

Introducción

El modelado y diseño orientado a objetos es una nueva forma de pensar acerca de los problemas usando modelos organizados alrededor de conceptos del mundo real. La noción fundamental es el objeto, el cual combina tanto estructura de datos como comportamiento en un solo ente. Los modelos orientados a objetos son útiles para el entendimiento de problemas, intercomunicación con los expertos de diferentes áreas, modelado de empresas, preparación de documentación, diseño de programas y bases de datos. En esta tesis presentamos una metodología de desarrollo de software orientado a objetos, la técnica de modelado de objetos, OMT en sus siglas en inglés (Object Modeling Technique), la cual se extiende desde el análisis, diseño e implementación.

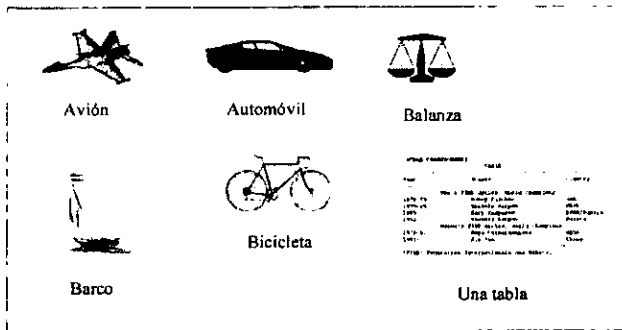


Figura 6 Ejemplos de objetos

OMT es una notación gráfica para expresar modelos orientados a objetos. Los objetos del dominio de la aplicación y del dominio de cómputo pueden ser modelados, diseñados e implementados usando la misma notación y conceptos orientados a objetos. Se utiliza la misma notación desde el análisis pasando por el diseño hasta la implementación, de tal manera que la información agregada en una etapa de desarrollo puede ser reutilizada y no se pierde o se necesita trasladar a la siguiente etapa.

Características de la tecnología orientada a objetos

En forma general el término "orientado a objetos" significa que se organiza el software como una colección de objetos discretos que incorporan tanto estructura de datos como comportamiento. Esto en contraste a la programación convencional en donde la estructura de datos y el comportamiento, o

procedimientos, se encuentran pobremente ligados. Existen diferencias acerca de qué características son requeridas para un enfoque orientado a objetos, pero en general se incluyen cuatro aspectos: Identidad, clasificación, polimorfismo y herencia.

La *identidad* significa que los datos están cuantificados dentro de entidades discretas e identificables denominadas objetos. Cada objeto tiene su propia identidad. En otras palabras, dos objetos son distintos aún cuando todos sus valores de sus atributos sean idénticos.

La *clasificación* significa que objetos con la misma estructura de datos (atributos) y comportamiento (procedimientos) son agrupados en clases. Una clase es una abstracción que describe las propiedades importantes e ignora el resto. Cualquier abstracción de clases es arbitraria y depende de la aplicación.

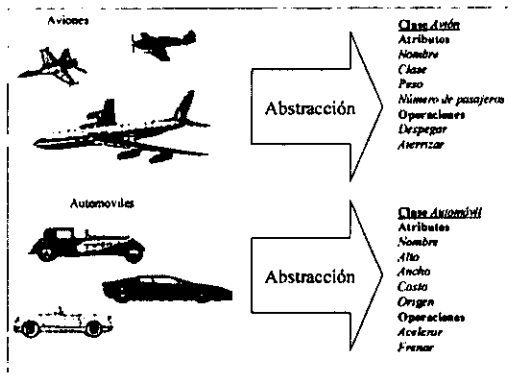


Figura 7 Ejemplos de clases

Cada clase define un conjunto con posibilidad infinita de objetos individuales. Cada objeto se dice ser una instancia de su clase. Cada instancia de la clase tiene su propio valor para cada atributo, pero comparte los nombres de los atributos y operaciones con otras instancias de la clase. Junto a esto, un objeto contiene una referencia a su propia clase, “conoce de que clase es”.

El *polimorfismo* significa que la misma operación se puede comportar en forma diferente sobre diferentes clases. Una operación es la acción o transformación que un objeto realiza o es sujeto de. Una implementación específica de una operación por cierta clase se le denomina método. Debido a que el operador orientado a objetos es polimórfico, puede tener más de un método que lo implemente.

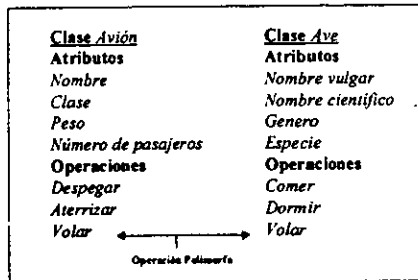


Figura 9 Ejemplo de polimorfismo

En el mundo real, una operación es simplemente una abstracción análoga del comportamiento a través de diferentes tipos de objetos. Cada objeto conoce cómo debe realizar su propia operación. En el

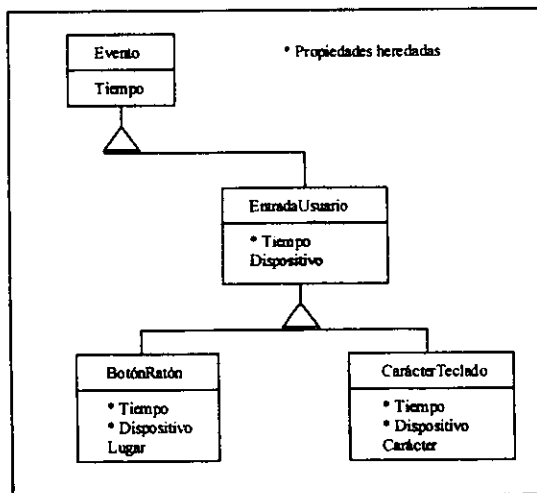


Figura 8 Ejemplo de herencia

lenguaje de programación orientado a objetos, sin embargo, el lenguaje automáticamente selecciona el método correcto para implementar la operación basada en el nombre de la operación y la clase del objeto sobre el cual está siendo operado. El usuario no tiene que preocuparse sobre cuantos métodos existen para implementar una operación polimórfica dada. Las nuevas clases pueden ser agregadas sin cambiar ningún código preexistente; pues se provee métodos nuevos sobre cada operación aplicable sobre las nuevas clases.

La *herencia* es compartir atributos y operaciones a través de las clases basadas en una relación jerárquica. Una clase puede ser definida someramente y después ser redefinida sucesivamente en subclases más finas. Cada subclase incorpora o hereda, todas las propiedades de su superclase o clase padre y agrega sus propias únicas propiedades. Las propiedades no necesitan ser repetidas en cada subclase. La facilidad de clasificar y factorizar las propiedades comunes de diversas clases dentro de una superclase común, y heredar las propiedades de la superclase puede reducir enormemente a la repetición de código dentro de los diseños y programas y es una de las principales ventajas del sistema orientado a objetos.

Temas de orientación a objetos

Existen diversos temas subyacentes en la tecnología orientada a objetos. Aún cuando estos temas no son únicos a los sistemas orientados a objetos, particularmente son bien soportados por los sistemas orientados a objetos.

✓ **Abstracción.** Consiste en enfocarse en los aspectos esenciales e inherentes de una entidad e ignorar las propiedades accidentales. En el desarrollo del sistema, esto significó enfocarse sobre que el objeto es y hace, antes de decidir como deberá ser implementado. El uso de la abstracción conserva la libertad de realizar decisiones tan tarde como sea posible, evitando compromisos prematuros en los detalles. El uso de la abstracción durante el análisis significó el manejo únicamente de conceptos del dominio de la aplicación, sin realizar diseño ni decisiones de implementación antes de que el problema fuera entendido. El uso apropiado de la abstracción permite al mismo modelo ser usado para el análisis, diseño de alto nivel, estructura de programa, estructura de base de datos y documentación.

✓ **Encapsulación o ocultación de información.** Consiste en separar los aspectos externos de un objeto, los cuales son accesibles a otros objetos, de los detalles de implementación internos del objeto, los cuales se encuentran ocultos a los demás objetos. La encapsulación previene a los programas de convertirse tan interdependientes que un pequeño cambio tenga efectos masivos. La implementación de un objeto puede ser cambiada sin afectar la aplicación que lo utiliza. La encapsulación no es un tema exclusivo de los lenguajes de orientación a objetos, pero la habilidad para combinar la estructura de datos y la conducta en una sola entidad hace a la encapsulación más limpia y poderosa que en los lenguajes convencionales que separan la estructura de datos y conducta (procedimientos).

✓ **Combinación de datos y procedimientos.** En los lenguajes de programación convencionales, para cada llamada a un procedimiento, es necesario verificar los parámetros de llamada para garantizar su correcto funcionamiento. En las operaciones orientadas a objetos, no es necesario conocer, las diferentes implementaciones que existen. El operador polimorfo escoge qué procedimiento va a ejecutar de acuerdo a la llamada de procedimiento en la jerarquía de clases. Así en el enfoque orientado a objetos se combinan datos y procedimientos en un solo árbol de clases jerárquica, en contraste con los dos árboles necesarios en el enfoque funcional, donde existe un árbol de datos jerárquico y un árbol de procedimientos separados.

✓ **Reutilización.** Las técnicas orientadas a objetos promueven la reutilización a diferentes niveles. La herencia de tanto estructura de datos como de conducta permite compartir una estructura común a diferentes subclases similares sin redundancia. La reutilización de código usando herencia es una de las principales ventajas de los lenguajes orientados a objetos. Más importante que los ahorros en código, es la claridad conceptual de reconocer que diferentes operaciones son todas realmente la misma cosa. Esto reduce el número de casos distintos que deben de ser entendidos y analizados. El desarrollo orientado a objetos no sólo permite a la información ser compartida dentro de la aplicación, sino también ofrece la posibilidad de reusar diseños y código en futuros proyectos. Aún cuando esta posibilidad ha sido sobreemfatizada como una justificación para la tecnología orientada a objetos, el desarrollo orientado a objetos provee de herramientas, tales como la abstracción, la encapsulación y la herencia, para construir librerías de componentes reutilizables. Sin embargo, la orientación a objetos no es una fórmula mágica para asegurar reusabilidad. El reuso no sucede espontáneamente; debe de ser planeada pensando más allá de la aplicación inmediata e invirtiendo un esfuerzo extra en un diseño más general.

✓ **Énfasis en la estructura de objetos contra la estructura de procedimientos.** La tecnología orientada a objetos obliga a la especificación sobre qué es el objeto, en contraste a cómo es usado. Los usos de un objeto dependen en alto grado sobre los detalles de la aplicación y frecuentemente cambian durante el desarrollo. Conforme los requerimientos evolucionan, las funciones otorgadas por un objeto son más estables que las formas en que son usadas, por lo tanto los sistemas de software construidos sobre estructura de objetos son más estables en el largo plazo. El desarrollo orientado a objetos tiene gran énfasis sobre la estructura de datos y menos énfasis sobre la estructura de procedimientos tradicionales de descomposición funcional.

✓ **Simergia.** La identidad, la clasificación, el polimorfismo y la herencia caracterizan la corriente principal de los lenguajes orientados a objetos. Cada uno de estos conceptos pueden ser usados aisladamente, pero unidos se complementan sinérgicamente. Los beneficios de un enfoque orientado a objetos son mayores de lo que pudiera parecer al principio. El gran énfasis sobre las propiedades esenciales de un objeto obliga al desarrollador de software a pensar más cuidadosa y profundamente acerca de qué es el objeto en cuestión y qué es lo que hace, con el resultado que el sistema es usualmente más limpio, más general y más robusto que si se hubiera hecho el énfasis sólo en el uso de datos y operaciones.

Metodología orientada a objetos

El concepto de *desarrollo orientado a objetos* se refiere a una nueva forma de pensar acerca del software basados en abstracciones que existen en el mundo real. En este contexto, desarrollo se refiere a la parte inicial del ciclo de vida del software: análisis, diseño e implementación. La esencia del desarrollo orientado a objetos es la identificación y organización de los conceptos del dominio de la aplicación, más que de su representación final en un lenguaje de programación, orientado a objetos o no. Se ha observado que la parte difícil del desarrollo de software es la manipulación de la esencia debido a complejidad inherente de un problema, más que a los accidentes de su representación en un lenguaje particular, los cuales se deben a las imperfecciones temporales de las herramientas para el desarrollo, las cuales pueden ser corregidas.

La mayoría de los esfuerzos a la fecha en la comunidad de orientación a objetos ha sido enfocada a los temas de lenguajes de programación. El mayor énfasis está sobre la implementación más que en el análisis y diseño. Los lenguajes de programación como C++ y Smalltalk, son útiles para quitar restricciones debido a la inflexibilidad de los lenguajes de programación tradicionales. Pero en sentido estricto, sin embargo, este énfasis es un paso atrás en la ingeniería de software por el enfoque excesivo sobre los mecanismos de implementación, más que en el proceso cognoscitivo inherente que lo soporta.

Las verdaderas ventajas provienen en direccionarse en los temas iniciales conceptuales, más que en los temas posteriores de implementación. Los defectos de diseño que surgen durante la etapa de implementación son más costosos de reparar que aquellos que son encontrados en etapas más tempranas. Enfocándose sobre temas de implementación demasiado rápido restringe las decisiones de diseño y frecuentemente conduce a un producto inferior. Un enfoque de desarrollo orientado a objetos fomenta a los desarrolladores de software a trabajar y pensar en términos del dominio de la aplicación a través de la mayoría del ciclo de vida de la ingeniería del software. Es solo cuando los conceptos inherentes de la aplicación son identificados, organizados y entendidos que los detalles de las estructuras de datos y funciones son direccionadas en forma efectiva.

El desarrollo orientado a objetos es un proceso conceptual independiente de un lenguaje de programación hasta sus últimas etapas. El desarrollo orientado a objetos es fundamentalmente una nueva forma de pensar y no únicamente una técnica de programación. Sus grandes beneficios provienen de la ayuda a los especificadores, desarrolladores, y clientes a expresar conceptos abstractos claramente y comunicarse entre sí. Puede servir como medio para la especificación, análisis,

documentación e interfaces, así como para programación. Aún como una herramienta de programación, puede tener varios objetivos, incluyendo lenguajes convencionales de programación y bases de datos, así como lenguajes orientados a objetos.

Desarrollo de un modelo orientado a objetos

En la presente tesis se presenta una metodología para el desarrollo orientado a objetos y notación gráfica para la representación de conceptos orientado a objetos. La metodología consiste en construir un modelo del dominio de una aplicación e ir agregando detalles de implementación durante el diseño. Esta metodología se denomina Técnica de Modelado de Objetos, OMT en sus siglas en inglés (Object Modeling Technique). La metodología contiene las siguientes etapas.

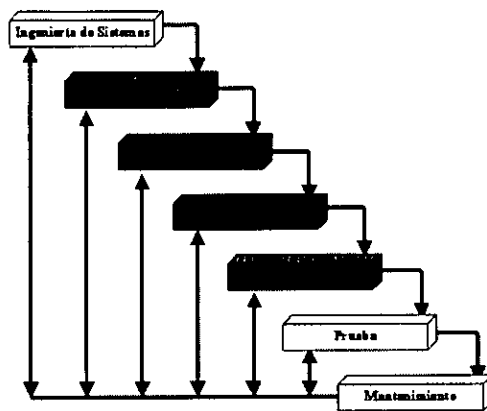


Figura 10 Ciclo de vida con enfoque orientado a objetos.

✓ **Análisis.** Comenzando con una sentencia del problema, se construye un modelo de una situación del mundo real mostrando sus propiedades más importantes. Se trabaja con el usuario para entender el problema porque las sentencias de los problemas rara vez se encuentran completas o correctas. El modelo del análisis es una abstracción concisa y precisa de qué se desea que haga el sistema, sin enfocarse en la forma que deberá ser hecho. Los objetos en el modelo son conceptos del dominio de la aplicación y no conceptos de implementación de cómputo, tales como estructuras de datos. Un buen modelo puede ser entendido y criticado por expertos de la aplicación quienes no son programadores. El modelo del análisis no debe de contener ninguna decisión de implementación.

✓ **Diseño del sistema.** Se realiza las decisiones de alto nivel acerca de la estructura general. Durante esta etapa, el sistema es organizado en subsistemas basados tanto en la estructura del análisis como en una estructura propuesta. Se decide que características de rendimiento optimizar, se escoge las estrategias de solución al problema y se realiza la elección de los recursos a utilizar. En este momento se eligen que módulos componen el sistema, cuanto usuarios van a utilizar el sistema, de que manera va a ser organizada la información.

✓ **Diseño de objetos.** Se construye un modelo de diseño basado en el modelo del análisis conteniendo detalles de implementación. Se agregan detalles al diseño del sistema de acuerdo a las estrategias establecidas durante el diseño del sistema. El enfoque del diseño de objetos es el diseño de la estructura de datos y de los algoritmos necesarios para implementar cada clase. Las clases de objetos del análisis contienen aún significado, pero son aumentadas con las estructuras de datos del dominio de cómputo y algoritmos elegidos para optimizar las medidas de rendimientos importantes. Tanto objetos del dominio de la aplicación como de cómputo son descritos usando los mismos conceptos orientados a objetos y la misma notación, aún cuando existen sobre diferentes planos conceptuales.

✓ **Implementación.** Las clases de objetos y relaciones entre las clases desarrolladas durante el diseño de objetos son finalmente trasladadas a un lenguaje de programación particular y/o base de datos. La programación debe ser una parte relativamente menor y mecánica del ciclo de desarrollo, debido a que todas las decisiones difíciles ya han sido hechas durante el diseño. Durante la implementación, es importante seguir buenas prácticas, de tal manera que el seguimiento del diseño sea directo y el sistema implementado se conserve flexible y extensible.

Los conceptos de orientación a objetos pueden ser aplicados a través del ciclo de vida del desarrollo del sistema, desde el análisis, diseño e implementación. Las mismas clases pueden ser llevadas de una etapa a otra sin necesidad de cambio de notación, aún ganando detalles de implementación en las etapas posteriores. Aún cuando el enfoque de análisis y un enfoque de implementación de un determinado objeto son correctos, sirven para diferentes propósitos y representan diferentes niveles de abstracción. Los mismos conceptos de orientación a objetos como identidad, clasificación, polimorfismo y herencia se aplican al ciclo de vida del desarrollo.

Algunas clases no son parte del análisis, pero son introducidas como parte del diseño o implementación. Se incluyen para soportar ciertos algoritmos particulares durante el diseño. Tales objetos de estructuras de datos son usados para implementar objetos del mundo real dentro del ámbito de cómputo y no derivan sus propiedades del mundo real.

Diferencias con respecto a la metodología funcional

El desarrollo orientado a objetos invierte la metodología orientada en funciones, ejemplificada por las metodologías de Yourdon y DeMarco. En estas metodologías, primariamente el énfasis se

coloca en la especificación y descomposición del sistema funcionalmente. Tal enfoque puede parecer la forma más directa para implementar la meta deseada, pero el sistema resultante puede ser frágil. Si los requerimientos cambian, el sistema basado sobre la descomposición funcional puede requerir una reestructuración masiva.

En contraste, el enfoque orientado a objetos enfatiza primeramente la identificación de objetos del dominio de la aplicación, y después acomodar procedimientos alrededor de éstos. Aún cuando pudiera parecer más indirecto, el software orientado a objetos alcanza mejor los requerimientos involucrados, pues está basado en la estructura del dominio de la aplicación en sí, más que en los requerimientos funcionales de un problema aislado.

Técnica de modelado de objetos OMT(Object Modeling Technique)

Es útil modelar un sistema por tres puntos de vistas diferentes pero relacionados, cada uno capturando aspectos esenciales del sistema, pero todos requeridos para una descripción completa. La técnica de modelado de objetos, OMT en sus siglas en inglés (Object Modeling Technique), es el nombre de la metodología propuesta por *James Rumbaugh*, en su libro *Object-Oriented Modeling and Design*, la cual combina estos tres puntos de vistas para el modelado de sistemas.

El *modelo de objetos* representa los aspectos estáticos, estructurales de los datos de un sistema. El *modelo dinámico* representa los aspectos temporales, de comportamiento y de control de un sistema. El *modelo funcional* representa los aspectos de transformación y funciones de un sistema. Un procedimiento de software tradicional incorpora los tres aspectos: Usa estructura de datos (modelo de objetos), da seguimiento a las operaciones en el tiempo (modelo dinámico), y transforma valores (modelo funcional). Cada modelo contiene referencias a entidades en los otros modelos. Por ejemplo, operaciones son adjuntadas a los objetos en el modelo de objetos pero son expandidos en el modelo funcional.

Los tres tipos de modelos separan un sistema en vistas ortogonales que pueden ser representados y manipulados con una notación uniforme. Los diferentes modelos no son completamente independientes – un sistema es más que una colección de partes independientes – sino cada modelo puede ser examinado y entendido por sí mismo a profundidad. Las interconexiones entre los diferentes modelos son limitadas y explícitas.

Modelo de objetos

Un modelo de objetos captura la estructura estática de un sistema mostrando los objetos en el sistema, relaciones entre los objetos. El modelo de objetos es el más importante de los tres modelos. Enfatiza la construcción del sistema alrededor de objetos en contraste alrededor de funciones, porque un modelo orientado a objetos corresponde más cercanamente al mundo real y consecuentemente es más resistente a los cambios. Los modelos de objetos proveen una representación gráfica intuitiva de un sistema y son valiosos para la comunicación con clientes y documentación de la estructura de un sistema.

Objetos y clases

Objetos

El propósito del modelado de objetos es el describir objetos. Un objeto se define como un concepto, abstracción o cosa con tenues límites y significado para un problema en particular. El objeto sirve para dos propósitos: promueven el entendimiento del mundo real y proveen de una base práctica para una implementación en computadora.

Todos los objetos tienen identidad y son distinguibles. El término *identidad* significa que objetos son distinguibles por su existencia inherente, y no por sus propiedades descriptivas que pudiera tener.

Clases

Una *clase de objeto* describe un grupo de objetos con propiedades similares (atributos), conducta similar (operaciones), relaciones similares con otros objetos y semántica similar. Los objetos y las clases de objeto frecuentemente aparecen como sustantivos en las descripciones de los problemas.

La abreviación de *clase* es frecuentemente usada en lugar de clase de objeto. Los objetos en la clase tienen los mismos atributos y patrones de conducta. La mayoría de los objetos derivan su individualidad de diferencias en sus valores de sus atributos y relaciones con otros objetos. Sin embargo objetos con valores de atributos idénticos y relaciones son posibles. Los objetos en una clase comparten un propósito semántico común sobre y más allá del requerimiento de atributos comunes y conducta. Definiciones comunes (tales como nombre de la clase y nombres de los atributos) son almacenadas una vez por clase más que por instancia. Operaciones pueden ser escritas una vez por clase, de tal manera que todos los objetos en la clase se beneficien del reuso de código.

Diagramas de objetos

Una vez comprendidos los conceptos de objeto y clase, éstos necesitan de una forma de expresión más precisa que las vagas definiciones en prosa, sobre todo para tópicos más complejos; Este formalismo para expresar modelos debe de ser coherente, preciso y fácil de formular.

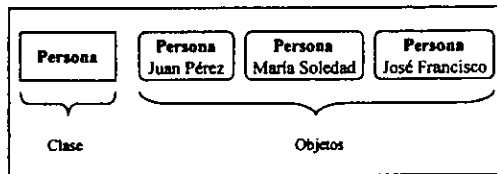


Figura 11 Clases y objetos

Los diagramas de objetos proveen de una notación formal gráfica para el modelado de objetos, clases y sus relaciones hacia otros. Existen dos tipos de diagramas de objetos: los diagramas de clases y los diagramas de instancias.

Un *diagrama de clases* describe clases de objetos; es un esquema, patrón o plantilla para describir muchas instancias posibles de datos. Un *diagrama de instancias* describe instancias de objetos; presenta cómo un conjunto particular de objetos se relacionan entre sí. Diagramas de instancia son útiles para documentar casos de prueba (en especial escenarios) y discutir ejemplos. Un diagrama de clases dado corresponde a un conjunto infinito de diagramas de instancias.

El símbolo OMT para una clase es una caja con el nombre de la clase en letra negrita. El símbolo OMT para una instancia de un objeto es una caja redondeada. El nombre de la clase entre paréntesis se encuentra en la parte superior de la caja en negritas. Los nombres de objetos son listados en letra normal.

Atributos

Un atributo es un valor de un dato retenido por los objetos en una clase. Cada atributo tiene un valor por cada instancia. Diferentes instancias pueden tener los mismos o diferentes valores en sus atributos dados. Un atributo debe de ser un dato de valor puro, no un objeto. A diferencia a los objetos, los datos de valor puro no tienen identidad.

Los atributos están listados en la segunda parte de la caja de clase. Cada nombre de atributo puede ser seguido de detalles opcionales, tales como el tipo de dato y valor por omisión. El tipo de dato es precedido por dos puntos (:). El valor por omisión es un signo de igual (=).

Operaciones

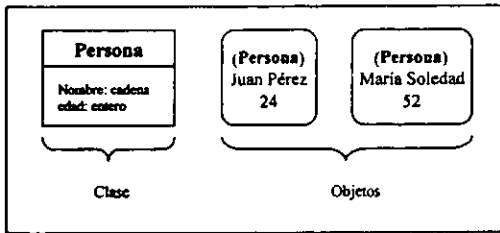


Figura 12 Atributos y valores

Una operación es una función o transformación que puede ser aplicada a o desde los objetos en una clase. Todos los objetos en una clase comparten las mismas operaciones.

Cada operación tiene un objeto destino como argumento implícito. La conducta de la operación depende de la clase que la implementa. La misma operación puede aplicar a muchas clases diferentes. Un método es la implementación de una operación para una clase en particular.

Cuando una operación tiene métodos en diversas clase, es importante que todos los métodos tengan la misma *forma*, es decir, el número y tipos de datos en los argumentos y el tipo de dato de retorno. La conducta de todos los métodos para una operación debe tener una intención consistente.

Las operaciones son listadas en la tercera parte de la clase. Cada nombre de operación es seguida con detalles opcionales, tales como la lista de argumentos y el tipo de dato de regreso. Una lista de argumentos es escrita en paréntesis seguida del nombre; los argumentos pueden ser separados por comas. El nombre y el tipo de dato de cada argumento pueden ser dados. El tipo de dato de regreso es precedido por dos puntos y debe de no ser omitido, pues es importante distinguir operaciones que regresan valores de aquellas que no lo regresan. Una lista de argumentos vacía en paréntesis muestra explícitamente que no existen argumentos; de otra manera no se puede sacar conclusiones.

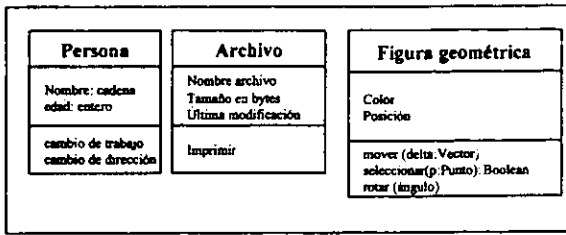


Figura 13 Ejemplo de notación de clases

Un objeto debe generalmente distinguir *atributos bases* independientes de los *atributos derivados* dependientes. En muchos casos, el objeto tiene un conjunto de atributos cuyos valores se encuentran interrelacionados, de los cuales sólo un número fijo de valores pueden ser escogidos independientemente. La elección de atributos base es arbitraria pero debe de ser evitada la sobreespecificación del estado del objeto.

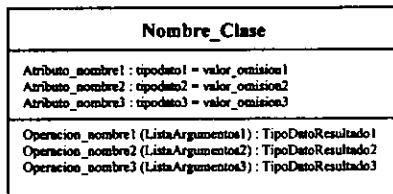


Figura 14 Notación de clases de objetos

Ligas y asociaciones

Las ligas y asociaciones son los medios para establecer relaciones entre objetos y clases. Una liga es una conexión física o conceptual entre instancias de objetos. Matemáticamente, una liga es definida como una tupla, esto es una lista ordenada de instancias de objetos. Una liga es una instancia de una asociación.

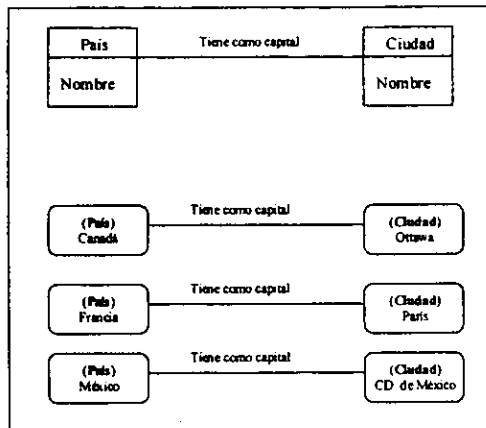


Figura 15 Asociación uno a uno, con ligas

Una asociación describe a un grupo de ligas con estructura común y semántica común. Todas las ligas en una asociación conectan a los objetos de las mismas clases. Una asociación describe el conjunto de ligas potenciales en la misma manera que una clase describe un conjunto de objetos potenciales.

La notación OMT para una asociación es una línea entre clases. Una liga es dibujada como una línea entre los objetos. Los nombres de las asociaciones se escriben en *italica*. El nombre de la asociación puede ser omitido si un par de clases tiene una simple asociación donde el significado es obvio. Los círculos en las ligas significan la multiplicidad. La multiplicidad especifica cuántas instancias de una clase pueden ser relacionadas con cada instancia de otra clase.

Las asociaciones pueden ser binarias, ternarias, o de orden mayor. El símbolo OMT para asociaciones ternarias y n-arias es un diamante con líneas conectadas a las clases relacionadas. El nombre de la asociación es escrito junto al diamante. Las asociaciones son frecuentemente dejadas sin nombre si pueden ser identificadas fácilmente por sus clases; a menos que si existen múltiples asociaciones entre los mismos objetos.

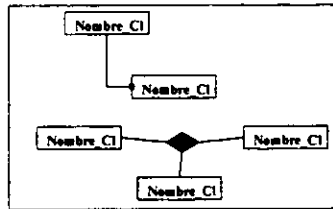


Figura 16 Ejemplo de asociación binaria y ternaria

Multiplicidad

La multiplicidad especifica cuantas instancias de una clase pueden ser relacionadas a una sola instancia de una clase asociada. Multiplicidad restringe el número de objetos relacionados. La multiplicidad frecuentemente se describe como uno o muchos, pero más general es un subconjunto (posiblemente infinito) de un conjunto de intervalos desconectados, por ejemplo el número de puertas en un automóvil. Los diagramas de objetos indican multiplicidad con símbolos especiales en los extremos de las líneas de asociación.

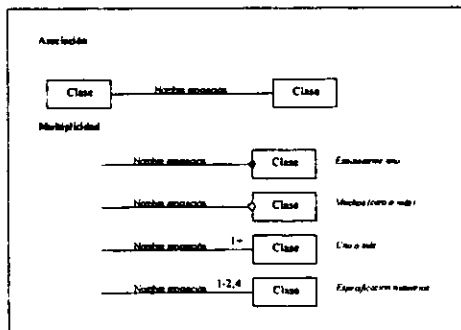


Figura 17 Notación para multiplicidad en asociaciones

La distinción más importante en multiplicidad es entre uno y muchos. Subestimando la multiplicidad puede restringir la flexibilidad de la aplicación. Por otro lado, sobrestimando la multiplicidad impone una sobrecarga y requiere de la aplicación proveer de información adicional para distinguir entre los miembros del conjunto de muchos.

Atributos en ligas

Un atributo es una propiedad de los objetos en una clase. Similarmente, un atributo de una liga es una propiedad de las ligas en una asociación. Cada atributo de la liga tiene un valor para cada liga al final de la figura. La notación OMT para un atributo de liga es una caja adjuntada a la asociación por medio de un medio elipse; uno o más ligas de atributos pueden aparecer en la segunda región de la caja. Esta notación enfatiza la similitud entre atributos para objetos y atributos para ligas.

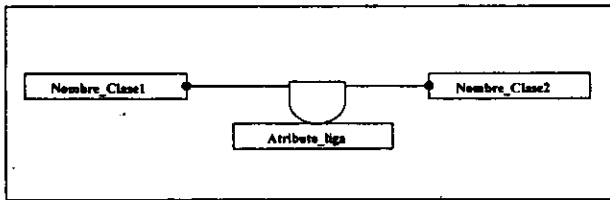


Figura 18 Ejemplo de atributos en una asociación

Las asociaciones muchos a muchos proveen el razonamiento más directo para los atributos de liga. Así cada atributo es una propiedad inherente de la liga y no puede ser adjuntada hacia ningún objeto.

Roles

Un rol es una terminal de una asociación. Una asociación binaria tiene dos roles, cada de los cuales puede tener un nombre de rol. El nombre de rol es un nombre que identifica una terminal de una asociación. Los roles proveen una forma de ver una asociación binaria a través de un objeto o un conjunto de objetos asociados con un objeto en la otra terminal. El nombre del rol es un atributo derivado cuyo valor es un conjunto de objetos relacionados.

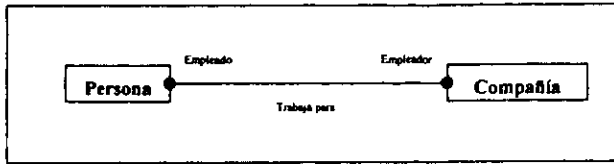


Figura 20 Ejemplo de roles

Los roles son necesarios para la asociación de dos objetos de la misma clase. Así mismo, son útiles cuando hay que distinguir dos asociaciones diferentes entre el mismo par de clases. Debido a que los nombres de los roles sirven para distinguir entre los objetos directamente conectados a un objeto dado, todos los nombres de los roles adjuntos en las terminales remotas a una clase deben de ser únicos. Aún cuando el nombre del rol es escrito cerca del objeto destino sobre una asociación, es realmente un atributo derivado de la clase fuente y debe de ser único dentro de este. Por la misma razón, el nombre del rol no debe de repetirse con algún nombre de atributo de la clase fuente.

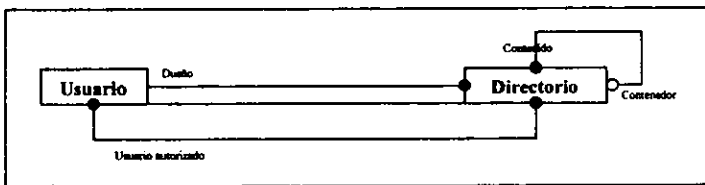


Figura 19 Ejemplo de roles varios

Ordenamiento

Usualmente los objetos del lado de “muchos” dentro de una asociación no tienen un orden específico y pueden ser considerado como un conjunto. Algunas veces, sin embargo, los objetos son explícitamente ordenados. El ordenamiento es una parte inherente de una asociación. Un conjunto ordenado de objeto del lado de “muchos” es indicado escribiendo “{ordenado}” siguiendo la multiplicidad del rol. Este es un tipo especial de restricción.

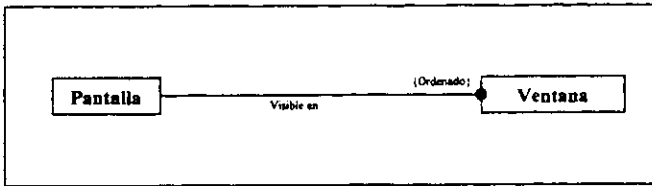


Figura 22 Ejemplo de ordenamiento

Calificativo

Una asociación calificada relaciona a dos clases y un calificador. El calificador es un atributo especial que reduce la multiplicidad efectiva de una asociación. Asociaciones uno a muchos y muchos a muchos pueden ser calificadas. El calificador distingue entre el conjunto de objetos del lado de

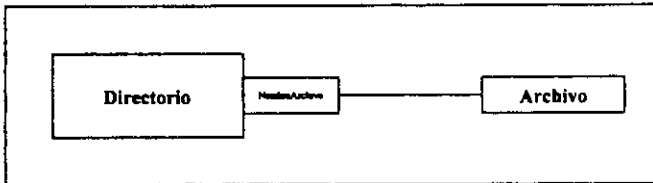


Figura 21 Ejemplo de calificativos

“muchos” en una asociación. Las asociaciones calificadas también pueden considerarse una forma de asociación ternaria.

Un calificador es dibujado como una pequeña caja en la terminal de la asociación cerca de la clase que califica. La calificación frecuentemente ocurre en problemas reales, a menudo debido a la necesidad de proveer de nombres. Normalmente existe un contexto dentro del cual el nombre tiene significado.

Agregación

La agregación es una relación de piezas a un ente mayor, en la cual los objetos representan los componentes de algo, asociados con el objeto que representa el ensamble total. Un ejemplo común es la lista de materiales, o un árbol de explosión de partes. La agregación es una forma de asociación

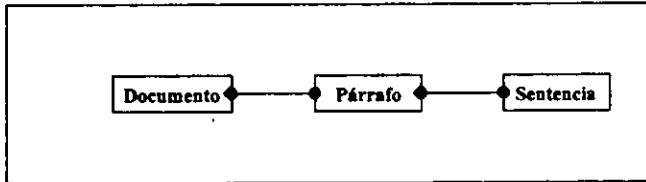


Figura 23 Ejemplo de agregación

estrechamente acoplada con alguna semántica extra. La propiedad más significativa es la transitividad, esto es, si A es parte de B y B es parte de C entonces A es parte de C. La agregación es también antisimétrica, esto es, que si A es parte de B, entonces B no es parte de A. Finalmente, algunas propiedades del ensamble se propagan hacia los componentes de alguna manera, con posibles modificaciones locales.

Definimos la relación de agregación como una relación de una clase ensamble a una clase componente. Un ensamble con muchos tipos de componentes corresponde a una muchas agregaciones. Definimos cada relación binaria como una agregación de tal manera que podemos especificar la multiplicidad de cada componente dentro del ensamble. Esta definición enfatiza que la agregación es una forma especial de asociación.

La agregación es dibujada como una asociación, excepto que un pequeño diamante indica la terminal del ensamble dentro de la relación.

Generalización y herencia.

La generalización y la herencia son abstracciones poderosas para compartir similitudes entre las clases preservando sus diferencias.

La generalización es la relación entre una clase y una o más versiones refinadas de la misma. La clase a ser refinada se denomina *Superclase* y cada versión refinada se denomina *Subclase*. Los atributos y las operaciones comunes a un grupo de subclases son adjuntados a una superclase y las comparten las subclases. La generalización es algunas veces denominada una relación “es del tipo” porque cada instancia de una subclase es una instancia de la superclase.

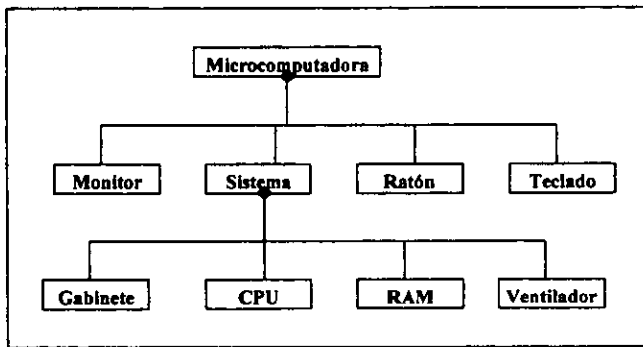


Figura 24 Ejemplo de agregación multinivel

La generalización y la herencia son transitivas a través de un número arbitrario de niveles. Los términos *antecesor* y *descendiente* se refieren a generalización de clases a través de múltiples niveles. Una instancia de una subclase es simultáneamente una instancia de todas sus clases antecesoras. El estado de una instancia incluye un valor para cada atributo de cada clase antecesora. Cualquier operación de cualquier clase antecesora puede ser aplicada a la instancia. Cada subclase no solo hereda todas las funciones de sus antecesores sino agrega sus propios atributos y operaciones.

La notación para la generalización es un triángulo conectando la superclase con sus subclases. La superclase es conectada por una línea a un vértice del triángulo. Las subclases son conectadas por líneas a una barra horizontal adjuntada a la base del triángulo. Por conveniencia, el triángulo puede ser invertido, y las subclases ser conectadas tanto arriba como abajo de la barra, pero si es posible deberán ser dibujada la superclase arriba y las subclases abajo.

Las palabras escritas cerca de los triángulos en el diagrama, tales como tipo de equipo, tipo de bomba y tipo de tanque, son discriminadores. Un discriminador es un atributo de tipo de enumeración que indica que propiedad del objeto es usada para abstraer una asociación de generalización en particular. Una propiedad debe de ser discriminada a la vez.

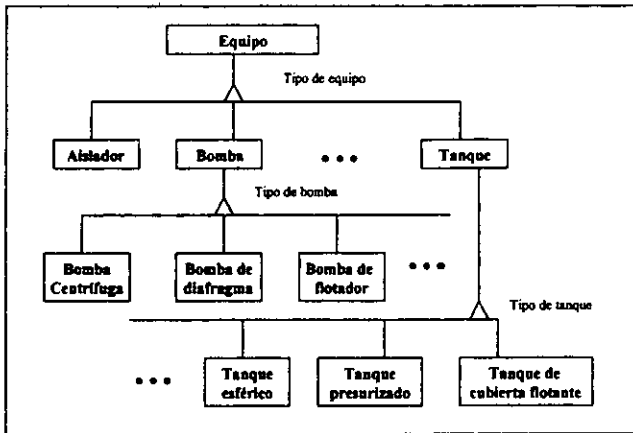


Figura 25 Ejemplo de generalización

Construcciones agrupadoras. Módulos

Un módulo es una construcción lógica para agrupar clases, asociaciones y generalizaciones. Un módulo captura una perspectiva o punto de vista de una situación. Un modelo de objetos consiste de uno o más módulos. Los módulos permiten la partición del modelo de objeto en piezas manejables. No existe una notación especial para módulos.

La misma clase puede ser referenciada en diferentes módulos. De hecho, referenciando la misma clase en diferentes módulos es el mecanismo para ligar los módulos. Deberán haber pocas ligas entre los módulos que las ligas internas del módulo.

Modelo dinámico

Las relaciones temporales son difíciles de entender. Los aspectos del sistema que conciernen al tiempo y cambios son el *modelo dinámico*, en contraste con el *modelo de objetos* estático. El *control* es ese aspecto del sistema que describe la secuencia de operaciones que ocurren en respuesta a estímulos externos sin la consideración de que hacen las operaciones o como son implementadas.

El modelo dinámico describe conceptos que manejan flujos de control, interacciones, y secuencias de operaciones en el sistema de objetos activos concurrentes. El concepto de modelado dinámico mayor es el evento, el cual representa un estímulo externo y los estados, los cuales representan valores de los objetos. El diagrama de estados es un concepto estándar en las ciencias de la computación. (Una representación gráfica de máquinas de estados finitos) que ha sido manejada en diferentes maneras en la literatura, dependiendo de su uso. Se enfatiza el uso de eventos y estados para especificar el control, más que estructuras algebraicas. Se muestra que los estados y eventos pueden ser organizados en jerarquías para compartir estructura y comportamiento.

Eventos y estados

Eventos

Un evento es algo que sucede en un momento determinado. Un evento no tiene duración. Por supuesto, nada es en realidad instantánea; un evento es simplemente una ocurrencia que es instantánea comparada con la granularidad de la escala del tiempo considerado.

Vuelo de avión (línea_aerea, numero_vuelo, ciudad_destino) Botón del ratón presionado(botón, zona) cadena de entrada(texto) Auricular levantado Dígito telefónico marcado (dígito) Velocidad de máquina entra en zona peligrosa

Figura 26 Ejemplo de eventos

Un evento puede preceder o suceder a otro en forma lógica, o dos eventos pueden estar no relacionados. Dos eventos que no están relacionados se dice que son concurrentes. Incluso si los lugares físicos de dos eventos no son distantes consideramos a los eventos concurrentes si éstos no se afectan. En el modelado del sistema no se trató de establecer un ordenamiento entre los eventos concurrentes puesto que éstos pueden ocurrir en cualquier orden. Un evento es una transmisión de

información de un solo sentido de un objeto a otro. Un evento que envía un evento a otro objeto puede esperar una respuesta pero la respuesta es un evento separado bajo el control del segundo objeto el cual puede elegir enviarlo o no.

Cada evento es una única ocurrencia, pero se agrupan en clases de eventos y se les da un nombre a cada una para indicar la estructura y comportamiento común. Esta estructura es jerárquica, de la misma manera que la estructura de objetos. Algunos eventos son simples señales, pero la mayoría de las clases de eventos tienen atributos indicando información que llevan. El tiempo en el cual ocurre un evento es un atributo implícito de todos los eventos.

Un evento transporta información de un objeto a otro, Algunas clases de eventos pueden ser simples señales de que algo ha ocurrido, mientras otras clases de eventos transportan valores de datos. Los valores de datos transportados por los eventos son sus atributos, como los valores de datos mantenidos por los objetos. Los atributos se muestran en paréntesis después del nombre de la clase del evento.

Escenarios y seguimiento de eventos

Un escenario es una secuencia de eventos que ocurren durante una ejecución particular del sistema. El alcance de un escenario puede variar, puede incluir todos los eventos en el sistema o puede incluir sólo esos eventos que repercuten o generados por ciertos objetos en el sistema. Los eventos incluyen condiciones de error al igual que ocurrencias normales.

En la figura 27 muestra cada objeto como una línea vertical y cada evento es una línea horizontal del origen al destino. El tiempo se incrementa de arriba hacia abajo, pero el espaciamiento es irrelevante; es sólo la secuencia de eventos los que son mostrados, no su ocurrencia en el tiempo en forma exacta.

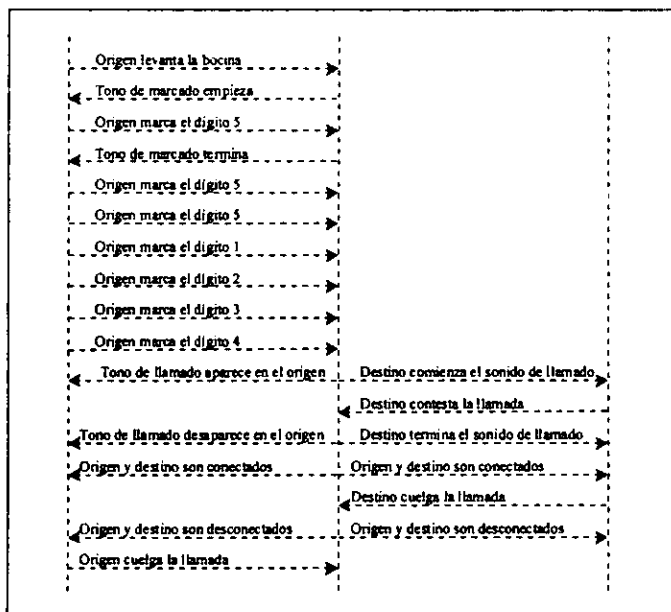


Figura 27 Ejemplo de un escenario con seguimiento de eventos

Estados

Un estado es una abstracción de los valores de los atributos y ligas del objeto. Conjuntos de valores son agrupados dentro de un estado de acuerdo con las propiedades que afectan el comportamiento en general del objeto. Un estado especifica la respuesta del objeto a los eventos de entrada. La respuesta a un evento recibido por un objeto puede variar cuantitativamente dependiendo de los valores exactos de sus atributos, pero la respuesta es cualitativamente la misma para todos los valores dentro del mismo estado, pero diferente para valores de diferentes estados. La respuesta de un objeto a un evento puede incluir una acción o un cambio del estado por el objeto.

Un estado corresponde a un intervalo entre dos eventos recibidos por un objeto. Los eventos representan momentos en el tiempo. El estado de un objeto depende de la secuencia anterior de eventos que ha recibido, pero en la mayoría de los casos los eventos pasados son eventualmente olvidados por los eventos subsecuentes.

Un estado tiene duración, ocupa un intervalo del tiempo. Un estado es asociado frecuentemente con una actividad continua que toma tiempo para completarse. Los eventos y estado son elementos duales; un evento separa dos estados, y un estado separa dos eventos. Un estado se asocia generalmente con el valor de un objeto que satisface alguna condición.

En la definición de estados, se ignora aquellos atributos que no afectan el comportamiento de los objetos y los agrupamos en un solo estado todas las combinaciones de los valores de los atributos y ligas que tienen la misma respuesta a eventos. Por supuesto que cada atributo tiene algún efecto en el comportamiento o sería sin sentido, pero algunos atributos frecuentemente no afectan el patrón de control y puede ser modelado como simples parámetros dentro de un estado dado. Algunas ocasiones todos los valores posibles de un atributo son importantes, pero usualmente el número de valores posibles es pequeño. Tanto eventos y estados dependen del nivel de abstracción usado.

Un estado puede ser caracterizado en diversas formas. El estado tiene un nombre sugestivo y el lenguaje de la descripción natural para su propósito. Una condición declarativa para el estado es dato en términos de parámetros

Estado:	Sonando alarma
Descripción:	La alarma del reloj esta sonando para indicar que ha llegado a la hora predeterminada.
Secuencia de eventos que producen el estado:	Configurar la alarma Cualquier secuencia que no limpie la alarma $TiempoActual = HoraPredeterminada$
Condición que caracteriza al estado	Alarma encendida $HoraPredeterminada \leq HoraActual$ $HoraActual \leq HoraPredeterminada + 20 \text{ segundos}$ Ningún BotónPresionado
Eventos que aceptan el estado	Evento: $HoraActual = HoraPredeterminada + 20$ Acción: Limpiar la alarma Siguiete Estado: Normal Evento: BotónPresionado Acción: Limpiar la alarma Siguiete Estado: Normal

Figura 28 Características de un estado

Un caso especial son las ligas. Desde el momento en que las ligas son consideradas objetos, las ligas pueden tener estado. En forma práctica, generalmente es suficiente asociar un estado solo a un objeto. El estado de un objeto puede incluir los valores de sus ligas.

Diagramas de estado

Un diagrama de estado muestra eventos y estados. Cuando un evento es recibido, el siguiente estado depende del estado actual así como del evento; un cambio de estado causado por un evento se denomina *transición*. Un diagrama de estado es un grafo donde los nodos son estados y los segmentos dirigidos son transiciones etiquetadas con nombres de eventos. Un estado es dibujado por una caja redondeada conteniendo un nombre opcionalmente. Una transición es dibujada como una flecha desde el estado emisor al estado receptor; la etiqueta de la flecha es el nombre del evento causado por la transición. Todas las transiciones dejando un estado deben corresponder a diferentes eventos.

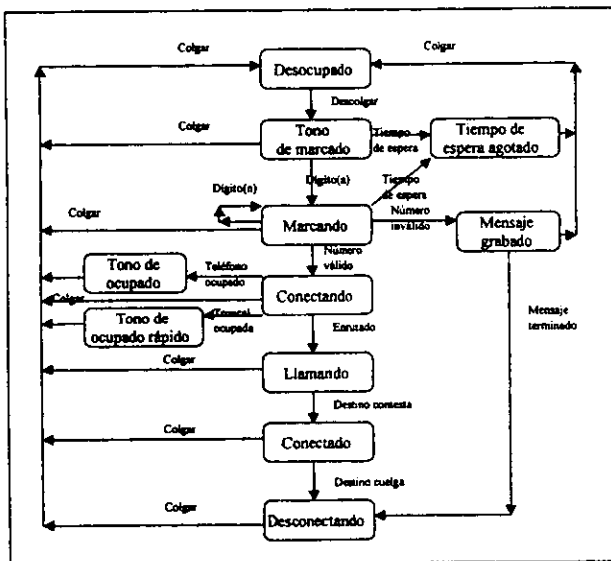


Figura 29 Diagrama de estados de una línea telefónica

El diagrama de estados especifica la secuencia de estados causado por una secuencia de eventos. Si un objeto está en un estado y un evento etiquetando una de sus transiciones ocurre, el objeto entra en el estado al final destino de la transición. Se dice que la transición se dispara. Si más de una transición sale del estado, entonces el primer evento que ocurre causa la transición correspondiente ser disparada. Si un evento ocurre que no tiene transición de salida del estado actual, entonces el evento es ignorado. Una secuencia de eventos corresponde a una ruta a través del grafo.

Un diagrama de estados describe la conducta de una sola clase de objetos. Debido a que todas las instancias de una clase tienen la misma conducta (por definición), todos comparten el mismo diagrama de estados, así como toman todas las funcionalidades de la misma clase. Sin embargo, como cada objeto tiene sus propios valores de atributos, por lo tanto el objeto es independiente de otros objetos y trabaja en su propio ámbito.

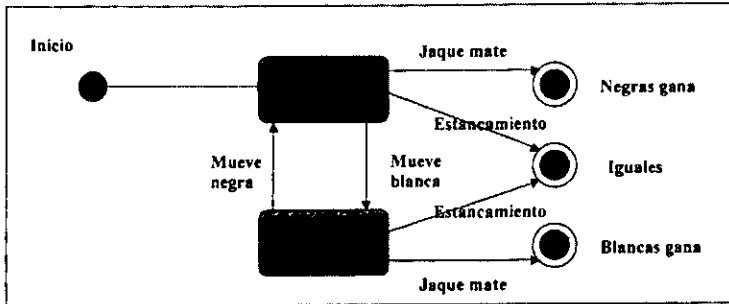


Figura 30 Ciclo de vida de un juego de ajedrez

Un diagrama de estados puede representar ciclos de vida lineales o de ciclos continuos. El diagrama del comportamiento de una línea telefónica es un ciclo continuo. En la descripción del uso ordinario de un teléfono, no se necesita saber o preocuparse de cómo el ciclo comienza (Si fuera la instalación de nuevas líneas, el estado inicial sería importante). Los diagramas de ciclos de vida lineales representan objetos con vida finita. Un diagrama de vida lineal tiene un estado inicial y un estado final. El estado inicial es entrado en la creación de un objeto. La llegada a un estado final implica la destrucción del objeto. Un estado inicial es mostrado como un círculo relleno. El círculo puede ser etiquetado con diferentes condiciones iniciales. Un estado final es mostrado mediante una figura denominada "ojo de buey", es decir un círculo vacío rodeando un círculo relleno. El estado final puede ser etiquetado para distinguir condiciones finales.

Condiciones

Una condición es una función booleana de valores del objeto. Una condición es válida sobre un intervalo de tiempo. Es importante distinguir las condiciones de los eventos, los cuales no tienen duración. Un estado puede ser definido en términos de una condición; análogamente, estar en un estado es una condición.

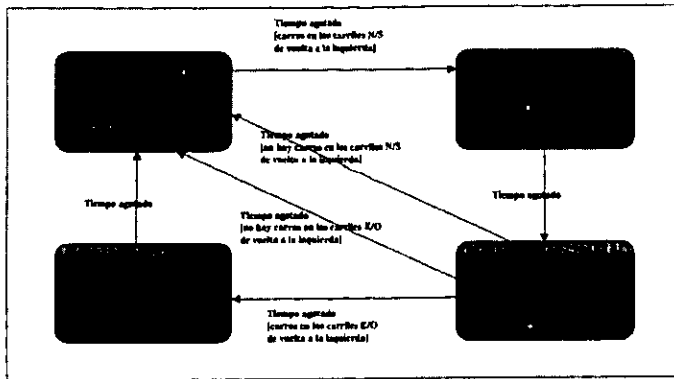


Figura 31 Diagrama de estado con transiciones verificadas

Las condiciones pueden ser usadas como verificadores sobre las transiciones. Una transición verificada se dispara cuando sus eventos ocurren, pero sólo si la condición se cumple. Una condición verificada sobre una transición se muestra como una expresión booleana en llaves siguiendo el nombre del evento.

Operaciones

Los diagramas de estados presentados hasta este párrafo describen los patrones de eventos y estados de una sola clase de objetos. En la siguiente sección se mostrará como las operaciones disparan los eventos.

Control de operaciones

Los diagramas de estados serían de poco uso si sólo describieran patrones de eventos. La descripción del comportamiento de un objeto debe especificar que hace el objeto en respuesta a eventos. Las operaciones adjuntadas a estados o transiciones son ejecutadas en respuesta a estados o eventos correspondientes.

Una *actividad* es una operación que toma un tiempo en ser completada. Una actividad esta asociada con un estado. Las actividades incluyen operaciones continuas, así como operaciones secuenciales que terminan por sí mismas después de un intervalo de tiempo. Un estado puede controlar una actividad continua, que persiste hasta que el evento termina causando un cambio de estado. La notación "*realizar: A*" dentro de la caja del estado indica que la actividad A empieza en el momento de entrar al evento y termina en el momento de salir del estado. Un estado también puede controlar una actividad secuencial, que va progresando hasta se completa o hasta que es interrumpida por un evento

que lo termina prematuramente. La misma notación "*realizar: A*" indica que la actividad secuencial *A* comienza en la entrada del estado y sale cuando se termina la actividad. Si un evento causa una transición del estado antes de que la actividad se complete, entonces la actividad es terminada prematuramente. Los dos usos no son realmente diferentes; una actividad continua puede ser vista como una actividad secuencial que dura indefinidamente.

Una *acción* es una operación instantánea, una acción es asociada con un evento. Una acción representa una operación cuya duración es insignificante comparada con la resolución de tiempo manejada por el diagrama de estado. Una operación del mundo real no es realmente instantánea, sin embargo, al modelarla como una acción, indicamos que no interesa su estructura interna para propósitos de control. Si es importante la estructura interna, entonces deberá ser modelada como una actividad, con un evento de inicio, final, y posibles estados intermedios.

Las acciones pueden también representar operaciones de control interno, tales como cambiar los atributos o la generación de otros eventos. Tales acciones no tienen contrapartes reales en el mundo real, sino son mecanismos para la estructuración del control dentro de una implementación.

La notación para una acción sobre una transición es un símbolo diagonal (/) y el nombre o descripción de la acción, siguiendo al nombre del evento que lo causa.

Diagramas de estado anidados

Los diagramas de estados pueden ser estructurados para permitir descripciones concisas de situaciones complejas. Las formas de estructuración de máquinas de estados son similares a las formas de estructuración de objetos: generalización y agregación. La generalización es equivalente a expandir las actividades anidadas. Permite que una actividad sea descrita en un alto nivel, después sea expandida a niveles inferiores agregando detalles, similar a una llamada por procedimientos anidada. Además, la generalización permite arreglar los estados y los eventos dentro de jerarquías de generalización, heredando estructura y conducta común, similar a la herencia de atributos y operaciones en las clases.

La agregación permite dividir un estado en componentes ortogonales, con interacción limitada entre ellos, similar a la jerarquía de agregación de objetos. La agregación es equivalente a la concurrencia de estados. Los estados concurrentes generalmente corresponden a objetos agregados, posiblemente dentro de un sistema completo, el cual tiene partes interactuantes.

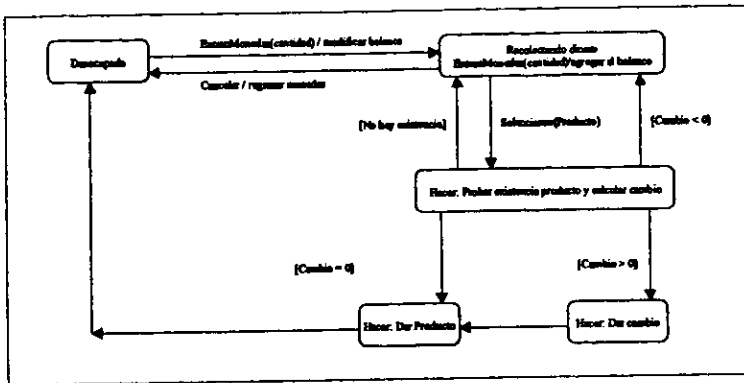


Figura 32 Modelo de máquina vendedora

Anidando diagramas de estado

Una actividad es un estado puede ser expandida como un nivel más inferior en un diagrama de estado, cada estado representando un paso de la actividad. Actividades anidadas son diagramas de estados de una sola vista, con una transición de entrada y una de salida, similar a una subrutina. El conjunto de diagramas de estados anidadas forma un enrejado o árbol, si se expande diferentes copias del mismo diagrama.

En la figura 32 se muestra un modelo de alto nivel de una máquina vendedora. Este diagrama contiene una actividad *DarProducto* y un evento *Seleccionar(Producto)* que son expandidos en mayor detalle en los diagramas de estados anidados.

La figura 33 muestra un subdiagrama para la actividad *DarProducto* de la figura 32. Esta

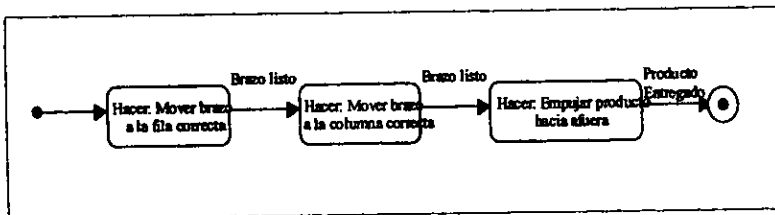


Figura 33 Actividad *DarProducto* de la máquina vendedora

actividad corresponde a una secuencia de los estados de un nivel más bajo y eventos que son invisibles al diagrama original de alto nivel.

Los eventos también pueden ser expandidos dentro de diagramas de estado subordinados. La figura 33 muestra el evento Seleccionar(Producto) de la figura 32, la cual actualmente involucra diversos eventos de bajo nivel. La etiqueta en el "ojo de buey" indica el evento generado en el diagrama de estados de alto nivel.

Generalización de estados

Un diagrama de estados anidado es en realidad una forma de generalización sobre estados. La generalización es una relación tipo "O-Exclusivo". Un objeto en un estado en un diagrama de alto nivel debe estar en exactamente un solo estado en el diagrama anidado. Debe de estar en el primer estado, o en el segundo estado, o en un de los otros estados. Los estados en el diagrama anidado son todos refinamientos del estado en el diagrama de alto nivel. En la sección previa, los estados en un diagrama anidado no son afectados por las transiciones en el diagrama de alto nivel, pero por lo general los estados en un diagrama de estados anidado pueden interactuar con otros estados.

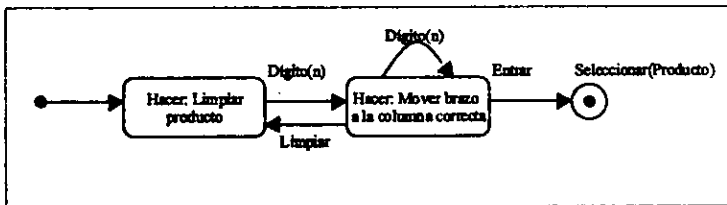


Figura 34 Transición Seleccionar(Producto) de la máquina vendedora

Los estados pueden tener subestados que hereden las transiciones de sus superestados, tal y como las clases tienen subclases que heredan los atributos y operaciones de sus superclases. Cualquier transición o acción que aplica a un estado aplica a todos sus subestados, a menos que se sobrespecifique una transición equivalente sobre el subestado.

La figura 35 muestra un diagrama de estados para una transmisión automática. La notación para la generalización para estados es diferente a la usada para clases, para evitar un número grande de líneas que puedan ser confundidas con transiciones. Un superestado es dibujado como una caja redondeada encerrando a todos sus subestados. Los subestados a su vez pueden encerrar a sus

subestados. Debido a que las cajas redondeadas representando diversos estados están anidadas, se les denominan contornos.

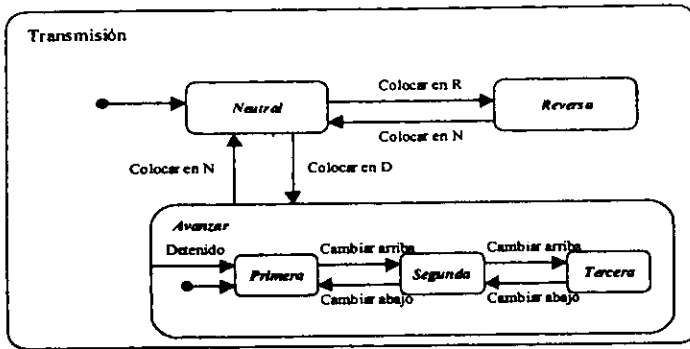


Figura 35 Diagrama de estados de una transmisión automotriz con generalización

Es posible representar situaciones más complicadas, tales como una transición explícita desde un subestado a un estado fuera del control, o una transición explícita hacia dentro del contorno. En tales casos, todos los estados deben aparecer sobre un diagrama usando la notación de contorno. En caso más simples, donde no existen interacciones excepto la inicialización y la terminación, los estados anidados puede simplemente dibujados como un diagrama separado y referenciado por el nombre en una sentencia "Hacer", como en el ejemplo de la máquina vendedora de la figura 26.

Generalización de eventos

Los eventos pueden ser organizados dentro de una jerarquía de generalización con herencias de eventos atributos. La figura 30 muestra parte de un árbol de eventos de entrada para una estación de trabajo.

Proveer de una jerarquía de eventos permite diferentes niveles de abstracción para ser usada en diferentes lugares del modelo. Por ejemplo, en algunos estados todos los caracteres de entradas pueden ser manejados igualmente y pueden conducir al mismo estado; en otros estados los caracteres de control pueden ser tratados en forma diferente a los caracteres de impresión; aún en otros pueden tener diferentes acciones sobre caracteres individuales.

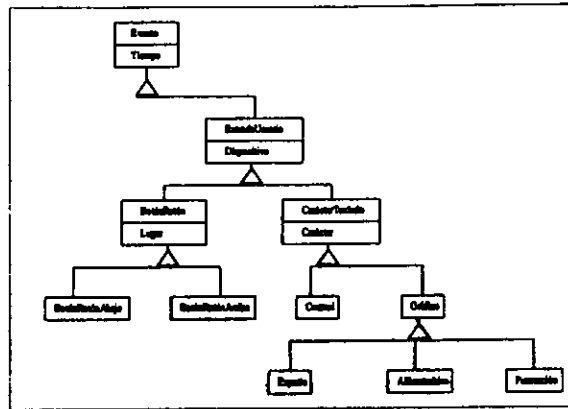


Figura 36 Jerarquía parcial de eventos para eventos de techado

Concurrencia

Concurrencia en la agregación

Un modelo dinámico describe un conjunto de objetos concurrentes, cada uno con su propio estado y diagrama de estado. Los objetos en un sistema son inherentemente concurrentes y pueden cambiar de estado independientemente. El estado de todo el sistema no puede ser representado por un solo estado en un solo objeto; es el producto de los estados de todos los objetos en él. En muchos sistemas, el número de objetos también puede cambiar dinámicamente.

Un diagrama de estado para un ensamble es una colección de diagrama de estado, uno por cada componente. La agregación implica concurrencia. El estado de agregación corresponde a los estados combinados de todos los diagramas de los componentes. La agregación es una relación tipo "Y". El estado de agregación es un estado desde el primer diagrama, y un estado del segundo diagrama, y un estado de cualquier otro diagrama. En casos más interesantes, los estados de los componentes interactúan. Las transiciones verificadas para un objeto pueden depender de que otros objetos estén en determinado estado. Esto permite la interacción entre diagramas de estado, manteniendo la modularidad.

La figura 31 muestra el estado de un auto como una agregación de estados de sus componentes.

Concurrencia dentro de un objeto

La concurrencia dentro del estado de un solo objeto sobresale cuando un objeto puede ser particionado en subconjuntos de atributos o ligas, cada cual con su propio subdiagrama. El estado del objeto conjunta un estado de cada subdiagrama.

Los subdiagramas no necesitan ser independientes; el mismo evento puede causar transiciones en más de un subdiagrama. La concurrencia dentro de un solo estado compuesto de un objeto es mostrando particionando el estado compuesto en subdiagrama con líneas punteadas. El nombre del estado compuesto total puede ser escrito en una región separada con una línea de los subdiagramas concurrentes. La figura 32 muestra el diagrama de estado para el juego de dos de tres ganados.

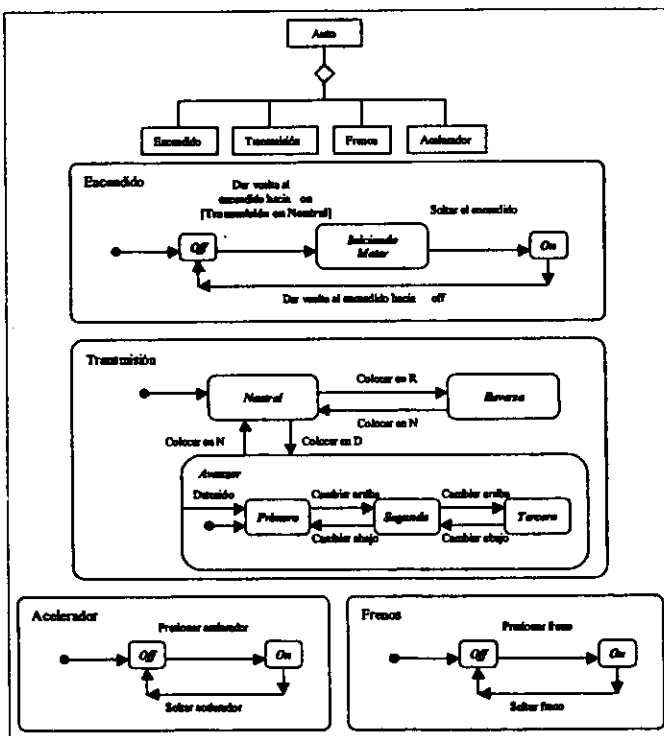


Figura 31 Una agregación y sus diagramas de estado concurrentes

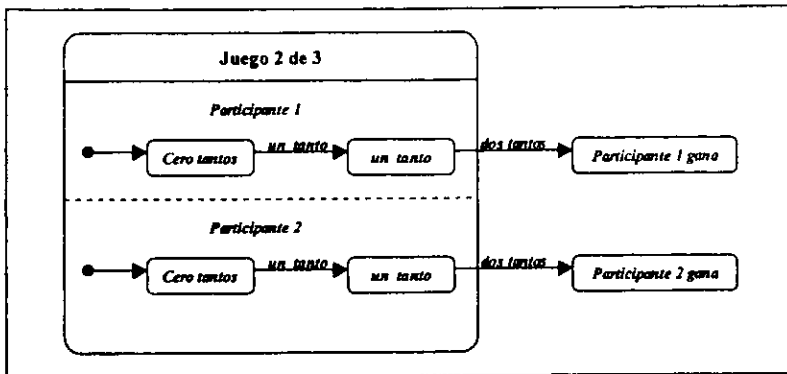


Figura 37 Juego 2 de 3 con estados concurrentes

Conceptos de modelado dinámico avanzados

En esta sección se presenta conceptos de modelado dinámico avanzados, así como algunos refinamientos en la notación.

Acciones de entrada y salida

Una alternativa a mostrar acciones en las transiciones son las acciones que pueden asociarse a la entrada o salida de un estado. Esto no representa una diferencia en el poder de expresión entre las dos notaciones, sino que frecuentemente todas las transiciones de entrada hacia un estado realizan la misma acción, ocasión en la cual la asociación de la acción a la entrada al estado es más concisa.

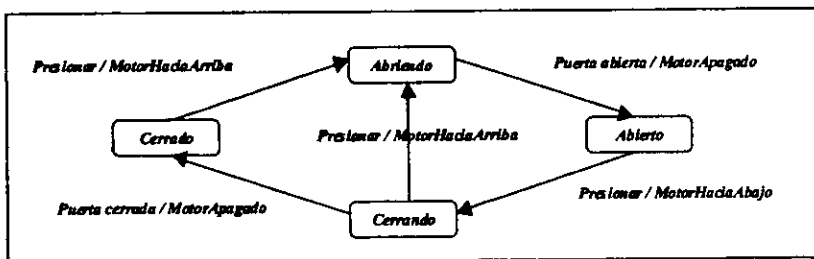


Figura 38 Acciones en transiciones

Por ejemplo, en la figura 33 muestra el control de un portero de puerta de cochera. El usuario genera eventos *presionó* con un botón para abrir y cerrar la puerta. Cada evento invierte la dirección del motor, pero por seguridad, la puerta abre completamente antes de que pueda ser cerrada. El control

genera acciones sobre un motor, *MotorHaciaAbajo* y *MotorHaciaArriba*. El motor genera los eventos *PuertaAbierta* y *PuertaCerrada* cuando el movimiento ha sido completado. Ambas transiciones en la entrada del estado Abriendo causan a la puerta abrirse.

La figura 34 muestra el mismo modelo usando acciones sobre la entrada a estados. Una acción de entrada se muestra dentro de la caja de estado seguida de la palabra reservada *Entrada* y un símbolo "/". Cuando al estado se entra, por cualquier transición, la acción de entrada se realiza. Una acción de entrada es equivalente a asociar la acción a cada transición de entrada. Si una transición de entrada tiene una acción predefinida, ésta es ejecutada antes que la acción de entrada.

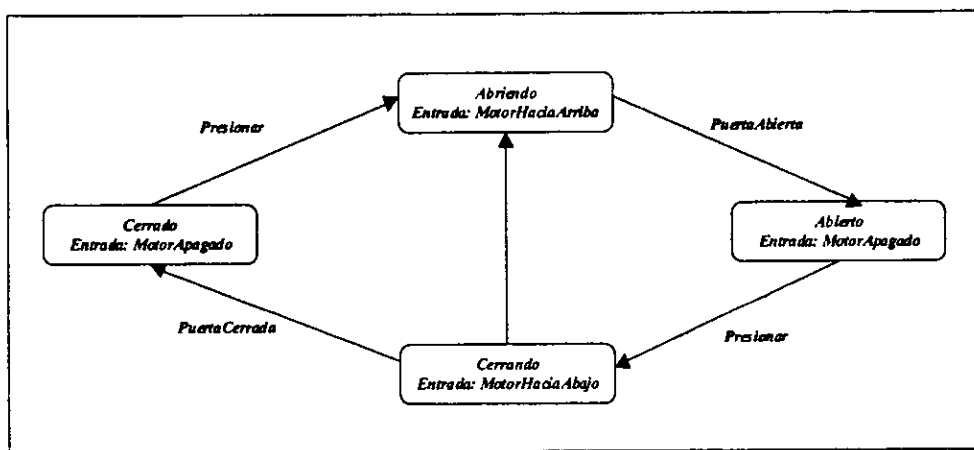


Figura 39 Acciones en la entrada a estados

Las acciones de salida son menos comunes que las acciones de entrada, pero son ocasionalmente útiles. Una acción de salida se muestra dentro de la caja de estado seguida por la palabra reservada *Salida* y un símbolo "/". Siempre que del estado se sale, por cualquier transición de salida, la acción de salida se realiza primero.

Si múltiples operaciones son especificadas sobre un estado, éstas son realizadas en el siguiente orden: Las acciones de la transición de entrada, las acciones de entrada, realizar actividades, acciones de salida, acciones de la transición de salida. Las actividades del estado pueden ser interrumpidas por eventos que causen la transición hacia afuera del estado, pero las transiciones de entrada y de salida son completadas sin excepción, puesto que éstas se consideran inmediatas. Si una actividad es interrumpida, las acciones de salida son realizadas.

Las acciones de entrada y salida son particularmente útiles en diagramas de estado anidados puesto que permite a un estado (posiblemente un subdiagrama completo) ser expresado en términos de las acciones de entrada salida equiparadas sin importar que suceda antes o después que del estado esté activo. Es posible usar acciones asociadas a transiciones así como de entrada y salida en un diagrama.

Hacer la transición hacia adentro o afuera de un subestado en un diagrama anidado puede causar la realización de diversas acciones de entrada y salida, si la transición se realiza a través de diversos niveles de generalización. Las acciones de entrada se realizan de afuera hacia adentro y las acciones de salida se realizan de adentro hacia afuera. Esto permite una conducta similar a las llamadas de subrutina.

Acciones internas

Un evento puede causar a una acción ser realizada son causar un cambio de estado. El evento es escrito dentro de la caja de estado, seguida de un símbolo “/” y el nombre de la acción. (Las palabras reservadas dentro de una caja de estado son *Entrada*, *Salida* y *Hacer*). Cuando tal evento sucede, su acción es ejecutada, pero no se realizan las acciones de entrada o de salida para el estado. Por lo tanto, existen diferencias entre un acción interna y una transición automática; la transición automática causa que las acciones de entrada y de salida para el estado sean ejecutadas. La figura 26 muestra una acción interna dentro del estado RecolectandoDinero.

La figura 35 resume la notación adicional para las acciones de entrada, salida e internas

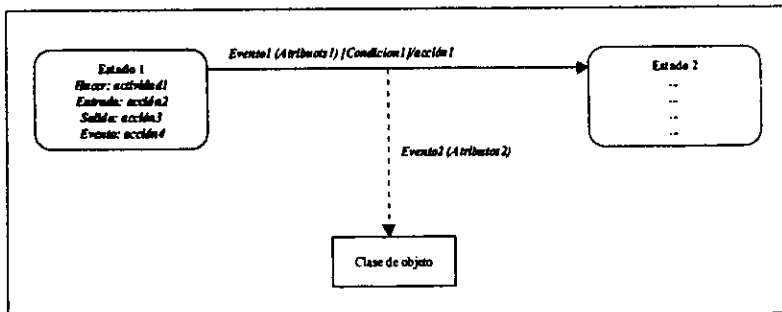


Figura 40 Resumen de la notación extendida para diagramas de estados

Transiciones automáticas

Frecuentemente el único propósito de un estado es el realizar una actividad secuencial. Cuando la actividad es completada, una transición a otro estado se dispara. Una flecha sin un nombre de evento indica una transición automática que se dispara cuando la actividad asociada con el estado inicial se completa. Si no existe ninguna actividad por realizar, la transición no etiquetada se dispara tan pronto como se entra al estado (pero las acciones de entrada y de salida se ejecutan siempre). Tales transiciones no etiquetadas son llamadas algunas veces transiciones lambda, la letra griega usada para indicar este tipo de transiciones. La figura 26 muestra cuatro transiciones no etiquetadas desde el estado conteniendo la actividad ProbarExistenciaProductoyCalcularCambio. Cada transición tiene una condición verificadora. Cuando la actividad se completa, la transición con una condición verificadora se dispara.

Si un estado tiene uno o más transiciones automáticas, pero ninguna de las condiciones verificadoras es satisfecha, entonces el estado permanece activo hasta que una de las condiciones es satisfecha o hasta un evento dispare otra transición. El cambio de valor de una condición es un evento implícito.

Envío de eventos

Un objeto puede realizar la acción de enviar un evento a otro objeto. Un sistema de objetos interactúa con el intercambio de eventos.

La acción *enviar E(atributos)* envía un evento E con los atributos dados al objeto o objetos que lo reciben. Un evento puede ser dirigido a un conjunto de objetos o a un solo objeto. Cualquiera o todos los objetos con transiciones sobre el evento pueden aceptarlo concurrentemente. La palabra *Enviar* puede ser omitida si es suficientemente claro que E es el nombre de un evento. En los diagramas, los nombres de los eventos están escritos con itálicas y los nombres de acciones con texto normal, para evitar confusiones.

La figura 35 muestra otra notación para enviar un evento de un objeto a otro. La línea punteada desde una transición a un objeto indica que un evento es enviado al objeto cuando la transición se dispara. La flecha puede ser conectada directamente a una transición dentro de un diagrama de estados al objeto destino para indicar que la transición destino depende del evento.

Si un estado puede aceptar eventos de más de un objeto, el orden en el cual los eventos concurrentemente son recibidos pueden afectar el estado final; esto se le llama una *carrera de condiciones*. Una carrera de condiciones no es necesariamente un error de diseño, pero los sistemas concurrentes frecuentemente contienen carreras de condiciones indeseadas las cuales deben ser evitadas con un diseño cuidadoso. Un requerimiento de dos eventos siendo recibidos simultáneamente no es una situación con significado en el mundo real, con la ligera variación en las velocidades de una transmisión que son inherentes a un sistema distribuido.

Cuando un objeto interactúa con un objeto externo, como una persona o dispositivo, enviar un evento es frecuentemente una forma indistinguible de una acción

Sincronización de actividades concurrentes

Algunas veces un objeto debe realizar dos o más actividades concurrentemente. Los pasos internos de las actividades no están sincronizados, pero ambas actividades deben ser completadas antes de que el objeto pueda progresar al siguiente estado.

En la figura 36 muestra un diagrama de estados concurrente para una actividad emisora. Actividades concurrentes dentro de una sola actividad compuesta son mostradas particionando un estado en regiones con líneas punteadas, como se explicó anteriormente. Cada región es un subdiagrama que representa actividades concurrentes dentro de la actividad compuesta. La actividad compuesta asume exactamente un estado para cada subdiagrama.

Separar el control en partes concurrentes es mostrado por una flecha que se divide. Las flechas divididas seleccionan un estado de cada subdiagrama concurrente. En el ejemplo, la transición sobre el evento listo se separa en dos partes concurrentes, uno a cada subdiagrama concurrente. Cuando esta transición se dispara, dos subestados concurrentes se convierten en activos y se ejecutan independientemente. Cada subestado puede ser un diagrama de estado completo.

Cualquier transición hacia dentro de un estado con subdiagramas concurrentes activa cada uno de los subdiagramas concurrentes. Si cualquiera de los subdiagramas es omitido de la transición, empezará en el estado inicial por omisión. Una transición puede ser dibujada al estado emisora, con el estado inicial por omisión indicado en cada subdiagrama.

Para conjuntar el control de la concurrencia es mostrado por una flecha con la cola dividida. El estado objetivo se convierte en el activo cuando ambos eventos ocurren en cualquier orden. Los

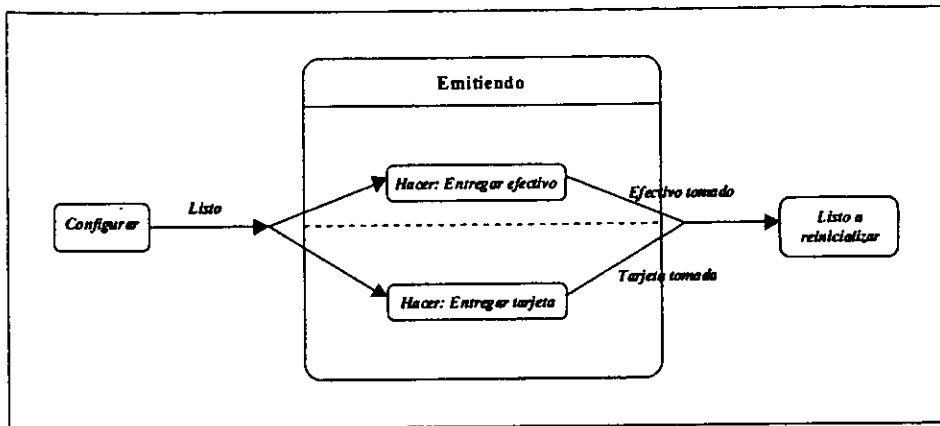


Figura 41 Control sincronizado

eventos no necesitan ser simultáneos. Cada subdiagrama termina tan pronto como sus partes de la transición se dispara, pero todas las partes de la transición se deben disparar antes de que la transición completa se dispare y el estado compuesto se termine. Si existe cualquier subdiagrama en el estado compuesto que no es parte de la conjunción, entonces éstos son automáticamente terminados cuando la transición conjuntada termina. Las acciones de salida (si existieran) de todos los subdiagramas son realizados cuando la transición conjuntada se dispara. En el ejemplo, las transiciones de efectivo tomado y tarjeta tomada son parte de una sola transición conjunta. Cuando ambas partes de la transición conjunta se dispara *Listo a reinicializar* se convierte en el estado activo. Dibujando una transición separada de cada subestado a un estado objetivo tendría un significado diferente: cualquier transición puede terminar el otro subdiagrama sin esperar por la otra transición.

Modelo funcional

El modelo funcional describe los cálculos realizados dentro de un sistema. El modelo funcional es la tercera parte del modelo tripartita, junto con el modelo de objetos y el modelo dinámico. El modelo funcional especifica que sucede, el modelo dinámico especifica cuando sucede y el modelo de objetos especifica a que o a quienes les suceden.

El modelo funcional muestra como los valores de salida son calculados y derivados a partir de los valores de entrada, sin tomar en cuenta el orden en el cual los valores son calculados. El modelo funcional consiste de múltiples diagramas de flujo de datos los cuales muestran los flujos de valores a partir de las entradas externas, a través de operaciones y datos almacenados internamente, hasta los datos de salida externos. El modelo funcional también incluye los rangos de valores para los atributos de los objetos. Los diagramas de flujo de datos no muestran la información de la estructura de control o de objetos; éstos pertenecen a los modelos dinámicos y de objetos.

Modelos funcionales

El modelo funciona especifica los resultados de un cálculo sin especificar como o cuando son calculados. El modelo funcional especifica el significado de las operaciones en el modelo de objetos y las acciones en el modelo dinámico, así como cualquier limitación en modelo de objetos. Programas no interactivos como un compilador, tienen un modelo dinámico trivial; su propósito es calcular una función. El modelo funcional es el modelo principal para tales programas, aún cuando el modelo de objetos es importante por cualquier problema con estructuras de datos no triviales. Muchos programas interactivos también tienen un modelo funcional. En contraste, las bases de datos frecuentemente tienen un modelo funcional trivial, puesto que su propósito es almacenar y organizar datos, no transformarlos.

Diagramas de flujo de datos

El modelo funcional consiste de múltiples diagramas de flujo de datos los cuales especifican el significado de operaciones y limitaciones. Un diagrama de flujo de datos (DFD) muestra las relaciones funcionales de los valores calculados por un sistema, incluyendo los valores de entrada, valores de salida y almacenamiento de datos internos. Un diagrama de flujo de datos es una gráfica mostrando el flujo de valores de datos desde sus fuentes en objetos por medio de procesos que los transforman hacia sus destinos en otros objetos. Un diagrama de flujo de datos no muestra la información de control, tal como los momentos en que los procesos son ejecutados o decisiones se hacen a través de diversas rutas

de ejecución; esta información pertenece al modelo dinámico. Un diagrama de flujo de datos no muestra la organización de los datos en objetos; esta información pertenece al modelo de objetos.

Un diagrama de flujo de datos contiene *procesos* que transforman datos, *flujos de datos* que mueven datos, *actores* que producen y consumen datos, y *almacenamiento de datos* que guardan datos en forma pasiva. La figura 37 muestra un diagrama de flujo de datos para el despliegue de un icono en un sistema por ventanas.

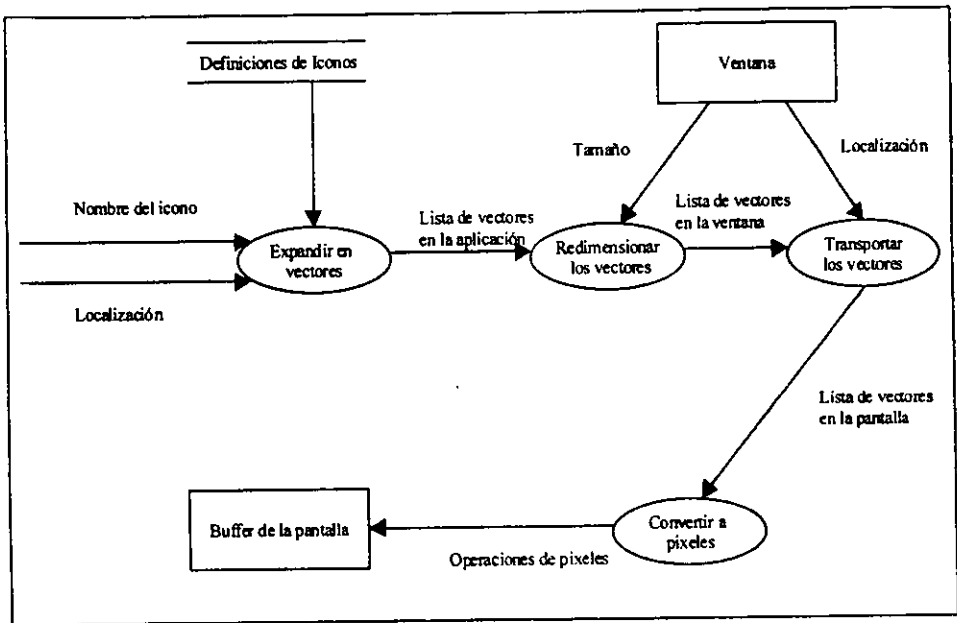


Figura 42 Diagrama de flujo de datos para un sistema gráfico por ventanas

Procesos

Un proceso transforma datos. Los procesos de bajo nivel son funciones puras sin efectos secundarios. Funciones típicas son la suma de dos números, la carga financiera de un conjunto de transacciones de tarjeta de crédito, el dibujo de una línea en una serie de puntos. Un diagrama de flujo de datos entero es un proceso de alto nivel. Un proceso puede tener efectos secundarios si no contiene componentes no funcionales, como los almacenamientos de datos y objetos externos. El modelo funcional solo indica las posibles rutas funcionales; no muestra la ruta que en realidad ocurrirá. Los resultados de tales procesos dependen de la conducta del sistema, como se especifica en el modelo dinámico.

Un proceso es una elipse conteniendo una descripción de la transformación, usualmente su nombre. Cada proceso tiene un número fijo de flechas de entrada y de salida, cada la cual transporta un dato de determinado tipo. Las entradas y salidas pueden ser etiquetadas para mostrar su funcionamiento

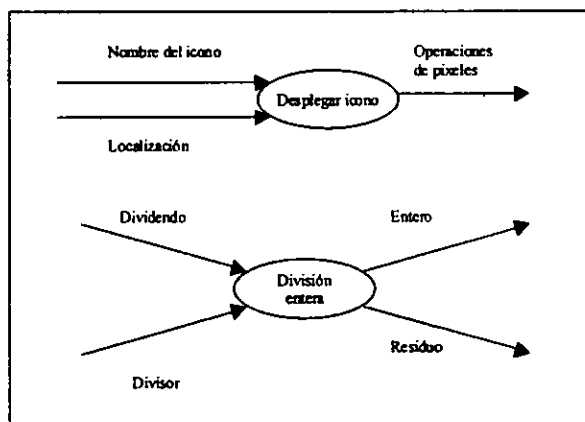


Figura 43 Procesos

dentro del cálculo, pero frecuentemente el tipo de valor del flujo de dato es suficiente. La figura 38 muestra dos procesos. Nótese que un proceso puede tener más de una salida. El proceso de desplegado de un icono representa el diagrama entero del flujo de datos de la figura 37 un nivel de abstracción mayor.

El diagrama solo muestra el patrón de entradas y salidas. El cálculo del valor de una salida desde sus valores de entrada debe ser especificado. Un proceso de alto nivel puede ser expandido en un diagrama entero de flujo de datos, de la misma manera que una subrutina puede ser expandida en subrutinas de un nivel inferior. Al final, lo recursivo deberá detenerse y el proceso atómico deberá ser descrito directamente, en lenguaje natural, ecuaciones matemáticas o por algún otro medio. Frecuentemente los procesos atómicos son triviales y simplemente acceden a un valor de un objeto, por ejemplo.

Los procesos son implementados como métodos (o como fragmentos de métodos) de operaciones en clases de objetos. El objeto destino es usualmente uno de los flujos de datos, especialmente si la misma clase de objetos es también uno de los flujos de salida. Sin embargo, en algunos casos, el objeto destino está implícito. Por ejemplo en la figura 38, el destino de *desplegar icono* es la ventana que recibe las *operaciones de pixeles*.

Flujos de datos

Un flujo de datos conecta una salida de un objeto o proceso a una entrada de otro objeto o proceso. Representa un valor de dato intermedio dentro de un cálculo. Los valores no se cambian por el flujo de datos.

Un flujo de datos se dibuja como una flecha entre el productor y el consumidor del dato. La flecha es etiquetada con una descripción de datos, usualmente por su nombre o tipo. El valor puede ser enviado a diferentes lugares; esto es indicado por una bifurcación con diversas flechas emergiendo de ésta. Las flechas pueden o no tener etiqueta pues representan el mismo valor como una entrada.

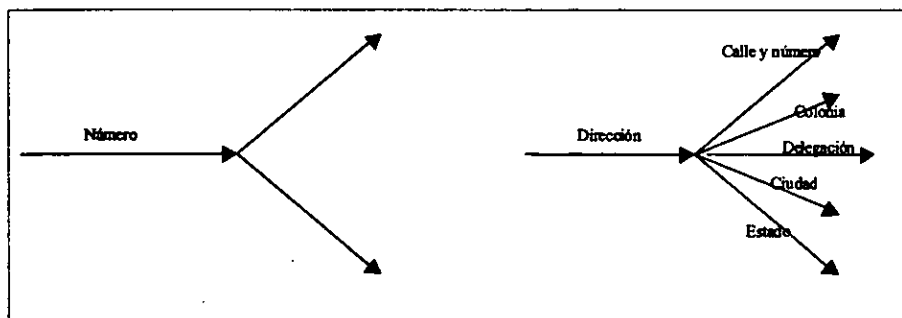


Figura 44 Datos agregados que se copian y datos agregados que se dividen

Algunas veces el valor de un dato agregado es dispersado en sus componentes, cada uno de los cuales van a diferentes procesos. Esto se muestra por una bifurcación en la ruta en la cual cada flecha de salida es etiquetada con el nombre del componente que conlleva. La combinación de diversos componentes en un valor agregado es exactamente su opuesto.

Cada flujo de dato representa un valor en algún punto en el cálculo. Los flujos de datos internos al diagrama representan valores intermedios del cálculo, y no tienen un significado en el mundo real necesariamente.

Los flujos en los límites de un diagrama de flujo de datos son sus entradas y sus salidas. Estos flujos pueden estar desconectados (si el diagrama es un fragmento de un sistema completo).

Actores

Un actor es un objeto activo que maneja el gráfico de un flujo de datos para producir o consumir valores. Los actores son asociados a entradas y salidas del gráfico de flujo de datos. En cierto sentido, los actores radican en los límites de los gráficos de flujos de datos, sirviendo como fuentes o destinos para la información, por lo tanto algunas veces se les denominan *terminales*. Las acciones de los actores están fuera del alcance del diagrama de flujo de datos, pero deben ser partes del modelo dinámico.

Los actores son dibujados con un rectángulo para mostrar que son un objeto. Las flechas entre el actor y el diagrama son entradas y salidas del diagrama. El buffer de la pantalla en la figura 37 es un actor que consume *operaciones de pixel*.

Almacenamiento de datos

Un almacenamiento de datos es un objeto pasivo dentro de un diagrama de flujo de datos que almacena datos para un acceso posterior. A diferencia de un actor, el almacenamiento de datos no genera ninguna operación sobre sí mismo, sino solamente responde a pedidos para almacenar datos y acceso a los mismos. Un almacenamiento de datos permite acceder los valores en diferente orden al que fueron generados. Almacenamiento de datos agregados, tales como listas y tablas, proveen acceso a datos por orden de inserción o mediante llaves indexadas.

Un almacenamiento de datos es dibujado como un par de líneas paralelas conteniendo el nombre del almacenamiento. Las flechas de entrada indican información o operaciones que modifican los datos

almacenados; esto incluye agregar elemento, modificar valores o eliminar elementos. Las flechas de salida indican información accedida desde el almacenamiento de datos; esto incluye acceso de todos los valores o solo algunos de sus componentes del mismo. La estructura actual del objeto debe describirse en el modelo de objetos, junto con una descripción de las operaciones de actualización y acceso permitidas.

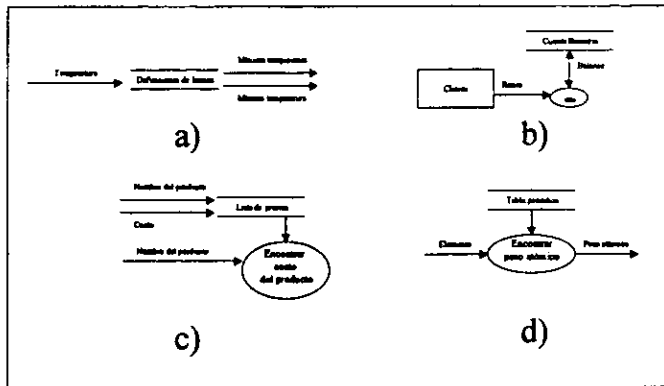


Figura 46 Almacenamiento de datos

La figura 46a muestra a un almacenamiento de datos para lectura de temperatura. La figura 46b muestra el almacenamiento de cuentas bancarias. La figura 46c muestra la lista de precios para productos. La figura 46d muestra la tabla periódica siendo accedida para encontrar el peso atómico de un elemento.

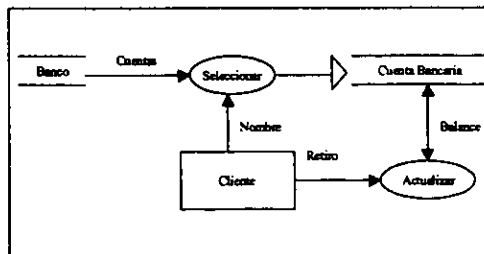


Figura 45 Creación de un objeto en un DFD

Tanto actores como almacenamiento de datos son objetos. Se distinguen puesto que su conducta y su uso son generalmente diferentes, aún cuando en un lenguaje orientado a objetos ambos sean implementados como objetos. Por otro lado, un almacenamiento de datos puede ser implementado como un archivo y un actor como un dispositivo externo. Algunos flujos de datos también son objetos; sin embargo, en muchos casos son valores puros. tales como un entero, con la falta de identidad individual. (En un lenguaje orientado a objetos, sin embargo, objetos y valores puros son frecuentemente implementados de la misma forma)

Existe una diferencia entre ver un objeto como un solo valor y un almacenamiento de datos conteniendo varios valores. En la figura 45, el nombre del cliente selecciona una cuenta de un banco. Un flujo de datos que genera un objeto usado como destino por otra operación se indica por medio de un triángulo vacío al final del flujo de datos. El triángulo vacío indica un valor del flujo de datos que subsecuentemente es tratado como un objeto, usualmente un almacenamiento de datos. (Esta es una nueva construcción que se introduce; La notación tradicional de flujo de datos no representa adecuadamente la creación dinámica o selección de un objeto para un uso posterior en el diagrama como un agregado.) La figura 47 muestra la creación de una nueva cuenta en un banco.

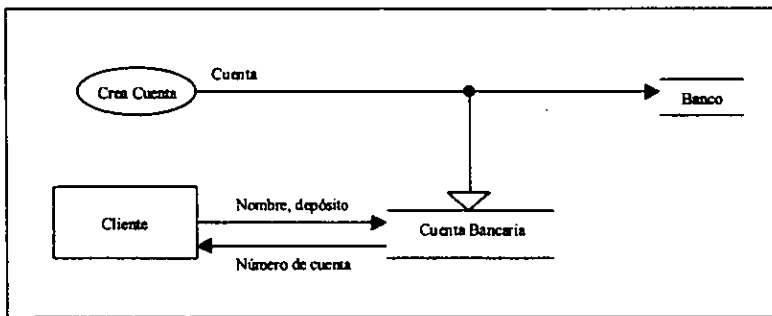


Figura 47 Creación de un nuevo objeto

Diagramas de flujo anidados

Un diagrama de flujo de datos es particularmente útil para mostrar funcionalidad de alto nivel de un sistema y su descomposición en unidades funcionales más pequeñas. Un proceso puede ser expandido en otros diagramas de flujo de datos. Cada entrada y salida del proceso es una entrada o salida del nuevo diagrama, El nuevo diagrama puede tener almacenamiento de datos que no son mostrados en el diagrama de alto nivel. El proceso *Desplegar icono* de la figura 37 corresponde al

diagrama de flujo de la figura 36. Los diagramas pueden estar anidados a un nivel arbitrario, y el conjunto completo de los diagramas anidados forman un árbol. El anidamiento de los diagramas de flujo de datos permite a cada nivel ser coherente y entendible aún cuando la funcionalidad total puede ser arbitrariamente completa. Un diagrama que se refiere a sí mismo representa un cálculo recursivo. (El anidamiento de diagramas también a sido llamado *nivelación*, puesto que los diagramas están organizados en diferentes niveles.)

Eventualmente el anidamiento de diagramas termina con simples funciones. Estas funciones deben ser especificadas como operaciones.

Control de flujos

Un diagrama de flujo de datos muestra todas las rutas posibles de un cálculo por valores; no muestra cuales rutas serán ejecutadas y en que orden. Las decisiones y secuencia son temas de control que forman parte del modelo dinámico. Una decisión afecta ya sea a una o varias funciones que son realizadas, más que entregar un valor a las funciones. Aún cuando las funciones no tienen valores de entrada de las funciones de decisión, es útil algunas veces incluirlas en el modelo funcional para que no se olviden y sus dependencias puedan ser mostradas. Esto se logra incluyendo flujos de control en el diagrama de flujo de datos.

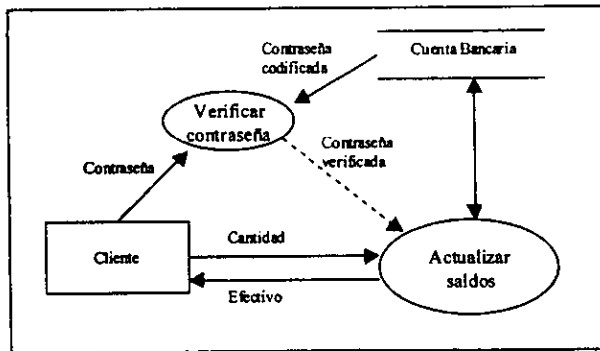


Figura 48 Flujo de control

Un flujo de control es un valor booleano que afecta a un proceso que es evaluado. El flujo de control no es un valor de entrada para el proceso en sí mismo. Un flujo de control se muestra por una línea punteada desde el proceso produciendo el valor booleano al proceso que está controlando.

La figura 48 muestra a un diagrama de flujo de datos para un retiro para una cuenta bancaria. El proceso de actualización puede ser expandido con un flujo de control similar para verificar un estado de sobregiro.

Los flujos de control pueden ser útiles ocasionalmente, pero duplican la información del modelo dinámico y deben ser usados con recelo.

Especificando operaciones

Los procesos en los diagramas de flujos de datos deben ser eventualmente implementados como operaciones sobre objetos. En cada nivel inferior final, cada proceso atómico es una operación. Los procesos de los niveles superiores pueden ser también considerados como operaciones, aún cuando una implementación puede ser organizada en forma diferente del diagrama de flujo de datos que lo representa debido a políticas de optimización. Cada operación puede ser especificada de varias maneras, incluyendo:

- ✓ Funciones matemáticas, tales como funciones trigonométricas
- ✓ Tabla de valores de entrada y salida (enumeración) para conjunto de valores finitos
- ✓ Ecuaciones especificando las salidas en función de las entradas.
- ✓ Pre condiciones y pos condiciones (definiciones axiomáticas)
- ✓ Tablas de decisión
- ✓ Pseudocódigo
- ✓ Lenguaje natural

La especificación de una operación incluye un formato y una transformación. El formato define la interfaz a la operación: los argumentos que requieren (número, orden y tipo de dato) y los valores que regresan (número, orden y tipo de dato). La operación es usualmente listada en el modelo de objetos para mostrar el patrón de herencia; el formato de todos los métodos implementando una operación deben ser iguales. La transformación define el efecto de una operación: los valores de salida como funciones de los datos de entrada y los efectos secundarios de la operación sobre los objetos sobre los que opera.

La especificación externa de una operación sólo describe los cambios visibles afuera de la operación. Durante la implementación de una operación, son creados valores internos por conveniencia o por optimización. Algunos valores pueden ser parte incluso del estado interno de un objeto. Tales detalles internos son privados a una operación (y posiblemente a una clase de objeto) y no aparecen en

su especificación interna. El propósito de la especificación es indicar que debe de hacer una operación lógicamente, no como ha sido implementada. Más aún el estado del arte en sí mismo debe dividir la información externa visible y la información interna privada. Los cambios al estado interno de un objeto que no son visibles externamente no cambian el valor del objeto.

Las operaciones de acceso son operaciones que leen y escriben atributos o ligas del objeto. Es innecesario listar o especificar las operaciones de acceso durante el análisis, puesto que son triviales. Durante el diseño, es necesario notar cuáles operaciones de acceso son públicas y cuáles son privadas a la clase de objeto. La razón para restringir el acceso no es por razones de exactitud lógica, sino por encapsulamiento de clases para la protección contra errores de codificación y permitir modificaciones a la implementación en el futuro. Operaciones de acceso son derivadas directamente de los atributos y asociaciones de una clase en el modelo de objeto.

Operaciones no triviales pueden ser divididas en tres categorías: *consultas*, *acciones* y *actividades*. Una *consulta* es una operación que no tiene efectos secundarios sobre el estado externo visible de ningún objeto; es sólo una función pura. Una consulta sin parámetros (excepto el objeto destino) es un atributo derivado; tiene la forma de un atributo (aunque no necesariamente la implementación). En el modelo de objetos, las operaciones de consulta pueden ser agrupadas con atributos pero su estado de derivación debe de ser indicado para no contribuir a información adicional del estado del objeto. En muchos casos, la elección de ciertos atributos como atributos base y otros como atributos derivados es arbitrario. Debido a que las operaciones de consulta no tienen efectos secundarios, son menos importantes cuando se analiza y se diseña un sistema que los atributos base y las acciones. Frecuentemente son especificadas con ecuaciones escritas en función de otros atributos y no requieren de un componente de control. Las operaciones de consulta son derivadas de las rutas del modelo de objetos o por el reempacamiento de datos en el modelo de objeto.

Una *acción* es una transformación que tiene efectos secundarios sobre el objeto destino o sobre otros objetos en el sistema alcanzables por el objeto destino. Una acción no tiene duración en el tiempo; es instantánea lógicamente (aún cuando cualquier implementación requiera algo de tiempo). Debido a que el estado de un objeto se define como sus atributos y ligas, todas las acciones deben de definirse en términos de actualizaciones a atributos base y ligas. Una acción puede ser definida en términos del estado del sistema antes y después de la acción; un componente de control es innecesario. Las acciones son derivadas usualmente de los procesos del modelo funcional.

Una *actividad* es una operación hacia o de un objeto que tiene una duración en el tiempo, como opuesto a las consultas y acciones, las cuales se consideran instantáneas. Una actividad inherentemente tiene un efecto secundario debido a su extensión en el tiempo. Las actividades solo tienen sentido para actores, objetos que generan operaciones por sí mismas, porque los objetos pasivos son meramente repositorios de datos. Los detalles de una actividad son especificados por el modelo dinámico, así como en el modelo funcional y no pueden ser consideradas solo como una transformación. En la mayoría de los casos, una actividad corresponde a un diagrama de estados en el modelo dinámico.

Restricciones

Una restricción muestra la relación entre dos objetos al mismo tiempo (tales como frecuencia y longitud de onda), o entre dos valores diferentes del mismo objeto en diferentes momentos (como los diferentes retiros de un fondo mutuo). Una restricción puede ser expresada como una función total (un valor es completamente especificado por otro). Por ejemplo, una transformación de coordenadas puede especificar el factor de escala para el eje x y el eje y sea el mismo. Esta restricción define un valor en término de otro. La segunda ley de la termodinámica expresa una restricción parcial, establece que la entropía (desorden) en el universo nunca disminuye.

Las restricciones pueden aparecer en cada tipo de modelo. Las restricciones de objetos especificando que algunos objetos dependen enteramente o parcialmente de otros. Las restricciones en el modelo dinámico especifican las relaciones entre los estado o eventos de diferentes objetos. Las restricciones en el modelo funcional especifican restricciones sobre las operaciones, tales como el escalamiento descrito líneas antes.

Una restricción entre valores de un objeto en el tiempo es frecuentemente llamado *invariable*. Las invariables son útiles en la especificación de la conducta de operaciones.

Función	Conexión de llamada
Entradas	Línea telefónica, número marcado, estado actual del conmutador
Salidas	Nuevo estado del conmutador, status de la conexión
Transformación	Conectar el teléfono origen al teléfono solicitado mediante el cerrado de un interruptor, tomando en cuenta las restricciones
Restricciones	Sólo dos líneas son conectadas a un tiempo por cualquier circuito. Conexiones previas no son molestadas. Si la línea solicitada está ocupada, ningún interruptor es cerrado y el status es reportado como línea ocupada. Si una conexión es imposible porque existen demasiados interruptores en uso, ningún interruptor es cerrado y el status es reportado como interruptor ocupado.

Figura 44 Acción de una conexión de un conmutador

Relación de los modelos de objetos, dinámico y funcional

El modelo funcional muestra lo que tiene que ser hecho en el sistema. Los procesos terminales son las operaciones en los objetos. El modelo de objetos muestra los realizadores – los objetos. Cada proceso es implementado por un método sobre algunos objetos. El modelo dinámico muestra la secuencia en la cual las operaciones serán realizadas. Cada secuencia es implementada como una secuencia, ciclo o serie de instrucciones dentro de algún método. Los tres métodos se usan juntos en la implementación de los métodos. El modelo funcional es una guía a los métodos.

Los procesos en el modelo funcional corresponden a las operaciones en el modelo de objetos. Frecuentemente existe una correspondencia directa a cada nivel de anidamiento. Un proceso de alto nivel corresponde a una operación en un objeto complejo, y los procesos de niveles inferiores corresponden sobre objetos más básicos que son parte del objeto más complejo o de ese que implementan. Algunos procesos corresponden a diversas operaciones, y algunas veces una operación corresponde a diversos procesos.

Los procesos en el modelo funcional muestran los objetos que están relacionados por su función. Frecuentemente una de sus entradas a un proceso puede ser identificada como el objeto destino, mientras el resto son parámetros de la operación. El objeto es un cliente de otros objetos (llamados proveedores) puesto que lo usan en la realización de operaciones. El destino conoce a los clientes, pero los clientes no necesariamente conocen al proveedor. La clase de objeto cliente es dependiente en los argumentos de la clase para sus operaciones. La relación cliente - proveedor establece dependencias de implementación entre clases; los clientes son implementados en términos de las clases proveedoras y por lo tanto son dependientes de éstas.

Un proceso es usualmente implementado como un método. Si la misma clases del objeto es la entrada y la salida, entonces el objeto es usualmente el destino, y las otras entradas son argumentos. Si la salida del proceso es un almacenamiento de datos, el almacenamiento de datos es el destino. Si la entrada al proceso es el almacenamiento de datos, el almacenamiento de datos es el destino. Frecuentemente un proceso con una entrada o una salida desde o hacia un almacenamiento de datos le corresponden dos métodos, uno de ellos siendo la selección implícita o actualización del almacenamiento de datos. Si una entrada o salida es un actor, entonces este es el destino. Si una entrada es un objeto y una salida es parte del objeto o está en las cercanías del objeto en el modelo de objetos, entonces el objeto es el destino. Si una salida es creada sin uso de las partes de entrada, entonces el

proceso representa un método de clase. Si ninguna de estas reglas aplica, entonces el destino frecuentemente está implícito y no es una de las entradas o salidas. Frecuentemente el destino del proceso es el destino del subdiagrama entero.

Los actores son objetos explícitos en el modelo de objetos. Flujos de datos desde o hacia los actores representan operaciones sobre o por los objetos. Los valores de los flujos de datos son argumentos o resultados de las operaciones. Debido a que los actores son objetos auto – motivados, el modelo funcional no es suficiente para indicar cuando actúan. El modelo dinámico para un objeto actor especifica cuando actúa.

Los almacenamientos de datos son objetos también en el modelo de objeto, o por lo menos un fragmento de un objeto.

Análisis y diseño de bases de datos

Bases de datos relacionales

Así como existieron diversas eras en la evolución de las computadoras, en el área de base de datos se realizaron constantes mejoras conforme fueron avanzando las tecnologías en hardware y software. El propósito de los sistemas de bases de datos es la gestión de grandes cantidades de información. Las primeras bases de datos surgieron del desarrollo de los sistemas de gestión de archivos. Estos archivos primero evolucionaron en bases de datos de red o en bases de datos jerárquicas y posteriormente en bases de datos relacionales.

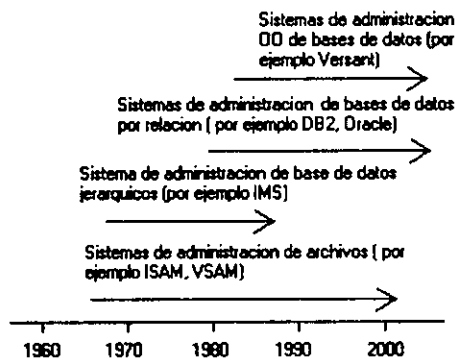


Figura 49 Evolución de las bases de datos

Actualmente el modelo de datos por relación, o bases de datos relacionales son el enfoque más utilizado en el mercado y está evolucionando hacia las bases de datos orientadas a objetos. Las bases de datos relacionales tienen las siguientes características:

- ✓ **Uniformidad.** Existe un gran número de datos estructurados de manera similar, todos del mismo tamaño.
- ✓ **Orientación de registros.** Los datos básicos constan de registros de longitud fija.
- ✓ **Datos pequeños.** Todos los registros son cortos. A menudo los registros tienen alrededor de 80 bytes, reflejando el tamaño de tarjetas perforadas como lo era en la década de los sesenta.
- ✓ **Campos atómicos.** Los campos de un registro son cortos y de longitud fija. No existe una estructura en los campos.

✓ **Transacciones cortas.** Las transacciones son programas con un tiempo de ejecución medido en fracciones de segundo. No hay una interacción humana con la transacción durante su ejecución. En vez de ello el usuario prepara una transacción, la somete a ejecución y espera la respuesta.

✓ **Esquemas de concepto estáticos.** El esquema de la base de datos se cambia con poca frecuencia. Cuando se cambia, los tipos de cambio permitidos son sencillos. En un sistema de relacional las únicas modificaciones de esquema permitidas normalmente son crear relaciones, eliminar relaciones, añadir atributos en un esquema de relaciones.

El modelo de datos por relación define 3 tipos de datos: la tabla (relación), la fila (registro) y la columna (atributo). El modelo especifica tres operadores de tablas: seleccionar proyectar y unir. *Seleccionar* aplica un operador booleano a cada hilera de una tabla y regresa una nueva tabla, la cual contiene solo aquellas hileras para los que el operador booleano regrese el valor de verdadero. *Proyectar*, especifica un subconjunto de las columnas definidas en una tabla. Regresa una nueva tabla que contiene todas las hileras originales con solo las columnas especificadas por la proyección. El operador *unir* combina dos tablas para producir una tabla única. Esta tabla representa el producto cartesiano de todas las tablas que confirman la unión. Así un operador booleano compara una columna de cada tabla original, y el resultado de la unión es una tabla que contiene aquellas hileras para las que el operador booleano regresa el valor de verdadero.

El modelo de datos relacionales ofrece varias ventajas significativas:

✓ **Independencia de datos.** La representación de los datos en la computadora es independiente de la interfaz con la aplicación.

✓ **Manejo declarativo.** SQL es el lenguaje más común utilizado para expresar el modelo relacional. SQL es un lenguaje declarativo que expresa el tipo de datos deseados. Los consultas o queries son mecanismos sencillos para el manejo de la base de datos. SQL expresa la filosofía de la independencia de datos. Permite al sistema de base de datos que elija entre mecanismos alternos y que obtenga los resultados deseados a partir de la realización física de la base de datos. El sistema para la administración de la base de datos (RDBMS) puede optimizar en forma dinámica la forma en que se ejecutan las consultas, lo que libera al programador de la aplicación de esta tarea.

✓ **Eliminación de la redundancia.** Al diseñar los datos por relación, se puede aplicar el proceso de normalización. La normalización total del modelo de datos produce una base de datos en la cual se ha eliminado la redundancia, eliminando la posibilidad de que partes de la base de datos se desincronicen por el almacenamiento repetitivo, y se minimiza la cantidad de datos almacenados, reduciendo el tamaño global de la base de datos y ahorra espacio en el disco.

✓ **Sencillez.** El modelo relacional tiene tres tipos básicos y tres operaciones, una mejora significativa sobre los lenguajes de programación. Fuesen programadores o no, la mayoría de las personas ya están familiarizadas con los conceptos básicos, puesto que han trabajado con tablas, renglones y columnas. El modelo relacional es más fácil de aprender y utilizar.

✓ **Tablas como vehículos de presentación.** El resultado de todos los operadores por relación es una tabla, la cual a menudo es la solución necesaria de la aplicación. Los modelos de datos anteriores requerían la extracción de la información y su traducción posterior a cierto formato de resultado, que por lo general era una tabla. La RDB eliminan ese último paso puesto que el resultado ya se encuentra en forma de tabla.

Todas las características anteriores permiten un desarrollo de aplicaciones más rápido y un mantenimiento de las aplicaciones más sencillo: a ello se debe la popularidad actual de las RDBMS. Otra razón más la constituyen los modernos ambientes de desarrollo de las aplicaciones originados por las RDBMS, sobre todo para los lenguajes de cuarta generación. Los lenguajes de cuarta generación se caracterizan por un lenguaje con operadores de una base de datos ya integrados y tal vez el formato de informe. Esto simplifica una vez más el proceso de desarrollo de aplicaciones y reduce el tiempo necesario para terminar una nueva aplicación.

Una base de datos consiste en un conjunto estructurado de datos registrados sobre soportes a los que una computadora puede acceder. Los sistemas de base de datos deben mantener la seguridad de la información almacenada pese a caídas o intentos de accesos no autorizados.

Bases de datos orientada a objetos.

Las bases de datos orientada a objetos se basa en la filosofía de la orientación a objetos. Actualmente la tecnología de base de datos se ha adaptado a aplicaciones fuera del ámbito del procesamiento de datos. Las nuevas aplicaciones incluyen entre otras cosas:

- ✓ **Diseño asistido por computadora (CAD).** Una base de datos CAD almacena datos pertenecientes a un diseño de ingeniería incluyendo las componentes del dato que se está diseñando, la relación de sus componentes y las versiones antiguas de los diseños.
- ✓ **Ingeniería de software asistido por computadora (CASE).** Una base de datos CASE almacena los datos requeridos para asistir a los que desarrollan el software. Estos datos incluyen código fuente, dependencias entre los módulos, definiciones y usos de variables y la historia del desarrollo del sistema de software.
- ✓ **Base de datos multimedia.** Una base de datos multimedia contiene datos variados, de audio, video y demás las bases de datos de este tipo derivan de datos geofísicos, sistemas de correo por voz y aplicaciones gráficas.
- ✓ **Sistemas de información de oficina(OIS).** Las OIS incluyen herramientas basadas en estaciones de trabajo para creación y recuperación de documentos, así como herramientas para mantener calendarios y citas, etc.
- ✓ **Sistemas expertos de bases de datos.** Un sistema experto de base de datos, además de datos incluye reglas explícitas que representan las restricciones de integridad, disparadores y otros conocimientos acerca de la empresa que está modelando la base de datos.

Las nuevas aplicaciones de las bases de datos no se consideraron anteriormente cuando se diseñaron la mayor parte de sistemas de base de datos relacionales actuales. Las nuevas aplicaciones requieren nuevos modelos de datos, nuevos lenguajes de consulta y nuevos modelos de transacciones, entre los cuales estan:

- ✓ **Objetos complejos.** Un objeto complejo es un dato que es visto como un simple objeto en el mundo real, pero que contiene estructura interna compleja arbitraria. A menudo los objetos están estructurados jerárquicamente, representando la relación entre ellos. El modelo de objetos complejos ha llevado al desarrollo de las bases de datos orientadas a objetos, las cuales están basadas en los conceptos orientados a objetos y a las bases de datos relacionales anidadas, en las que las relaciones pueden almacenarse dentro de otras.
- ✓ **Datos de comportamiento.** Puede que distintos objetos necesiten responder de diferentes formas a la misma orden. Por ejemplo la eliminación de ciertas tuplas requiere eliminar otras tuplas, como el caso de entidades débiles. Los métodos de los sistemas de base de datos orientados a objetos y la regla base de los sistemas basados en comportamientos proporcionan la capacidad de almacenar el comportamiento en código ejecutable.
- ✓ **Meta conocimiento.** A menudo los datos más importantes sobre aplicaciones son reglas generales acerca de la aplicación más que de las tuplas específicas. Las reglas forman una parte importante de los sistemas expertos de bases de datos.
- ✓ **Transacciones de larga duración.** Las aplicaciones CAD y CASE implican interacción humana con los datos. Los esfuerzos de diseño concurrente que implican varios diseñadores pueden conducir a conflictos entre transacciones. Debido a que estas transacciones implican interacción humana con el sistema, las consecuencias de abortos de transacciones no interactivas cortas comúnmente usadas en aplicaciones tipo negocio.

Con estos nuevos requerimientos de las bases de datos, se realizan actualmente esfuerzos por la comunidad de bases de datos de ofrecer OODBMS, sistema de administración de base de datos orientadas a objetos que cumplan con las nuevas expectativas.

Bases de datos distribuidas.

En un sistema de bases de datos centralizados, todos los componentes del sistema residen en una sola computadora. Los componentes consisten en los datos, el software del sistema de gestión de bases de datos (DBMS) y los dispositivos de almacenamiento secundario asociados, como disco para el almacenamiento en línea de la base de datos y cintas para las copias de seguridad. Podemos tener acceso remoto a una base de datos centralizada a través de terminales conectadas al nodo; sin embargo, los datos del software del DBMS, residen principalmente en un solo nodo. En los últimos años se ha observado una marcada tendencia hacia la distribución de los sistemas de cómputo, en múltiples nodos que se interconectan a través de una red de comunicaciones. El software con que se implementa un sistema así se denomina Sistema de Gestión de Base de Datos Distribuidas (DDBMS).

Introducción a los conceptos de DDBMS distribuidos

Una base de datos distribuida es una colección de datos que pertenece lógicamente al mismo sistema pero que está dispersa físicamente entre los nodos de una red de computadoras. Varios factores han dado pie a la creación de los DDBMS. Entre las ventajas potenciales DDBMS están las siguientes:

- ✓ **La naturaleza distribuida de algunas aplicaciones de bases de datos.** Muchas de estas aplicaciones están distribuidas naturalmente en varios lugares. Por ejemplo, una compañía puede tener oficinas en varias ciudades, o un banco pueden tener múltiples sucursales. Es natural que las bases de datos empleadas estén distribuidas en esos lugares. Muchos usuarios locales tienen acceso exclusivamente a los datos que están en el lugar. Pero otros usuarios globales – como la oficina central de la compañía – pueden requerir acceso ocasional a los datos almacenados en varios de los lugares.
- ✓ **Mayor fiabilidad y disponibilidad.** Estas son dos de las ventajas potenciales de las bases de datos distribuidas que se citan más comúnmente. La fiabilidad se define a grandes rasgos como la probabilidad de que un sistema esté en función en un momento determinado, y la disponibilidad es la probabilidad de que el sistema esté disponible continuamente durante un intervalo de tiempo. Cuando los datos y el software del DDBMS están distribuidos en varios nodos, un nodo puede fallar mientras que los demás siguen operando. Sólo los datos y el software que residen en el nodo que falló están accesibles. Esto mejora tanto la fiabilidad como la disponibilidad. Se logran mejoras adicionales si se replican, con un criterio adecuado los datos y el software en más de un nodo. En un sistema centralizado, el fallo de un solo nodo hace que el sistema completo deje de estar disponible para todos los usuarios.
- ✓ **Posibilidad de compartir los datos al tiempo que se mantiene un cierto grado de control local.** En algunos tipos de DDBMS, es posible controlar los datos y el software localmente en cada nodo. Sin embargo los usuarios de los otros nodos remotos pueden tener acceso a ciertos datos a través del software del DDBMS. Esto hace posible el compartimiento controlado de los datos en todo el sistema distribuido.

✓ **Mejor rendimiento:** Cuando una base de datos grande está distribuida en múltiples nodos, hay bases de datos más pequeñas en cada uno de éstos. En consecuencia, las consultas locales y las transacciones que tienen acceso a datos de un solo nodo tiene un mejor rendimiento por que las bases de datos locales son más pequeñas. Además, cada nodo tiene un menor número de transacciones en ejecución que si todas las transacciones se enviaran a una sola base de datos centralizada. En el caso de transacciones que impliquen acceso a más de un nodo, el procesamiento en los diferentes nodos puede efectuarse en paralelo, reduciéndose así el tiempo de respuesta.

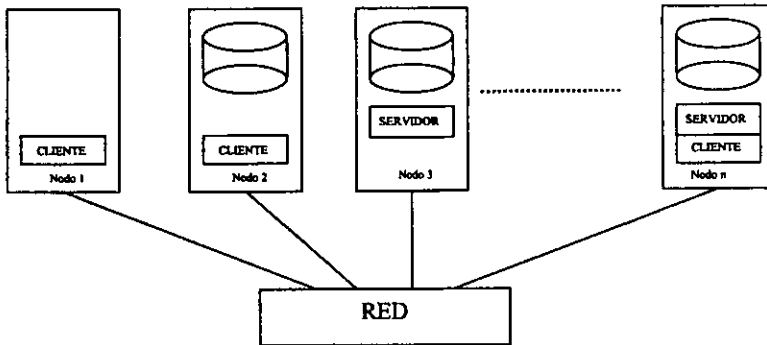


Figura 50 Arquitectura física cliente – servidor simplificada para un SBDD

La distribución produce un aumento en la complejidad de diseño y en la implementación del sistema. Para obtener las ventajas potenciales el software de DDBMS debe contar con las funciones de un DBMS centralizado y además con las siguientes:

- ✓ La capacidad de tener acceso a nodos remotos y transmitir consulta de datos entre los diversos nodos a través de una red de comunicaciones.
- ✓ La capacidad de seguir la pista a la distribución y la replicación de los datos en el catalogo del DDBMS.
- ✓ La capacidad de elaborar estrategias de ejecución para consultas y transacciones que tienen acceso a los datos de más de un nodo.
- ✓ La capacidad de decidir a cual copia de un elemento de información replicado se tendrá acceso.
- ✓ La capacidad de mantener la consistencia de copias de un elemento de información replicado.
- ✓ La capacidad de recuperarse de caídas de nodos individuales y de nuevos tipo de fallo como fallo de un enlace de comunicación.

Por si solas estas funciones elevan la complejidad de un DDBMS en comparación de un DDBMS centralizado. Cuando consideramos el diseño de una base de datos distribuida surgen complejidades adicionales, se debe decidir como distribuir los datos entre los nodos y cuales datos replicar.

En el nivel físico del hardware los principales factores que distinguen un DDBMS de un sistema centralizado son los siguientes:

- ✓ Hay múltiples computadoras llamadas sitios o nodos.
- ✓ Estos sitios deben estar comunicados por medio de algún tipo de red de comunicaciones para transmitir datos y órdenes entre los nodos como en la figura 50.

Los nodos pueden estar cercanos físicamente, digamos dentro del mismo edificio o grupo de edificios adyacentes, y conectados a través de una red de área local, o pueden estar distribuidos geográficamente a grandes distancias y conectados a través de una red de larga distancia. Las redes de área local por lo regular usan cables en tanto que las redes de larga distancia emplea líneas telefónicas o satélites. También es posible usar una combinación de ambos tipos de redes.

Las redes pueden tener diferentes topologías que definen los caminos de comunicación directa entre los nodos. El tipo y la topología de la red empleada pueden tener un efecto significativo sobre rendimiento y las estrategias para el procedimiento de consultas distribuidas y el diseño de las mismas. Sin embargo en los aspectos arquitectónicos de alto nivel, no importa que tipo de red se use sólo importa que cada uno de los nodos pueda comunicarse directa o indirectamente con todos los demás.

Arquitectura Cliente – Servidor para DBMS

La arquitectura cliente servidor se creó para manejar los nuevos entornos de cómputo en los que un gran número de computadoras personales, estaciones de trabajo, servidores de archivos, impresoras y otros equipos están interconectados a través de una red. La idea es definir servidores especializados con funciones específicas por ejemplo es posible conectar como clientes varias estaciones de trabajo o computadoras personales sin disco a una máquina servidora de archivos que mantenga los archivos de los usuarios de los clientes. Se podría designar otra máquina como servidora de impresoras conectándola a diversas impresoras, de aquí todas las solicitudes de impresión se enviarán a esta máquina. De esta manera los recursos que proveen los servidores especializados están a disposición de muchos clientes. Esta idea puede aplicarse al software al almacenar programas especializados como un DDBMS o un paquete CAD en máquinas servidoras específicas al fin de ponerlo a disposición de múltiples clientes.

La arquitectura cliente servidor se están incorporando cada vez más en los paquetes de DDBMS comerciales conforme se van orientando hacia la distribución. La idea consiste en dividir el software de DDBMS en dos niveles, cliente y servidor, para reducir su complejidad.

Todavía no se ha establecido una forma precisa de dividir la funcionalidad del DDBMS entre cliente y servidor. Una posibilidad consiste en incluir la funcionalidad de un DDBMS centralizado en el nivel del servidor. Varios productos de DDBMS relacionales han adoptado este enfoque, en el que se proporciona un servidor SQL a los clientes. Cada cliente debe entonces formular las consultas SQL apropiadas y proveer la interfaz con el usuario y las funciones de interfaz con los lenguajes de programación. Puesto que SQL es una norma relacional, diversos servidores SQL diferentes posiblemente proveniente de distintos proveedores pueden aceptar estas ordenes. El cliente también puede hacer referencia a un diccionario de datos que tenga información sobre la distribución de los datos entre diferentes servidores SQL, así como módulos para descomponer una consulta global en varias consultas locales que se pueden ejecutar en los diferentes nodos. Interacción entre el cliente y el servidor podría efectuarse como sigue durante el procesamiento de una consulta:

- ✓ El cliente analiza sintácticamente la consulta del usuario y la descompone en varias consultas de nodos independientes cada consulta de una computadora se envía al servidor apropiado.
- ✓ Cada servidor procesa la consulta local y envía la relación al cliente.
- ✓ El cliente combina los resultados de las subconsultas para producir el resultado de la consulta original.

En este enfoque el servidor SQL recibe también el nombre de procesador de base de datos (DP database processor) o máquina dorsal, en tanto que el cliente se denomina procesador de aplicaciones (AP application processor) o máquina frontal. El usuario puede especificar la interacción entre cliente y servidor en el nivel de cliente o través de un módulo cliente especializado, por ejemplo el usuario podría saber que datos están almacenados en cada servidor, descomponer manualmente una consulta en subconsultas de clientes y presentar subconsultas individuales a los diferentes clientes. Las tablas resultantes podrían combinarse explícitamente mediante otra consulta del usuario en el nivel del cliente. La alternativa es hacer que el módulo cliente lleve a cabo estas acciones automáticamente.

Otro enfoque adoptado por algunos DDBMS divide los módulos de software del DDBMS entre cliente y servidor de una manera más integrada, por ejemplo, el nivel servidor podría incluir la parte del software del DDBMS encargada de manejar el almacenamiento de los datos en páginas de disco, el control de concurrencia junto con la recuperación local, el almacenamiento intermedio y en caché de

disco y otras funciones similares. Por su lado en nivel cliente podría manejar la interfaz con el usuario, las funciones de diccionarios de datos, la interacción de DDBMS con los compiladores de lenguajes de programación, la optimización y control de concurrencia o recuperación de consultas globales, la estructura de objetos complejos a partir de los datos en almacenamiento intermedio y otras funciones similares.

- ✓ En un DDBMS representativo, se acostumbra dividir los módulos de software en tres niveles:
- ✓ El software servidor se encarga de la gestión local de los datos en un nodo, en forma muy parecida al software DDBMS centralizado.
- ✓ El software cliente se encarga de todas funciones de distribución, tiene acceso a información almacenada en el DDBMS sobre la distribución de los datos y procesa todas las solicitudes que requieren acceso a más de un nodo.
- ✓ El software de comunicaciones, a veces en colaboración con un sistema operativo distribuido proporciona la base de comunicación con que el cliente transmite ordenes y datos entre los diferentes nodos según las necesidades

El cliente se encarga de generar un plan de ejecución distribuido para una consulta o transacción en múltiples nodos y de supervisar la ejecución distribuida enviando órdenes a los servidores, estas órdenes son las consultas y transacciones locales que han de ejecutarse, así como órdenes para transmitir datos a otros clientes o servidores, todos los nodos en los que se introduzcan consultas a múltiples nodos deberán contar con el software de cliente. Otra función que esta bajo el control del cliente es la de asegurar que las replicas de un elemento de información sean consistentes para lo cual se vale de técnicas de control de concurrencias distribuidas. El cliente también debe asegurar que las transacciones globales sean atómicas realizando acciones de recuperación global cuando ciertos nodos fallen.

Una posible función del cliente es ocultar al usuario los detalles de la distribución de los datos, permite al usuario escribir consultas y transacciones globales como si la base de datos fuera centralizada sin tener que especificar los nodos en donde residen los datos a los que se hace referencia en la consulta; esta propiedad se llama transparencia de distribución. Como algunos DDBMS no cuentan con transparencia de distribución exige que los usuarios conozcan los detalles de la distribución de los datos. Cuando hay transparencia de distribución se presenta al usuario un enfoque que no incluye información de distribución y el mismo DDBMS registra los nodos donde están ubicados los datos en el catálogo de distribución. Con ayuda de esta información el cliente puede descomponer una consulta en varias subconsultas que se pueden ejecutar en diversos nodos y puede

planear la forma de transmitir los resultados de las subconsultas a otros nodos para un procesamiento adicional.

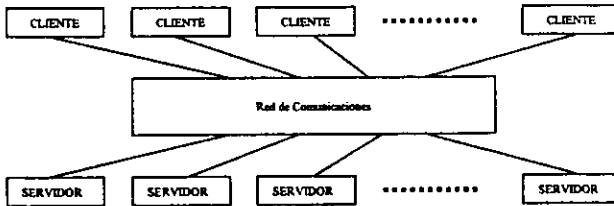


Figura 51 Arquitectura lógica cliente – servidor simplificada para un SBDD.

Fragmentación, replicación y reparto de los datos para el diseño de bases de datos distribuidas

Con la replicación de los datos es posible almacenar ciertos datos y el proceso de repartir fragmentos para almacenar en diferentes nodos. La información concerniente a la fragmentación de los datos reparto y replicación se almacena en un catálogo global del sistema al que tiene acceso el cliente cuando es necesario.

Fragmentación de datos

En un DDBMS se tiene que tomar la decisión respecto a los nodos en los que se almacenarán las porciones de la base de datos. Se partirá de un esquema de base de datos relacional en el que hacemos referencia al esquema.

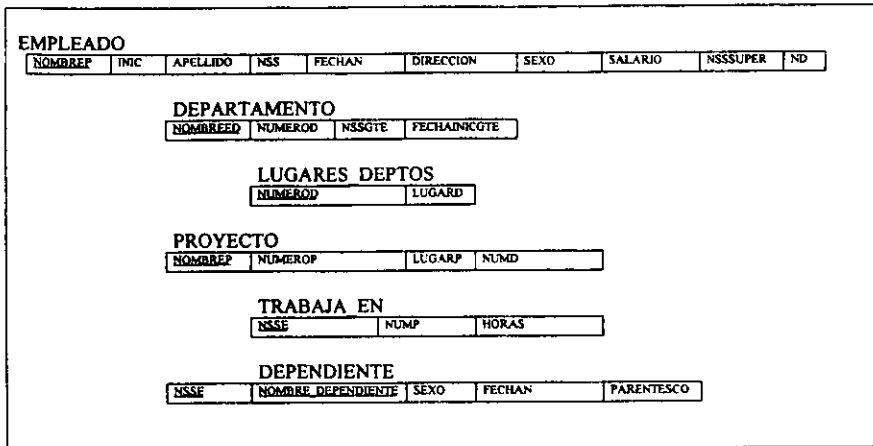


Figura 52 Esquema de base de datos relacional. (Claves primarias subrayadas)

Las unidades lógicas más simples son las relaciones mismas es decir cada relación se almacenará en un nodo específico. En el ejemplo se decide en que nodo se almacenará cada una de las relaciones EMPLEADO, DEPARTAMENTO, TRABAJA_EN y DEPENDIENTE. Es posible dividir una relación en unidades lógicas más pequeñas para su distribución. Por ejemplo consideremos la base de datos de una compañía donde hay tres nodos, uno para cada departamento de una compañía donde se almacenará información referente a cada departamento en el nodo referente al departamento para ello se divide la relación mediante una técnica llamada fragmentación horizontal.

✓ Fragmentación horizontal.- Un fragmento horizontal de una relación es un subconjunto de las tuplas de esa relación. Las tuplas que pertenecen al fragmento horizontal se especifican mediante una condición sobre uno más de los atributos de la relación. Con frecuencia sólo interviene un atributo, por ejemplo, se podría definir tres segmentos horizontales en la relación EMPLEADO con las siguientes condiciones: (ND=5), (ND=4) y (ND=1), cada fragmento contiene tuplas de EMPLEADO que pertenecen a un departamento en particular.

✓ Fragmentación vertical.- Un fragmento vertical de una relación mantiene sólo ciertos atributos de la relación. Por ejemplo se podría dividir la relación EMPLEADO en dos fragmentos verticales, el primero incluiría información personal (NOMBRE, FECHA, DIRECCIÓN Y SEXO) y el segundo información relacionada con el trabajo (NSS, SALARIO, NSSUPER, ND). Este fragmento vertical no es muy apropiado ya que si ambos fragmentos se almacenan por separado no se podrán juntar las tuplas originales ya que no existe un atributo común entre los dos fragmentos. Es necesario incluir el atributo de clave primaria en todo fragmento vertical para que sea posible reconstruir la relación completa a partir de los fragmentos. Por tanto habrá que añadir el atributo NSS al fragmento de información personal. La fragmentación vertical divide la relación por columnas.

Replicación y reparto de los datos.

La replicación resulta útil para mejorar la disponibilidad de los datos el caso más extremo es la replicación de toda la base de datos en todos los nodos de sistemas distribuidos creando así una base de datos distribuida totalmente replicada. Esto puede mejorar la disponibilidad notablemente por que el sistema puede seguir operando mientras por lo menos unos de los nodos este activo. También mejora el rendimiento de la obtención de datos en consultas globales, por que el resultado se puede obtener en cualquier nodo. La desventaja de la replicación completa es que puede reducir drásticamente la rapidez de las operaciones de actualización, pues una sola actualización lógica se deberá ejecutar con todas y cada una de las copias de la base de datos para mantener la consistencia. Con la replicación completa las técnicas de control de concurrencia y recuperación se tornan más costosa de lo que serían sin replicación.

El extremo opuesto a la replicación completa es no tener ninguna replicación, esto es cada fragmento se almacena exactamente en un nodo. En este caso todo los fragmentos deben ser disjuntos con excepción de la repetición de claves primarias entre los fragmentos verticales o mixtos.

Entre los dos extremos se tiene una amplia gama de replicación parcial de los datos, es decir, algunos fragmentos de la base de datos pueden estar replicados y otros no. El número de copias de cada fragmento puede ir desde una hasta el número total de nodos en el sistema distribuido.

Cada fragmento se debe asignar a un nodo determinado en el sistema distribuido. Este proceso se denomina distribución de los datos; la elección de nodos y el grado de replicación de los objetivos de rendimiento y disponibilidad para el sistema y de los tipos y frecuencias de transacciones introducidas en cada nodo. Por ejemplo si se requiere una alta disponibilidad y si las transacciones se pueden introducir en cualquier nodo y si la mayoría de ellas son de la obtención de datos entonces una base de datos completamente replicada sería una buena opción. Sin embargo si por lo regular ciertas

transacciones que tienen acceso a partes específicas de la base de datos se introducen en un solo nodo, se podría asignar el conjunto de fragmentos correspondientes exclusivamente a ese nodo. Los datos que se utilizan en múltiples nodos se pueden replicar en estos nodos. Si se efectúan muchas actualizaciones es conveniente limitar la replicación.

Tipos de sistemas de bases de datos distribuidas

El término Sistema de Gestión de Bases de Datos Distribuidas (DDBMS) describe diversos sistemas que presentan muchas diferencias entre sí, donde tienen en común el hecho en que los datos y el software están distribuidos entre múltiples nodos conectados por alguna especie de red de comunicaciones. El primer factor que se considera es el grado de homogeneidad del software de DDBMS. Si todos los servidores, o DDBMS locales, utilizan software idéntico y todos los clientes también se dice que el DDBMS es homogéneo; en caso contrario se le caracteriza como heterogéneo. Otro factor relacionado con el grado de homogeneidad es el grado de autonomía local. Si todo acceso al DDBMS debe hacerse a través de un cliente, el sistema no tiene autonomía local.

En un extremo de la autonomía tenemos un DDBMS que da al usuario la impresión de ser un DDBMS centralizado. Solo hay un esquema conceptual, y todo acceso al sistema se hace a través de un cliente, de modo que no hay autonomía local. En el otro extremo existe un tipo de DDBMS denominado DDBMS federado, o también sistema de multibases de datos. En sistema así, cada servidor es un DDBMS centralizado independiente y autónomo que tiene sus propios usuarios locales, transacciones locales y DBA, y por ende posee un alto grado de autonomía local. Cada servidor puede autorizar el acceso a porciones específicas de su bases de datos definiendo un esquema de exportación, el cual especifica la parte de la base de datos a la cual puede tener acceso una cierta clase de usuarios no locales. En esencia un cliente en un sistema así es una interfaz adicional para varios servidores, DDBMS locales que permite al usuario de multibases de datos, o global, tener acceso a información almacenada en varias bases de datos autónomas. Un sistema federado es un híbrido entre los sistemas distribuidos y los centralizados; es un sistema centralizado para los usuarios autónomos locales y es un sistema distribuido para los usuarios globales.

Otro aspecto para clasificar las bases de datos distribuidas es el grado de transparencia de la distribución, o sea el grado de integración de los esquemas. Si el usuario percibe un solo esquema integrado sin información alguna relativa a la fragmentación, replicación o distribución, se dice que el DDBMS tiene un alto grado de transparencia de distribución, o integración de sistemas. Por otro lado si

el usuario puede ver toda la fragmentación, reparto y replicación, el DDBMS no tiene transparencia de distribución ni integración de esquemas. En este caso el usuario debe referencia a copias específicas de fragmentos en sitios específicos cuando formule una consulta, anexando el nombre del nodo como prefijo de cada nombre o de fragmento.

Costo por transferencia de datos en consultas distribuidas

En un sistema distribuido varios factores adicionales complican aún más el procesamiento de consultas. El primero es el costo de transferir datos por la red. Estos datos incluyen los archivos intermedios que se transfieren a otros sitios para continuar su procesamiento, así como los archivos de resultado final que tal vez deban transferirse al sitio donde se necesita el resultado de la consulta. Aunque es posible que estos costos no sean demasiado altos si los sitios están conectados en una red de área local de alto rendimiento, adquieren una importancia considerable en otros tipos de redes. Por ello los algoritmos de optimización de consulta de los DDBMS consideran el objetivo de reducir la cantidad de transferencia de datos como criterio de optimización al elegir una estrategia de ejecución de una consulta distribuida.

Control de concurrencia y recuperación en base de datos distribuida

Respecto al control de la concurrencia y recuperación en un entorno DBMS distribuido existen numerosos problemas que no se encuentran en los entornos de DBMS centralizados como son:

- ✓ **Manejar múltiples copias de los elementos de información.**- El control de concurrencia tiene la obligación de mantener la consistencia entre estas copias; recuperación debe cuidar que una copia sea consistente con todas las demás si el nodo en que la copia estaba almacenada falla y se recupera posteriormente.
- ✓ **Fallo de nodos individuales.**- El DDBMS debe continuar operando con sus nodos activos, si es posible cuando fallen uno u más nodos individuales. Cuando un nodo se recupere, su base datos local se deberá poner al día con los demás nodos antes de incorporarse al sistema.
- ✓ **Fallo de enlaces de comunicación.**- El sistema debe ser capaz de manejar el fallo de uno o más de los enlaces de comunicación en los nodos. Un caso extremo de este problema es que puede haber partición de la red. Esto divide los nodos en dos o más particiones dentro de las cuales los nodos pueden comunicarse entre sí pero no con nodos de otras particiones.
- ✓ **Confirmación distribuida.**- Puede haber problemas para confirmar una transacción que esta teniendo acceso a bases de datos almacenadas en múltiples nodos si algunos de éstos fallan en el proceso de confirmación. Regularmente se utiliza el protocolo de confirmación de dos fases para resolver este problema.

✓ **Bloqueo mortal distribuido.**- Puede ocurrir un bloqueo mortal en varios nodos lo que hace necesario extender las técnicas de manejo de bloqueos mortales para que se tenga en cuenta.

Existen varios métodos de control de concurrencia que manejan los elementos de información replicados en una base de datos distribuida que se basan en la extensión de las técnicas de control de concurrencia de las bases de datos centralizadas. La principal idea es designar una copia determinada de cada elemento de información como copia distinguida. Las llaves para este elemento de información se asocian a la copia distinguida y todas las solicitudes de bloqueo y desbloqueo se envían al nodo que contienen esa copia.

El proceso de recuperación en las bases de datos distribuidas es bastante complicado; en ciertos casos incluso es difícil determinar si un nodo está inactivo sin intercambiar un gran número de mensajes con otros nodos supóngase que el nodo "X" envíe un mensaje al nodo "Y" y espera una respuesta de Y pero no la recibe; donde se tienen varias explicaciones posibles:

- El mensaje no llega a "Y" debido a un fallo de comunicación.
- El nodo "Y" está inactivo y no pudo responder.
- El nodo "Y" está activo y envía una respuesta pero esta no llega.

Sin información adicional y sin el envío de mensajes adicionales es difícil determinar que sucedió realmente. Otro problema con la recuperación distribuida es la recuperación distribuida; cuando una transacción esta actualizando datos en varios nodos, no puede confirmarse hasta estar segura de que el efecto de las transacciones de todos los nodos no puede perderse. Esto significa que cada nodo debe haber asentado primero permanentemente los efectos locales de la transacción en la bitácora local en el disco del nodo. A menudo se usa el protocolo de confirmación de dos fases para garantizar la corrección de la conformación distribuida la cual este protocolo de 2 fases menciona que si todas las bases de datos participantes contestan "correcto" la transacción tiene éxito y el coordinador envía a las participantes una señal "confirmar para esta transacción. Como todos los efectos locales de la transacción se han asentado en las bitácoras de la base de datos participantes, ya es posible recuperarse de un fallo. Cada base de datos participante completa la confirmación de la transacción escribiendo en la bitácora una entrada para esa transacción y actualizando permanentemente la base de datos si es necesario. Por otro lado si una o más de las bases de datos participantes envió respuesta "incorrecto" al coordinador, la transacción habrá fallado, y el coordinador enviará a cada participante un mensaje de "revertir" o "deshacer" el efecto local de la transacción donde la anulación se efectúa usando la bitácora.

Análisis del sistema

Requerimientos

El Grupo GEO se encuentra formado por empresas, delegaciones y un corporativo. En las empresas se realizan todos los procesos operativos del negocio, desde la adquisición de terrenos por parte de promoción, hasta la entrega de la vivienda por parte de titulación. En las delegaciones se realizan actividades de control y ayuda a las empresas que concentra. Así mismo el corporativo se encarga de las mismas actividades de control y ayuda a nivel grupo.

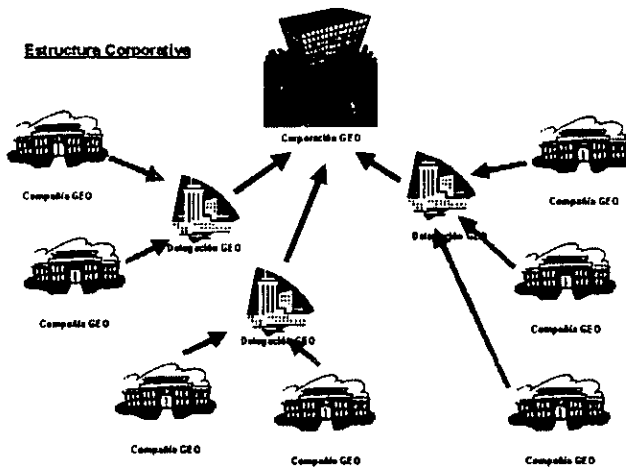


Figura 53 Estructura corporativa

Cada compañía cuenta con sus oficinas, en cambio las delegaciones son entes lógicos que residen en alguna compañía de las que tienen a su cargo; el corporativo cuenta con sus oficinas centrales, donde se realiza todas las actividades corporativas, consolidación de información y puesta en servicio de información a las delegaciones.

El grupo GEO está formado por las siguientes empresas:

Corporación GEO, S.A. de C.V.		
Empresa	Ubicación	Región de negocio
Delegación Centro		
PHISA, S.A. de C.V.	México, D.F.	Área metropolitana, Zona Norte
GEO Edif, S.A. de C.V.	México, D.F.	Área metropolitana, Zona Oriente
GEO Morelos, S.A. de C.V.	Cuernavaca, Morelos	Cuernavaca, Cuautla
Delegación Suroeste		
GEO Puebla, S.A. de C.V.	Puebla, Puebla	Puebla
GEO Acapulco, S.A. de C.V.	Acapulco, Guerrero	Acapulco
GEO Ixtapa, S.A. de C.V.	Ixtapa, Guerrero	Ixtapa
Delegación Noroeste		
GEO Tijuana, S.A. de C.V.	Tijuana, Baja California Norte	Tijuana
GEO Mexicali, S.A. de C.V.	Mexicali, Baja California Norte	Mexicali
GEO Ensenada, S.A. de C.V.	Ensenada, Baja California Norte	Ensenada
Delegación Noreste		
GEO Monterrey, S.A. de C.V.	Monterrey, Nuevo León	Monterrey
GEO Reynosa, S.A. de C.V.	Reynosa, Tamaulipas	Reynosa
GEO Tampico, S.A. de C.V.	Tampico, Tamaulipas	Tampico
GEO Torreón, S.A. de C.V.	Torreón, Coahuila	Torreón
Delegación Bajío		
GEO Jalisco, S.A. de C.V.	Guadalajara, Jalisco	Guadalajara
GEO Queretaro, S.A. de C.V.	Queretaro, Queretaro	Queretaro
GEO Guanajuato, S.A. de C.V.	Guanajuato, Guanajuato	Guanajuato
Delegación Sur		
GEO Oaxaca, S.A. de C.V.	Oaxaca, Oaxaca	Oaxaca
GEO Veracruz, S.A. de C.V.	Veracruz, Veracruz	Veracruz
Internacionales		
GEO Beizer Inc.	Houston, Texas, EUA	Houston
GEO SAL, S.A. de C.V.	Buenos Aires, Argentina	Buenos Aires

Cada compañía se estructura mediante una oficina central, y diferentes puntos remotos para la operación en general. Cada uno de estos puntos cumplen con diferentes actividades a lo largo del proceso comercial, ya que las áreas que se encuentran físicamente en estos puntos remotos por la naturaleza de sus trabajos. Los puntos remotos pueden ser puntos de venta o obras. La operación comercial se realiza mayormente en los puntos de venta y en oficina centrales, pero requiere de información otorgada por áreas que residen en la obra, como son promoción, proyectos, etc.

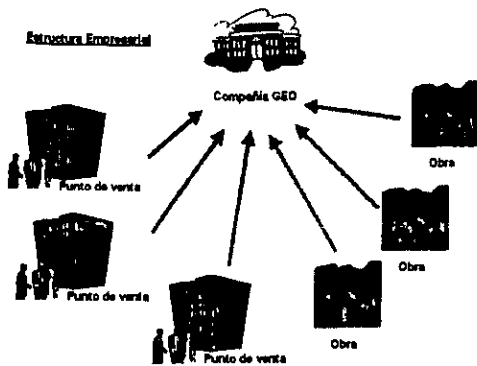


Figura 54 Estructura empresarial. Puntos remotos de venta y obras

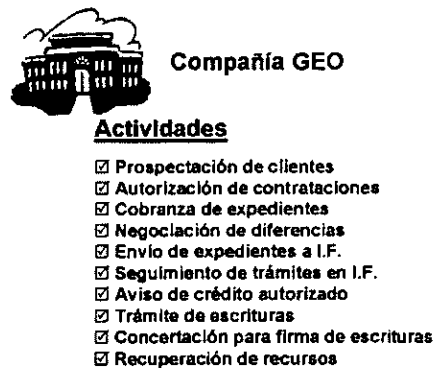


Figura 55 Actividades de la compañía

En las oficinas centrales de la compañía, o solo compañía, se realizan las actividades que por su naturaleza requieren de un lugar fijo para realizarlo, tal como las tareas de análisis, acuerdos con las instituciones financieras, atención especializada de clientes, etc

Dentro del punto de venta, se realiza el primer contacto con el cliente, así como la contratación y primeros trámites, así como consulta y cobro por concepto de Buró de crédito.

En la obra, practicamente se realiza las actividades relacionadas con la vivienda, como son la captura del sembrado, preparación de la viviendas y la firma de escrituras.



Actividades

- Captación de afluencia
- Consultas a Buró de crédito
- Contrataciones
- Cobranza de expedientes
- Recepción de documentos
- Atención a clientes

Figura 56 Actividades en el punto de venta



Actividades

- Captura del sembrado (Viviendas)
- Preparación y entrega de vivienda
- Firma de escrituras

Figura 57 Actividades en la obra

Durante este análisis y levantamiento de requerimientos, fue necesario visitar cada una de las empresas, con el fin de obtener un detalle claro de la operación. El medio fue mediante entrevistas con las personas operativas y directivas de cada área involucrada.

Para la presentación de los requerimientos y análisis realizados, se decidió sintetizar los resultados, de tal manera que se muestran sólo las áreas de Ventas, Cobranza y Titulación.

Ventas

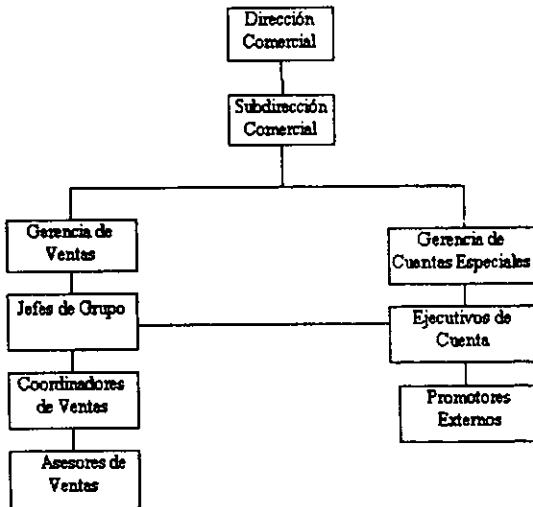
Descriptivo del área

La gestión de ventas realiza las tareas necesarias para lograr que las personas interesadas en la compra de una vivienda se vuelvan clientes de GEO. Este proceso necesita de la conjugación de tareas realizadas por diferentes áreas de Comercialización, Titulación y Crédito y Cobranza, siendo: Cobranza, Ventas, Ejecutivos de Cuenta, Telemarketing, Atención a Clientes, Dictamen y Mercadotecnia.

Objetivo del área.

Lograr generar las ventas metas establecidas en el Plan de Ventas, al captar como clientes a la afluencia que llega a cada punto de venta. Y por otro lado busca realizar ventas a futuro y a gran escala, sin tener que depender de la afluencia a los puntos de venta, para ello se enfoca a realizar labor de venta directamente en las empresas y así obtener un acercamiento directo con el cliente.

Estructura del área



Cobranza

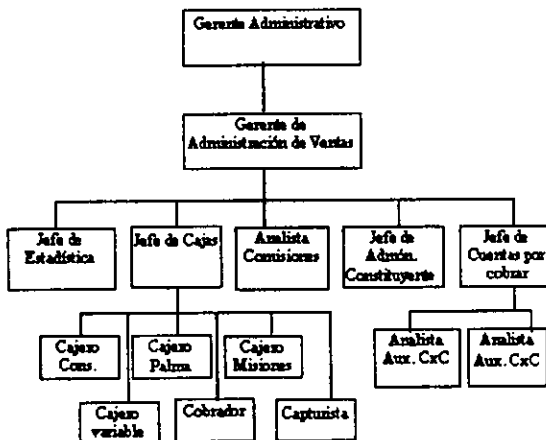
Descriptivo del área

El área de cuentas por cobrar se encarga de coordinar la cobranza del enganche, los gastos de investigación relacionados con la adquisición de la vivienda y el pago de comisiones, así como de generar la información concerniente al área; realiza las negociaciones necesarias con los clientes que reportan diferencias o que reportan retrasos en sus pagos y que necesitan una reestructuración de su deuda.

Objetivo del área.

Recuperar oportunamente el total del enganche y los gastos derivados de la compra de una vivienda.

Estructura del área



Titulación

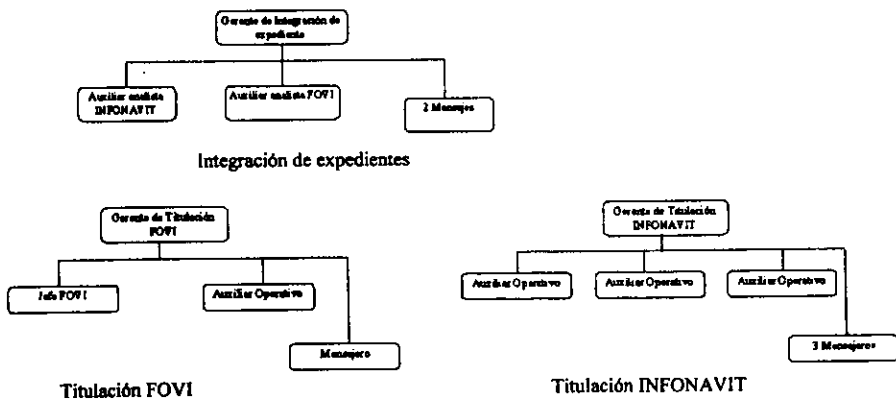
Descriptivo del área

La área de titulación lleva acabo los diferentes trámites en la obtención de un crédito para la adquisición de una vivienda. Estos trámites se llevan acabo en diferentes instituciones de crédito gubernamentales y no gubernamentales. Al finalizar el trámite de obtención del crédito, el área se dispone a realizar los pasos subsecuentes para la asignación, escrituración y entrega de la vivienda. Posteriormente se procede a realizar la recuperación del crédito, lo cual es el cobrar el monto de la casa ya vendida a las instituciones que otorgaron el crédito.

Objetivo del área.

Lograr la integración de los documentos solicitados por las Instituciones Financieras, para el financiamiento de la vivienda, de tal manera que se cubran los requisitos establecidos por las Instituciones para el buen fin del trámite. Asimismo, una vez obtenido el crédito para el financiamiento de la vivienda (a excepción de las viviendas adquiridas al contado), es responsabilidad de esta área lograr la escrituración de las viviendas.

Estructura del área



Análisis general

Durante el desarrollo del análisis de requerimientos, se llegó a la conclusión que existían diferentes ámbitos de necesidades de la información. Los diferentes ámbitos fueron:

Ámbito	Descripción
Empresarial operativo	En el ámbito empresarial operativo se requiere de la información de la operación en sí, con poco o análisis de la misma, pues a este nivel solo se utiliza para dar continuidad al trabajo diario. En realidad el personal que ocupa esta información es como apoyo a su trabajo, y generalmente son las mismas personas que capturaron la información en el sistema
Empresarial ejecutivo	En el ámbito empresarial ejecutivo se requiere del análisis de la información. Generalmente el personal a este nivel tiene las funciones de supervisión de que el trabajo se realice, estadísticas del trabajo realizado y tendencias en la operación. La mayoría de las consultas realizadas en este ámbito es revisar los números de las operaciones realizadas (ventas, contrataciones, escrituraciones, cancelaciones, etc)
Empresarial Directivo Delegacional Directivo Corporativo Directivo	En estos ámbitos la consulta de la información es muy similar, pues en este caso requieren también saber de las operaciones realizadas en cada compañía, delegación o todo el corporativo respectivamente, junto a las tendencias de las operaciones (ventas, contrataciones, escrituraciones, cancelaciones). Así mismo requieren de comparar esta información real con la estimada al inicio del año, ya que dentro de Grupo GEO se estiman las operaciones al inicio de cada año y se utilizan para las comparativas entre el real y lo estimado.

De acuerdo a esto se tomó la decisión de iniciar con el ámbito operativo empresarial, y una vez definida y establecida la operación de las empresas, aprovechar las bases de datos creadas para las empresas para la creación de la información ejecutiva y directiva, en los ámbitos empresarial, delegacional y directiva.

Ámbito empresarial operativo

Dentro de este ámbito se realizó el estudio de los diferentes objetos (Modelo de objetos) de la aplicación en las áreas de comercialización (Ventas, Cobranza y Titulación), junto a conocer la operación que éstos tienen (Modelo dinámico) y los flujos de información que de cada proceso establecido se tiene (Modelo funcional).

Modelo de objetos

En el modelo de objetos se procedió a encontrar los objetos que se encontraban en la aplicación, junto a las relaciones que tenían entre ellos. Durante la realización del análisis se realizaron diferentes modelos que correspondían a los diferentes momentos en que el cliente se encuentra en contacto con el Grupo. De esta manera primeramente se llegaron a los siguientes submodelos de objetos

- ✓ Afluencia
- ✓ Prospectación
- ✓ Cliente
- ✓ Contratación
- ✓ Cobranza y caja
- ✓ Trámites
- ✓ Construcción

Modelo de objetos - Afluencia

El modelo de afluencia se formó a partir del momento en que un cliente llega a un punto de venta de una compañía GEO. En este se toman los datos necesarios para poder contactarlo posteriormente. Así mismo, los datos son utilizados para poder realizar la prospección de la afluencia. Para el control de la afluencia se contempla este modelo.

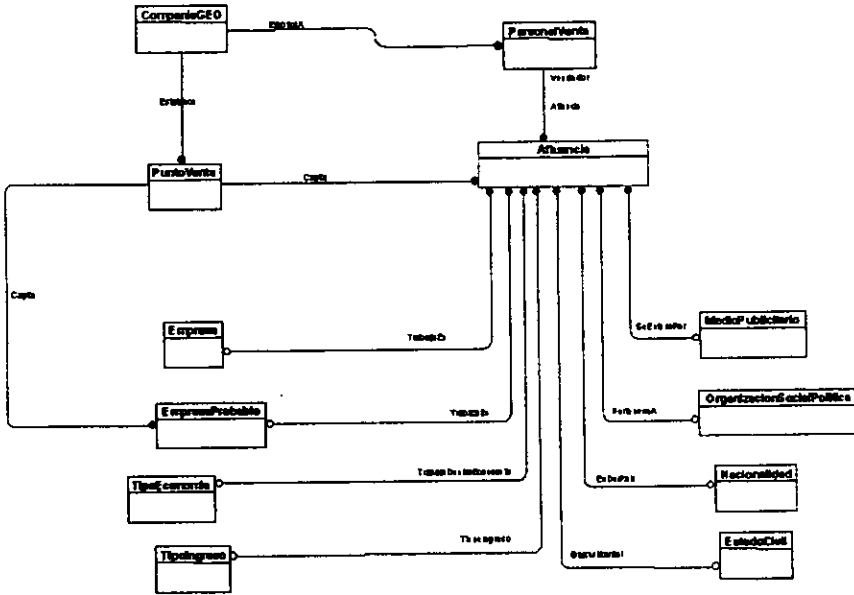


Figura 58 Diagrama de objetos - Afluencia

Modelo de objetos - Prospección

En la prospección se realiza un estudio de la capacidad crediticia de la afluencia para saber que personas pueden ser sujetos de crédito actualmente o en el futuro, debido al perfil que presenta. A los prospectos que cumplen con el perfil y son sujetos de crédito, se les realiza una venta por medio telefónico, invitándoles a conocer nuestras casas e incluyéndolos en promociones realizadas dentro de los desarrollos constructivos. Para el control de la prospección se contempla este modelo.

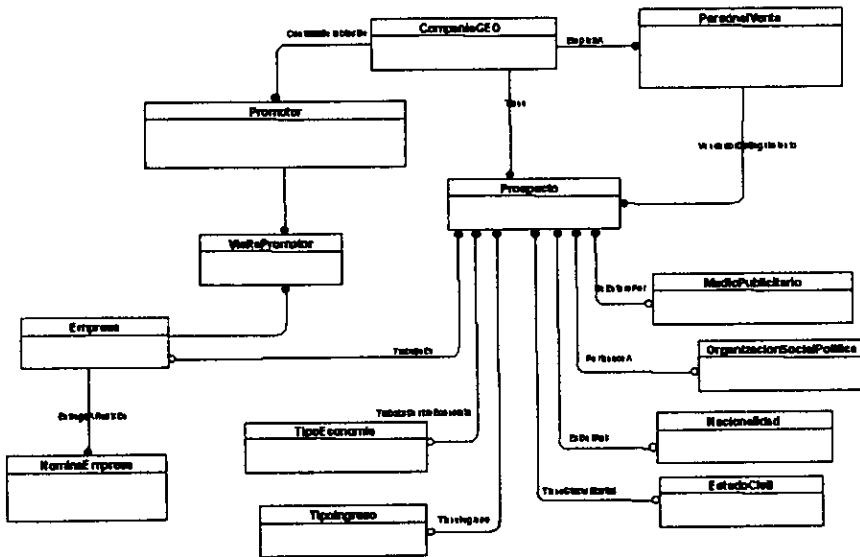


Figura 59 Diagrama de objetos – Prospección

Modelo de objetos - Cliente

Cuando una persona inicia un trámite de adquisición de vivienda, se le denomina cliente. Para el manejo de todos los datos relacionados al cliente, así como personas y empresa donde trabaje, se contempla este modelo.

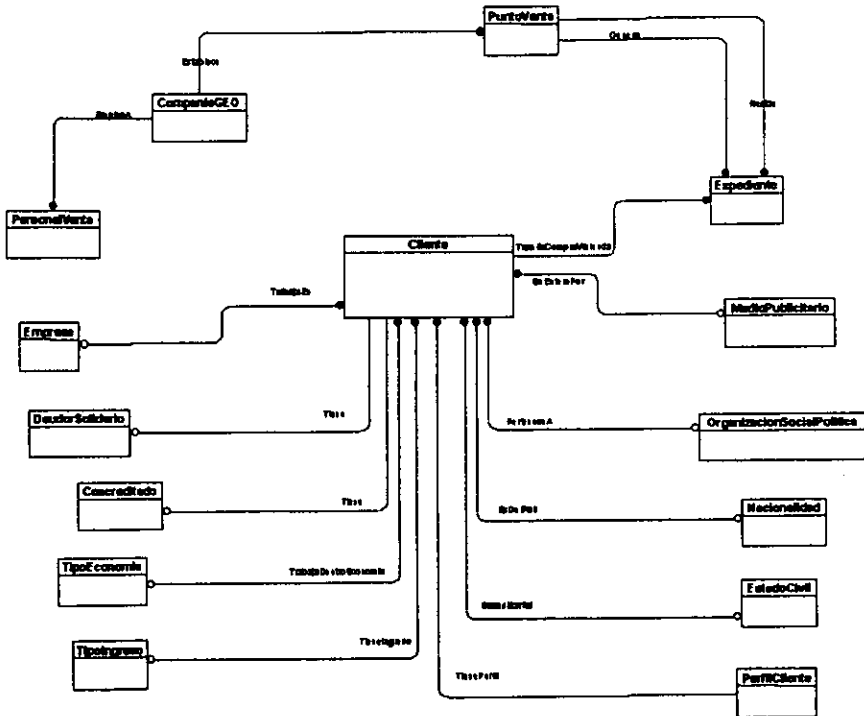


Figura 60 Diagrama de objetos – Cliente

Modelo de objetos - Trámites

Durante la tramitación de un crédito ante una institución financiera, se llevan una serie de pasos a realizar, junto a requisitos a cumplir y documentos a entregar. Así mismo cada institución financiera tiene sus propios trámites, requisitos y documentos. Para cada cliente que tiene un expediente con un trámite debe elegir la institución financiera, tipo de crédito y el producto a comprar. Conforme avanza en el trámite, el seguimiento es importante por parte del personal de titulación y ventas, con el fin de ayudar a tener el expediente siempre actualizado y preparar los trámites gubernamentales y documentos necesarios para la posterior escrituración y entrega de vivienda. Para el control del avance de expedientes en titulación y actualización de requisitos y documentos, se contempla este modelo.

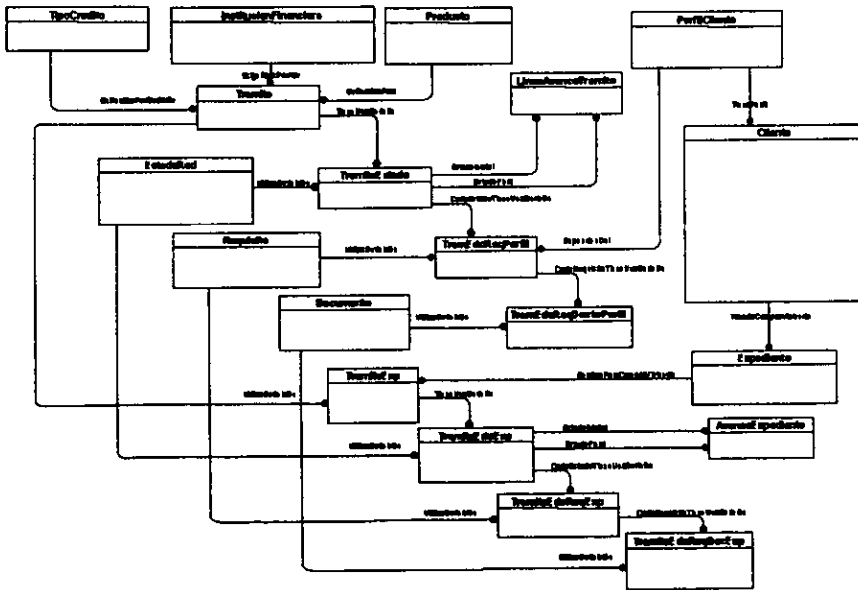


Figura 63 Diagrama de objetos - Trámites

Modelo de objetos - Construcción

La parte de construcción es el centro de la operación de Grupo GEO, pues es en esta parte se encuentra el negocio de la empresa. Para el control de los terrenos adquiridos como reserva territorial, los diversos desarrollos constructivos (promoción) llevados a cabo, las zonas constructivas en que se divide una promoción y las viviendas que afecta, junto los productos (veces salarios mínimos) y prototipos de las mismas, que posteriormente se venden al cliente, se contempla este modelo.

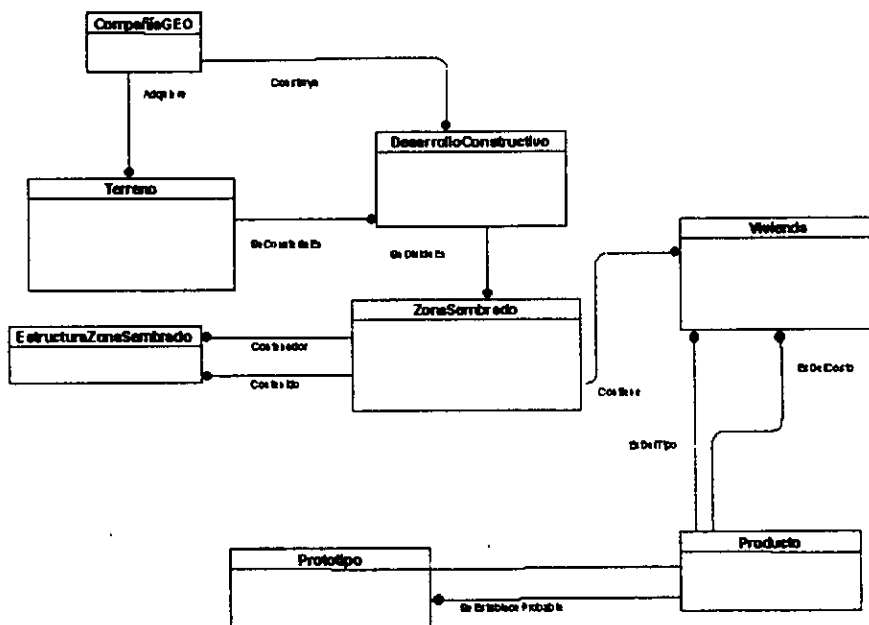


Figura 64 Diagrama de objetos - Construcción

Modelo dinámico

Dentro del modelo dinámico se procedió a obtener la secuencia de eventos y diagramas de estado que indican la forma en que se realiza la operación de la empresa. Cabe mencionar que en este momento se toma en cuenta simplemente la secuencia temporal de los eventos sin importar la escala del tiempo, junto a omitir los procedimientos realizados para cada evento.

Durante la realización del análisis se realizaron diferentes escenarios que correspondían a los diferentes momentos en que el cliente se encuentra en contacto con el Grupo.

Escenarios

- ✓ Captación de afluencia
- ✓ Labor de venta
- ✓ Contratación
- ✓ Cobranza y caja
- ✓ Trámites
- ✓ Avance de expediente
- ✓ Autorización del crédito
- ✓ Negociación de diferencias
- ✓ Escrituración

Escenario Captación de afluencia

El escenario típico de la captación de afluencia inicia con el acercamiento por parte del cliente a un punto de venta de la empresa GEO, donde existe una recepción donde se da información de las casas GEO. La recepción se encarga de entregar la información solicitada. Así mismo solicita datos básicos a la persona (llamado ya desde aquí Afluente). Los datos básicos se conforma de su nombre completo, teléfono particular (en algunas empresas solicitan así mismo los datos como la dirección, sueldo actual, etc., las cuales quedan a la consideración de la dirección de ventas de la empresa). Después de ingresar los datos al sistema, se le pregunta al afluente si desea ser atendido por un vendedor, que en caso de que se le responda afirmativamente, se le asigna a un vendedor al cual se informa oportunamente.

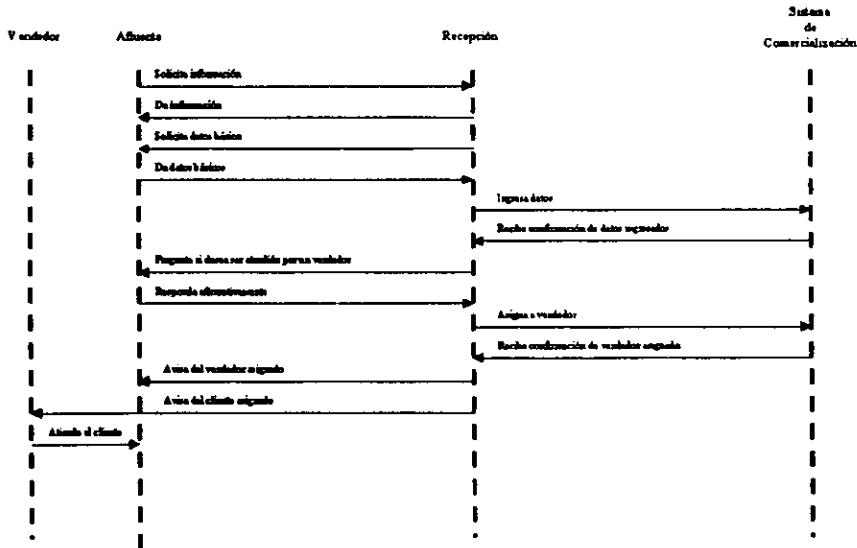


Figura 65 Secuencia de eventos – Captación de afluencia

Escenario Labor de venta

La labor de venta se realiza mediante el recorrido y convencimiento que realiza el vendedor con el afuente para convertirlo en cliente. En caso de que el cliente esté interesado en las casas GEO, entonces se procede a solicitarle la autorización y pago por concepto de consulta al Buró Nacional de Crédito (BNC) el cual nos proporciona la información necesaria para saber si el afuente es sujeto de crédito.

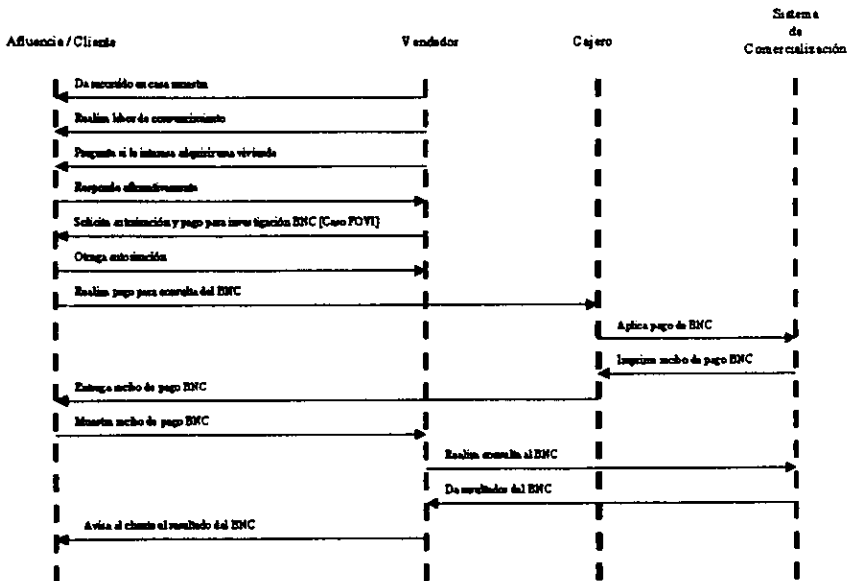


Figura 66 Secuencia de eventos – Labor de venta

Escenario Contratación

La contratación se realiza después de haber obtenido que el cliente es sujeto de crédito y se encuentra interesado en las casas GEO. Primeramente, el vendedor procede a mostrar los diferentes planes de venta y planes de pago al cliente, de los cuales el cliente elige el plan de ventas y plan de pagos deseados. Dentro de estos planes se elige el prototipo de vivienda que va adquirir, el tipo de crédito el cual va a contratar, la institución financiera que el corresponde, así como la forma de pago del enganche, gastos administrativos y de escrituración, etc. los cuales cubre el cliente en la entrega de vivienda. Los montos de esta contratación son estimados, por lo cual solo se genera información probable que ayuda al GEO a adelantar los trámites de la institución financiera, como son que el cliente tenga pagado su enganche para poder ejercer el crédito

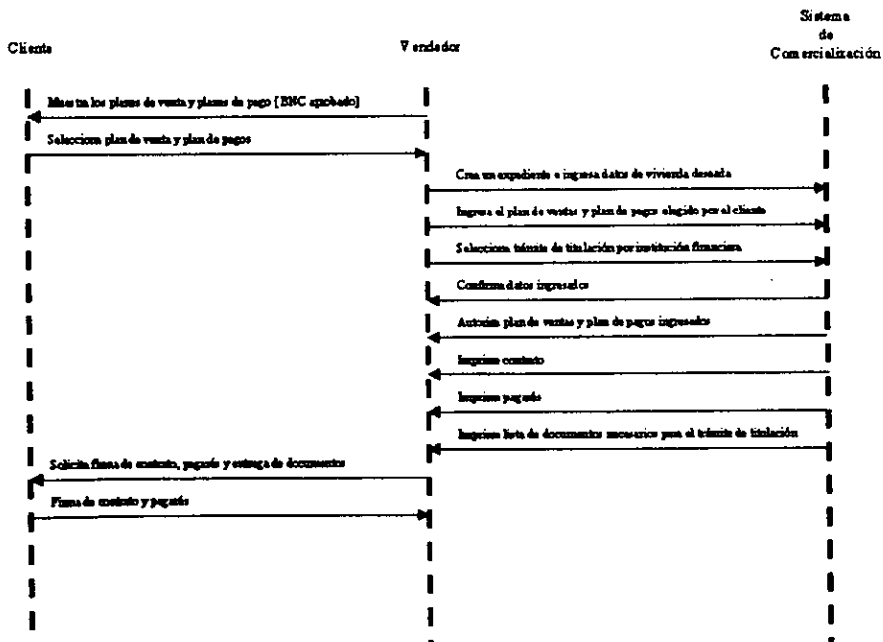


Figura 67 Secuencia de eventos – Contratación

Escenario Cobranza y caja

La cobranza se realiza una vez que el cliente ya ha iniciado un trámite con GEO y se ha obligado con la empresa a liquidar el enganche y los otros gastos. Aquí primeramente se solicita el estado de cuenta del cliente en su expediente y el cajero lo imprime para entregárselo al cliente. Este realiza su pago y aplica ese pago a la deuda contraída con GEO e imprime el recibo de pago y los cargos afectados por el pago, entregando recibo y un nuevo estado de cuenta para el cliente.

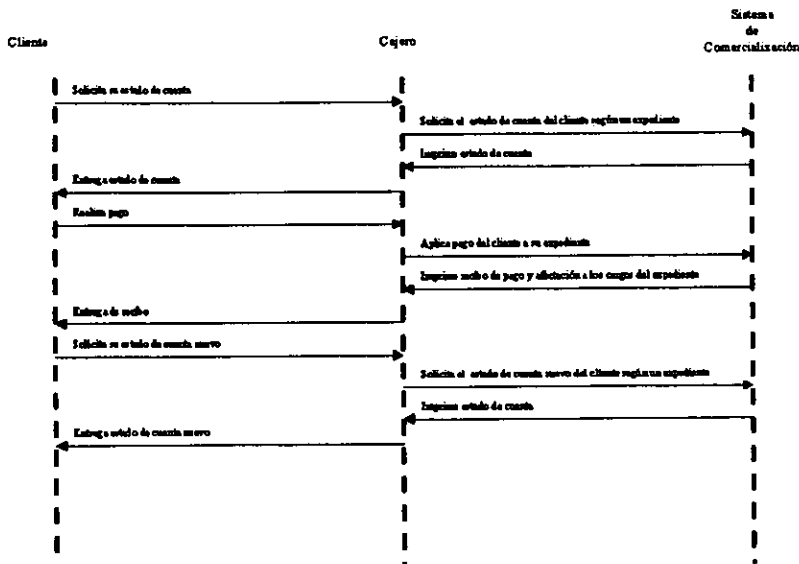


Figura 68 Secuencia de eventos – Cobranza

Escenario Trámites

El trámite es seleccionado por el cliente durante la contratación. De allí en adelante, el personal de titulación se encarga de solicitarle los requisitos a cumplir así como los documentos a entregar, junto a informar al cliente del avance del expediente dentro de la institución financiera.

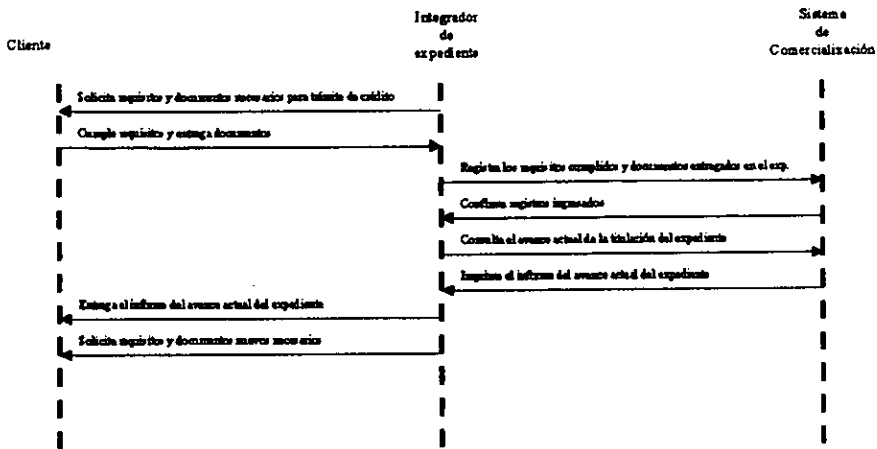


Figura 69 Secuencia de eventos – Trámites

Escenario Avance del expediente

Durante el trámite, se realizan el cumplimiento de todos los requisitos y los documentos requeridos para llevar a cabo la autorización del crédito. La institución financiera informa de los avances en los trámites, los cuales a su vez son reportados al sistema por medio del auxiliar. En caso de haber faltantes o necesidad de nuevos documentos se procede a ingresarlos al sistema para avisar al integrador que a su vez los solicitará al cliente.

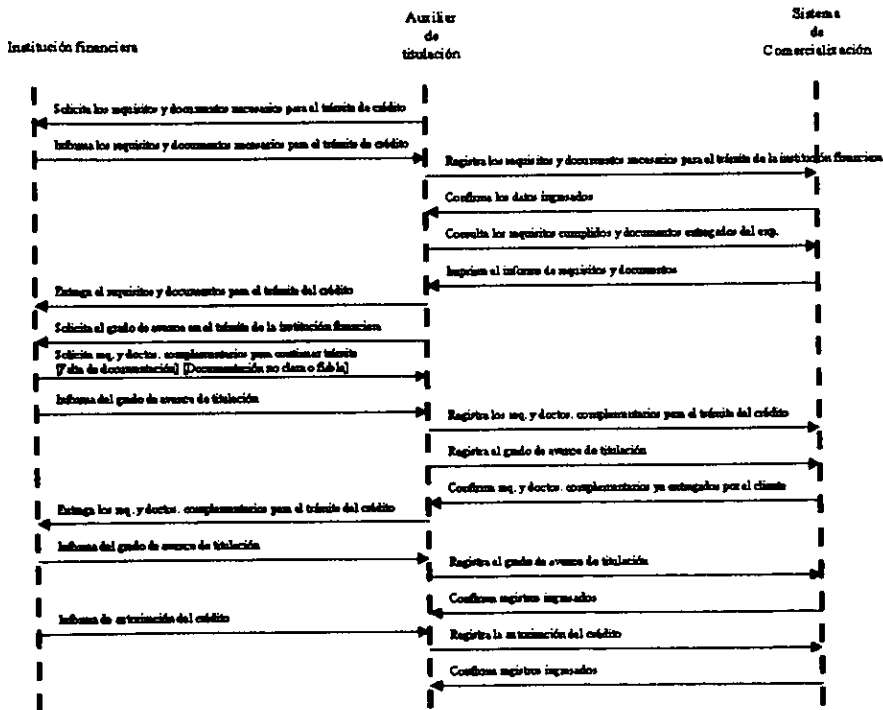


Figura 70 Secuencia de eventos - Avance de expediente

Escenario Autorización del crédito

Cuando se dá la autorización del crédito, el integrador de expediente, personal de titulación, avisa al cliente de los requisitos o documentos necesarios para completar su trámite. Así mismo registra todo los requisitos y documentos en el sistema, junto a enviarlos a la institución financiera. Por otro lado revisa los documentos necesarios y trámites requeridos por la notaría a fin de preparar la vivienda a entregar, pidiéndole al cliente seleccionar la vivienda definitiva a adquirir (hasta antes solo se tenía el tipo de vivienda a comprar), apartando la casa para la entrega a ese expediente

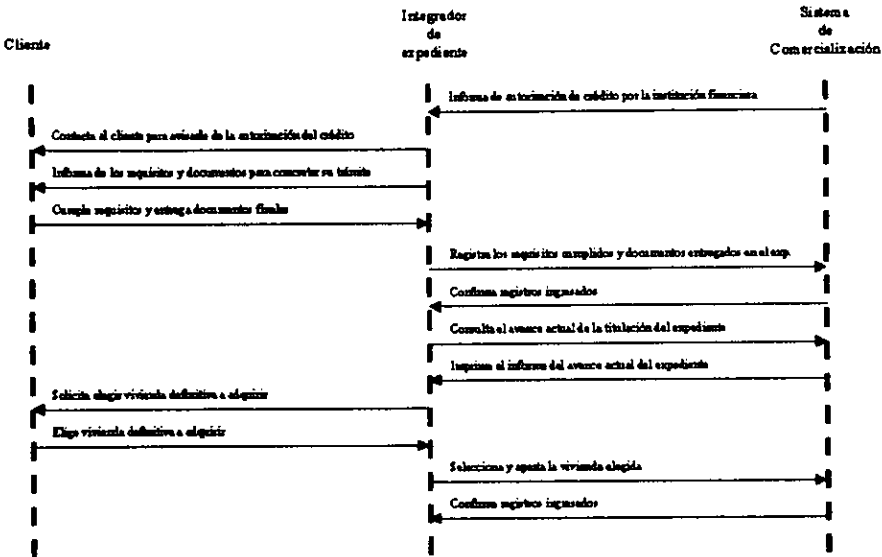


Figura 71 Secuencia de eventos -- Autorización del crédito

Escenario Negociación de diferencias

Al autorizarse el crédito, también se debe de ejecutar la negociación de diferencias, lo cual es el ver el monto de crédito otorgado por la institución financiera al cliente de acuerdo a los perfiles y estudios del cliente hechos por la institución financiera. Al contratar, los montos de enganches, gasto administrativos, etc., fueron estimados. En este paso se procede a revisar los montos reales ya con el crédito autorizado y ver las diferencias, las cuales se negocian con el cliente y se crea un nuevo plan de pagos para realizar la escrituración y entrega de vivienda

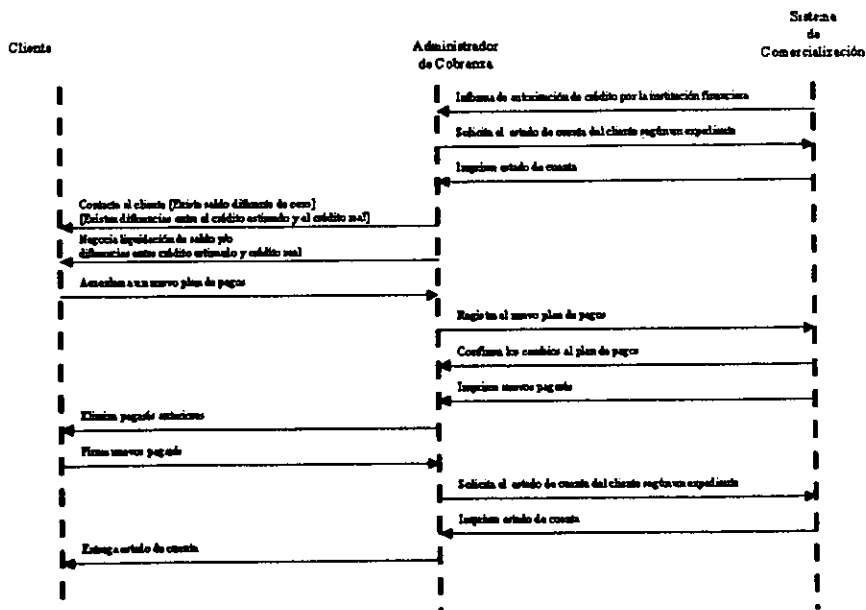


Figura 72 Secuencia de eventos – Negociación de diferencias

Escenario Escrituración

Al autorizarse el crédito, se confirma la cobranza del cliente ya negociada y/o totalmente cubierta, así como una consulta de que todos los requisitos y documentos estén completos, después de lo cual se proceden a realizar los trámites necesarios con la notaría y solicita la escritura, que después de la entrega se procede notificar al cliente con su expediente de la fecha de firma de escrituras, a la cual acudirá el cliente, un representante de la institución financiera, un representante de la empresa GEO y el notario. Generalmente ese mismo día se realiza la entrega de la vivienda.

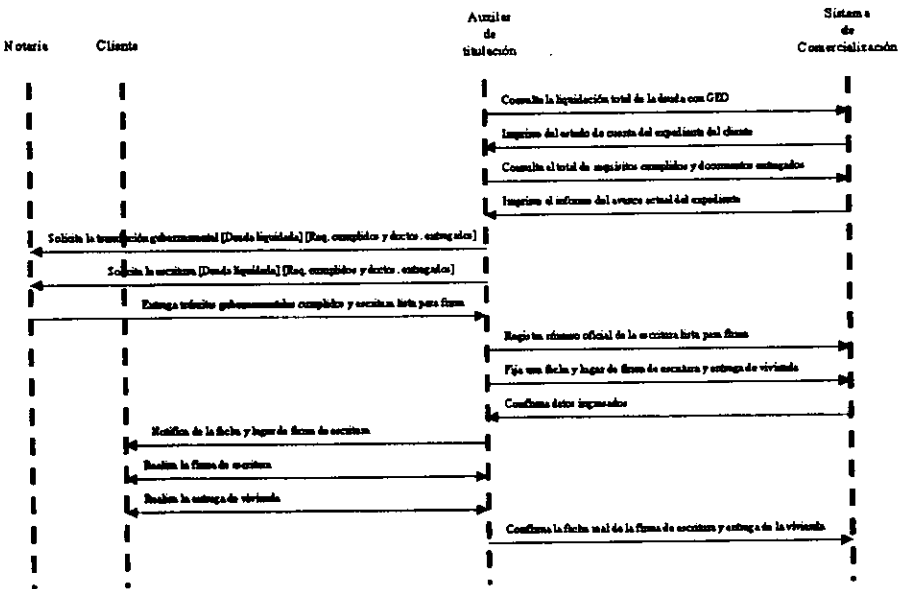


Figura 73 Secuencia de eventos -- Escrituración

Diagrama de estados

A partir de las secuencias de estados se prosiguió a realizar el análisis para obtener los diagramas de estados de los objetos más importantes, con el objeto de tener ampliamente definido el modelo dinámico del sistema.

De una serie de escenarios se obtiene el diagrama de estados del cliente completo, involucrando los diferentes pasos durante la relación con Grupo GEO.

- ✓ Cliente
- ✓ Cliente – Persona como cliente
- ✓ Cliente – Cliente con trámite
- ✓ Cliente – Cliente con crédito autorizado
- ✓ Cliente – Cliente sin negociación de diferencias
- ✓ Cliente – Cliente para firma de escrituras
- ✓ Vendedor
- ✓ Institución financiera

Subdiagrama Cliente con crédito autorizado

Diagrama de estados Cliente

Subdiagrama de estados Cliente con crédito autorizado

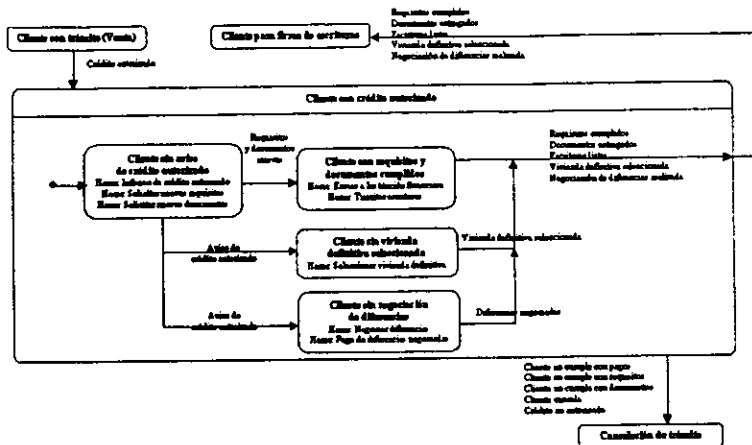


Figura 77 Diagrama de estados – Subdiagrama Cliente con crédito autorizado

Subdiagrama Cliente sin negociación de diferencias

Diagrama de estados Cliente

Subdiagrama de estados Cliente con crédito autorizado

Subdiagrama de estados Cliente en negociación de diferencias

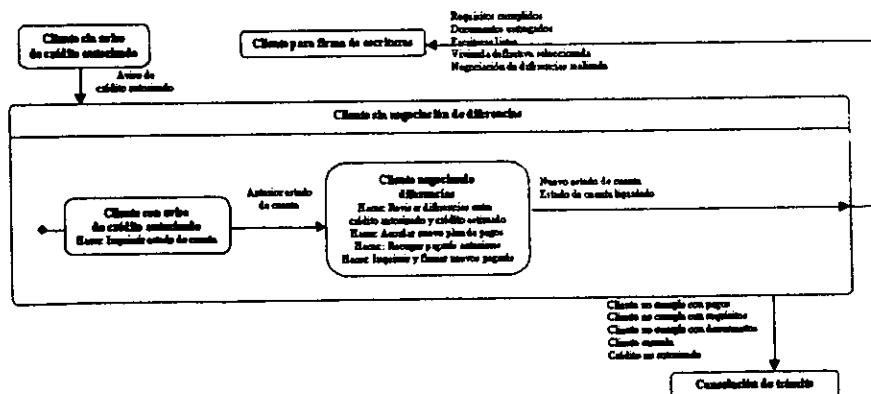


Figura 78 Diagrama de estados – Subdiagrama Cliente sin negociación de diferencias

Subdiagrama Cliente para firma de escrituras

Diagrama de estados Cliente

Subdiagrama de estado de Cliente para firma de escrituras

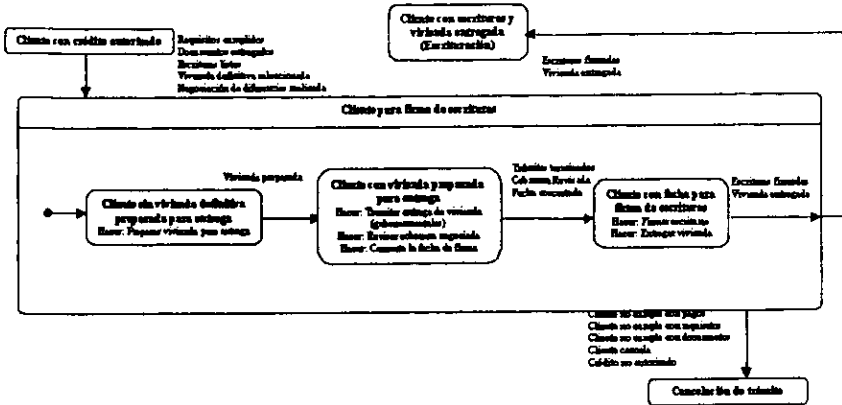


Figura 79 Diagrama de estados – Subdiagrama Cliente para firma de escrituras

Diagrama de estados Vendedor

Diagrama de estados Vendedor

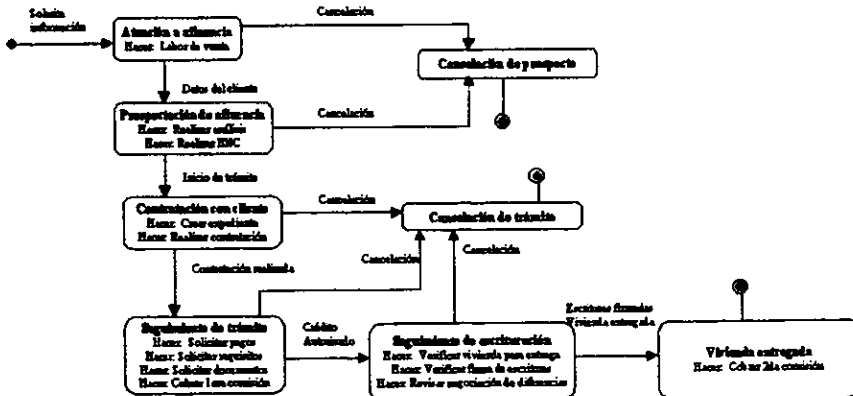


Figura 80 Diagrama de estados – Vendedor

Diagrama de estados Institución financiera

Diagrama de estados Institución financiera

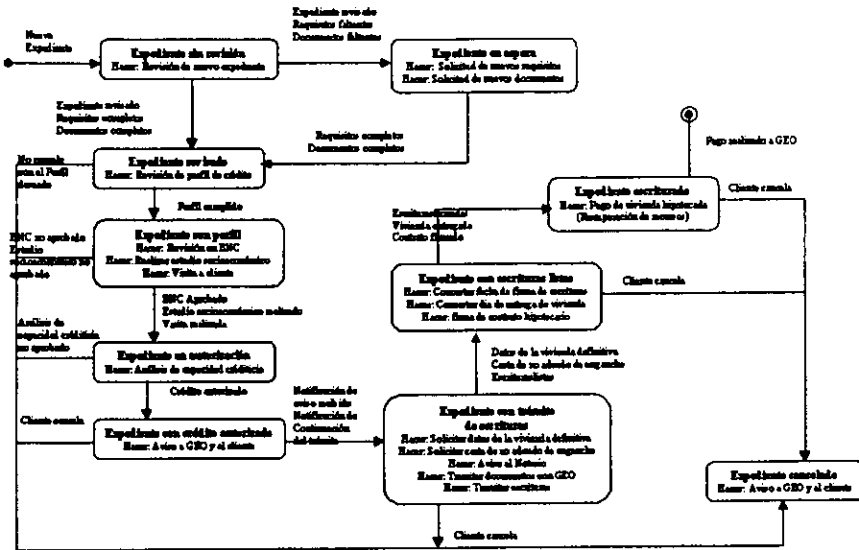


Figura 81 Diagrama de estados - Institución financiera

Modelo funcional

En el modelo funcional, se obtuvo la serie de procesos jerárquicos que componen la operación de la empresa, preocupándose principalmente en los insumos y productos de cada proceso, así como la serialización de los mismos.

Debido a que inicialmente las entrevistas se siguieron con una metodología diferente a OMT, se procedió a transcribir los resultados obtenidos en la entrevistas a modelos OMT funcionales, con el fin de tener el modelo completo del sistema en OMT.

- ✓ Ventas
- ✓ Cobranza
- ✓ Titulación

Ventas

En el modelo funcional, nos enfocamos a esquematizar el modelo de entradas, proceso, salidas, junto a las fuentes de datos y destinos de los mismo. Para el caso de Ventas, se realiza la gestión de ventas, donde se realiza desde el plan de ventas de acuerdo a la información suministrada por la Institución financiera como por parte de construcción, hasta la entrega del contratos, negociación y expediente a Cobranza y Titulación

Proceso *Gestionar Ventas* Nivel 0

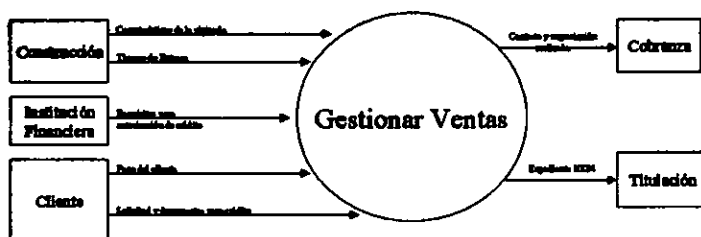


Figura 83 Proceso Gestión Ventas - Diagrama de Contexto

Proceso *Gestionar Ventas* Nivel 1

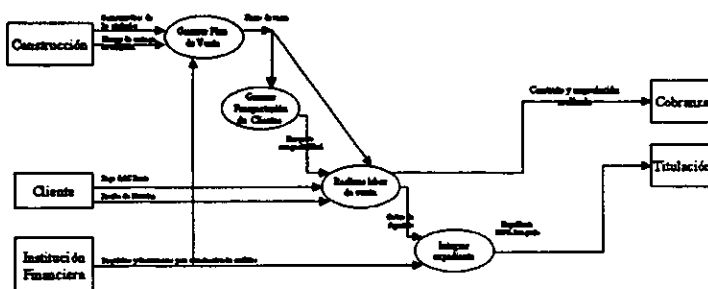


Figura 82 Proceso Gestión Ventas - Diagrama Nivel 1

Cobranza

En el modelo funcional de cobranza, se revisan las entradas y salidas, en general, los cargos y abonos realizados a los expedientes de los clientes; los primeros como reflejo de los planes de venta contratados por el cliente, y los segundos como los pagos realizados por el cliente. Junto a esto la negociación de diferencias también forma parte importante de este modelo.

Proceso *Realizar Cobranza*.

Nivel 0

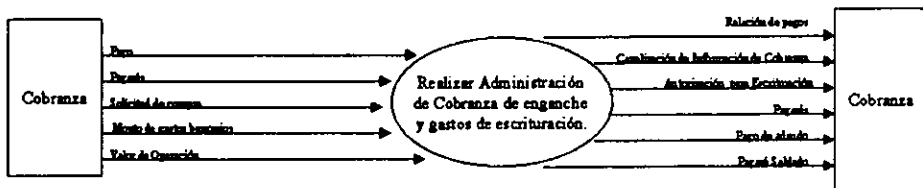


Figura 85 Proceso Realizar Cobranza - Diagrama de Contexto

Proceso *Realizar Cobranza*.

Nivel 1

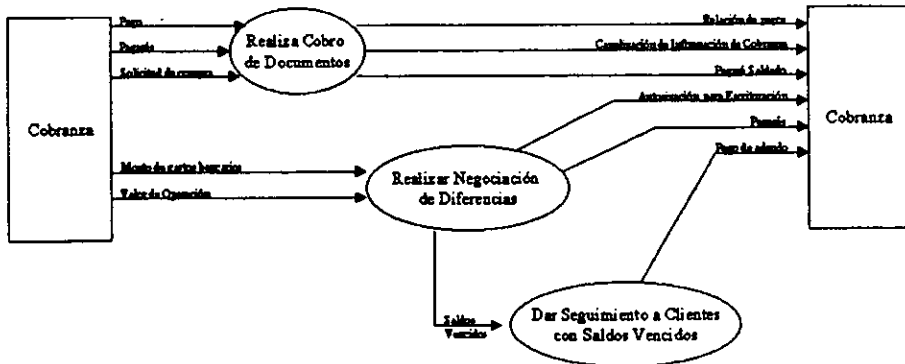


Figura 84 Proceso Realizar Cobranza - Diagrama Nivel 1

Titulación

Dentro del proceso de titulación, se realiza una serie de pasos que llevan al cliente a completar todos los documentos y requisitos solicitados por la institución financiera para analizar la autorización de un crédito. Junto a esto, se autoriza o niega el crédito, procediéndose a la escrituración y entrega de vivienda.

Proceso *Realizar titulación* Nivel 0

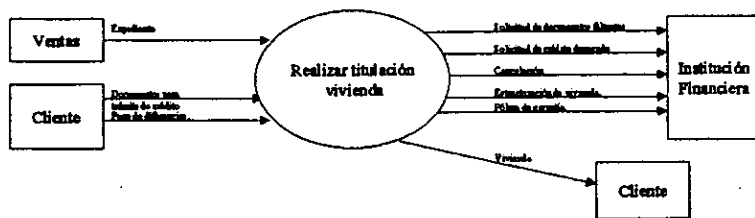


Figura 87 Proceso Realizar Titulación- Diagrama de Contexto

Proceso *Realizar titulación* Nivel 1

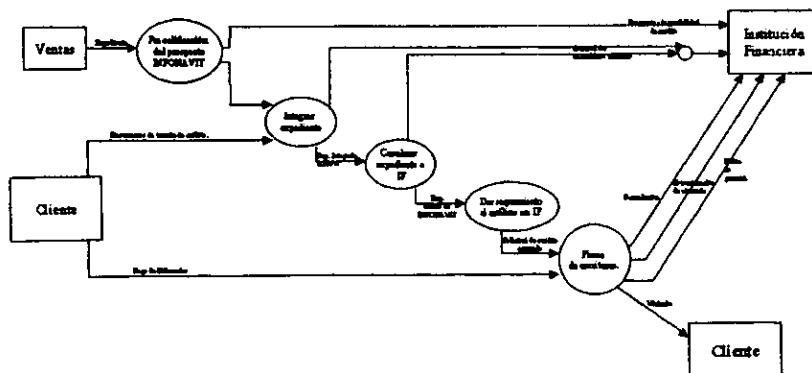


Figura 86 Proceso Realizar Titulación – Diagrama Nivel 1

Ámbito empresarial ejecutivo, empresarial directivo, delegacional directivo, corporativo directivo

En estos ámbitos se realizó el estudio de los objetos necesarios para crear el análisis de información.

Para el ámbito empresarial ejecutivo, mucha de la información encontrada en este ámbito son consultas simples a la información capturada en el ámbito operativo, de tal manera que muchas se convierten en programas de explotación de la información.

Dentro de este ámbito se realizó el estudio de los modelos de objetos, dinámico y funcional para agilizar la consulta de información consolidada, así como crear los mecanismos necesarios para el análisis de la información generada por el ámbito operativo

Modelo de objetos

En el modelo de objetos se procedió a encontrar los objetos que se encontraban en la aplicación, junto a las relaciones que tenían entre ellos. Durante la realización del análisis se realizaron diferentes modelos que correspondían a las diferentes formas de consolidación de la información, así como el análisis de información operativa

- ✓ Estructura organizacional
- ✓ Resumen operativo
- ✓ Estadísticas

Modelo de objetos - Estructura organizacional

Grupo GEO se divide en delegaciones que a su vez se subdivide en Compañías. Para el control organizacional de las delegaciones y compañías, junto a su uso posterior para la consolidación de información, se contempla este modelo.

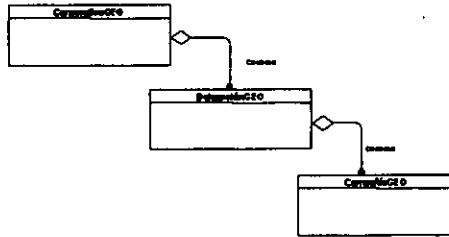


Figura 88 Diagrama de objetos - Estructura organizacional

Modelo de objetos - Resumen operativo

Los resúmenes operativos son aquellos que son resultado de la operación de la empresa, reflejando los resultados en cada una de las áreas, así como la información consolidada de los clientes, expedientes que llevan una tramitación, adquisición de vivienda, en las diferentes etapas.

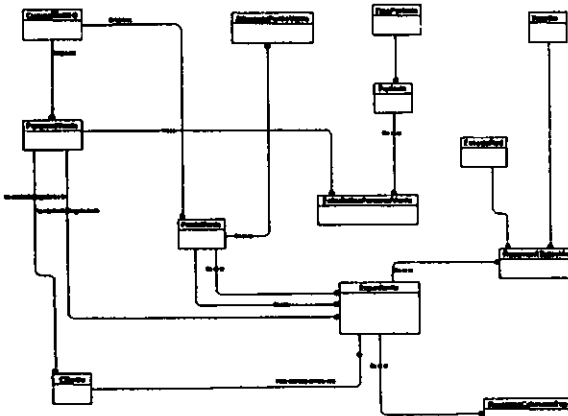


Figura 89 Diagrama de objetos - Resumen operativo

Modelo de objetos - Estadísticas

Las estadísticas son la primera información requerida por el ámbito ejecutivo directivo a cualquier nivel de Grupo GEO. Dentro de éstas es necesario tener periodos de tiempo bien definidos para la comparación de la información, junto a una definición de periodicidad a revisar la información (Tipos de periodo semestral, mensual, quincenal, semanal, etc), para medir avances de resultados con respecto a un estimado y periodos anteriores. Para el manejo de estadísticas de diferentes periodos, junto a la información estimada al inicio de año, se contempla este modelo.

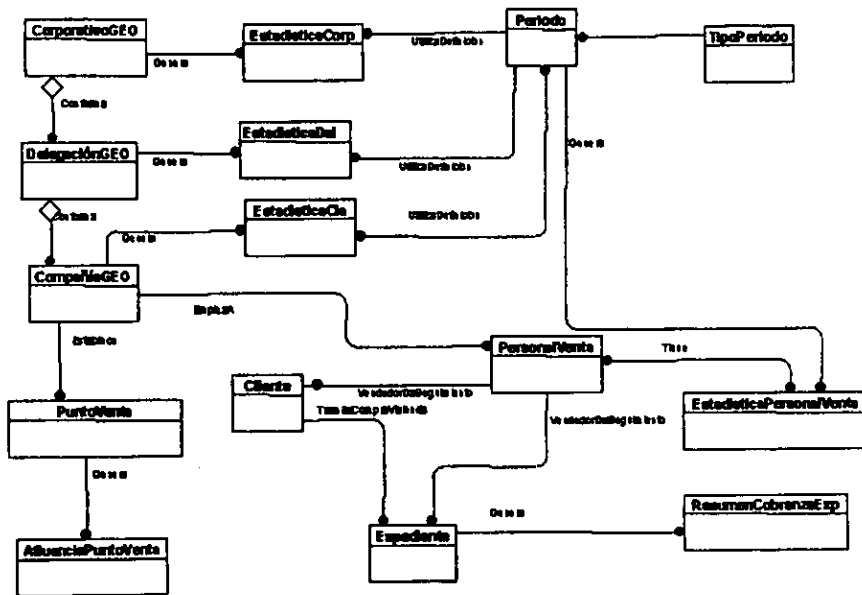


Figura 90 Diagrama de objetos – Estadísticas

Modelo dinámico

En el ámbito ejecutivo directivo, los modelos dinámicos encontrados se limitan al cálculo de información y su explotación.

Durante el análisis, se encontraron los siguiente modelos, con la reserva que tal módulo se encontraba en fase de desarrollo, pues los módulos de explotación de información fueron considerados como una segunda fase dentro del desarrollo del sistema. Sin embargo, los modelos aquí mostrados, ofrecen la explotación de información más importante del Grupo, al ser la que refleja los resultados obtenidos de la operación.

Escenarios

- ✓ Generación de resumen operativo
- ✓ Generación de Estadísticas de compañía, delegación y corporativo.

Escenario Generación de resumen operativo

En la generación de resumen operativo, se obtienen las diferentes consolidaciones consideradas en el modelo de objetos, como son la estadística del personal de ventas (afluencia atendida, ventas por vendedor, escrituraciones por vendedor), y la afluencia de los puntos de venta.

Secuencia de eventos Generación de Resumen operativo

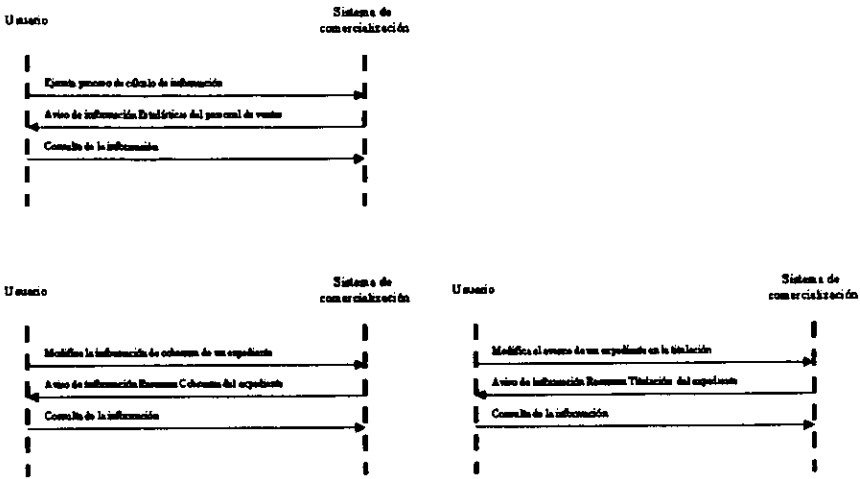


Figura 91 Secuencia de eventos – Generación de resumen operativo

Escenario Generación de estadísticas

La principal información necesaria para el control del Corporativo de Grupo GEO se basa en los resultados obtenidos en ventas y escrituraciones alcanzadas por las compañías y delegaciones, con el objeto de garantizar el correcto funcionamiento de las áreas comerciales.

**Secuencia de eventos
Generación de Estadísticas de compañía, delegación y corporativo**

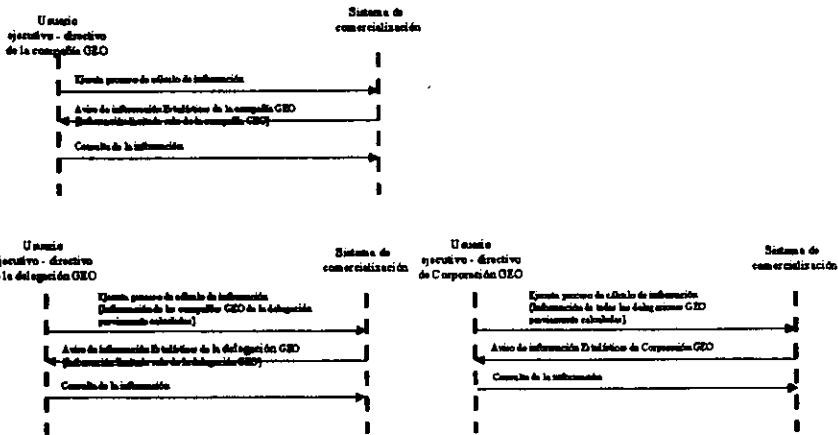


Figura 92 Secuencia de eventos – Generación de estadísticas de compañía, delegación y corporativo

Diagrama de estados

A partir de las secuencias de estados se prosiguió a realizar el análisis para obtener los diagramas de estados de los objetos más importantes, con el objeto de tener ampliamente definido el modelo dinámico del sistema.

De una serie de escenarios se obtiene el diagrama de estados de la generación de información ejecutiva y directiva con el fin de tenerla disponible para su consulta.

- ✓ Generación de información

Generación de información consolidada

La generación de la información consolidada se logra mediante sumalizaciones directas sobre la información operativa, relacionándola al nivel necesario de compañía, delegación, o corporativo.

Diagrama de estados Generación de Información consolidada

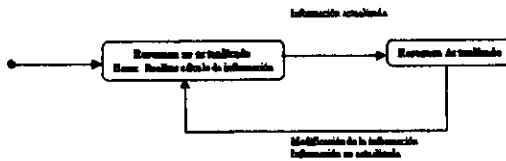


Figura 93 Diagrama de estados Generación de información consolidada

Modelo funcional

El modelo funcional indica la forma en que se consolidará la información, tanto en el ámbito de la compañía, como de las delegaciones y corporativo. En este caso, tal diagrama es muy simple y se decidió no incluirlo

Diseño del sistema

Diseño de los procesos y flujos de información

Los procesos diseñados para las empresas fueron realizados sobre la base de realizar las herramientas de cómputo que ayuden al desarrollo del trabajo diario del personal operativo, con el fin de convertirse en un elemento de apoyo durante la operación. De esta manera, todos los procesos están fundamentados en los flujos de trabajo del negocio, sirviendo el sistema de comercialización de un repositorio de datos inteligente, que además mantener la información disponible y consistente, también conservara las reglas de información que el mismo negocio requiere. El diseño de los procesos principales y flujos de la información del sistema de comercialización se basa en el flujo de trabajo normal para la adquisición de una vivienda en cualquiera de las compañías del Grupo GEO.

Operación comercial

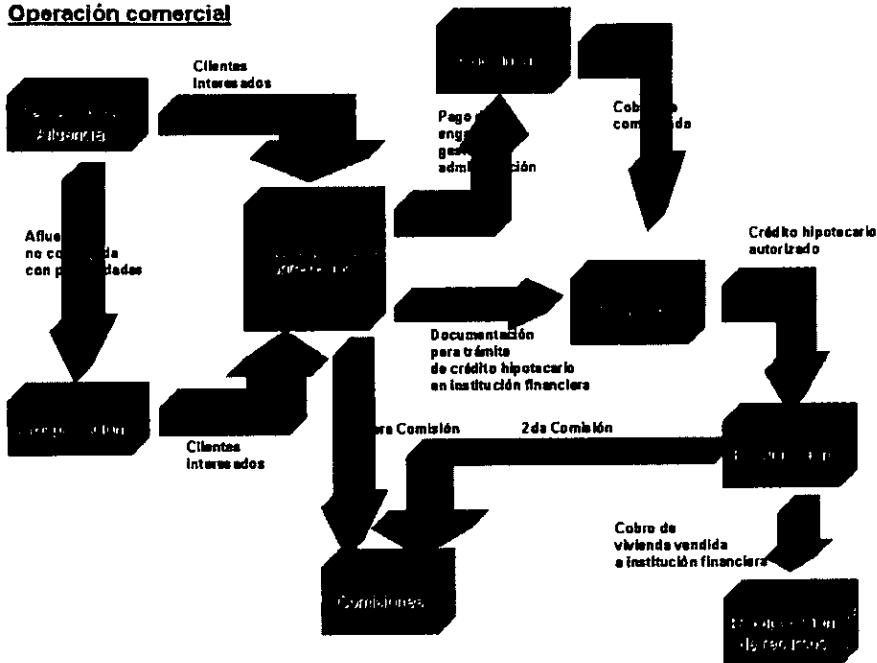


Figura 94 Operación comercial de una Compañía GEO típica

A partir de la recopilación de los requerimientos en las empresas, delegaciones y corporativo, se observó dos ámbitos de acción del sistema: el ámbito operativo y el ámbito ejecutivo – directivo.

Diseño de módulos

Ámbitos

El ámbito operativo se ocupa de todas las tareas propias de la operación del sistema; se ocupa de la serie de tareas a llevar a cabo por la naturaleza del negocio, ayudando al personal operativo en su trabajo, permitiéndole tener un repositorio de datos para su consulta posterior. Cumple generalmente con la necesidad de información de los casos particulares dentro del quehacer diario. El ámbito ejecutivo – directivo se ocupa de las necesidades de información para la toma de decisiones, comparativas estratégicas y información

general del grupo. En este caso, la información se utiliza como herramienta para basar las estrategias de comercialización del Grupo en datos, conociendo las ventas y escrituraciones realizadas, el análisis de afluencia en los puntos de venta, y el avance de los expedientes en titulación, que posteriormente se traducen en ventas y escrituraciones.

Ámbito empresarial operativo

Dentro de los ámbitos descritos arriba, cada uno de ellos se estructuraron mediante módulos, los cuales correspondían a uso dentro de la aplicación del sistema. De esta forma, el ámbito operativo se estructuró mediante cinco módulos básicos, en los cuales se realizaba conforme al flujo natural del negocio, con funciones específicas dentro del proceso, y el manejo distribuidos de los datos en cada uno de los punto remotos, en una base de datos distribuida.

Sistema de comercialización

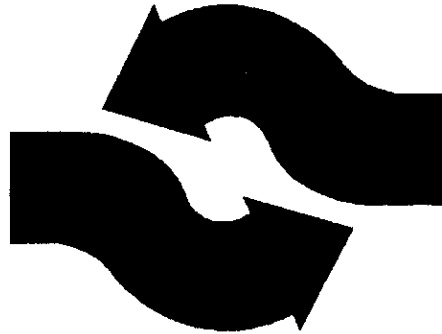


Figura 95 Sistema de comercialización Ámbitos de trabajo

Sistema de comercialización

Ámbito empresarial operativo



Figura 96 Módulos del sistema de comercialización

Acceso de los módulos del sistema de comercialización a la base de datos distribuida

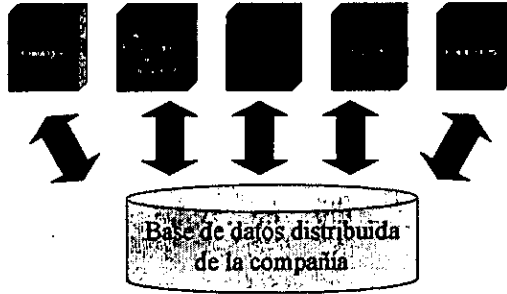


Figura 97 Acceso a la base de datos

Los módulos accesan a esta base de datos en cualquiera de los puntos en que se encuentren, ya sea en el punto central de la compañía o en los puntos remotos de venta u obra.

Catálogos

Sistema de comercialización



Catálogos

- Corporativo, Delegaciones y empresas
- Puntos de venta, desarrollos constructivos
- Afluencia, prospección y clientes
- Contratación (Contratos, planes de venta, planes de pago)
- Cobranza (Cajas, conceptos CXC, formas de pago)
- Trámites (Estados, recibos, documentos)
- Comisiones (conceptos CXP, porcentajes, cobros)

El módulo de catálogos se encarga de realizar toda la configuración necesaria para el funcionamiento del sistema. En este se dan de alta los catálogos de corporativo, delegaciones y compañías, junto a otros de uso general para todo el sistema.

Figura 98 Módulo de Catálogos

Afluencia, prospección y clientes. Contratación

Sistema de comercialización



Afluencia, Prospección, Clientes, Contratación

- Captación de afluencia
- Prospección de clientes
- Captación de clientes (desde afluencia o prospección)
- Contratación (Expedientes, contratos, planes de venta y pagos)
- Autorización de contratos (de planes de venta y pagos)
- Emisión de contratos
(Contratos y pagarés, afectación a cobranza, titulación y comisiones)
- Selección de vivienda probable

Figura 99 Módulo de Afluencia, prospección y clientes.
Contratación

como contrataciones especiales, incluyendo la autorización e impresión del contrato y pagarés, con la afectación a Cobranza, Titulación y Comisiones.

Cobranza

Sistema de comercialización



Cobranza

- Caja
- Estados de cuenta
- Negociación de diferencias
- Pronóstico de cobranza
- Antigüedad de saldos
- Avisos automáticos de falta de pago
- Reporte de cobranza por tipo de trámite

Figura 100 Módulo de cobranza

institución financiera y el estimado en el inicio de la contratación. Emite así mismo la liberación de la cobranza para su posterior uso en la escrituración.

El módulo de afluencia, prospección y clientes se encarga de cubrir los requerimientos de captación de datos de estas actividades, así como permitir la configuración de los catálogos respecto a medios publicitarios, tipos de ingreso, etc. para el posterior análisis de la información. Junto a esto, también aquí se contemplan la configuración de los planes de venta y pagos para su posterior uso en la contratación, la cual incluye tanto contrataciones preestablecidas

El módulo de cobranza, junto al de titulación es uno de los más grandes dentro del sistema. En este se realizan todas las actividades de caja, como son los cobros por los diferentes conceptos a clientes así como los pagos por BNC, manejo de multimonedas, aplicación de pagos a pagarés, etc. Conjunto a estas actividades, también cubre el siguiente proceso en la etapa de autorización del crédito, negociando las diferencias entre el crédito real otorgado por la

Titulación

Sistema de comercialización



Titulación

- Asignación de proceso de trámite
- Seguimiento de requisitos y documentos
- Avisos automáticos para actualización de exp.
- Avance de expediente en trámite
- Autorización de créditos hipotecarios
- Asignación de vivienda definitiva
- Seguimiento y escrituración (Afectación a comisiones)
- Recuperación de recursos

Figura 101 Módulo de Titulación

vivienda y recuperación del crédito por parte de GEO por la institución financiera, con la afectación a comisiones. Este módulo se vuelve fuente de información para la *información ejecutiva*, sistema explicado posteriormente.

Comisiones

Sistema de comercialización



Comisiones

- Revisión de comisiones
- Estados de cuenta
- Pronóstico de comisiones futuras
- Antigüedad de comisiones no cobradas

Figura 102 Módulo de Comisiones

de cada comisionista e información consolidada de todo el manejo de comisiones.

El módulo de titulación realiza las actividades de inicio y seguimiento de trámites ante la institución financiera a la cual se le solicita la autorización del crédito. El proceso integra la solicitud de los requisitos y documentación al cliente para el armado de un expediente y su ingreso a la institución financiera, seguimiento de los pasos en la revisión del expediente, la posterior notificación del crédito autorizado, escrituración, entrega de

El módulo de comisiones se encarga, como su nombre lo indica, de todo el manejo de comisiones por pagar y pagadas a los vendedores y promotores de la venta de casas, junto a la estructura comercial que se le considera para comisiones, tomando en cuenta los diferentes tiempos en los que se va avanzando una venta a escrituración, para el pago parcial de las comisiones, así como el estado de cuenta

Ámbito ejecutivo – directivo

En el ámbito ejecutivo – directivo, de la misma manera que en el ámbito empresarial operativo, se estructuró en módulos de acuerdo a las necesidades de análisis de información. Los procesos empresariales operativos son la base de información para cualquier análisis posterior dentro del Grupo GEO. Las diferentes áreas de las compañías, debido a su ejercicio, generan información de las operaciones particulares de venta, trámites, etc., de tal manera que es posible crear medios para la consolidación de esta información, mediante los cuales es posible tener información general en los ámbitos empresarial, delegacional y corporativo. A este segundo ámbito de información ejecutiva – directiva se le denominó *Información ejecutiva*, dado el perfil de usuarios que tendrían uso de esta herramienta.

Sistema de información ejecutiva

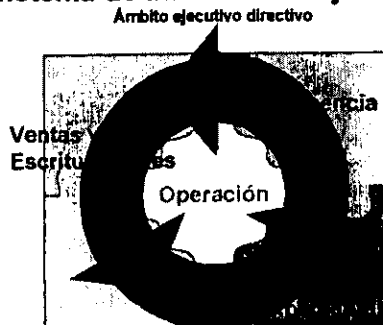


Figura 103 Sistema de comercialización Información ejecutiva

En este segundo ámbito de información ejecutiva – directiva se le denominó *Información ejecutiva*, dado el perfil de usuarios que tendrían uso de esta herramienta.

Módulos del sistema de información ejecutiva

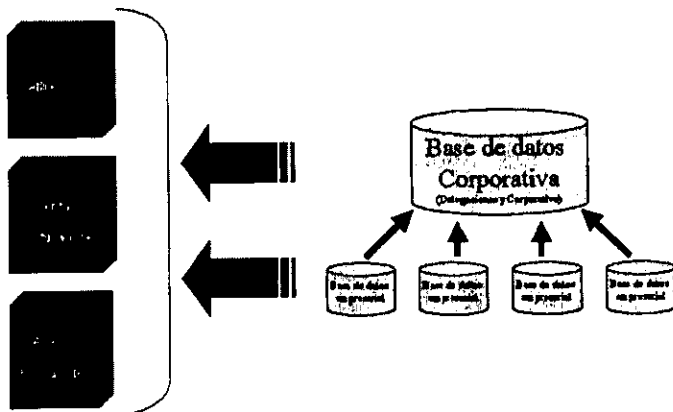


Figura 104 Módulos del sistema de información ejecutiva

Durante el desarrollo del sistema y por las estructuras de comunicaciones en las cuales se apoya el sistema de comercialización, se definió que las consultas de información consolidada en la empresa sería en la base de datos de la empresa y las consultas de información consolidada por delegación y corporativo, serían directamente en el corporativo, dada la forma de trabajo de estos usuarios.

Los módulos accesan a la información por medio de un punto central en la compañía en el caso de información empresarial, y mediante un acceso remoto para la información delegacional y corporativo, residiendo tal información en las oficinas generales del corporativo

Grupo GEO se encuentra organizado mediante un Corporativo que agrupa delegaciones y estas delegaciones a su vez agrupan compañías. Esto da como resultado una limitación al acceso de los datos con el fin de permitir la consulta de información solo a las personas autorizadas a ello, sin comprometer la flexibilidad del sistema, permitiéndolo un cambio rápido del tipo de consulta al usuario con nuevos permisos.

Afluencia

El módulo de Afluencia se encarga del análisis completo del tipo de persona que solicita información para la adquisición de una vivienda, evaluando la efectividad de medios publicitarios, tipo de ingresos (asalariado, por honorarios) y tipo de economía (formal o informal), de los afluentes, con el fin de conocer el perfil de la persona a la cual que atraen los mensajes, conformando una herramienta de estrategia publicitaria, reforzada con el mercadeo telefónico. El módulo maneja el esquema empresarial, junto al delegacional y corporativo.

Sistema de información ejecutiva



Afluencia

- Información de corporativo, delegaciones y empresas**
- Volumenes de afluencia**
(medios publicitarios, ingresos, economía)
- Gráficas de tendencia de afluencia**
(medios publicitarios, ingresos, economía)
- Proyecciones de afluencia**
(medios publicitarios, ingresos, economía)

Figura 105 Sistema de Información ejecutiva Módulo de Afluencia

En este módulo se incluyen gráficas de comparativas de medios, volumenes de afluencia por empresa, delegación y corporativo, gráficas de tendencias de los medios y proyecciones a futuro tanto de la afluencia total así como por medio publicitario.

Ventas y Escrituraciones

El módulo de ventas y escrituraciones se conforma del análisis completo de las ventas, escrituraciones, cancelaciones, y reposiciones (una forma especial de venta), evaluando la efectividad del cumplimiento del presupuesto anual en ventas y escrituraciones, junto a la revisión de las cancelaciones, dado que un volumen alto de cancelaciones implica una mala atención a los clientes por parte de las áreas involucradas. La opción presentada

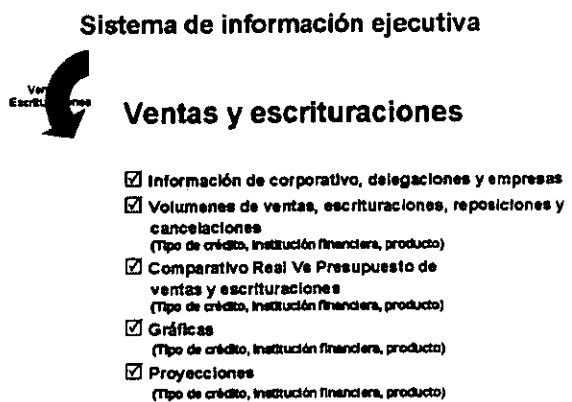


Figura 106 Sistema de Información ejecutiva Módulo de Ventas y Escrituraciones

de análisis por institución financiera, tipo de crédito y producto, informa de la efectividad de las negociaciones con una institución financiera, el tipo de crédito (FOVI o INFONAVIT), y el tipo de producto (Veces salario mínimo) que se está colocando en el mercado. El módulo maneja el esquema empresarial, junto al delegacional y corporativo.

Aquí se tienen la información tabular como gráfica entre presupuesto y real, también por tipo de crédito, institución financiera y producto, así como proyecciones a futuro para la revisión de las expectativas de cumplimiento de las empresas, delegaciones y corporativo.

Avance de expedientes

El módulo de avance de expedientes es un módulo intermedio entre los ámbitos empresarial operativo y el ejecutivo – directivo, ya que tiene una información media entre ambos ámbitos. Informa del avance de los expedientes a través del tiempo en la empresa, delegación o corporativo, para la asistencia corporativa ante las instituciones



Sistema de información ejecutiva

Avance de expedientes

- Información de corporativo, delegaciones y empresas
- Gráficas de volúmenes de expedientes por estado de red
- Futuras ventas probables
- Futuras escrituraciones probables
- Futuras reposiciones probables

Figura 107 Sistema de Información ejecutiva Módulo de Avance de expedientes

financieras de carácter nacional, proporcionando los medios para controlar a las áreas funcionales de titulación directamente desde la delegación o corporativo. Así mismo permite realizar estimaciones de las ventas y escrituraciones a corto, mediano y largo plazo, de acuerdo al informe de avance de los expedientes, de acuerdo a la experiencia y capacidad de los ejecutivos de delegación y corporativos.

Diseño de bases de datos

El diseño de la base de datos distribuida se realizó de acuerdo a los objetos encontrados dentro del dominio de la aplicación, junto a aquellos necesarios en el dominio del cómputo para una implementación eficiente del sistema. Dado que el acceso principal es el acceso persistente al mismo conjunto de datos, la base de datos es la forma apropiada de implementación. La principal enfoque en la base de datos es la estructura y limitaciones en los datos. Los comandos en una base de datos típicamente operan sobre un conjunto de datos dentro de la base, incorporando un grado de paralelismo, mientras que la mayoría de los lenguajes son seriales. Las bases de datos proveen operaciones concurrentes sobre los datos a diferentes usuarios como parte de su estructura fundamental.

El enfoque orientado a objetos es versátil. No solo provee de una base sólida para el diseño de sistemas y código de programación, sino puede ser utilizado para el diseño de las bases de datos. Los diseños son eficientes, coherentes y menos propensos a los problemas de actualización que suceden en otras técnicas de diseño de bases de datos. Como un beneficio adicional, el uso de una técnica de diseño uniforme mejora la integración entre la base de datos y el código de programación.

Arquitectura de triple esquema para Modelo de objetos.

En la figura 108 se muestra como el modelado de objetos se relaciona con la arquitectura de triple esquema. Primeramente, formulamos los objetos para un esquema externo y un esquema interno. Las vistas o programas de interfaz comunican al modelo externo con el interno. Después se traslada cada objeto a tablas ideales, esto es, a un modelo de tablas conceptuales. Las tablas conceptuales se trasladan a un esquema interno.

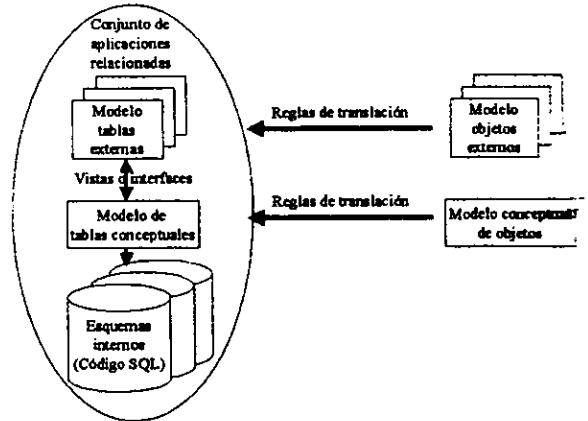


Figura 108 Modelado de objetos y arquitectura de triple esquema para el RDBMS

Cada modelo de tablas consiste de varias tablas ideales. Son genéricas e independientes del DBMS. El modelo de tablas conceptuales se desliga de las características intrínsecas del DBMS, pasando del modelo de objetos a reglas de correspondencia en el modelo de tablas.

Uso de identificadores para los objetos

En la mayoría de los objetos implementados en el sistema hacen uso de ID como claves primarias. Toda tabla derivada de una tabla tiene una ID como clave primaria; las ID de uno o más objetos forman la clave primaria para las tablas derivadas de la asociación.

Correspondencia entre objetos y tablas

Toda clase le corresponde una o más tablas. La clase *Empresa externa* tiene como atributos la razón social, el teléfono y un comentario. El modelo conceptual de tablas lista estos atributos, y le agrega el atributo *ClaveEmpresa*. Como parte del diseño, se agregan detalles. Se especifica que la *ClaveEmpresa* no puede ser nula, al igual que *RazonSocial*; una razón social debe existir por cada registro en *EmpresaExterna*.

El código SQL crea la Tabla *EmpresaExterna*. Se agrega también un índice por *RazonSocial* para asegurar un acceso rápido por este atributo por el cual es frecuentemente consultada. Así mismo el SQL mapea los diferentes dominios en tipos de datos.

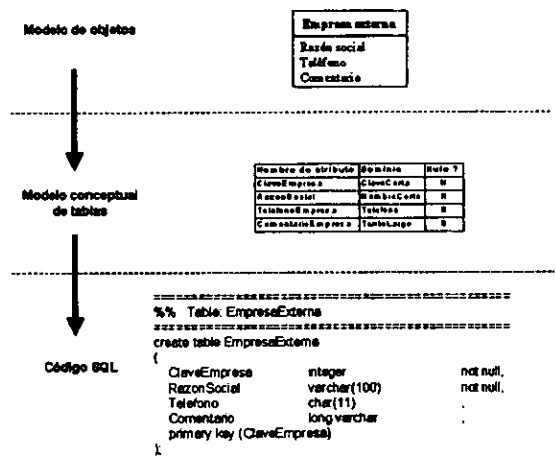


Figura 109 Mapeo de un objeto a una tabla

Mapeo de asociaciones a tablas

En una asociación puede o no mapear a una tabla. Depende generalmente del tipo y multiplicidad en la asociación. En figura XX se muestra una relación muchos a muchos, tipo de relación que siempre mapea a una tabla distintiva. El esquema satisface la tercera forma normal. Las llaves primarias de las dos clases relacionadas se convierten en atributos de la tabla asociativa. Los atributos *ClaveEmpresa* y *ClavePromotor* se combinan junto con *FechaVisita* para formar la llave primaria de la tabla

VisitaPromotor. En general, la asociación puede ser utilizada con EmpresaExterna o con Promotor y ser accesada frecuentemente. Las sentencias de llaves foráneas en el código SQL indican que cada registro de VisitaPromotor debe tener una referencia en un EmpresaExterna y a un Promotor que deben a su vez estar definidas en sus tablas respectivas.

Una asociación siempre cambia sus llaves foráneas de los objetos referidos a no nulos; por definición, una liga entre dos objetos requiere que ambos objetos se conozcan.

En una relación uno a muchos, existen dos formas de crear la relación. En

la primera se genera una tabla con las llaves primarias de las tablas y se agregan los atributos extra. En la segunda se incluye los atributos de la relación en el lado de Muchos y se hereda la llave del lado de uno.

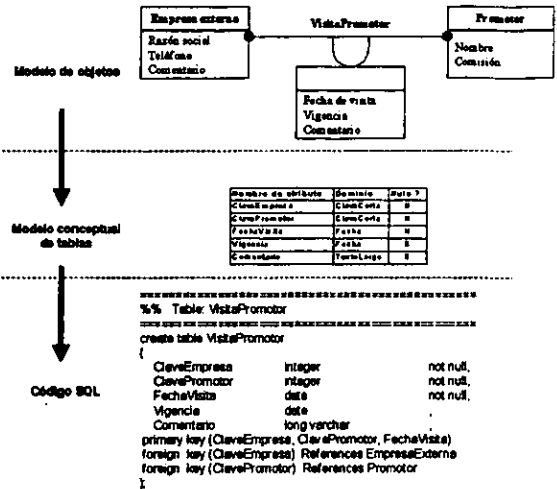


Figura 110 Mapeo de una relación muchos a muchos

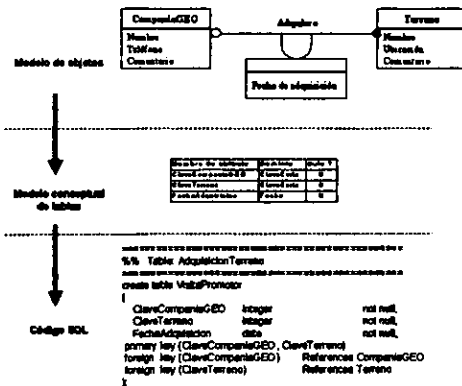


Figura 112 Mapeo uno a muchos - tabla generada

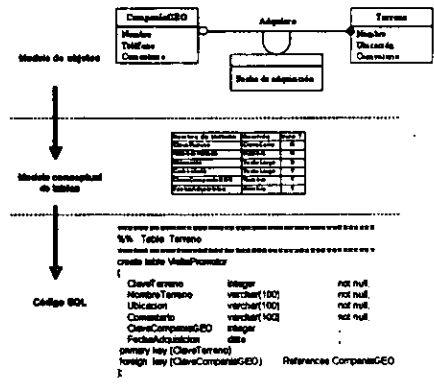


Figura 111 Mapeo uno a muchos - atributos heredados

Las ventajas de heredar los atributos a una tabla son:

- ✓ Menor numero de tablas
- ✓ Mejor actuación debido al menor numero de tablas a navegar

Las desventajas son

- ✓ Menor rigor en el diseño. Las asociaciones son objetos independientes con un mismo peso. En general vemos inapropiado contaminar a los objetos con los atributos heredados de relaciones.
- ✓ Reduce la posibilidad de crecimiento. Es difícil obtener la multiplicidad correcta en los primeros pasos de diseño. Una asociación uno a uno o uno a muchos puede ser aumentada.
- ✓ Mayor complejidad. Una representación asimétrica de la asociación complica la búsqueda y la actualización de datos.

Mapeo de generalización a tablas

En el mapeo de generalización a tablas, existen 4 formas de generalización, aunque para nuestro sistema solo ocupamos uno. En la generalización se tienen dos clases, la superclase persona, y las subclases Afluencia, Cliente, Prospecto, Coacreditado, etc.

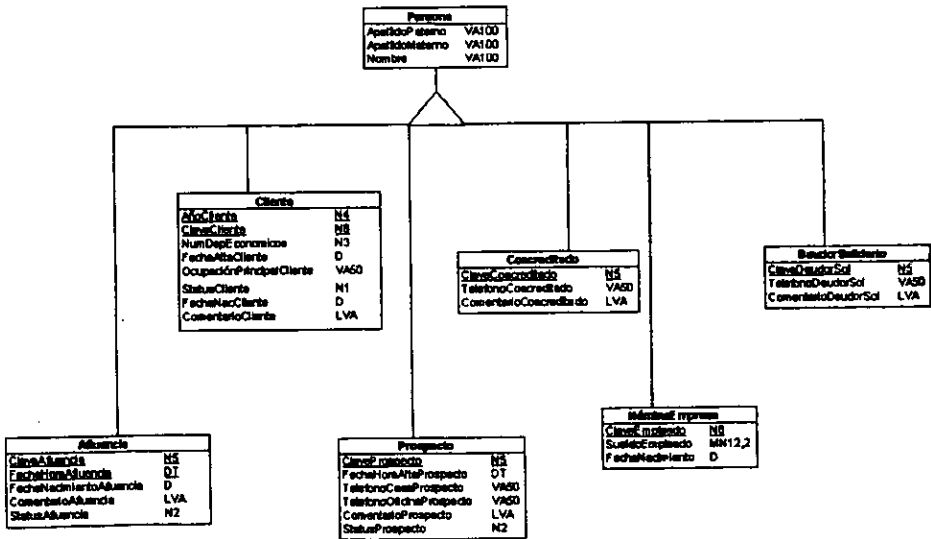


Figura 113 Modelo de objetos de la generalización de personas

El enfoque utilizado se denomina "muchas subclases". Este enfoque elimina a la tabla de la superclase y duplica todos sus atributos a cada tabla subclase. Se usa este enfoque cuando la subclase tiene muchos atributos, y la superclase tiene pocos atributos, junto a que la aplicación conoce que subclase debe buscar. Este enfoque cumple con la tercera forma normal, pero tiene una implementación un poco mas pobre que otras utilizadas. No se puede garantizar que cada nuevo nombre de persona ya se encuentra capturado en la base de datos, dado que el RDBMS no proporciona un mecanismo contra la duplicidad entre diferentes tablas.

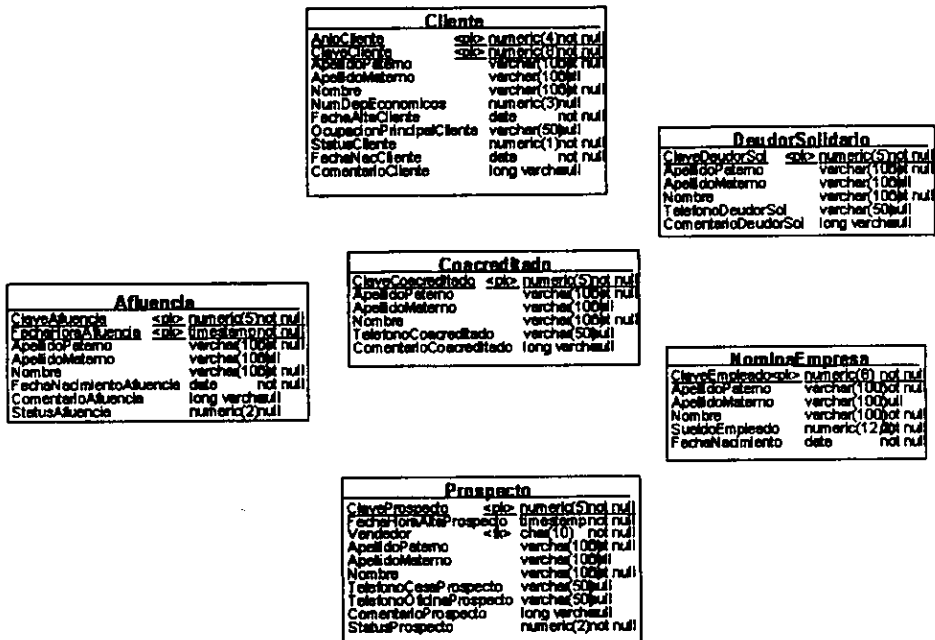


Figura 114 Mapeo de generalización - Enfoque "Muchas subclases"

Cientes

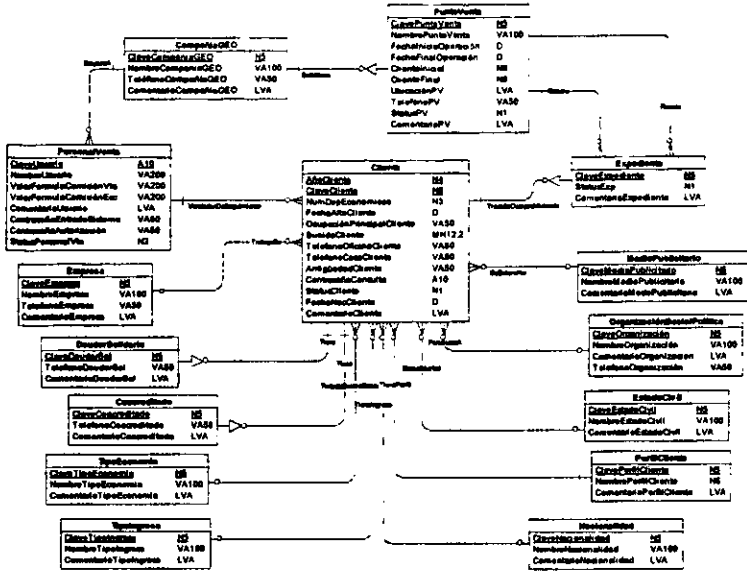


Figura 120 Modelo lógico - Clientes

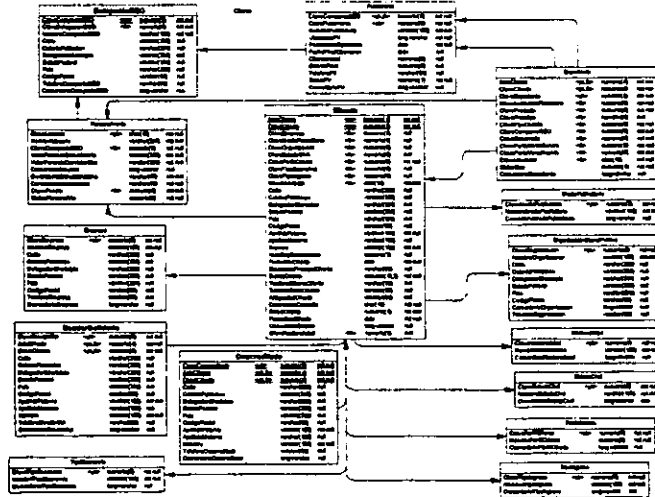


Figura 119 Modelo físico - Clientes

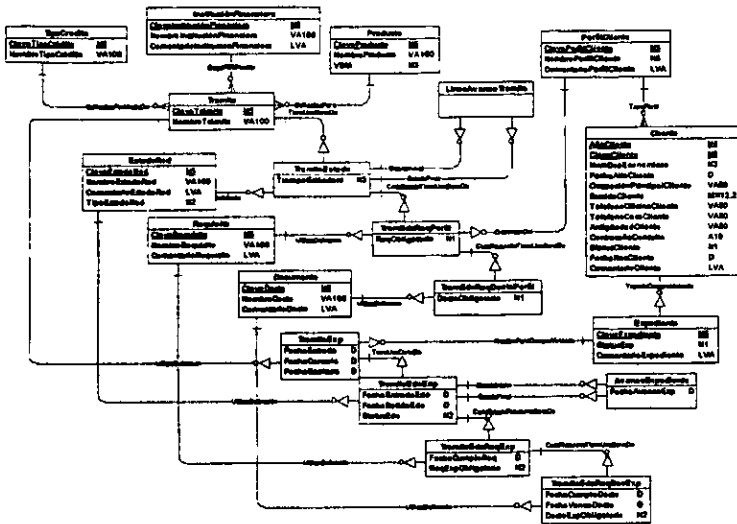


Figura 124 Modelo lógico - Titulación

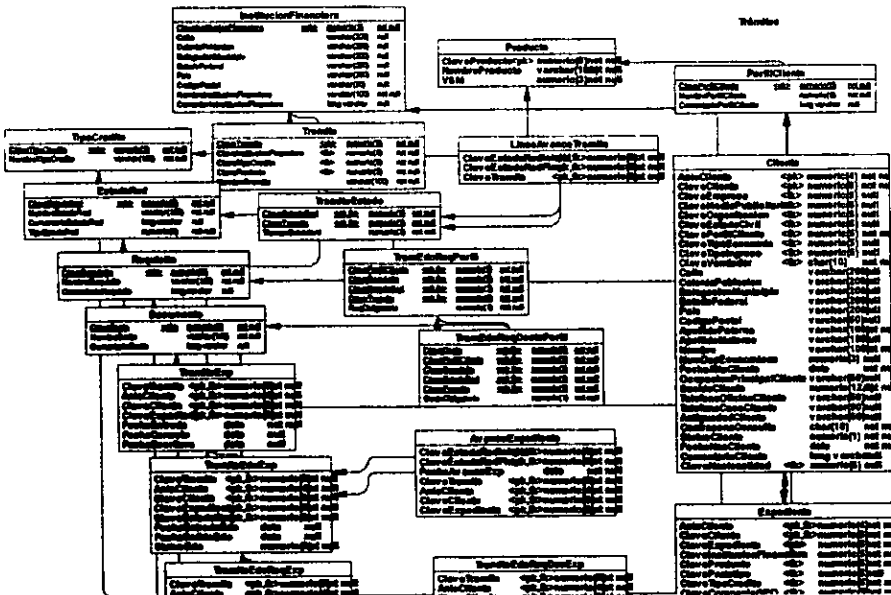


Figura 123 Modelo físico - Titulación

Diseño de interfaz

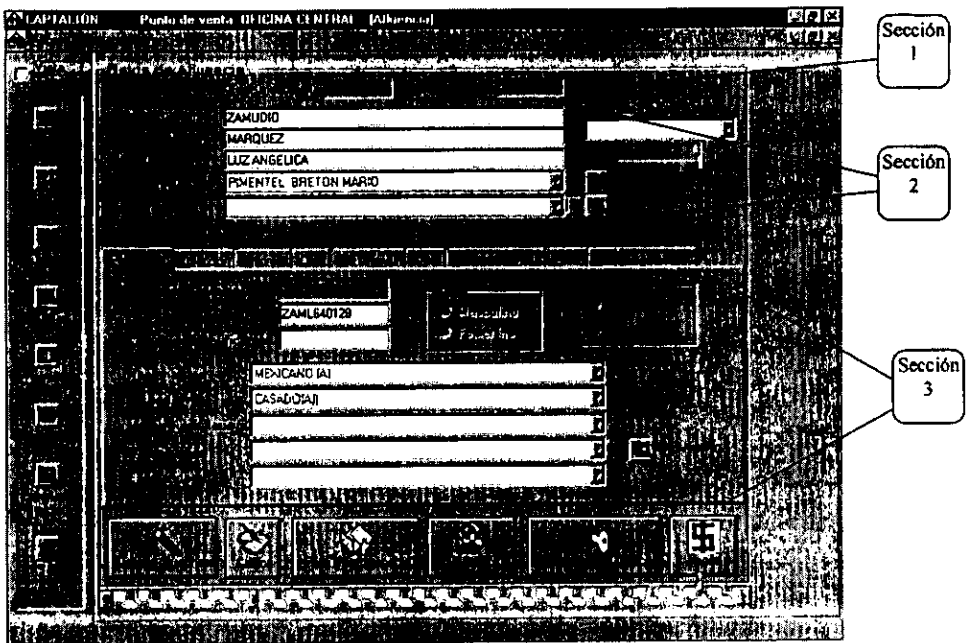
En las siguientes páginas se mostrarán algunas de las pantallas del sistema, las cuales siguen el proceso normal de captación, contratación, cobranza y titulación de una escrituración.

Captación

Dentro de captación se realizará la captura de los datos principales de toda persona que vaya a adquirir una vivienda o únicamente a pedir información acerca de la misma.

Es dividida en dos submenús : Afluencia, Cliente.

Afluencia



El propósito de este menú es realizar la captura de toda persona que llegan por medio de cualquier medio publicitario a solicitar información acerca de las viviendas.

Es importante tener capturados la mayor parte de la información. En caso de carecer de algún dato importante, el sistema manda un aviso en el cual se especifica la información obligatoria que necesita el sistema para poder generar la inserción.

Los 6 principales datos que solicita esta pantalla son :

- ✓ 1. Apellido paterno
- ✓ 2. Nombre
- ✓ 3. Nacionalidad
- ✓ 4. Medio publicitarios
- ✓ 5. Sexo
- ✓ 6. País
- ✓ 7. Estado

Se divide en 3 secciones:

- ✓ Sección 1.- Son campos que el sistema proporciona por default (hora y fecha), da como posibilidad de cambio ambos campos, esto con el fin de poder editar la hora y fecha para tener lo real de la visita y no los datos de captura.
- ✓ Sección 2.- Es aquella en donde se captura el nombre del cliente probable. La inserción se puede llevar a cabo comenzando por el apellido paterno, materno (opcional) y, nombre. Se procede a solicitar al comisionista y el promotor (en caso de existir) los cuales son seleccionado mediante lista desplegable.
- ✓ Sección 3.- Es dividida en 10 pestañas de trabajo donde es almacenada la información general de la afluencia:

Personales

ALEGRE
NAJERA
ALEJANDRO

AENAG50413

MEXICANO (A)
SOLTERO (A)
GALLARDETES
CTM
CARRERA PROFESIONAL Y SUPERIOR

Esta área captura los datos personales correspondientes al prospecto a cliente; dentro de la organización, proporciona una lista de valores la cual muestra la información activando el botón de buscar ubicado a un costado de dicho campo.

El RFC lo calcula automáticamente en base a la fecha de nacimiento anteriormente capturada.

Dirección

ALEGRE
NAJERA
ALEJANDRO

MEXICO
DISTRITO FEDERAL

MAQDALENA CONTRERAS LA
CERRO DEL JUDIO
CERRO DEL JUDIO #57

015 0759548

Dentro de Dirección se realizará la captura de la ubicación de la vivienda proporcionada por el prospecto a cliente.

Para hacer más eficiente esta área la captura se realiza de la siguiente forma:

- ✓ En caso de que no tener el Código Postal el Sistema por medio de SEPOMEX lo inserta automáticamente después de haber seleccionado el País, Estado, Delegación y la Colonia o población para finalmente capturar la calle y número, independientemente del teléfono.
- ✓ Si se tiene únicamente el Código Postal, se procede a seleccionar el País correspondiente y a capturar el código, de esta manera el sistema desglosa el resto de la información para finalizar con la captura de la calle, número, lada y teléfono.

Empresa

ALEGRE
NAJERA
ALEJANDRO

ACTUARIO
30807505348 0
9526 1 AÑO 3 MESES
ANALISTA FINANCIERO
NISSAN MEXICANA S.A. DE C.V.
015 6597421

Dentro de esta área se capturaran todos los campos que en ella existen. Es importante dar el número de IMSS debido a que en base a este dato se calculan las aportaciones hechas a dicha institución.

Para cargar las pantallas mostradas con anterioridad se realiza el siguiente QUERY el cual realiza un SELECT de la información consultada

```

SELECT
A.ClaveAfluencia, A.ApellidoPaterno, A.ApellidoMaterno, A.Nombre, DateFormat(A.FechaVisita, 'DD/MM/YYYY' ),
DateFormat(A.HoraVisita,'hh:mm:ss'),A.FechaNacimiento, A.Sexo, ARGN(A.Sexo, 'Masculino', 'Femenino'),
A.ClaveEstadoCivil,EC.NombreEstadoCivil,A.RFC,A.CedulaIdentificacion, A.ClaveMedioPublicitario, M.NombreMedioPublicitario, A.Ocupacion,
A.Sueldo, A.AfiliacionIMSS, A.LadaEmpresa, A.TelefonoEmpresa, A.ExtensionEmpresa, A.Puesto,
A.Antiguedad, A.ClavePais, P.NombrePais, A.ClaveEstado, E.NombreEstado, A.DelegacionMunicipio, A.ColoniaPoblacion,
A.CalleNumero, A.CodigoPostal, A.Lada, A.Telefono, A.ClaveComisionista,
(Com.Nombre + ' ' + Com.ApellidoPaterno + ' ' + Com.ApellidoMaterno), A.Com_ClaveComisionista,
(ComP.Nombre + ' ' + ComP.ApellidoPaterno + ' ' + ComP.ApellidoMaterno), IFNULL(A.Observacion, 'A.Observacion), A.ClaveNacionalidad,
N.NombreNacionalidad, A.ClaveEmpresa, EE.NombCortoRazonSocial, EE.RazonSocial, A.NumeroDependientes,
A.AportacionesINFONAVIT1, A.MontoSAR, A.CreditoMaximoVSMM, A.FechaCalifinonavit, A.PuntosFactor1, A.PuntosFactor2, A.PuntosFactor3,
A.PuntosFactor4, (A.PuntosFactor1 + A.PuntosFactor2 + A.PuntosFactor3 + A.PuntosFactor4 ) AS Total, A.AportacionesINFONAVIT2,
A.ClaveOrganizacion, NombreOrganizacion, A.Escolaridad, ARGN(Convert(Char(1),A.Escolaridad), 'PRIMARIA Y SECUNDARIA', 'CARRERA
SUBPROFESIONAL', 'CARRERA PROFESIONAL Y SUPERIOR'), A.ClavePuntoVenta, MontoCreditoP, FechaCalificaP, PuntosFactor1P,
PuntosFactor2P, PuntosFactor3P, PuntosFactor4P, (A.PuntosFactor1P + A.PuntosFactor2P + A.PuntosFactor3P + A.PuntosFactor4P ) AS
TotalP,Programa,Subas_Paq,Coacreditado, A.ClaveTipoPeriodo,PE.NombreTipoPeriodo, A.ClaveInstitucionFinanciera,InstF.RazonSocial,PagoRenta,
DateFormat( A.FechaSolicitudBC, 'DD/MM/YYYY' ), A.ResultadoBC,DateFormat( A.FechaResultadoBC, 'DD/MM/YYYY' ), A.NumReporteBC,
A.FolioBC,IFNULL(A.ObservacionBC, "A.ObservacionBC), A.FolioAutorizadoAfluenciaBC, A.NumReporteBC2, A.ResultadoBC2, A.FolioBC2,
DateFormat( A.FechaSolicitudBC2, 'DD/MM/YYYY' ),DateFormat( A.FechaResultadoBC2, 'DD/MM/YYYY' ), A.FolioAutorizadoAfluenciaBC2,
IFNULL(A.ObservacionBC2, "A.ObservacionBC2), A.NumReporteCoacreditado, A.ResultadoCoacreditado, A.FolioCoacreditado, DateFormat(
A.FechaSolicitudCoacreditado, 'DD/MM/YYYY' ),DateFormat( A.FechaResultadoCoacreditado, 'DD/MM/YYYY' ),
A.FolioAutorizadoAfluenciaCoacreditado, IFNULL(A.ObservacionCoacreditado, "A.ObservacionCoacreditado), A.NumReporteCoacreditado2,
A.ResultadoCoacreditado2, A.FolioCoacreditado2,DateFormat( A.FechaSolicitudCoacreditado2, 'DD/MM/YYYY' ), DateFormat(
A.FechaResultadoCoacreditado2, 'DD/MM/YYYY' ), A.FolioAutorizadoAfluenciaCoacreditado2, IFNULL(A.ObservacionCoacreditado2,
"A.ObservacionCoacreditado2), IFNULL(A.ClaveCalificacion, "A.ClaveCalificacion), IFNULL(RBC.Calificacion, "RBC.Calificacion),
IFNULL(A.ClaveCalificacion2, "A.ClaveCalificacion2), IFNULL(RBC1.Calificacion, "RBC1.Calificacion), IFNULL(A.ClaveCalificacionCoa,
"A.ClaveCalificacionCoa), IFNULL(RBC2.Calificacion, "RBC2.Calificacion), IFNULL(A.ClaveCalificacion2Coa, "A.ClaveCalificacion2Coa),
IFNULL(RBC3.Calificacion, "RBC3.Calificacion)
FROM
Afluencia A Left Outer Join EstadoCivil EC Left Outer Join MedioPublicitario M Left outer join organization O Left Outer Join Estado E
Left Outer Join Pais P, Comisionista COM, Comisionista ComP, Nacionalidad N, EmpresaExterna EE ,TipoPeriodo PE, InstitucionFinanciera InstF,
CalificacionBC RBC, CalificacionBC RBC1 , CalificacionBC RBC2 , CalificacionBC RBC3
WHERE
A.ClaveCorporativo = 1 AND A.ClaveDelegacion = 1 AND A.ClaveCompaniaGEO = 2 AND A.ClaveComisionista *= COM,
ClaveComisionista AND A.Com_ClaveComisionista *= ComP.ClaveComisionista AND A.ClaveNacionalidad *= N.ClaveNacionalidad
AND A.ClaveEmpresa *= EE.ClaveEmpresa AND A.ClaveInstitucionFinanciera *=InstF.ClaveInstitucionFinanciera AND
A.ClaveTipoPeriodo *=PE.ClaveTipoPeriodo AND A.ClaveCalificacion *= RBC.ClaveCalificacion AND
A.ClaveCalificacion2 *= RBC1.ClaveCalificacion AND A.ClaveCalificacionCoa *= RBC2.ClaveCalificacion AND
A.ClaveCalificacion2Coa *= RBC3.ClaveCalificacion ORDER BY A.ApellidoPaterno,
A.ApellidoMaterno, A.Nombre, A.ClaveAfluencia

```

Cliente

En el catálogo que se muestra a continuación se insertaran los datos personales de todas aquellas personas que se considere clientes.

The image shows a screenshot of a data entry form for a client. The form is divided into four sections, labeled on the right side as Sección 1, Sección 2, Sección 3, and Sección 4. The data entered in the form includes:

- Sección 1:** ACEVEDO
- Sección 2:** CASTRO
- Sección 3:** ALMA MINERVA
- Sección 4:** 0001 / A / Venta / FOMI / AHUATLAN III / HIPOTECARIA N.

Below these sections, there are several fields and dropdown menus:

- AED: 59Q920TL4
- MEXICANO(A) (dropdown menu)
- SOLTERO(A) (dropdown menu)
- CAMBACED
- CTM

There are also some icons and a small window titled 'Masculino' and 'Femenino' visible on the right side of the form.

En algunas ocasiones esta información será transferida desde Afluencia.

El objetivo de este catálogo es revisar y completar la información que proviene de afluencia,

Si el cliente no se registró con anterioridad se debe capturar toda la información requerida para comenzar la elaboración del expediente.

La finalidad de este catálogo es almacenar todos los datos necesarios del cliente para cualquier consulta.

Está dividido en 4 secciones:

- ✓ Sección 1.- Consta de una clave única y fecha de alta la cual es proporcionada por default. Dichos datos no constan de una probable posibilidad de cambio.

- ✓ Sección 2.- Se puede realizar la captura de 2 formas : Si el prospecto inmediatamente paso a cliente y no requirió registrarse dentro de afluencia debe darse de alta en éste catálogo anotando su nombre completo. Si llegará a existir comisionista y/o promotor proceden a ser seleccionados según corresponda. Consta de dos campos de información, en los que se pueden observar los datos principales del expediente (número de expediente, status, tipo de venta , promoción e institución financiera) y un campo por el cual se visualiza el nombre de la persona por la que fue atendido, la fecha, el número de días que han transcurrido desde la alta del cliente y número de expediente. Si incluye información capturada desde afluencia no necesita insertar dato alguno. El "comisionista y promotor" son asignados dentro de la pantalla de expediente. Dentro de esta misma sección se aprecia un campo nombrado "expediente", dicho campo aparece después de haberlo asignado. El campo nombrado "atendido por", se visualiza después de la inserción de información.
- ✓ Sección 3.- Esta sección contiene 2 áreas más que en el catálogo de afluencia : foto e INFONAVIT. Si se da de alta en éste catálogo el cliente, es decir, no fue enviado, necesitará la captura completa de los datos necesitados. Dentro de estas se requiere el complemento de la información en caso de haberla transferido de afluencia, debido a que en el catálogo de donde fue enviada no se detallan algunas partes que dentro de éste catálogo son necesarias.

A screenshot of a computer application form with several text input fields. The fields contain the following text from top to bottom:

- ALFORE
- NAGERA
- ALEJANDRO
- INVESTIGACIONES ALFONSO
- 0001 / A / POC / FOM / RINCORNADAS VILLAS MODO / BANAM
- INVEST ESCATEL LUIS ALFONSO 03/04/98 0
- MENCO
- DISTRITO FEDERAL
- MATEVALEN CONTRERAS LA
- CERRO DEL JUDIO
- CERRO DEL JUDIO 37
- 15
- 0758841

Proyecto - Cliente

ALEGRE
NAGERA
ALEJANDRO
REYES ESCATEL LUIS ALFONSO
0001 / A / PDD / FOM / RINCONADAS VILLAS XOXO / BANAM
REYES ESCATEL LUIS ALFONSO 03/06/98 0


ACTUARIO
2000 ASALARIADO
NISSAN MEXICANA S.A. DE C.V.
ANALISTA FINANCIERA
1 AÑOS 3 MESES
FORMAL
15 6697421
1
12934589754

OTROS EMPLEOS

Es un submenú en el que se captura la información completa de la empresa en la que esta elaborando, consta de los mismos campos de captura que el catálogo de Afluencia, excepto por la antigüedad.

En la parte inferior derecha se puede observar un botón llamado otros empleos.



Contiene una sección exclusiva para los clientes que tengan más de 1 fuente de ingresos. Dentro de este catálogo se lleva el control de los diferentes empleos que puede tener el cliente consta de un campo llamado empleo,  el cual sirve para controlar las diferentes ocupaciones que pudiera tener el cliente. Es un campo noeditable.

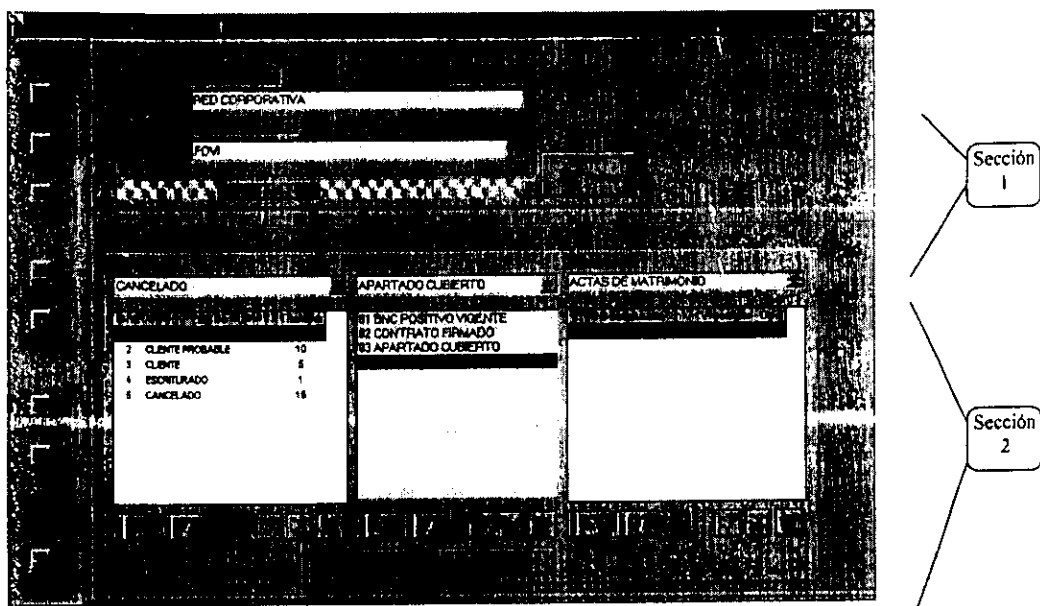
Para cargar las pantallas mostradas con anterioridad se realiza el siguiente QUERY el cual realiza un SELECT de la información consultada

```

SELECT DISTINCT
STRING(C.ClaveCliente, C.ApellidoPaterno, C.ApellidoMaterno, C.Nombre, C.Lada, C.Telefono, IFNULL(C.Observacion, "C.Observacion"), C.RFC,
C.CedulaIdentificacion, DateFormat(C.FechaNacimiento, 'DD/MM/YYYY'), C.ClaveEstadoCivil, EC.NombreEstadoCivil, C.Sexo, ARGN(C.Sexo,
'Masculino', 'Femenino'), " ", C.ClavePais, P.NombrePais, C.ClaveEstado, E.NombreEstado, C.DelegacionMunicipio, C.ColoniaPoblacion,
C.CalleNumero, C.CodigoPostal, C.ClaveMedioPublicitario, M.NombreMedioPublicitario, C.ClaveTipoEconomia, TE.NombreTipoEconomia,
C.ClaveOrganizacion, O.NombreOrganizacion, C.Password,
DateFormat(C.FechaAlta, 'DD/MM/YY'), C.Ocupacion, C.TipoCliente, ARGN(Concat(TinyInt, C.TipoCliente), 'Cliente', 'Prospecto'),
C.ClaveNacionalidad, N.NombreNacionalidad, C.Status, ARGN(Concat(TinyInt, C.Status+1), 'Cancelado', 'Activo', 'Terminado'),
C.Sueldo, C.ClaveTipoIngreso, TI.NombreTipoIngreso, C.ClaveEmpresa, EX.RazonSocial, C.Puesto, C.Antiguedad,
C.TelefonoEmpresa, C.LadaEmpresa, C.ExtensionEmpresa, C.AfiliacionMSS, C.NumeroDependientes,
C.AportacionesINFONAVIT1, c.MontoSAR, C.CreditoMaximoVSM, C.FechaCalifonavit, C.PuntosFactor1, C.PuntosFactor2,
C.PuntosFactor3, C.PuntosFactor4, (C.PuntosFactor1 + C.PuntosFactor2 + C.PuntosFactor3 + C.PuntosFactor4) AS Total,
C.AportacionesINFONAVIT2, IFNULL(C.LugarNacimiento, "C.LugarNacimiento"), C.NumReporteBC, C.ResultadoBC,
DateFormat(C.FechaSolicitudBC, 'DD/MM/YYYY'), C.FolioBC, DateFormat(C.FechaResultadoBC, 'DD/MM/YY'), C.FolioAutorizadoAfluenciaBC,
IFNULL(C.ObservacionBC, "C.ObservacionBC"), C.NumReporteBC2, C.ResultadoBC2, DateFormat(C.FechaSolicitudBC2, 'DD/MM/YYYY'),
C.FolioBC2, DateFormat(C.FechaResultadoBC2, 'DD/MM/YY'), C.FolioAutorizadoAfluenciaBC2, IFNULL(C.ObservacionBC2, "C.ObservacionBC2"),
C.NumReporteCoacreditado, C.ResultadoCoacreditado, C.FolioCoacreditado, DateFormat(C.FechaSolicitudCoacreditado, 'DD/MM/YYYY'),
DateFormat(C.FechaResultadoCoacreditado, 'DD/MM/YYYY'), C.FolioAutorizadoAfluenciaCoacreditado, IFNULL(C.ObservacionCoacreditado,
"C.ObservacionCoacreditado"), C.NumReporteCoacreditado2, C.ResultadoCoacreditado2, C.FolioCoacreditado2,
DateFormat(C.FechaSolicitudCoacreditado2, 'DD/MM/YYYY'), DateFormat(C.FechaResultadoCoacreditado2, 'DD/MM/YYYY'),
C.FolioAutorizadoAfluenciaCoacreditado2, IFNULL(C.ObservacionCoacreditado2, "C.ObservacionCoacreditado2"), IFNULL(C.ClaveCalificacion, "C.ClaveCalificacion"), IFNULL(RBC.Calificacion, "RBC.Calificacion"), IFNULL(C.ClaveCalificacion2, "C.ClaveCalificacion2"),
IFNULL(RBC1.Calificacion, "RBC1.Calificacion"), IFNULL(C.ClaveCalificacionCoa, "C.ClaveCalificacionCoa"), IFNULL(RBC2.Calificacion, "RBC2.Calificacion"), IFNULL(C.ClaveCalificacion2Coa, "C.ClaveCalificacion2Coa"), IFNULL(RBC3.Calificacion, "RBC3.Calificacion")
FROM
(Cliente C LEFT OUTER JOIN Estado E LEFT OUTER JOIN Pais P), Organizacion O, Nacionalidad N, TipoEconomia TE, MedioPublicitario M,
EmpresaExterna EX, EstadoCivil EC, TipoIngreso TI, Comisionista Com, Comisionista Ptr, CalificacionBC RBC, CalificacionBC RBC1,
CalificacionBC RBC2, CalificacionBC RBC3
WHERE
C.ClaveOrganizacion *= O.ClaveOrganizacion AND C.ClaveMedioPublicitario *= M.ClaveMedioPublicitario AND
C.ClaveEstadoCivil *= EC.ClaveEstadoCivil AND C.ClaveTipoEconomia *= TE.ClaveTipoEconomia AND
C.ClaveNacionalidad *= N.ClaveNacionalidad AND C.ClaveEmpresa *= EX.ClaveEmpresa AND C.ClaveCalificacion *= RBC.ClaveCalificacion
AND C.ClaveCalificacion2 *= RBC1.ClaveCalificacion AND C.ClaveCalificacionCoa *= RBC2.ClaveCalificacion AND
C.ClaveCalificacion2Coa *= RBC3.ClaveCalificacion AND C.ClaveTipoIngreso *= TI.ClaveTipoIngreso AND C.ClaveCorporativo=1 AND
C.ClaveDelegacion=1 AND C.ClaveCompaniaGEO=2 AND C.ClaveComisionista *= Com.ClaveComisionista
AND C.Com_ClaveComisionista *= Ptr.ClaveComisionista ORDER BY C.ApellidoPaterno, C.ApellidoMaterno, C.Nombre

```

Configuración de red corporativo



El objetivo de este catálogo es configurar las red de Titulación.

La imagen que se encuentra en la parte superior se va a dividir en 2 secciones (1 y 2), así mismo, en la sección 2 se verá un apartado: guardar como.

- ✓ **Sección 1.-** En esta parte se le da el nombre a la red como *Red Corporativa*, se captura el Nmonico y el tipo de crédito para el cual la red va a ser válida.
- ✓ **Sección 2.-** Esta sección se despliega en el momento de oprimir el botón de *Detalle*. Se debe de seleccionar el tipo de ingreso.

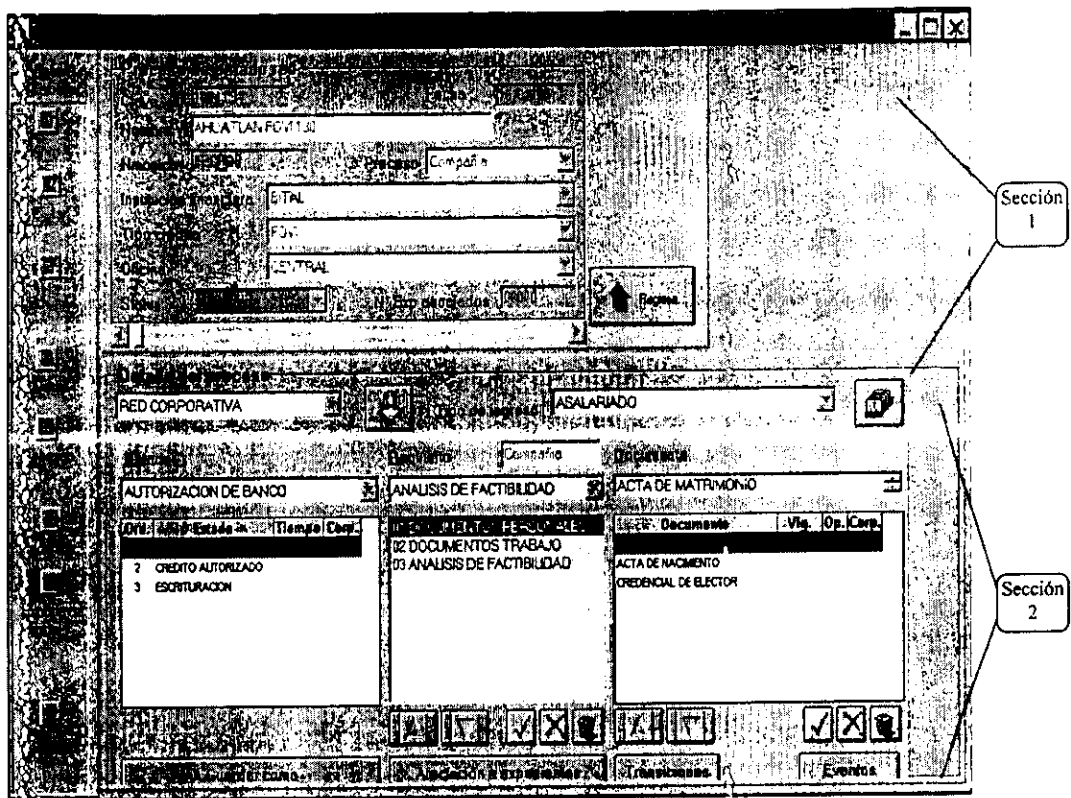
Estado.- Se refiere a los estados por los que debe pasar el expediente, como la información ya fue previamente capturada únicamente se seleccionará y se dará de alta para poder utilizar las áreas restantes que son requisito y documento.

Requisito.- Se van a seleccionar los requisitos que necesite cada estado, y se van a dar de alta, al escoger el requisito se procede a documento.

Documento.- Se van a tomar el o los documentos que sean necesarios para cada requisito antes seleccionado, se podrá mencionar su vigencia y si son opcionales.

Guardar como.- Sirve para optimizar tiempo, es decir, no es necesario crear otra red con los mismos componentes; en el momento de dar guardar mantiene las mismas propiedades pero con otro nombre.

Configuración de red



El objetivo de este catálogo es configurar las redes.

La imagen que se encuentra en la parte superior se va a dividir en 2 secciones (1 y 2), así mismo en la sección 2 se verán 4 apartados: guardar como, afectación de expedientes, transiciones y eventos.

- ✓ Sección 1.- En esta parte se le dará un nombre a la red para después seleccionar, institución financiera, tipo de crédito y oficina, en el campo autorización se podrá ver si la red es o no autorizada, el número de expedientes que tenga asociados y proceso que se refiere a la configuración de la red, ya sea, por la compañía o por el corporativo; después de contar con los datos de la sección 1, se procederá a la siguiente etapa donde se trabajará la red a detalle.
- ✓ Sección 2.- Esta parte del submenú se dividirá en 3 áreas, estado, requisito y documento. Consta de un campo en el que se debe seleccionar el tipo de ingreso al que pertenecerá la red de titulación.

Estado.- Se refiere a los estados por los que debe pasar el expediente, como la información ya fue previamente capturada únicamente se seleccionará y se dará de alta para poder utilizar las áreas restantes que son requisito y documento.

Requisito.- Se van a seleccionar los requisitos que necesite cada estado, y se van a dar de alta, al escoger el requisito se procede a documento.

Documento.- Se van a tomar el o los documentos que sean necesarios para cada requisito antes seleccionado, se podrá mencionar su vigencia y si son opcionales.

Afectación a expedientes.- En caso de necesitar un requisito adicional para uno o más expedientes en especial se podrá hacerlo en la parte inferior del área de requisito. Al visualizar la pantalla se podrá seleccionar, todos, algunos o ningún expediente, en caso de que se proceda de manera positiva se creara una red alterna donde se registrarán las modificaciones, con la finalidad de no afectar la red principal.

Transiciones.- En esta pantalla se podrán ver los estados iniciales y finales para hacer las transiciones necesarias en la red, solo se relacionarán las columnas.

Eventos.- Este catálogo se dividirá en estados de red, requisitos, documentos y transiciones. Todos los apartados mostrarán la información correspondiente a cada área, se podrá seleccionar el evento correspondiente y si son de tipo corporativo o compañía. Es importante señalar el tipo de ingreso, por que si se quiere generar una red con un tipo diferente al de la red original se podrán hacer las modificaciones en requisito y documento no así en estado, siguiendo el mismo proceso.

Guardar como.- Tiene dos funciones las cuales llevan el mismo objetivo : Principalmente tiene como finalidad guardar el estado de red corporativo, en base a él se agregan los estados, requisitos o documentos que sean necesarios. Sirve para optimizar tiempo, es decir, no es necesario crear otra red con los mismos componentes; en el momento de dar guardar mantiene las mismas propiedades pero con otro nombre.

Query que carga las redes de titulación.

```
SELECT
PR.*
FROM
ProcesoRed PR
WHERE
PR.EstCorporativo = 1
ORDER BY PR.NombreProcesoRed
```

Query que carga las Estado de red de titulación.

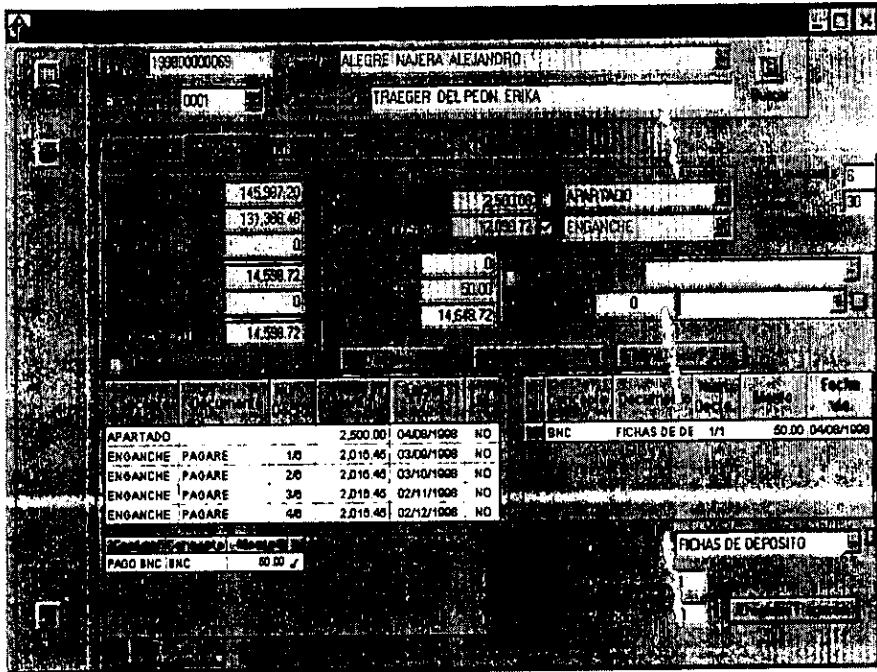
```
SELECT
PRER.*, ER.*, PRER.TiempoEstimar AS TiempoDeRed
FROM
ProcesoRedEstadoRed PRER, EstadoRed ER
WHERE
PRER.ClaveEstadoRed = ER.ClaveEstadoRed AND PRER.ClaveProcesoRed= 1000
ORDER BY PRER.Orden
```

Query que carga los requisitos de la red de titulación.

```
SELECT
PERTL.*, Re.*
FROM
ProcesoEdoRequisitoTipolograso PERTL, Requisito Re
WHERE
PERTL.ClaveRequisito = Re.ClaveRequisito AND PERTL.ClaveTipolograso = 900 AND PERTL.ClaveProcesoRed = 1000 AND PERTL.ClaveEstadoRed = 000
ORDER BY PERTL.Orden
```


En la parte inferior de la sección se muestran los grupos, sus conceptos asociados y los valores numéricos. Una vez seleccionado el plan de ventas se pasa al área plan de pagos.

Plan de pagos.



Aquí aparecerán el precio de la vivienda y crédito; de esto se puede deducir el remanente, el apartado y los cargos extras si los hubiera. En esta pantalla se podrá modificar la información con respecto al número de pagarés, periodo u otra que sea necesaria para ajustar el plan de pagos. Cabe señalar que si se selecciona el apartado se desglosará la información asociada con la del remanente.

También se desglosan los montos Gastos F_Eng.; se les asigna numero de pagarés, periodo y tipo de documento.

Otros cargos.

En esta pestaña se podrá agregar el o los conceptos que no estén incluidos en el plan de venta por ejemplo, (BNC, gastos de investigación, etc.). Se puede incluir dentro de los pagarés considerados en el plan de pagos.

Query que carga lo que contrato el cliente

```
SELECT *
FROM
Contratacion
WHERE
ClaveCorporativo = 1 AND ClaveDelegacion = 1 AND ClaveCompaniaGEO = 2 AND ClaveCliente = 199901000064
AND ClaveExpediente = 1
ORDER BY CampoNegociacion
```

Query que realiza la suma de otros cargos del cliente

```
SELECT
SUM(Monto)
FROM
OtrosCargos
WHERE
ClaveCorporativo= 1 AND ClaveDelegacion= 1 AND ClaveCompaniaGEO= 2 AND ClaveCliente= 199901000064
AND ClaveExpediente= 1
```

Query que carga el plan de venta seleccionado por el cliente

```
SELECT
ClaveDefPlusVenta, Opcion
FROM
Contrato
WHERE
ClaveCorporativo = 1 AND ClaveDelegacion = 1 AND ClaveCompaniaGEO = 2 AND ClaveCliente = 199901000064
AND ClaveExpediente = 1
```

Query que carga la promoción seleccionada por el cliente

```
SELECT
E.VivClavePromocion, E.ClaveTipoCredito, E.ClaveInstitucionFinanciera
FROM
expediente E
WHERE E.
ClaveCorporativo = 1 AND E.ClaveDelegacion = 1 AND E.ClaveCompaniaGEO = 2 AND E.ClaveCliente = 199901000064 AND E.ClaveExpediente = 1
```

Conclusiones

Beneficios

Durante la implantación del sistema se ha logrado una mejora dentro de la productividad de la compañía, ya que la información introducida al sistema y su posterior explotación, ha dado como consecuencia una mejor atención al cliente, una mayor confianza en la información manejada por los niveles directivos y menor índice de duplicidad de datos generados manualmente.

Junto a esto, las personas también se ha beneficiado del sistema, ya que anteriormente reinaba una serie de reportes generados en hojas de cálculo, lo cual sobrecargaba al personal operativo. Actualmente, la dirección obtiene información directamente del sistema, sin necesidad de solicitarlo a los niveles operativos, liberando a estos últimos de una tarea extra.

En la distribución de datos, se ha observado una mayor disponibilidad de la información, tanto a nivel operacional para la atención a clientes en las diferentes áreas como Ventas, Cobranza o Titulación, así como en las directivas al utilizar la información para conocer el estado de las compañías en particular y del corporativo en general.

Implementación

En el análisis del sistema, dado lo grande del mismo, se necesitaba utilizar una metodología. Inicialmente, se trabajó con la metodología IDEF0 y IDEF1, las cuales provienen de su uso en los EUA. Posteriormente, los documentos se utilizaron para la realización de los diagramas OMT para su mejor comprensión y mejorar la calidad del producto a realizar. Después de esto, se procedió a realizar la implementación de los modelos de objetos, dinámico y funcional en una base de datos relacional y en un lenguaje de programación.

En el grupo se cuenta con una plataforma Windows NT con TCP/IP Frame relay. Las comunicaciones se realizan en general mediante la WAN mediante Enlaces dedicados, y para los puntos de venta, se utiliza modems para la comunicación.

Se decidió utilizar Sybase SQL Anywhere como plataforma de base de datos debido a que conocíamos a Sybase de proyectos anteriores, y económicamente era accesible a las posibilidades del corporativo. Esta herramienta permite la distribución de datos mediante un sistema de mensajes que utilizan de transporte el correo de Lotus Notes 4.5 (VIM – Vendor Independent Messaging), o archivos

codificados (FILES). Mediante estos medios es posible trasportar la información a través de una WAN, una LAN o Módem, lo cual es bastante moldeable para las necesidades del grupo, ya que en la mayoría de los puntos de venta, solo son Modem los recursos disponibles.

Junto a ello, se utilizó Visual Basic 4.0 para el desarrollo de la interfaz del front end, utilizando como Middleware ODBC (Open DataBase Connectivity) de Microsoft. Esta herramienta se eligió por la facilidad de aprendizaje para el nuevo personal de programación, así como el desarrollo rápido que ofrece esta herramienta, ya que no es necesario conocer a detalle el sistema operativo (DOS/Windows) para realizar tareas complejas. Era solo necesario conocer programación por eventos, programación con objetos y Lenguaje de consulta SQL.

En general el sistema se puede definir como dos sistemas en uno. El primero se enfoca a la solución del trabajo diario operativo o lo que es un sistema de transacciones. El segundo se enfoca a colocar a disponibilidad los datos cargados por el sistema de transacciones y trasformarlo para entregar información útil en las esferas directivas.

Comparación de metodologías

En la actualidad, las metodologías de ingeniería de software que tiene un mayor grado de aceptación son las que están basadas en diagramas de flujo de datos. En la práctica, se utilizan distintas variaciones de la aproximación de flujo de datos. Describiremos el análisis estructurado /Diseño estructurado (SA/SD) como representante de la aproximación de flujo de datos. Yourdon, DeMarco y otros que han escrito acerca de SA/SD.

Las metodologías OMT y SA/SD contienen unos componentes de modelo similares, ambas metodologías admiten los tres puntos de vista ortogonales del sistema: Los modelos de objetos, funcional y dinámico. Las metodologías OMT y SA/SD difieren en el énfasis relativo que se hace en los distintos componentes del modelado. Los diseños de OMT están dominados por el modelo de objetos. El paradigma del mundo real formado por objetos y relaciones proporciona el contexto para comprender el comportamiento dinámico y funcional.

Comparación con OMT.

El modelo de SA/SD y OMT tiene mucho en común. Ambas metodologías utilizan estructuras de modelado similares y admiten los tres puntos de vista ortogonales del sistema. La diferencia entre SA/SD y OMT es sobre todo una cuestión de estilo y de énfasis, en la aproximación SA/SD, el modelo

funcional es dominante; va a continuación el modelo dinámico, y el modelo de objetos es menos importante, por contraste el modelado OMT considera al modelo de objetos como el más importante; viene a continuación el modelo dinámico y el modelo funcional está en último lugar.

SA/SD organiza el sistema en torno a los procedimientos, por contraste, una técnica de diseño orientada a objetos como OMT organiza el sistema en torno a objetos del mundo real, o en torno a objetos conceptuales que existían en la visión que el usuario tiene del mundo real. La mayoría de los cambios habidos en los requisitos son cambios de función más que cambios de objetos, así que el cambio puede resultar desastroso para el diseño basado en procedimientos. Por contraste, los cambios de función se admiten con facilidad en diseño orientado a objetos añadiendo y modificando operaciones y manteniendo intacta la estructura de objetos básica.

La analogía directa entre objetos dentro de un diseño orientado a objetos y objetos del dominio del problema da lugar a sistemas más fáciles de entender. Esto hace que el sistema sea más intuitivo y simplifica la consistencia entre los requisitos y el código de software. Además hace que el diseño sea más coherente para personas que no fueran parte del equipo original de diseño.

Futuro del sistema

Actualmente el sistema continua en revisión, mantenimiento y mejora, dado que debe cumplir un amplio espectro de necesidades del grupo. Junto a esto, a nivel técnico, existen ciertas líneas de acción a seguir en lo subsecuente.

La primera de ellas es hacer transportable el sistema a cualquier base de datos, sea cual sea la marca del proveedor (Sybase, Microsoft, Oracle). Haciendo uso extensivo del ODBC, se plantea tener una sola definición de datos o estructura de la base y aplicarla al manejador seleccionado para poder ejecutar el sistema en ese manejador. La segunda, es la introducción de los llamados Geographical Interface Systems (GIS) para la selección visual de los datos. En particular, tenemos actualmente las estructuras básicas necesarias para poder introducir esta tecnología en la selección de vivienda, reportes gráficos de las promociones por tipo de crédito, institución financiera y producto seleccionado. Esto sirve en la mayoría de los casos para hacer más palpable la información que contiene la base de datos. La tercera y última, es realizar totalmente el sistema multilinguaje, ya que en el presente cuenta con multimonedas, para el manejo de pagos en otras monedas en cobranza. Multilinguaje se desarrollara

mediante un modelo particular para la modificación de las etiquetas del front end y su posterior implementación.

Bibliografía

Object-Oriented Modeling and Design

Rumbaugh James / Blaha Michel / William Premerlani / Frederick Eddy / William Lorenzen

Prentice Hall

1991

Análisis y Diseño de sistemas

Kendall Kenneth

Prentice Hall

1998

Sistemas de bases de datos – Conceptos fundamentales

Elmasri Ramez

Addison Wesley Ibero /Edivision

1997

Ingeniería del Software

Roger S. Pressman

McGraw Hill de México

1989

Apéndice

Metodología OMT – Notación

Notación del Modelo de Objetos Conceptos Básicos

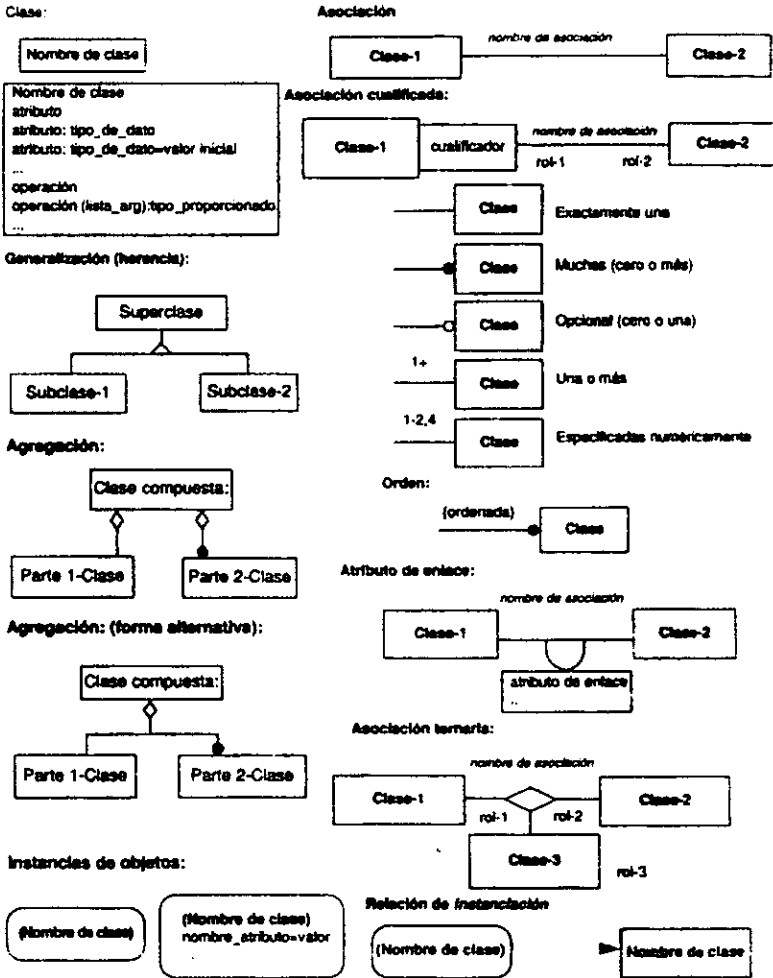


Figura 125 Notación de Modelo de objetos - Conceptos Básicos

Notación del Modelo de Objetos Conceptos Avanzados

Notación del Modelo Dinámico

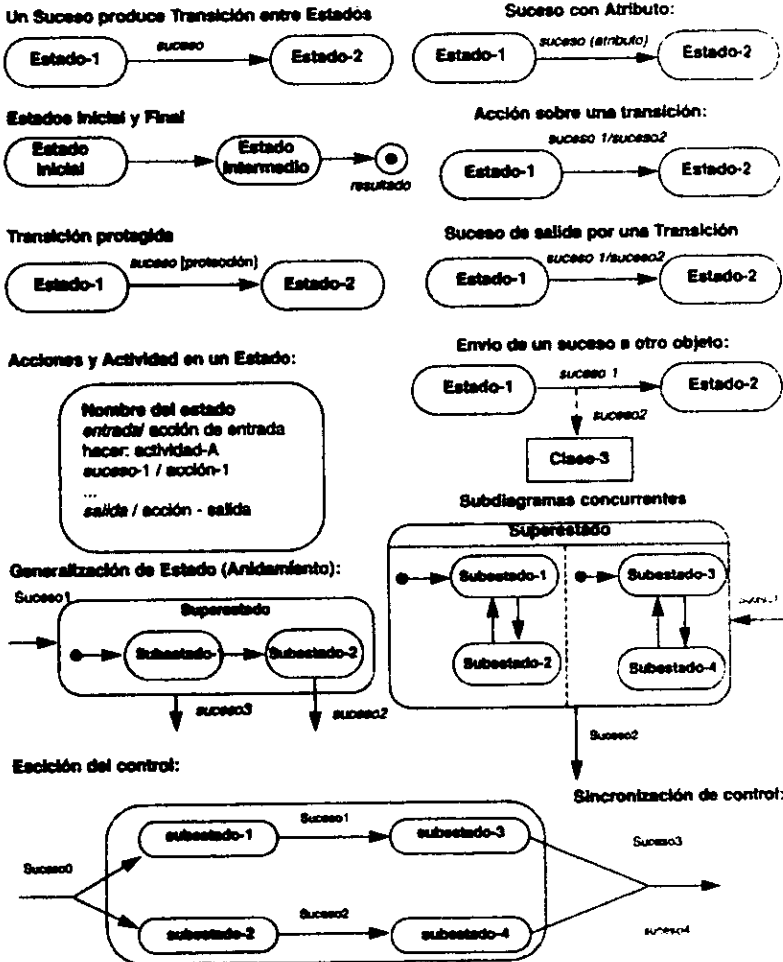


Figura 127 Notación de Modelo dinámico

Notación del Modelo Funcional

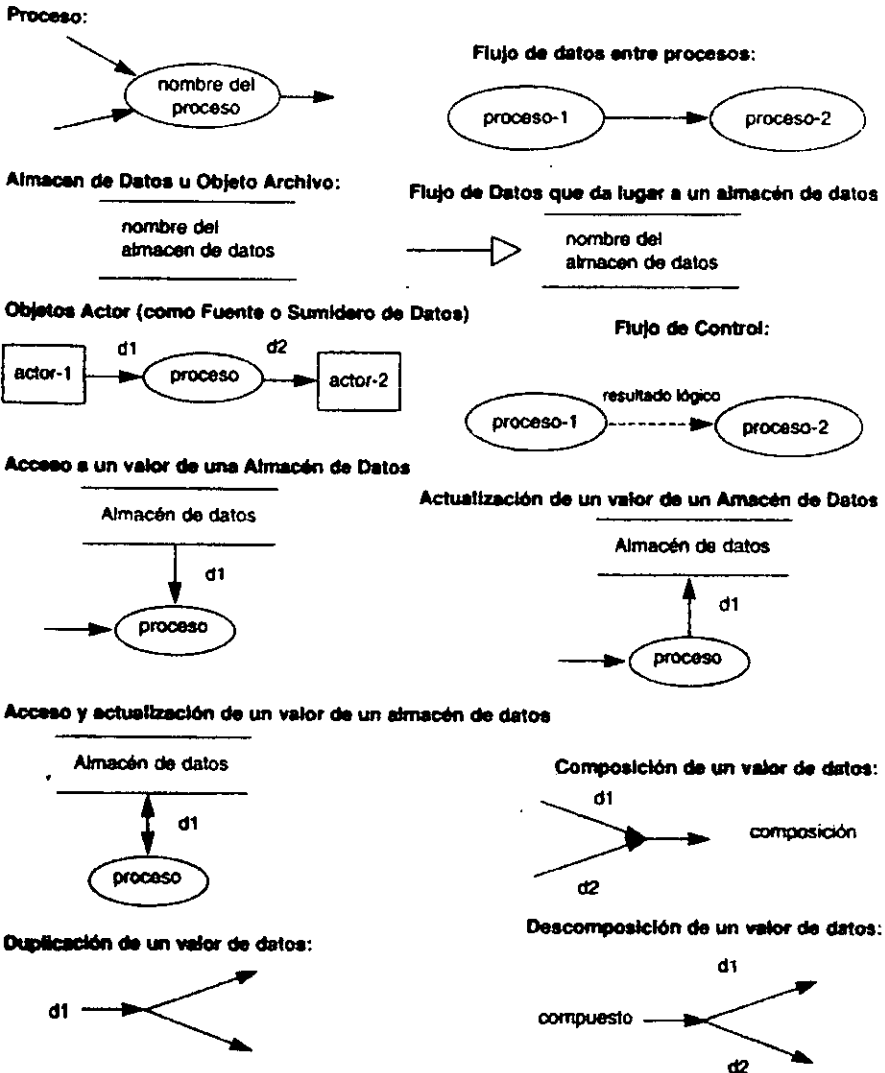


Figura 128 Notación de Modelo funcional

Código Fuente

Generalmente los objetos funcionan bien cuando la relación es de uno a uno, sin embargo ocurre con frecuencia que un objeto con un tipo contiene varios objetos de otro tipo la propiedad de afluentes está implementada como un objeto Collection para que el objeto Afluencia pueda contener varios objetos Afluentes.

```
Option Explicit
Public Afluentes as New Collection

Public Function NuevoAfluente(Nombre, FechaVisita, ClaveAfluente) as Afluente
Dim AfluenteNew as New Afluente

    AfluenteNew.Nombre = Nombre ' creación implícita
    AfluenteNew.FechaVisita
    AfluenteNew.ClaveAfluente = ClaveAfluente
    'Agrega la Clave del afluente a la colección.
    SdMain.Afluentes.add AfluenteNew, Cstring(ClaveAfluente)
    'Devuelve la clave del afluente
    set NuevoAfluentes = Afluente New
End Function
```

Figura 129 Función que carga un afluente por primera vez

Puede darse el caso que algún parte del programa hay variables que siguen manteniendo referencia al objeto Afluente o a cualquiera de los objetos de nuestro programa. Por lo que se desarrollo un código para solucionar en su totalidad este problema

```
'Colección global para dar a cada objeto un clave única
Public gcolDebug as New Collection

'Función Global para dar a cada objeto una clave única
Public Function DebugSerial() as Long
    Static lngSerial As Long
    lngSerial = lngSerial + 1
    DebugSerial = lngSerial
End Function
```

Figura 130 Código de depuración

```
'Almacenamiento para la clave de depuración
Private mlngDebugClaveAfluente as Long
Property Get DeoebgClaveAfluente() as Long
    DebugClaveAfluente = mlngDebugClaveAfluente
End Property

Private Sub Class_Initialize()
    mlngDebugClaveAfluente = DeudbSerial
    'Agrega una entrada a la cadena a la colección global
    gcolDebug.Add "Afluente; DebugClave = " & DebugClaveAfluente, Cstring(DebugClaveAfluente)
End Sub
```

Figura 131 En el modulo de clase puede ser proporcionado su nombre a Afluente

```

Private Sub CargaInfo(ByVal SClasulaWhere As String)
Dim SSqQry As String
Dim SdDatos As Recordset
Dim sCAdAux As String
Dim nRow As Long
Dim nCol As Long
Dim bOficial As Boolean
Dim bOversidero As Boolean
Dim iCodigoError As Integer
bOficial = False
bOversidero = True
Screen.MousePointer = vbHourglass
On Error GoTo CargaInfo_Error
SSqQry = "SELECT DISTINCT STRING(C.ClaveCliente), C.ApellidoPaterno, C.ApellidoMaterno, C.Nombre, " & _
        " C.Lada, C.Telefono, IFNULL(C.Observacion, 'C.Observacion'), C.RFC, " & _
        " C.CedulaIdentificacion, DateFormat(C.FechaNacimiento, 'DD/MM/YYYY'), C.ClaveEstadoCivil, " & _
        " EC.NombreEstadoCivil, C.Sexo, ARGN(C.Sexo, 'Masculino', 'Femenino'), " & _
        " C.ClavePais, P.NombrePais, C.ClaveEstado, E.NombreEstado, " & _
        " C.DelegacionMunicipio, C.ColoniaPoblacion, C.CalleNumero, C.CodigoPostal, " & _
        " C.ClaveMedioPublicitario, M.NombreMedioPublicitario, C.ClaveTipoEconomia, " & _
        " TE.NombreTipoEconomia, C.ClaveOrganizacion, O.NombreOrganizacion, "
SSqQry = SSqQry & _
        " C.Password, DateFormat(C.FechaAloj, 'DD/MM/YYYY'), C.Ocupacion, " & _
        " C.TipoCliente, ARGN(Convert(TinyInt, C.TipoCliente), 'Cliente', 'Prospecto'), " & _
        " C.ClaveNacionalidad, N.NombreNacionalidad, C.Status, " & _
        " ARGN(Convert(TinyInt, C.Status+1), 'Cancelado', 'Activo', 'Terminado'), C.Sueldo, " & _
        " C.ClaveTipoIngreso, TI.NombreTipoIngreso, C.ClaveEmpresa, EX.RazonSocial, " & _
        " C.Puesto, C.Antiguedad, C.TelefonoEmpresa, C.LadaEmpresa, " & _
        " C.ExcasionEmpresa, C.AfiliacionIMSS, C.NumeroDependientes, "
SSqQry = SSqQry & " IFNULL(C.ClaveCalificacion, 'C.ClaveCalificacion'), IFNULL(RBC1.Calificacion, 'RBC1.Calificacion),"
SSqQry = SSqQry & " IFNULL(C.ClaveCalificacion2, 'C.ClaveCalificacion2'), IFNULL(RBC2.Calificacion, 'RBC2.Calificacion),"
SSqQry = SSqQry & " IFNULL(C.ClaveCalificacionCoo, 'C.ClaveCalificacionCoo'), IFNULL(RBC3.Calificacion, 'RBC3.Calificacion),"
SSqQry = SSqQry & " FROM (Clientes C LEFT OUTER JOIN Estado E LEFT OUTER JOIN Pais P), Organizacion O, " & _
        " Nacionalidad N, TipoEconomia TE, MedioPublicitario M, " & _
        " EmpresaExterna EX, EstadoCivil EC, TipoIngreso TI, Comisionista Com, Comisionista Pr, CalificacionBC RBC, CalificacionRC RBC1, "
SSqQry & " WHERE C.ClaveOrganizacion " & O.ClaveOrganizacion "
        & " AND C.ClaveMedioPublicitario " & M.ClaveMedioPublicitario "
        & " AND C.ClaveEstadoCivil " & EC.ClaveEstadoCivil "
        & " AND C.ClaveTipoEconomia " & TE.ClaveTipoEconomia "
        & " AND C.ClaveNacionalidad " & N.ClaveNacionalidad "
        & " AND C.ClaveEmpresa " & EX.ClaveEmpresa "
        & " AND C.ClaveCalificacion " & RBC.ClaveCalificacion "
        & " AND C.ClaveCalificacion2 " & RBC1.ClaveCalificacion "
SSqQry = SSqQry & " AND C.ClaveCalificacionCoo " & RBC3.ClaveCalificacion "
        & " AND C.ClaveCalificacion2Coo " & RBC3.ClaveCalificacion "
        & " AND C.ClaveTipoIngreso " & TI.ClaveTipoIngreso "
        & " AND C.ClaveCorporativo " & iGM.ClaveCorporativo "
        & " AND C.ClaveDelegacion " & iGM.ClaveDelegacion "
        & " AND C.ClaveCompaniaGEO " & iGM.ClaveCompaniaGEO "
        & " AND C.ClaveComisionista " & Com.ClaveComisionista "
        & " AND C.Com_ClaveComisionista " & Pr.ClaveComisionista "
SSqQry = SSqQry & SClasulaWhere
SSqQry = SSqQry & " ORDER BY C.ApellidoPaterno, C.ApellidoMaterno, C.Nombre "
Call LimpiaGrid(F1Info)
nRow = 1: nCol = 1
sCAdAux = "DSN=" & sGdbServer & ";UID=" & sGdbUser & ";PWD=" & sGdbPassword
F1Info.ODBCConnect sCAdAux, True, iCodigoError
F1Info.ODBCQuery SSqQry, nRow, nCol, bOficial, bOficial, bOficial, bOversidero, iCodigoError
F1Info.OBDCDisconnect
Screen.MousePointer = vbDefault
GoTo CargaInfo_OK
CargaInfo_Fin:
On Error Resume Next
SdDatos.Close
Screen.MousePointer = vbDefault
GoTo CargaInfo_OK
CargaInfo_Error:
Screen.MousePointer = vbDefault
Call MuestraError("Aviso")
Resume CargaInfo_Fin
CargaInfo_OK:
End Sub

```

Figura 132 Procedimiento que carga la información de Afluencia en la cual se utiliza un OCX llamado Formula ONE

Para cargar la información del afluente se genera un procedimiento simple con funciones SQL.

Para agregar un Plan de venta a un cliente realizamos dentro del evento CmdAsignaPlanVenta_Click() agrega un miembro a la colección Cliente

```
Opcion Explicit
Public Cliente as New Cliente

Public Promocion as New Collection
Public InstitucionFinanciera as New Collection
Public ClavePlanVenta as New Collection

Private Sub CmdAsignaPlanVenta_Click()
    Dim PlanventaNew as New Cliente
    Static intPlanventaNum as Integer
    With PlanventaNew
        intPlanventaNum = intPlanventaNum + 1
        .ClavePlanVenta = "Plan de Venta: " & Format$(intPlanventaNum, "0000")
        .Promocion = TxtPromocion.Text
        .InstitucionFinanciera = Cdbl(TxtInstitucionFinanciera)
        'Agrega la referencia del objeto empleado a la collection ClavePlanVenta como ella misma
        sdomain.Cliente.Add PlanventaNew, .ClavePuntoVenta
    End With
    Call AsignaPlanVentaCliente(ClavePuntoVenta, Cliente, Promocion)
End Sub
```

Figura 133 Procedimiento CmdAsignaPlanVenta

Después de Asignar plan de venta inserta en las tablas correspondientes así como carga la información con el siguiente procedimiento.

```

Private Sub AsignaPlanVentaCliente(ByVal iClaveConceptoEsquemaVenta As Integer, ByVal cOpcion As String, ByVal DValorInicial As Double)
Dim sSqlQry As String
Dim sDatos As Recordset
Dim iClaveEsquemaVenta
On Error GoTo SubError
**** tras la clave faltante(iClaveEsquemaVenta),para la tabla PlanVentaCliente
sSqlQry = " SELECT ClaveEsquemaVenta "
sSqlQry = sSqlQry & " FROM DefinicionPlanVenta "
sSqlQry = sSqlQry & " WHERE ClaveDefPlanVenta = " & ObtieneValorDesdeCombo(CboDefPlanVenta)
Set sDatos = DbGblDB.OpenRecordset(sSqlQry, dbOpenSnapshot, dbSQLPassThrough)
iClaveEsquemaVenta = sDatos("ClaveEsquemaVenta")
sSqlQry = ""

sSqlQry = " INSERT INTO PlanVentaCliente"
sSqlQry = sSqlQry & "(ClaveCorporativo, ClaveDelegacion, ClaveCompaniaGEO, ClaveCliente,"
sSqlQry = sSqlQry & " ClaveExpediente,ClaveDefPlanVenta, "
sSqlQry = sSqlQry & " ClaveEsquemaVenta, ClaveConceptoEsquemaVenta, Opcion, ValorInicial )"
sSqlQry = sSqlQry & " VALUES("
sSqlQry = sSqlQry & iGblClaveCorporativo & ", "
sSqlQry = sSqlQry & iGblClaveDelegacion & ", "
sSqlQry = sSqlQry & iGblClaveCompaniaGEO & ", "
sSqlQry = sSqlQry & CboClaveCliente.List(CboCliente.ListIndex) & ", "
sSqlQry = sSqlQry & CboExpediente.ColList & ", "
sSqlQry = sSqlQry & ObtieneValorDesdeCombo(CboDefPlanVenta) & ", "
sSqlQry = sSqlQry & iClaveEsquemaVenta & ", " & " Por definir
sSqlQry = sSqlQry & iClaveConceptoEsquemaVenta & ", "
sSqlQry = sSqlQry & "" & cOpcion & "" & ", "
sSqlQry = sSqlQry & DValorInicial & ") "
DbGblDB.Execute sSqlQry, dbSQLPassThrough
If boGblAbiertaTraspaso Then
FrmTraspasoExp.CmdSeleccionarPlanVenta.Enabled = False
FrmTraspasoExp.CmdAutorizacion.Enabled = True
End If
GoTo SubOK
SubFin:
On Error Resume Next
GoTo SubOK
SubError:
Call MuestraError("Aviso")
Resume SubFin
SubOK:
End Sub

'Función que genera la cantidad del plan seleccionado con letra
Public Function MontoEnLetra(ByVal DMonto As Double) As String
Dim cNum2Let As clsNum2Let
Set cNum2Let = New clsNum2Let
cNum2Let.Numero = DMonto
MontoEnLetra = cNum2Let.ALetra
End Function

```

Figura 134 Procedimiento Asignar plan de venta