



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE INFORMACION DE FLUJOS DE CAJA DE UN BANCO CENTRAL

T E S I S

QUE PARA OBTENER EL TITULO DE: INGENIERO EN COMPUTACION

P R E S E N T A N :

- ALFONSO ARROYO
JOSE ALFREDO ALMARAZ
JOSE LUIS CABALLERO
RICARDO GARCIA BENITEZ
RAUL MENA GAYTAN

Extra
Rosa
Ortega

DIRECTOR DE TESIS: M. I. LAURO SANTIAGO CRUZ



MEXICO, D.F.

2000

284421



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.

A mi Madre:

Sabiendo que no existirá una forma de agradecer toda una vida de sacrificios y esfuerzos, quiero que sienta que el objetivo logrado también es suyo, gracias a su gran apoyo, he logrado conseguirlo.

José Luis Caballero

A mis padres y hermanos sin cuya valiosa ayuda y apoyo no hubiera logrado terminar mi carrera universitaria.

A la Universidad Nacional Autónoma de México a quién le debo mi formación profesional.

Ricardo

A mi Padres por todas las oportunidades que me han dado para realizarme

A mis hermanos por todo su apoyo en todos estos años

A Jenny por que su cariño ha sido fundamental en mi vida

Alfonso

A Jesús por su amor y misericordia.

A mis padres que siempre creyeron en mí y siempre me han apoyado en cuanta decisión he tomado.

A mis hermanos y sus familias que siempre han hecho divertida la vida, aún cuando tenga que ir a buscarlos a tierras inhóspitas.

A todos los compañeritos de *Radio Pasillo*, que han hecho agradable y divertida la carga de trabajo en el Banco y que como dieron lata para que escribiera esta tesis.

A la Universidad Nacional Autónoma de México y muy especialmente a la Facultad de Ingeniería donde viví grandes momentos y donde fui formado profesionalmente.

A Banco de México por darme los elementos para desarrollarme profesionalmente, aprender a trabajar en equipo y a analizar con mayor detalle los problemas a los que nos enfrentamos.

A mis hermanitos de la Iglesia por formar parte de mi vida.

A todos los que no pudieron aparecer en esta lista pero que, en su momento, fueron un apoyo invaluable, incluyendo a nuestros amigos de Sanborns que nos atendieron mientras discutíamos este trabajo.

Con mucho cariño a mis pequeñuelas hijas Ditzza y Tamara (y si hay más, también a ellos o ellas) por su cariño y por todos los momentos felices que hemos vivido juntos.

Y muy especialmente

A Gabys por su amor, apoyo y comprensión, supongo que no ha de ser fácil estar casada conmigo.

Raúl

A MI MADRE POR TODO EL APOYO
Y SACRIFICIOS REALIZADOS,
PARA QUE PUDIERA ALCANZAR
LA CULMINACION DE ESTA
META TAN VALIOSA EN MI VIDA.

A MIS HERMANOS POR EL APOYO,
MOTIVACION Y COMPRENCION.

A MIS AMIGOS POR LOS MOMENTOS
DIFICILES Y AGRADABLES VIVIDOS
EN SU COMPAÑIA.

José Alfredo

Índice

INTRODUCCIÓN	iii
1 ANTECEDENTES.....	1
1.1 FUNCIONES DE UN BANCO CENTRAL	1
1.2 CASO MÉXICO.....	4
1.2.1 <i>Historia de Banco de México</i>	4
1.2.2 <i>Funciones del Banco de México</i>	8
1.3 CICLO DE VIDA DEL EFECTIVO.....	9
1.4 CONCEPTOS BÁSICOS	12
1.4.1 <i>Diseño de Bases de Datos</i>	12
1.4.2 <i>Arquitectura Cliente - Servidor</i>	21
1.4.3 <i>Metodologías</i>	23
2 REQUERIMIENTOS	29
2.1 SITUACIÓN ACTUAL	29
2.2 PROBLEMÁTICA.....	33
2.3 CONTROL DE EXISTENCIAS.....	35
2.4 CONTROL DE OPERACIONES	36
2.5 CONTROL DE SEGURIDAD	39
2.6 VALIDACIONES GENERALES.....	41
2.7 DISPONIBILIDAD	43
2.8 INFORMACIÓN DIARIA E HISTÓRICA.....	44
3 ANÁLISIS	45
3.1 ANÁLISIS DEL FLUJO DE EFECTIVO.....	45
3.2 CONTROL DE EXISTENCIAS.....	57
3.3 CONTROL DE OPERACIONES	58
3.4 ESQUEMA DE SEGURIDAD.....	76
3.5 ESQUEMA DE VALIDACIONES.....	82
3.6 PROPUESTAS DE SOLUCIÓN	90
3.6.1 <i>Propuestas y factibilidad</i>	97
3.6.2 <i>Selección de alternativa</i>	99
3.6.3 <i>Plan de trabajo</i>	99

4 DISEÑO	102
4.1 PRINCIPALES COMPONENTES DE UN RDBMS	102
4.2 DISEÑO ENTIDAD – RELACIÓN	106
4.3 NORMALIZACIÓN	142
4.4 DICIONARIO DE DATOS	146
4.5 RELACIÓN CON OTROS SISTEMAS	158
5 CODIFICACIÓN	161
5.1 LINEAMIENTOS DE CODIFICACIÓN	161
5.2 ESTRUCTURA DE LA CODIFICACIÓN	163
5.3 CODIFICACIÓN	164
6 PRUEBAS, LIBERACIÓN Y MANTENIMIENTO	190
6.1 PRUEBAS	190
6.2 LIBERACIÓN Y CAPACITACIÓN	191
6.2.1 Liberación	191
6.2.2 Capacitación	194
6.3 MANTENIMIENTO	195
7 RESULTADOS Y CONCLUSIONES	197
7.1 RESULTADOS	197
7.2 CONCLUSIONES	198
APÉNDICE A. COMANDOS SQL	200
COMANDOS BÁSICOS	200
ELEMENTOS DE UNA BASE DE DATOS	202
MANEJO DE TRANSACCIONES	207
CICLOS DE CONTROL	207
PERMISOS	208
APÉNDICE B. GLOSARIO DE TÉRMINOS	210
BIBLIOGRAFÍA	213

Introducción

Los Bancos Centrales tiene como objetivo primordial el regular a los sistemas financieros de su país, determinando las políticas monetarias que lleven a una estabilidad económica. También son los encargados de la emisión de billete y moneda de curso legal, y una de sus finalidades es la de proveer moneda nacional a la economía del país.

Para el abastecimiento de efectivo, los Bancos Centrales cuenta con una amplia infraestructura que les permite distribuir el efectivo dentro del territorio nacional, sin embargo, la distribución de la moneda nacional es solamente un punto del ciclo de vida del efectivo, por esta razón, los Bancos Centrales controlan todas las etapas que forman este ciclo que va desde la autorización de emisión hasta la destrucción de billetes o monedas inservibles.

Normalmente los Bancos Centrales ponen el efectivo en circulación a través de las instituciones bancarias del país, así, los bancos son quienes se presentan ante el Banco Central para realizar depósitos y retiros, en otras palabras convertir dinero electrónico en dinero físico y viceversa. Ellos a su vez ofrecen este servicio de conversión a sus clientes, llegando el efectivo a la sociedad.

Dentro de los Bancos Centrales se genera una gran cantidad de movimientos de varios tipos que es necesario controlar para así contar con un inventario de billetes y monedas confiable. En todo momento es necesario conocer el número piezas por cada denominación y la cantidad de dinero que representa.

Problemática

Los Bancos Centrales requieren controlar los diferentes procesos involucrados con la emisión de dinero, así como, identificar la ubicación de cada uno de los billetes o monedas de su propiedad. Adicionalmente requiere conocer cuánto dinero entra o sale a circulación; todos los movimientos que se realizan internamente y cuántos billetes o monedas han sido destruidos.

Todos los movimientos deben de apegarse a las reglas del negocio establecidas, las cuales pueden cambiar repentinamente.

Propuesta de solución

La propuesta es desarrollar un sistema en arquitectura cliente - servidor de tres capas, utilizando como servidor a un manejador de bases de datos relacional, en donde se controlarán las existencias y se llevará un registro de las operaciones realizadas en el día, almacenando esta información en una base de datos histórica para consultas posteriores. Este sistema permitirá al usuario establecer las reglas del negocio relativas a políticas de movimientos y administración de su infraestructura.

En un modelo de tres capas, la primera capa corresponde al repositorio de datos, que es dónde se almacena la información, la segunda capa corresponde a un conjunto de servicios que contienen la lógica del sistema y la tercera capa corresponde a la interfaz con el usuario. Considerando que, en una institución con cierto grado de sistematización, es común que los sistemas se interrelacionen, es decir, que varios sistemas sean clientes de otros sistemas, por esta razón es importante que los servicios de comunicación sean los mismos sin importar el tipo de cliente; el incrustar la lógica del sistema o las reglas del negocio en la capa del cliente, obligaría a "contaminar" a otros sistemas con unas funciones que no son propias de ellos, complicando el mantenimiento y la uniformidad de la información. Por esto, en este trabajo solamente trataremos las dos primeras capas, dejando la interfaz con el usuario para otro estudio.

La capa intermedia a desarrollar contará con todos los servicios necesarios para realizar la operación diaria en materia de flujos de caja y la obtención de información sin importar la forma en la cuál son invocados los servicios, restringiendo el acceso a los datos en forma directa.

Objetivo

Este sistema controlará las existencias propiedad de un Banco Central y llevará el control de las operaciones que originan flujos de efectivo ya sea interno o externo. Además proporcionará la información pertinente para la administración del mismo y servirá de gran ayuda en la toma de decisiones.

Estructura de la tesis

Este trabajo de tesis consta de 7 capítulos, los cuales están organizados de la siguiente manera:

En el capítulo 1 se mencionan las funciones generales de los Bancos Centrales, también se describe el ciclo de vida del efectivo y se mencionan los antecedentes teóricos necesarios para el desarrollo del sistema.

En el capítulo 2 encontramos el estudio de requerimientos del sistema y la descripción de la infraestructura de los Bancos Centrales.

En el capítulo 3 se realiza un análisis formal, para profundizar sobre los requerimientos del sistema y sentar las bases del desarrollo, también se muestran las alternativas de solución y el estudio de factibilidad.

En el capítulo 4 encontramos el diseño del modelo Entidad – Relación, así como la normalización del mismo y el diccionario de datos.

En el capítulo 5 se muestran los lineamientos de codificación, así como las políticas de desarrollo empleadas, también se muestra una parte del código generado.

En el capítulo 6 se mencionan las pruebas realizadas al sistema, incluyendo las pruebas de año 2000, así como las actividades necesarias para la liberación del sistema.

En el capítulo 7 encontramos los resultados y conclusiones del presente.

Finalmente presentamos un apéndice que menciona las principales características del lenguaje utilizado para la codificación. También presentamos el glosario de términos y la bibliografía.

1 Antecedentes

En este capítulo se describirán las principales funciones de un Banco Central, así como los conceptos básicos que sirvieron de cimiento para la construcción de uno de los sistemas informáticos que da soporte a una de las funciones fundamentales de un Banco Central: el control de existencias de billetes y monedas.

Los conceptos básicos tratados en este capítulo engloban la descripción del modelo "cliente – servidor", así como conceptos de bases de datos relacionales y metodologías para el diseño de sistemas computacionales.

1.1 Funciones de un Banco Central

Desde la utilización del trueque como medio de intercambio de bienes para la satisfacción de necesidades, hasta el surgimiento de las monedas y billetes como vehículo para facilitar dichas transacciones, se hizo patente la necesidad de la existencia de un órgano rector capaz de imponer un orden y planificar la política monetaria de una Nación, necesidad que se ha materializado en la figura de un Banco Central.

Los Bancos Centrales, en general, tienen como objetivo prioritario procurar la estabilidad del poder adquisitivo de la moneda nacional. Para cumplir con su encomienda, son facultados para diseñar y ejecutar las políticas monetarias convenientes. Por esta razón, las funciones que realizan los Bancos Centrales están relacionadas con la regulación del sistema financiero interno, la administración de las reservas ya sea en divisas, oro o plata y el control de la balanza comercial; otra función importante de los Bancos Centrales es su representación ante organismos financieros internacionales.

Dentro de la regulación del sistema financiero interno, podemos observar que los Bancos Centrales se encargan de vigilar su buen funcionamiento, ya que un desequilibrio interno de alguno de los participantes puede ocasionar un estado de insolvencia que podría generalizarse y esto, a fin de cuentas, repercutir en las actividades productivas de la nación. Por ejemplo, un sector perteneciente al

sistema financiero es la banca comercial, si un banco no pudiera cumplir con sus compromisos de pago, podría ocasionar que uno o más de sus acreedores se encuentren imposibilitados de liquidar sus propios compromisos, lo que podría ocasionar que un tercer banco que sea acreedor de estos dos o más bancos tuviera que absorber, aunque fuera temporalmente, el impacto de esta insolvencia, y como consecuencia, a las instituciones morosas se les podrían negar los financiamientos de otros miembros del sistema, y dado que gran parte del capital que manejan los bancos provienen de las empresas y público en general, estas instituciones podrían tener incumplimientos con sus clientes.

Los mecanismos utilizados por los Bancos Centrales para prevenir esta situación varían de país a país, en general solicitan a las instituciones participantes del sistema financiero un depósito que, normalmente, es en papel gubernamental, el cual se toma como base para restringir las líneas de crédito que les son otorgadas. Algunos Bancos Centrales se auxilian también de los saldos promedios en las cuentas que sostienen en el Banco Central, de tal forma que pueden analizar el comportamiento de cada institución y tomar las medidas necesarias cuando se detectan problemas dentro de las mismas.

Un aspecto importante a cuidar en los sistemas financieros son los sistemas de pagos. Los sistemas de pagos son los medios por los cuales se liquidan los compromisos adquiridos por el consumo de bienes y servicios que realizan las instituciones, empresas y el público en general.

A través de los sistemas de pagos, podemos ver como las naciones efectúan sus transacciones internas. Existen muchos y muy varios sistemas de pagos, por ejemplo, los pagos interbancarios (pagos que realizan un banco o sus clientes a sus proveedores con cuentas en otros bancos), pagos con tarjetas de crédito y/o débito, cheques, etc.; sin embargo, aún cuando la tecnología ha permitido que se realicen cada vez más transacciones en forma electrónica, el más popular de los sistemas de pago sigue siendo el dinero en efectivo.

Dentro de las funciones de los Bancos Centrales se encuentra la de emitir y abastecer a la nación de moneda nacional, ya sea en billete o en moneda, aunque existen excepciones como los países de la Comunidad Económica Europea que ya están utilizando una moneda común, por lo que la emisión ya no formará parte de las funciones de estos Bancos Centrales, sin embargo, aún ahí permanece la necesidad de la existencia de una institución centralizada que se encargue del proceso de emisión y regulación de los "EUROS" (nombre de la moneda de la Comunidad Económica Europea), por esto fue creado un órgano supremo BCE (Banco Central Europeo) conformado por los Bancos Centrales pertenecientes al SEBC (Sistema Europeo de Bancos Centrales), que por consenso determinarán las políticas monetarias generales aplicables a las naciones participantes. El BCE como cualquier otro Banco Central se encargará de estudiar los requerimientos de efectivo que debe haber en circulación y emitir o retirar cuando sea conveniente.

La reglamentación de las funciones de cada Banco Central varía de país a país, por ejemplo en México es una ordenanza constitucional incluida en los párrafos sexto y séptimo del artículo 28, en otros países forma parte de la ley orgánica.

Cabe señalar que los Bancos Centrales regulan la cantidad de dinero que hay en circulación, aunque no todo el dinero exista como billete o moneda. Para explicar la importancia de la regulación del dinero veamos al dinero como una mercancía que utilizamos como medida de intercambio. Como toda mercancía está sujeta a la ley del mercado de la oferta y la demanda, si hay mucha oferta su valor disminuye y por el contrario, si la demanda es mayor su valor se incrementa, este fenómeno se refleja directamente en las tasas de interés y en los precios de los bienes y servicios. Los Bancos Centrales se preocupan por encontrar el punto de equilibrio que permita a la nación por un lado fomentar la inversión y el ahorro y por otro un sano comercio interno.

Como mencionamos anteriormente, no todo el dinero existe en forma física, sin embargo sí debe estar respaldado para que tenga valor. Una parte considerable del dinero de un país se encuentra en cuentas contables, ya sea por depósitos en cuentas o por la adquisición de papel gubernamental¹ o acciones. Normalmente los pagos son traspasos de una cuenta a otra, estas operaciones se realizan por lo general sin que exista un intercambio de efectivo físico, un ejemplo muy común es realizar el pago de algún servicio utilizando un cheque o realizando una llamada telefónica, en donde lo que sucede es que descuentan la cantidad a pagar de la cuenta del cliente y la agregan a la cuenta del beneficiario, otro ejemplo son las cámaras de compensación donde los bancos compensan sus pagos y solamente traspasan la diferencia de lo que van a recibir y lo que van a pagar aquellas instituciones cuyo diferencial sea negativo, como estos ejemplos podríamos mencionar una gran cantidad de situaciones donde se realizan pagos sin dinero físico. Sin embargo sigue existiendo el comercio con dinero en efectivo sobre todo en pagos de bajo valor y son los Bancos Centrales los encargados de su emisión.

Cuando decimos que la emisión está a cargo de los Bancos Centrales no significa que sean ellos los que la tengan que fabricar, a lo que nos referimos es al control de cuánto dinero y de qué denominación hay que fabricar. Algunos países han optado por contratar los servicios de otros Bancos Centrales o empresas que cuentan con la infraestructura y los controles de calidad, necesarios para la impresión de billetes y/o acuñación de moneda.

La responsabilidad de los Bancos Centrales es distribuir el efectivo entre la población para que pueda funcionar el comercio. Las ventanillas de los bancos y los cajeros automáticos son dos medios a través de los cuales la gente "convierte" dinero electrónico en físico y viceversa, estos dos medios son servicios que la banca ofrece a sus clientes. Las instituciones bancarias

¹ Son bonos que emiten los gobiernos federales para financiar sus proyectos, estos bonos forman parte de la deuda interna del país.

mantiene una cuenta en los Bancos Centrales, en otras palabras, son clientes de estos institutos centrales y como tales, para poder ofrecer los servicios de disposición de efectivo requieren que los Bancos Centrales los provean de efectivo, así la banca acude al Banco Central para que, al igual que sus clientes, transformen dinero de su cuenta en efectivo.

Un control que debe llevar cada Banco Central para poder regular el sistema financiero es el conocimiento de cuanto dinero en efectivo está en circulación. Con esta información complementan el total de dinero que está circulando, que es un elemento que observan las regulaciones monetarias.

El dinero en efectivo, como cualquier mercancía sufre desgastes ocasionados por el uso, así que al pensar en la emisión de dinero, los Bancos Centrales están conscientes que cada pieza (billete o moneda) tiene una vida útil y habrá que sustituirla, por lo que esta situación es considerada en sus proyectos de emisión.

1.2 Caso México

Tomaremos a México como ejemplo para comprender la importancia que tienen los Bancos Centrales dentro una nación. Comenzaremos con una breve historia del Banco de México donde se destacan las acciones que trascendieron en la vida económica del país, mencionando al final las funciones que desempeña actualmente este organismo.

1.2.1 Historia de Banco de México

El Banco de México abre sus puertas en enero de 1925. Los antecedentes de esta Institución se remontan al menos hasta principios del siglo XIX, cuando en 1822, durante el reinado de Agustín de Iturbide, fue presentado un proyecto para crear una institución con la facultad para emitir billetes que se denominaría "Gran Banco del Imperio Mexicano".

Hacia 1884 en un juicio legal celebrado en la capital del país triunfa la postura que favorecía la libre concurrencia de los bancos comerciales en cuanto a la emisión de billetes, mecanismo que se extendió hasta principios de la Revolución.

Con la destrucción del sistema bancario porfirista durante la Revolución, la polémica se centró en las características que debería tener el Banco Único de Emisión, cuyo establecimiento se consagró en el Art. 28 de la Carta Magna promulgada en 1917. La disyuntiva consistía en proponer el establecimiento de un banco privado, o un banco bajo control gubernamental. Los Constituyentes reunidos en Querétaro optaron por esta segunda fórmula, aunque la Carta

Magna sólo estableció que la emisión de moneda se encargaría en exclusiva a un banco que estaría "bajo el control del Gobierno".

Sin embargo, a pesar del *mandato* consagrado en la Constitución, siete años demoró la fundación del entonces llamado Banco Unico de Emisión. Reiteradamente, la escasez de fondos públicos fue el obstáculo insuperable para poder integrar el capital de la Institución. En el interin, se fue consolidando en el mundo la tesis sobre la necesidad de que todos los países contasen con un Banco Central. Tal fue el mensaje de un comunicado emitido en 1920 por la Sociedad de las Naciones, durante la Conferencia Financiera Internacional celebrada en Bruselas.

El establecimiento del Banco de México no se hace realidad hasta 1925, gracias a los esfuerzos presupuestarios y de organización del Secretario de Hacienda, Alberto J. Pani, y al apoyo recibido del Presidente Plutarco Elías Calles.

El Banco de México se inaugura el 1º de septiembre de 1925, acto presidido por el presidente Plutarco Elías Calles. Al recién creado Instituto se le entregó, en exclusiva, la facultad de crear moneda, tanto mediante la acuñación de piezas metálicas como a través de la emisión de billetes. Como consecuencia correlativa de lo anterior, se le encargó la regulación de la circulación monetaria, de los tipos de interés y del cambio sobre el exterior. Asimismo, se convirtió al nuevo órgano en agente, asesor financiero y banquero del Gobierno Federal, aunque se dejó en libertad a los bancos comerciales para asociarse o no con el Banco de México.

El Banco Central nace en momentos de grandes retos y aspiraciones para la economía del país. A la necesidad de contar con una institución de esa naturaleza la acompañaban otros imperativos: propiciar el surgimiento de un nuevo sistema bancario, hacer renacer el crédito en el país y reconciliar a la población con el uso del papel moneda. Esta última no se presentaba como una tarea sencilla después de la traumática experiencia inflacionaria con los "bilimbiques" del periodo revolucionario. Según las crónicas, con ese nombre bautizó el ingenio popular al papel moneda que emitieron los distintos bancos revolucionarios y que se depreció como una exhalación. Por todo ello al Banco de México, además de los atributos propios de un banco de emisión, se le otorgaron también facultades para operar como institución ordinaria de crédito y descuento.

Durante sus primeros seis años de vida, el Banco obtuvo un éxito razonable en cuanto a promover el renacimiento del crédito en el país. Sin embargo, las dificultades que enfrentó en su propósito de consolidarse como Banco Central resultaron formidables. Aunque su prestigio creció y logró avances, la circulación de sus billetes fue débil y pocos bancos comerciales aceptaron asociarse con él mediante la compra de sus acciones.

La primera gran reforma en la vida del Banco de México ocurre hacia 1931 y 1932. En julio de 1931 se promulga una controvertida Ley Monetaria por la cual se desmonetiza el oro en el país. En cuanto al Banco de México, dicha Ley confirió ciertas características de moneda a sus billetes, aun cuando se conservó la libre aceptación de los mismos. Sin embargo, tan sólo ocho meses después sobrevienen otras reformas de mayor trascendencia: la de la mencionada Ley Monetaria y la promulgación de una nueva Ley Orgánica para el Banco de México. Mediante esta última reforma se le retiraron al Banco las facultades para operar como banco comercial, se hizo obligatoria la asociación de los bancos con el Instituto Central y se flexibilizaron las reglas para la emisión de billetes.

Así, una vez expedida la nueva Ley, a la par que se iniciaba un periodo de libre fluctuación para que el tipo de cambio tomara su nivel de equilibrio, se autorizó al Banco a comprar oro a precios de mercado. Con ello, no sólo se dio lugar al aumento de la reserva monetaria, sino que se consolidó uno de los principales canales para la emisión del billete. Dicha fórmula y la recuperación de la economía después de la postración de 1929 y 1930, fue lo que arraigó en definitiva al papel moneda como el principal instrumento de pagos en el país.

Una vez conseguida la aceptación del billete, se abrió el camino para que el Banco pudiera cumplir las funciones señaladas en su nueva Ley Orgánica. Estas fueron las siguientes: regular la circulación monetaria, la tasa de interés y los cambios sobre el exterior; encargarse del servicio de la Tesorería del Gobierno Federal; centralizar las reservas bancarias y convertirse en banco de los bancos y en prestamista de última instancia.

En 1935 se enfrenta una crisis sin precedentes, cuando el precio de la plata se eleva ininterrumpidamente. Se llega a correr riesgo grave de que el valor intrínseco de las monedas de ese metal --las cuales integran el grueso de la moneda fraccionario o de apoyo-- supere a su valor impreso. El peligro se cierne en especial sobre las piezas de un peso, que gozaban de gran arraigo entre la población. La crisis se conjura mediante dos medidas: emitir piezas con menor contenido de plata, y poner en circulación billetes de esa denominación que por largos años se conocieron como "camarones".

A finales de 1939, el entorno de la economía mexicana se modifica drásticamente con el inicio de la Segunda Guerra Mundial. México se ve inundado con los capitales flotantes o "golondrinos" que buscan refugio bancario en nuestro país. Es en este contexto que en junio de 1941 se promulga un nuevo marco legal para las materias bancaria y financiera. De ahí surgieron una nueva Ley Bancaria, así como una nueva Ley Orgánica del Banco de México.

El periodo bélico que se extendió aproximadamente de 1940 a 1945 fue de gran trascendencia en la evolución del Banco de México. Nunca antes desde su fundación, el Banco se había enfrentado al reto de tener que aplicar una política de contención monetaria.

Como se ha dicho, durante este periodo el influjo de capitales hinchó la reserva monetaria del Banco, dando lugar a una acelerada expansión de los medios de pago, expansión preñada de un ominoso potencial inflacionario. Para conjurar esa posibilidad, las autoridades se embarcaron en un proceso de aproximaciones en el que gradualmente se fueron definiendo las herramientas más idóneas para realizar la requerida contención monetaria. El instrumento que mejores resultados arrojó fue la elevación de los "encajes", o sea, de los depósitos obligatorios que la banca tenía que abonar en el Banco Central. Este recurso se llevó a un extremo nunca antes experimentado en país alguno, pues el coeficiente de encaje se elevó hasta el 50% de los depósitos para los bancos del Distrito Federal, y hasta el 45% para los bancos de provincia.

Tan probó su eficacia el expediente de elevar el encaje legal que, una vez concluida la Guerra y durante muchos años, se recurrió en México a la manipulación del requisito de reserva obligatoria, no sólo para fines de regulación monetaria, sino también para otros dos objetivos: como método de financiamiento para los déficit del Gobierno y para fines de "control selectivo del crédito". En 1949, al producirse nuevamente una preocupante entrada de divisas, se reforma la Ley Bancaria y se otorga al Instituto Central la facultad potestativa de elevar el encaje de la banca comercial hasta el 100% sobre el crecimiento de sus pasivos. Sin embargo, esta obligación se graduó en función de la forma en que las instituciones integrasen sus carteras de crédito o, en otras palabras, de acuerdo a la manera en que canalizaran su financiamiento a los distintos sectores de la economía.

En 1948 y 1949, México sufre dos severas crisis de balanza de pagos, atribuibles en muy buena medida, a los reacomodos y ajustes de la economía mundial típicos de la postguerra.

En junio de 1944, México había sido uno de los países suscriptores del convenio de Bretton Woods, mediante el cual se acordó, entre otras cosas, un sistema de tipos de cambio fijos para las monedas del mundo. En 1948, a pesar de que dicho convenio no aceptaba los tipos de cambio fluctuantes, México decidió dejar en flotación la tasa de cambio del peso. Sin embargo, en 1949 se decidió ensayar una nueva paridad al nivel de 8.65. Esta fue la tasa de cambio que prevaleció hasta 1954, cuando el país tuvo que efectuar un nuevo ajuste cambiario.

Durante los años setentas y parte de los ochentas se extiende una época de dificultades para el Banco de México. Hasta 1982, los problemas tuvieron su origen en la aplicación de políticas económicas excesivamente expansivas, y en la obligación que se impuso al Banco de extender amplio crédito para financiar los deficientes fiscales en que entonces se incurrió. Todo ello dio lugar al deterioro de la estabilidad de los precios y fue causa de que ocurrieran dos severas crisis de balanza de pagos en 1976 y 1982. De 1983 en adelante, el sentido de las acciones ha sido de signo distinto. A partir de ese año, los esfuerzos han estado dirigidos, en lo fundamental, a controlar la inflación, a

corregir los desequilibrios de la economía y a procurar la recuperación de la confianza de los agentes económicos.

La gran transformación en la historia reciente del Banco de México ocurrió en 1993, con la reforma constitucional mediante la cual se otorgó autonomía a esta Institución. La autonomía concedida al Banco de México - explicada en la Exposición de Motivos de la reforma constitucional respectiva - tiene como principal objeto construir una salvaguarda contra futuros brotes de inflación. De ahí la importancia de que en el texto Constitucional haya quedado precisado el criterio rector al cual debe sujetarse en todo tiempo la actuación del Banco de México: la procuración de la estabilidad del poder adquisitivo de la moneda nacional.

1.2.2 Funciones del Banco de México

El Banco de México tiene por finalidad proveer a la economía del país de moneda nacional. Teniendo como objetivo prioritario procurar la estabilidad del poder adquisitivo de dicha moneda. Son también finalidades del Banco promover el sano desarrollo del sistema financiero y propiciar el buen funcionamiento de los sistemas de pagos.

El Banco desempeña las funciones siguientes:

- Regular la emisión y circulación de la moneda, los cambios, la intermediación y los servicios financieros, así como los sistemas de pagos.
- Operar con las instituciones de crédito como banco de reserva y acreditante de última instancia.
- Prestar servicios de tesorería al Gobierno Federal y actuar como agente financiero del mismo.
- Fungir como asesor del Gobierno Federal en materia económica y, particularmente, financiera.
- Participar en el Fondo Monetario Internacional y en otros organismos de cooperación financiera internacional o que agrupen a Bancos Centrales.
- Operar con los organismos a que se refiere el punto anterior, con Bancos Centrales y con otras personas morales extranjeras que ejerzan funciones de autoridad en materia financiera.

1.3 Ciclo de vida del efectivo

Una de las funciones de un Banco Central es la emisión y abastecimiento de moneda nacional a la economía del país. Los mecanismos para la autorización de emisión de dinero son muy variadas en el mundo, sin embargo para los fines de este trabajo bastará con suponer que la autoridad competente, que incluso puede ser el mismo Banco Central, autoriza la emisión, con lo que da inicio el ciclo de vida del dinero. Dado que los billetes y monedas son objetos físicos, en lo sucesivo vamos a hacer referencia a ellos como piezas, donde cada pieza tiene atributos propios como tipo de efectivo (billete o moneda), la denominación o valor facial y su valor real².

Una vez autorizada la emisión de piezas por el cause legal correspondiente, ya debe y de hecho es considerado en la contabilidad de los Bancos Centrales. Normalmente las autorizaciones de emisión no involucran la emisión en un instante del tiempo de todas las piezas, los Bancos Centrales basándose en sus estudios de requerimientos de efectivo por parte de la población establece los programas de tal forma que evite la escasez de alguna denominación y también evite un exceso de efectivo o simplemente evitar mantener grandes inventarios.

Una vez autorizada la emisión de dinero, se instruye a la institución encargada de la impresión de billete o acuñación de moneda, según sea el caso, de la fabricación de un número determinado de piezas de la denominación correspondiente. El proceso de fabricación encierra por sí mismo todo un ciclo de vida, sin embargo, para los Bancos Centrales esta fase no forma parte del ciclo de vida del efectivo.

Hasta este momento podemos considerar al dinero como no físico, aún cuando ya esté en espera de ser entregado, mientras las piezas no sean entregadas oficialmente al Banco Central, no son consideradas como parte de las existencias físicas, por tanto su control únicamente se lleva en cuentas contables.

Cuando los fabricantes realizan entregas de billete o moneda al Banco Central inmediatamente forman parte de las existencias, así que dejan de pertenecer al dinero "no materializado". Los departamentos encargados del seguimiento de éstas ordenes van verificando que se les entregue el total de cada orden de fabricación, pueden llegar a saldar total o parcialmente una orden en una entrega, sin embargo para el control de flujos de efectivo realmente no importa a que orden de fabricación corresponde, lo que sí es trascendente es que ingresaron las piezas al banco.

² En algunas ocasiones el valor facial puede no ser el valor real de una pieza, recordemos, a manera de ejemplo, que en México en 1993 se hizo un ajuste a la moneda en la cual se dividió entre 1000, creando los nuevos pesos que después de un tiempo volvieron a llamarse pesos, en este caso el valor facial de un billete de \$1000.00 emitido antes de esta fecha tiene un valor real hoy de \$1 00

Las piezas entregadas al Banco Central son almacenadas en bóvedas, que son espacios físicos destinados al almacenamiento de efectivo, donde pueden estar sujetas a algunos procesos de recuento y verificación.

Las piezas, como cualquier objeto físico, se desgastan por el uso, las piezas cuyo grado de desgaste permita que sigan en circulación les llamaremos Aptas y cuando su grado de desgaste sea tal que ya no resulte conveniente que siga en circulación le llamaremos Deteriorada.

Como el objetivo de la emisión de dinero es abastecer al país de billetes y monedas, los Bancos Centrales utilizan a las instituciones bancarias para este fin. Los bancos acuden al Banco Central a retirar dinero en efectivo, normalmente y por política de los Bancos Centrales, acuden unas cuantas sucursales de cada banco a realizar estas operaciones y estas sucursales se encarga de abastecer a las demás sucursales de su institución y así poder ofrecerlo a sus clientes.

Mientras las piezas se encuentren dentro del Banco Central son consideradas como de su propiedad y por tanto forman parte de sus existencias. Cuando las piezas son retiradas por una institución se considera que están en circulación, sin importar si las piezas están en posesión de las personas o almacenadas en las bóvedas de éstas.

Cada institución bancaria pone a disposición de sus clientes las piezas en diferentes denominaciones, los clientes, según sus necesidades acuden a las ventanillas de su banco o a los cajeros automáticos para realizar retiros y poder adquirir bienes y/o servicios, pagar sus compromisos, etc. El dinero pasa de mano en mano al realizar transacciones comerciales, y regresa a los bancos en forma de depósitos. Las piezas depositadas normalmente son entregadas a otros usuarios que las soliciten vía retiro y las piezas siguen en circulación, salvo cuando las piezas están deterioradas, en cuyo caso estas piezas ya no puestas en circulación.

Cuando las instituciones bancarias tienen demasiado efectivo en su poder hacen lo que cualquier persona, acuden a su banco a depositar. Para las instituciones tener el dinero almacenado en sus bóvedas es similar a "guardarlo bajo el colchón", ya que no genera intereses y si representa un gasto por la infraestructura que requiere. Las instituciones acuden al Banco Central a realizar operaciones de depósito que se ven reflejadas en la cuenta que mantienen con el Banco Central.

Cuando el dinero ya está en su cuenta, las instituciones bancarias pueden invertir, comprar, prestar, etc., en otras palabras pueden hacer que su dinero sea productivo. Por otra parte las piezas depositadas son sujetas a varios procesos dentro de los Bancos Centrales, ahí se recueñtan para verificar que la cantidad que indicaron sea la depositada. Las piezas deterioradas son almacenadas en otras bóvedas donde esperaran el momento de su destrucción. Las piezas aptas

CAPÍTULO 1. Antecedentes

son almacenadas en bóvedas acondicionadas para esto y volverán a la circulación cuando un banco se presente a realizar un retiro y seguirán en circulación.

El proceso de destrucción marca el fin del ciclo de vida de una pieza. Los métodos de destrucción son muy variados, pero todos garantizan que esa pieza nunca va a volver a circular. Las piezas destruidas son sustituidas por piezas nuevas.

En resumen, el ciclo de vida de un billete o moneda comienza con la autorización de emisión, luego, la pieza se fabrica, se entrega al Banco Central, éste la pone a disposición de los bancos, los bancos las retiran con cargo a su cuenta, éstos lo ponen a disposición del público en general, donde permanecen en una interacción entre los bancos y clientes hasta que las piezas se desgastan a tal grado que son retiradas de la circulación y son destruidas. Y en su lugar entran nuevas piezas que tendrán el mismo ciclo de vida. En la figura 1-1 se ilustra el proceso descrito anteriormente.

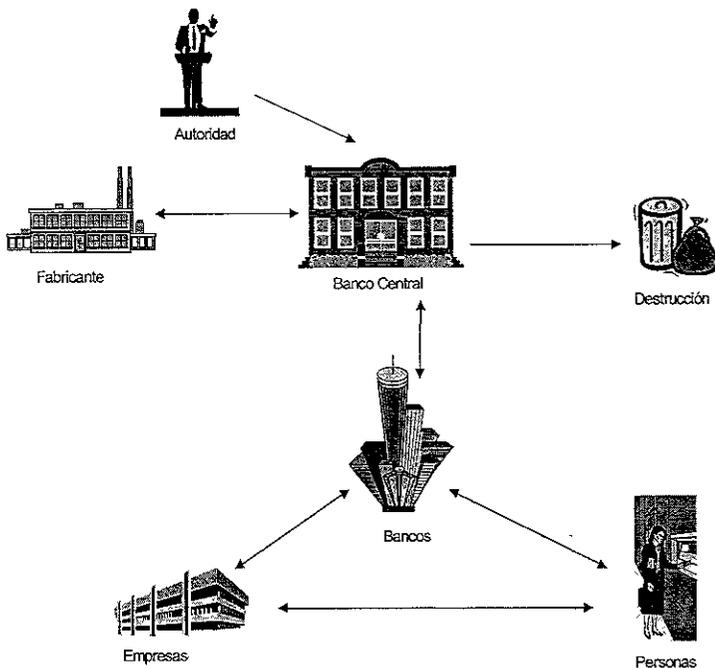


Figura 1-1. Ciclo de vida del efectivo.

1.4 Conceptos básicos

Se presentan a continuación algunos de los conceptos que se tomaron como base de partida para resolver la problemática que planteaba el desarrollo del Sistema de Control de Flujos de Caja de un Banco Central. Los temas tratados incluyen : el Diseño de Bases de Datos, una introducción a la arquitectura Cliente - Servidor y una descripción de las metodologías comunes para la construcción de sistemas computacionales.

1.4.1 Diseño de Bases de Datos

El diseño de una Base de Datos requiere de una gran atención en la naturaleza y propósito de los datos, así como la finalidad e impacto de la Base de Datos dentro de la organización.

El proceso de diseño persigue las siguientes metas:

- Proveer de una estructura sólida para representar los componentes y sus interrelaciones en el sistema de información modelado.
- Especificar que tipo de información se debe coleccionar, como hacerlo y el procedimiento para mostrar los resultados del análisis de la misma.
- Garantizar que ninguna de las funciones del negocio que refleja el modelo sea excluida.

Existen 5 fases para el diseño de una Base de Datos:

1. Recolección y entendimiento de los requerimientos del negocio.

En esta fase se recolectan la descripción y objetivos de la empresa, las reglas del negocio así como las funciones que hacen uso de los datos.

2. Creación de un modelo conceptual

En esta fase se desarrolla una representación rigurosa y detallada que describe el sistema de información, a dicha representación se conoce como modelo conceptual.

3. Creación de un modelo lógico

En esta fase se normaliza el modelo conceptual de acuerdo a la metodología impuesta por el modelo abstracto de datos elegido (por ejemplo, jerárquico, relacional ó multidimensional). El modelo lógico obtenido encaja en cualquier sistema manejador de bases de datos construido para dar soporte al modelo abstracto de datos seleccionado.

4. Creación del modelo físico

En esta fase se completan los siguientes pasos:

- Identificar el DBMS (*Database Management System*, Sistema Administrador de Bases de Datos) a utilizar.
- Identificar los índices
- Considerar y modelar las capacidades específicas DBMS.
- Identificar los puntos en donde se requiere un mejor desempeño.

5. Implementación

En esta fase se traduce el modelo físico al lenguaje de definición de datos del DBMS. Se toman en cuenta las características y limitaciones del lenguaje de programación, y configuración del hardware.

En la figura 1-2 se observan cada una de las fases que integran el diseño de Bases de Datos y como este proceso sigue una estructura en forma de espiral.

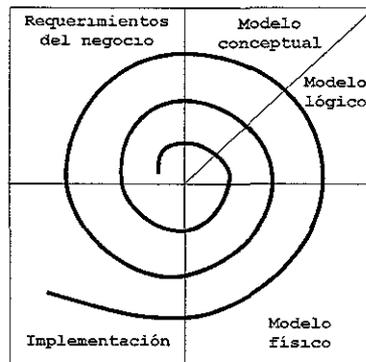


Figura 1-2. Ciclo de vida del diseño de bases de datos.

Modelo Conceptual

La creación de un modelo conceptual, un modelo lógico y un modelo físico son los pasos previos para llevar a cabo una buena definición de la Base de datos antes de su implementación. El diagrama entidad - relación es un tipo de modelo conceptual. Hay que señalar que el modelo conceptual es independiente del modelo de Bases de Datos (jerárquico, multidimensional o relacional). Por ejemplo, el modelo conceptual es independiente de la definición de llaves primarias, llaves foráneas, índices, al igual que la normalización; elementos propios del modelo de Bases de datos Relacional.

El modelo lógico refina el modelo conceptual, agregando la definición de llaves primarias y foráneas al igual que la normalización. El modelo físico implementa la funcionalidad específica del sistema administrador de bases de datos. Esta funcionalidad incluye la creación de índices, localización de datos, etc.

Debido a que el diagrama entidad-relación se basa en una metodología lo suficientemente sólida para obtener el modelo conceptual, se tratará con más detalle a continuación.

Diagrama entidad - relación

El diagrama entidad - relación engloba la definición de “entidades”, “atributos”, “llaves” y “relaciones”:

Entidad : Una entidad es el conjunto definido de elementos similares pero distinguibles entre ellos cuyas características y estado son importantes para el negocio. Los elementos pueden ser tangibles (como el inventario, partes, personal) o abstractos (como las metas de una empresa). Cada elemento particular en el conjunto es llamado instancia de la entidad.

Una entidad debe consistir de elementos que sean similares (todos ellos son descritos por el mismo conjunto de características), pero distinguibles (dos instancias no pueden tener exactamente el mismo valor para todas las características compartidas).

Atributo : Un atributo es una característica que describe a una entidad o representa su estado. Cada entidad tiene su propio conjunto de atributos. Por ejemplo la entidad “Parte” puede tener atributos como numero de inventario, peso, volumen, altura y color al igual que la entidad “Empleado” tiene los atributos nombre, oficina y teléfono.

Dominio de atributo : El dominio es el conjunto de valores a partir del cual uno o más atributos toman su valor. Por ejemplo, si una compañía lleva el registro del número telefónico de cada empleado, así como el número telefónico de cada uno de sus clientes, entonces existe un dominio común: números telefónicos. A partir de este dominio toman sus valores estos dos atributos, aun cuando cada uno de ellos pertenece a una entidad distinta.

Valor : El valor es el dato almacenado en un determinado atributo para una instancia en particular. Por ejemplo, el valor para el atributo color para una determinada instancia de “Parte” es “azul”.

Un atributo puede tener cualquiera de los siguientes tipos de valores:

- Un valor único

- Un valor NULO
- Un grupo de valores (atributo multivaluado)
- Valores derivados

Un valor único implica que el atributo acepta solamente un valor en todo momento, no pudiendo ser éste un dato nulo.

Un valor NULO puede significar:

- El valor del atributo se desconoce en ese momento. Por ejemplo, un empleado nuevo no puede tener asignado su número de extensión telefónica el primer día de trabajo.
- No hay un valor apropiado de atributo para una determinada instancia. Por ejemplo la entidad Empleado contiene un atributo Comisión que solo aplica para algunos empleados, digamos aquellos que pertenecen al departamento de ventas, pero no aplica para otros como aquellos que pertenecen al departamento de contabilidad.

Un atributo multivaluado acepta cualquier número de valores no incluyendo el valor nulo.

Un valor derivado es aquel que ha sido resultado de realizar alguna operación entre los valores de otros atributos.

Llave : Una llave es el conjunto de atributos susceptibles de ser utilizados para distinguir una instancia del resto de las demás. La llave también es conocida como "identificador", debido a que una llave se utiliza para identificar una instancia particular.

En la fase conceptual sólo es necesario identificar los atributos que pueden llegar a ser llaves primarias. A este conjunto de atributos se le llama llaves candidatas y cumplen con las siguientes características:

- La llave debe tener un valor único para cada instancia de la entidad.
- El valor de la llave debe estar disponible todo el tiempo, en otras palabras nunca puede tener un valor NULO.

Puede darse el caso en donde sea necesario utilizar una combinación de dos o más atributos para identificar a cada instancia. Una llave que está formada por más de un atributo es llamada llave compuesta.

Relación : Una relación es un enunciado que describe como una entidad se asocia o conecta con otra. Por ejemplo el enunciado "El empleado trabaja en

proyectos" describe la relación "trabaja en ", entre las entidades Empleado y Proyecto.

Cardinalidad de la relación : La cardinalidad describe el número de instancias de dos entidades que pueden ocurrir en una relación.

Se dan los siguientes valores para la cardinalidad de una relación:

- Uno a uno: se permite solo una instancia de cada entidad para una instancia de la relación.
- Uno a muchos o muchos a uno: una instancia de la relación permite muchas instancias de una entidad pero sólo una instancia de la otra entidad.
- Muchos a muchos: ambas entidades pueden tener muchas instancias participando en una instancia de la relación.

Opcionalidad de una relación: Además de especificar la cardinalidad de una relación, se debe descubrir si la relación es opcional o no.

Considere el ejemplo en donde la relación que existe entre la entidad Libro y la entidad Copia es "Libro-tiene-Copia". Las posibles combinaciones de cardinalidad y opcionalidad son:

- Exactamente uno: La cardinalidad es 1 y la relación se requiere, "Un libro debe tener exactamente una copia".
- A lo más uno: La cardinalidad es 1 pero la relación es opcional, "Un libro puede tener a lo más una copia".
- Al menos una: La cardinalidad es M y la relación se requiere, "Un libro debe tener al menos una copia".
- Cualquier número La cardinalidad es M pero la relación es opcional, "Un libro puede tener cualquier número de copias, incluso ninguna".

Simbología utilizada en un diagrama entidad - relación

Una entidad se representa por el nombre de la entidad dentro de un rectángulo.

Los atributos se listan dentro del rectángulo y debajo de la línea que los separa del nombre de la entidad. Los atributos que son llaves candidatas se listan primero y subrayados. La figura 1-3 muestra la representación gráfica de una entidad.

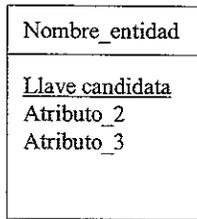
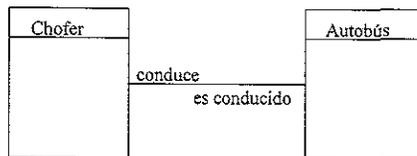


Figura 1-3. Representación de una entidad.

Para representar la relación entre dos entidades binarias simples, basta con dibujar una línea entre ambas. Sobre la línea se colocan dos etiquetas que indican la relación en un sentido o en otro, como se muestra en la Figura 1-4.



En este ejemplo se da la relación "chofer-conduce-autobus" equivalente a "autobus-es conducido-chofer"

Figura 1-4. Representación de una relación.

En la Figura 1-5 se muestran los símbolos para representar las diferentes combinaciones cardinalidad - opcionalidad:

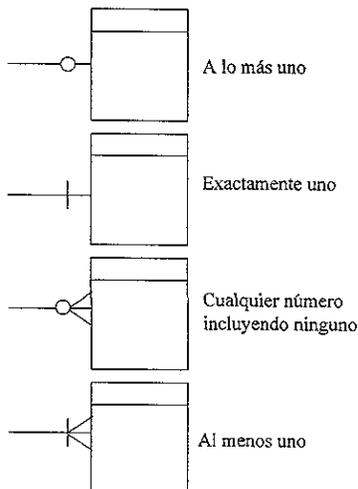


Figura 1-5. Simbología utilizada para representar cardinalidad – opcionalidad

Modelo Lógico

El modelo lógico es la transformación del modelo conceptual utilizando la metodología de alguno de los modelos de datos existentes (relacional, jerárquico o multidimensional).

En este caso únicamente se describirá el modelo de datos relacional.

El proceso de transformación incluye:

- La normalización del modelo conceptual
- Identificar llaves primarias
- Utilizar llaves foráneas para representas las relaciones

Modelo de Datos Relacional

El modelo relacional fue formalmente introducido en 1970 por E.F. Codd. El modelo proporciona un simple concepto de como los usuarios perciben los datos. El modelo relacional representa a los datos en forma de tablas de dos dimensiones, cada tabla representa una entidad.

La organización de datos en tablas relacionales es conocida como la vista lógica de la base de datos, esto es, la forma en la cual una base de datos relacional presenta los datos al usuario y al programador. El modo en que son almacenados los datos físicamente es llamado vista interna.

En el modelo relacional, una base de datos es una colección de tablas relacionales. Una tabla relacional es un archivo compuesto por un conjunto de columnas nombradas (atributos) y un número arbitrario de renglones (instancias de entidad). Las columnas contienen información acerca de la tabla, los renglones de la tabla representan sucesos de la "cosa" representada en la tabla.

Cada valor es insertado en la intersección de un renglón y una columna. Cada columna pertenece a un dominio, el cual es el conjunto de valores permitidos que pueden aparecer en esa columna. Cada renglón es único, no pueden haber dos renglones con valores idénticos. La secuencia de las columnas y de los renglones es irrelevante.

En resumen la estructura de tabla del modelo de datos relacional tiene las siguientes características:

- Los renglones y las columnas no tienen un orden determinado, es decir, no se puede extraer ningún tipo de información considerando el orden de los renglones o columnas.
- No pueden existir dos renglones duplicados.

- Una tabla no puede contener nombres de columnas duplicados.
- Sólo se permite un valor en cada intersección renglón - columna.

Otros términos son usados para describir las tablas relacionales como campos y registros en lugar de columnas y renglones, respectivamente.

Los componentes de una base de datos relacional son:

Llaves primarias : Una llave primaria es una columna cuyos valores identifican un renglón-o registro dentro de una tabla. Una llave primaria puede consistir en una o más columnas o campos. Todas las columnas que contengan valores únicos son llamadas llaves y la llave primaria puede ser escogida de cualquiera de ellas, las restantes son nombradas llaves alternas. Una llave primaria compuesta por una columna es una llave simple mientras que una llave primaria hecha de varias columnas es una llave compuesta.

Una llave primaria debe cumplir con los siguientes criterios:

- Poseer un valor definido: El valor de una llave primaria no podrá ser NULO.
- El valor debe ser persistente: El valor nunca debe cambiar durante toda la vida de la instancia.
- Mínima: Si la llave primaria es compuesta debe ser el menor número de atributos.
- Accesible a todos los usuarios de la base de datos: Todos los usuarios que tengan acceso a la Base de datos podrán consultar la llave primaria.

Llave foránea : Una llave foránea es una columna cuyos valores son los mismos que los de la llave primaria de otra tabla. Al igual que las llaves primarias, las foráneas pueden consistir en una o más columnas.

Reglas de integridad : La integridad de la entidad establece que el valor de una llave primaria no puede ser nulo (Un valor nulo es aquel que no tiene valor, no es lo mismo que un espacio en blanco.)

La integridad de referencia indica que si una tabla relacional tiene una llave foránea, entonces cada valor de la llave foránea debe ser nulo o corresponder con los valores dentro de una tabla relacional en la cual esa llave foránea sea una llave primaria.

Relaciones : Una relación es una asociación entre dos o más tablas. Existen tres tipos de relaciones, uno a uno, uno a muchos y muchos a muchos.

En una relación uno a uno cada registro en una tabla tiene a lo mas un registro relacionado en otra tabla.

La relación uno a muchos es la más usual de todas, en ella un registro en una tabla está relacionado con varios registros en otra tabla.

Una relación muchos a muchos significa que por cada registro de una tabla pueden existir varios registros en otra tabla relacionados entre sí y por cada registro de la segunda tabla pueden existir varios registros relacionados en la primera tabla.

Normalización : Normalización es una técnica ampliamente usada como guía en el diseño de bases de datos relacionales. El objetivo de la normalización es crear un conjunto de tablas relacionales que estén libres de datos redundantes y que puedan ser modificados consistentemente y correctamente.

Existen cuatro reglas básicas en la normalización:

1. Cada tabla debe tener una llave única.
2. Eliminar grupos repetitivos.
3. Examinar relaciones de llaves compuestas. Esencialmente esto simplemente nos dice que donde tengamos una llave compuesta debemos ser capaces de eliminar esa tabla. Una llave compuesta está dada por una llave primaria basada en dos o más campos.
4. Examinar cualquier grupo de información restante. Si la información descriptiva no está directamente relacionada con la llave en esa tabla, entonces será necesario mover esa información a otra tabla diferente.

Las 5 formas de normalización están definidas por las siguientes reglas:

- 1ª. Forma normal: una tabla se encuentra en su primera forma normal cuando cada columna en cada renglón contiene un solo valor. En otras palabras, cada atributo de una entidad debe contener un valor atómico mismo que ya no puede ser descompuesto en unidades más pequeñas. En cualquier caso, los requerimientos del negocio indican cuando un atributo es atómico.
- 2ª. Forma normal: una tabla se encuentra en su segunda forma normal cuando además de estar en su primera forma normal, cada atributo que no forma parte de la llave primaria es funcionalmente dependiente de la llave primaria. Se tiene la siguiente definición de dependencia funcional.- Considere dos atributos, A y B. A determina funcionalmente B y B es funcionalmente dependiente de A si cuando dos renglones coinciden en el valor de A también coinciden en el valor de B. La definición es válida cuando se consideran los atributos por separado o en conjunto.
- 3ª. Forma normal: una tabla se encuentra en su tercera forma normal si está en su segunda forma normal y además para cada atributo que no es parte de

la llave primaria éste es directamente dependiente de la llave primaria completa.

- 4ª. Forma normal: una tabla se encuentra en su cuarta forma cuando todas las dependencias multivaluadas en la tabla son también dependencias funcionales cuyo determinante es una llave candidata.
- 5ª. Forma normal: una tabla está en su quinta forma normal cuando todas las dependencias *join* que pueden ser descompuestas han sido de hecho descompuestas.

1.4.2 Arquitectura Cliente - Servidor

La arquitectura Cliente - Servidor es una combinación de *hardware* y *software* que permite al usuario dividir el procesamiento de los datos entre la estación de trabajo del usuario y el servidor. La interface de usuario se ejecuta en el cliente mientras que la base de datos reside en el servidor. Representa la fusión de varios elementos, los tres principales componentes son el cliente, el servidor y la red. Juntos permiten procesamiento distribuido, análisis de datos y presentación de los mismos. En la figura 1-6 se muestran los componentes de la arquitectura cliente-servidor y la forma en que se relacionan.

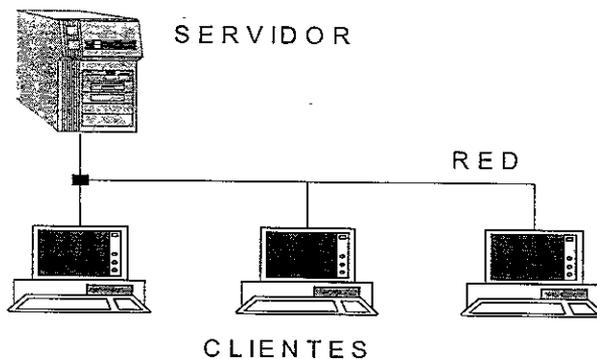


Figura 1-6. Elementos de la arquitectura Cliente - Servidor.

Un cliente es una computadora o estación de trabajo conectada a una red que permite el acceso a los recursos de la red. Presenta una interfaz de usuario que

permite obtener las instrucciones o comandos directamente del usuario para propósitos de recolección y análisis de datos, así como también es el medio para presentar los resultados al usuario.

Las instrucciones o comandos son ejecutadas en un lenguaje predefinido. Estas instrucciones no son necesariamente enviadas al servidor para su ejecución, es posible que el cliente mismo procese algunas de éstas para de esta forma reducir la carga de trabajo del servidor o realizar un control de acceso y seguridad; sin embargo, es recomendable que el cliente no realice estas funciones. Las aplicaciones de los clientes pueden ser bases de datos, hojas de cálculo, procesamiento de texto o cualquier otra aplicación específica.

Un servidor es un procesador que provee servicio a uno o más clientes, almacena la información, procesa las solicitudes de información de los clientes y controla el acceso a los datos y la seguridad, únicamente responde a los comandos de los clientes, por lo que el servidor no inicia la conversación con el cliente. Un servidor ideal esconde el esquema completo Cliente - Servidor a los usuarios, un cliente no deberá estar consciente de los elementos que integran al servidor. Un servidor puede ser una microcomputadora, mimicomputadora o un *mainframe*, es posible tener varios servidores conectados a través de la red.

Existen varios tipos de servidores dependiendo de las funciones que proporcionan a los clientes. En la figura 1-7 se ilustran algunos tipos de servidores que incluyen : servidor de aplicaciones, servidor de datos, servidor de impresión, servidor de archivos y programas.

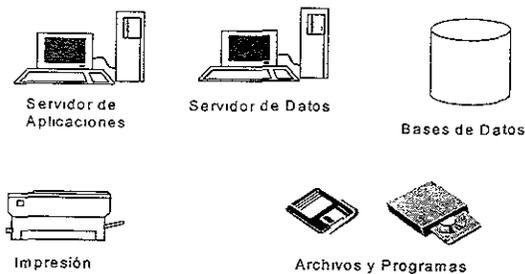


Figura 1-7. Ejemplos de tipos de servidores

La red provee un canal de comunicación entre los clientes y el servidor, consiste de hardware (cables e interfaces de red) y de software (sistema operativo de red) y transmite los datos a lo largo de todos los clientes y servidores.

La red puede ser vista como un modelo de tres capas:

- Aplicación
- Transporte
- Red

La transparencia entre el cliente y el servidor la proporciona la capa de transporte, dentro de la capa de red y de transporte tenemos a los protocolos de comunicaciones que son los que establecen la comunicación entre las partes, el protocolo más comúnmente utilizado es TCP/IP (*Transport Control Protocol / Internet Protocol*, Protocolo de control de transporte / Protocolo de Internet). En un ambiente LAN (*Local Area Network*, Red de Area Local) las responsabilidades del servidor a menudo están divididas entre servidor de datos y servidor de aplicaciones.

Dentro de un ambiente de bases de datos uno o más clientes dentro de la red hacen requerimientos al servidor. El servidor tiene un control absoluto sobre el acceso a los datos, así como la responsabilidad de mantener la integridad y consistencia de los mismos. Por tanto, se hace necesario proveer a los clientes de una interface bien definida para formular los requerimientos y enviarlos al servidor. Desde hace varios años SQL (*Structured Query Language*, Lenguaje de Requerimientos Estructurados) es el lenguaje más popular para la arquitectura Cliente – Servidor.

Diferentes proveedores de aplicaciones utilizan SQL para comunicarse entre sí, utilizando comandos para entrar, recolectar y manipular información. Una interface de usuario acepta la entrada de comandos del usuario y genera un código apropiado en SQL para procesar los datos. El comando es enviado a través de la red o cualquier otro medio de comunicación hacia el servidor, el cual procesa el requerimiento. El resultado es enviado de vuelta hacia el cliente y es desplegado por la interface de usuario del cliente.

1.4.3 Metodologías

El desarrollo de Software involucra el seguimiento de alguna metodología que proporcione una base teórica e imponga un orden en los pasos a seguir durante todo el proceso del desarrollo. Las metodologías tratadas incluyen el Ciclo de Vida y el Modelo de Prototipos.

Ciclo de Vida

La Figura 1-8 ilustra el modelo del Ciclo de Vida para la Ingeniería de Software. Algunas veces llamado el “Modelo en Cascada.”

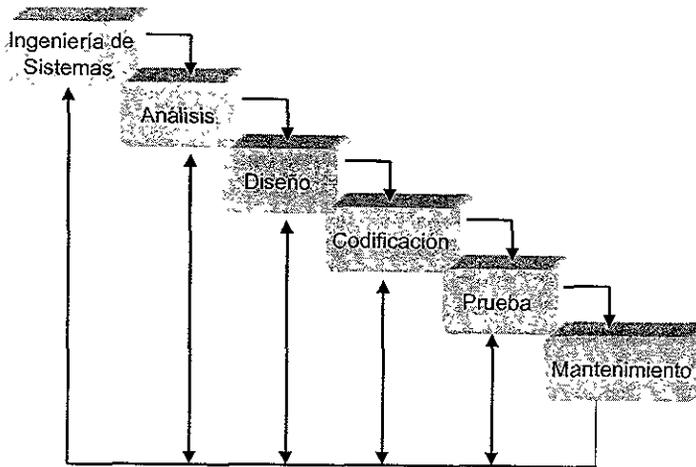


Figura 1-8. Modelo Ciclo de Vida o Modelo en Cascada.

El modelo del Ciclo de Vida exige un enfoque sistemático y secuencial del desarrollo del software y abarca las siguientes actividades : Ingeniería y análisis del sistema, análisis de los requerimientos de software, diseño, codificación, prueba y mantenimiento.

Ingeniería y análisis del sistema

Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requerimientos de todos los elementos del sistema y luego asignado algún subconjunto de estos requerimientos del software. Esta visión del sistema es esencial cuando el software debe interrelacionarse con otros elementos tales como hardware, personas y bases de datos. La ingeniería y el análisis del sistema abarcan los requerimientos globales a nivel del sistema con una pequeña cantidad de análisis y diseño a nivel superior.

Análisis de los requerimientos del software

El proceso de recogida de los requerimientos se centra e intensifica especialmente en el software. Para comprender la naturaleza de los programas que hay que construir, el Ingeniero de software debe comprender de dominio de la información del software, así como la función, rendimiento e interfaces requeridas. Los requerimientos tanto del sistema como del software se documentan y se revisan con el cliente.

Diseño

El diseño del software es realmente un proceso multipaso que enfoca sobre tres atributos distintos del programa: estructura de datos, arquitectura del software y detalle procedural. El proceso de diseño traduce los requerimientos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Como los requerimientos, el diseño se documenta y forma parte de la configuración del software.

Codificación

El diseño debe traducirse en una forma legible para la máquina. El paso de la codificación ejerce esta tarea. Si el diseño se ejecuta 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 enfoca sobre la lógica interna del software, asegurando que todas las sentencias se han aprobado, y sobre las funciones externas, esto es, realizando pruebas para asegurar que la entrada definida producirá los resultados que realmente se requieren.

Mantenimiento

El software sufrirá indudablemente cambios después de que se entregue al cliente (una posible excepción es el software empotrado). Los cambios ocurrirán debido a que se ha encontrado errores, debido a que el software debe adaptarse por cambios del entorno externo (por ejemplo, un cambio solicitado debido a que se tiene un nuevo sistema operativo o dispositivo periférico), o debido a que el cliente requiere aumentos funcionales o del rendimiento. El mantenimiento del software se aplica a cada uno de los pasos precedentes del ciclo de vida a un programa existente en vez de uno nuevo.

El ciclo de vida es el más viejo y más ampliamente usado en la ingeniería del software. Sin embargo, con el paso de algunos cuantos años, se han producido críticas al modelo, incluso por seguidores activos que cuestionan su aplicabilidad a todas las situaciones. Entre los problemas que se presentan algunas veces, cuando se aplica el modelo de ciclo de vida, se encuentran los siguientes:

1. Los proyectos reales raramente siguen el flujo secuencial que propone el modelo. Siempre ocurren iteraciones y se crean problemas en la aplicación del modelo.

2. Normalmente es difícil para el cliente establecer explícitamente al principio todos los requerimientos. El ciclo de vida requiere esto y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos proyectos.
3. El cliente debe tener paciencia. Una versión funcionando del programa no estará disponible hasta las etapas finales del desarrollo del proyecto. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso.

Cada uno de estos problemas es real. Sin embargo, el modelo del ciclo de vida tiene un lugar definido e importante en el trabajo sobre ingeniería del software. Suministra una plantilla en la que puede colocarse los métodos para el análisis, diseño, codificación, prueba y mantenimiento. Además, los pasos del modelo son muy similares a otros modelos aplicables a la ingeniería del software. Sin embargo, es el modelo procedural más usado por los ingenieros de software.

Construcción de prototipos

Normalmente un cliente definirá un conjunto de objetivos generales para el software, pero no identificará los requerimientos detallados de entrada, procesamiento o salida. En otros casos el programador puede no estar seguro de la eficiencia de un algoritmo, la adaptabilidad de un sistema operativo o la forma en que debe realizarse la interacción "hombre – máquina". En estas y muchas otras situaciones, puede ser mejor método de ingeniería del software realizar un prototipo.

La construcción del prototipo es un proceso que facilita al programador la creación de un modelo del software a construir. El modelo tomará una de las tres formas siguientes: un prototipo en papel que describa la interacción "hombre – máquina" de forma que facilite al usuario la comprensión de cómo de producirá tal interacción; un prototipo que funcione que implementa algunos subconjuntos de la función requerida al software deseado; o un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

La secuencia de sucesos para el modelo de construcción de prototipos se muestra en la Figura 1-9.

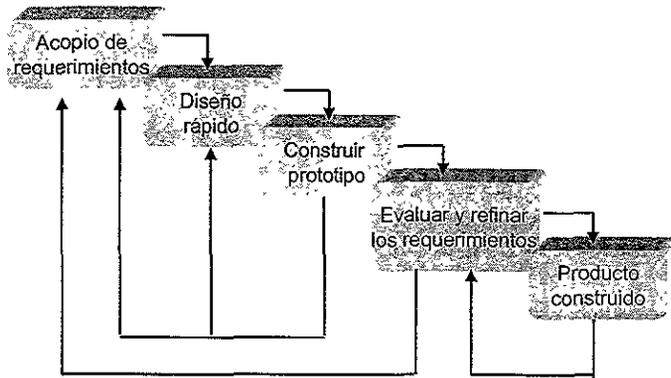


Figura 1-9. Modelo "Construcción de Prototipos".

Como todos los métodos de desarrollo de software, la construcción de prototipos comienza con la recolección de los requerimientos. El técnico y el cliente se reúnen y definen los objetivos globales para el software, identifican todos los requerimientos conocidos y perfilan las áreas en donde será necesario una mayor definición. Luego se produce un "diseño rápido" se enfoca sobre la presentación de los aspectos del software, visibles para el usuario (por ejemplo, métodos de entrada y formatos de salida). El diseño rápido conduce a la construcción de un prototipo. El prototipo es evaluado por el cliente/usuario y se utiliza para refinar los requerimientos del software a desarrollar. Se produce un proceso interactivo en el que el prototipo es "afinado" para que satisfaga las necesidades del cliente, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer. Idealmente, el prototipo sirve como un mecanismo para identificar los requerimientos del software. Si se construye un prototipo que funciona, el realizador intenta hacer uso de fragmentos de programas existentes o aplica herramientas (por ejemplo, generadores de informes, gestores de ventana, etc.) que faciliten la rápida generación de programas que funcionen.

El prototipo puede servir como "el primer sistema". Pero esto puede ser una visión idealizada. Como en el ciclo de vida, la construcción de prototipos, como modelo para ingeniería del software, puede ser problemática por las siguientes razones:

1. El cliente ve funcionando lo que parece ser una versión del software, ignorando las prisas en hacer que funcione, no hemos considerado los aspectos de calidad o mantenimiento a largo plazo del software. Cuando se

le informa que el producto debe ser reconstruido, el cliente por lo general se molesta y solicita que se apliquen las mejoras necesarias para hacer el prototipo un producto final que funcione. Aunque muchas veces el gestor del desarrollo del software cede.

2. El técnico de desarrollo realiza frecuentemente ciertos compromisos de implementación en orden a obtener un prototipo de función rápidamente. Puede utilizarse un sistema operativo o lenguaje de programación inapropiado simplemente porque está disponible y es conocido; un algoritmo ineficiente puede implementarse de forma sencilla para demostrar su capacidad. Después de pasar algún tiempo en el que el técnico estaba familiarizado con estas elecciones, olvida las razones por las que eran inapropiadas. La elección menos ideal forma ahora parte integral del sistema.

Aunque se pueden presentar problemas, la construcción de prototipos es un modelo efectivo para la ingeniería del software. La clave está en definir al comienzo las reglas del juego; esto es, el cliente y técnico deben estar de acuerdo en que el prototipo se construya para servir sólo como un mecanismo de definición de los requerimientos. Posteriormente ha de ser descartado (al menos en parte) y debe construirse el software real, con los ojos puestos en la calidad y mantenimiento.

Una vez mencionados los antecedentes necesarios, tanto históricos como técnicos, para entender la problemática de los sistemas de flujos de efectivo, en el siguiente capítulo señalaremos los requerimientos por parte de los Bancos Centrales en esta materia.

2 Requerimientos

En este capítulo mencionaremos la forma en que operan los sistemas de flujos de efectivo, la problemática actual y señalaremos las necesidades para conseguir un sistema de flujo que se adecue a las prioridades de los Bancos Centrales.

2.1 *Situación actual*

Los Bancos Centrales se encargan de controlar las emisiones de billetes y monedas. Como hemos visto, una autoridad, que puede ser ajena al Banco Central, autoriza la emisión de un determinado número de billetes o monedas de cierta denominación. Cada emisión, a pesar de ser de la misma denominación, puede ser distinta; en la emisión se especifican las características de las piezas involucradas, en el caso de los billetes, se menciona el diseño tanto en el anverso como en el reverso, las leyendas que incluye, etc.; en el caso de las monedas además, se mencionan los materiales que las conforman, indicando el porcentaje utilizado en cada aleación. Dado que los Bancos Centrales solamente pueden emitir hasta el número de piezas que les hallan autorizado, con las características señaladas, se cuenta con un registro contable donde controlan cuántas piezas de cada denominación fueron autorizadas, cuántas piezas se han mandado fabricar y cuántas aún no han sido entregadas.

Cuando las instituciones encargadas de la fabricación concluyen su tarea, entregan las piezas a los Bancos Centrales, a partir de este momento forman parte de las existencias de los Bancos Centrales.

Los Bancos Centrales cuentan con bóvedas donde almacenan las piezas. Por la naturaleza de las piezas, existen varias bóvedas, en unas bóvedas se almacenan billetes nuevos, en otras, billetes usados, en otras, billetes deteriorados, en otras, monedas nuevas, etc., obedeciendo a las políticas de cada Banco Central.

También cuentan con espacios destinados a los procesos propios del manejo y la verificación de las piezas a las que llamaremos ventanillas. En estas ventanillas se llevan las piezas para ser recontadas y verificar que el número de

piezas reportadas como entregadas, corresponda a la realidad. También existe otro tipo de ventanillas de atención a la banca. La razón por la cual llamamos ventanilla indistintamente es por la característica común que tienen estos dos espacios, al cierre del día operativo las ventanillas no deben tener ninguna pieza, todas deben estar almacenadas en las bóvedas.

Como el objetivo de la emisión de dinero es abastecerlo al país, los Bancos Centrales se auxilian de las instituciones bancarias, éstas acuden al Banco Central a retirar dinero en efectivo, utilizando para esto, la cuenta que mantienen en el banco.

Si pensamos los efectos que causaría que todas las sucursales bancarias acudieran al Banco Central para retirar dinero, podemos imaginarnos que serían insuficientes los espacios con que cuentan los Bancos Centrales para realizar esta labor, por otra parte, si imaginamos la complejidad de los controles de los bancos para controlar cuanto dinero han retirado en cada una de sus sucursales, el problema se incrementa, ya que los bancos mantienen una cuenta por institución en el Banco Central y cada sucursal tendría que reportar cuanto dinero van a retirar para que la tesorería de cada banco hiciera las operaciones necesarias para contar con los fondos. Así, la solución implantada por los Bancos Centrales es reducir el número de sucursales por banco que pueden retirar o depositar dinero. Las sucursales que acuden al Banco Central por dinero en efectivo se encargan posteriormente de abastecer a las demás sucursales y así poder ofrecer el servicio a sus clientes.

Debido a las dimensiones territoriales de cada país encontramos que es difícil que una sola sucursal pueda abastecer oportunamente de efectivo a todas las sucursales del país, algunos Bancos Centrales cuentan con sucursales en varias ciudades dividiendo el territorio nacional en sectores, así cada sucursal se encarga de abastecer a las sucursales bancarias de su jurisdicción.

En países con una extensión considerable, esta medida no resulta suficiente para lograr el abasto, ya que las distancias entre las ciudades y la sucursal que les corresponda pueden ser considerables, por otro lado, resulta inadecuado que el Banco Central cuente con sucursales en todas las plazas¹, ya que esto provocaría una infraestructura de altas dimensiones.

Para solucionar este problema, los Bancos Centrales que se han enfrentado a esta situación, han recurrido a las propias sucursales bancarias para que funjan como oficinas o sucursales del Banco Central, a estas sucursales les llamaremos corresponsales. La forma de identificar a un corresponsal es considerando la institución bancaria y la plaza donde se localiza. Puede darse el caso que una institución tenga más sucursales en la misma plaza, sin embargo los Bancos Centrales no ocupan más de una sucursal, así que la información referente a la ubicación física del corresponsal es importante para el control de

¹ Llamaremos plaza a un poblado que cuente con al menos una sucursal bancaria.

corresponsales pero no para el control de flujos de efectivo. El objetivo de los corresponsales es recibir las operaciones de retiro de las sucursales bancarias en la demarcación que le corresponde. De esta forma el abastecimiento de efectivo a todas las sucursales dentro del territorio nacional está garantizado.

Los Bancos Centrales son los encargados de enviar las piezas hacia sus sucursales o sus corresponsales. Normalmente, cuando un Banco Central tiene sucursales cada sucursal se encarga de controlar y abastecer a los corresponsales de su zona.

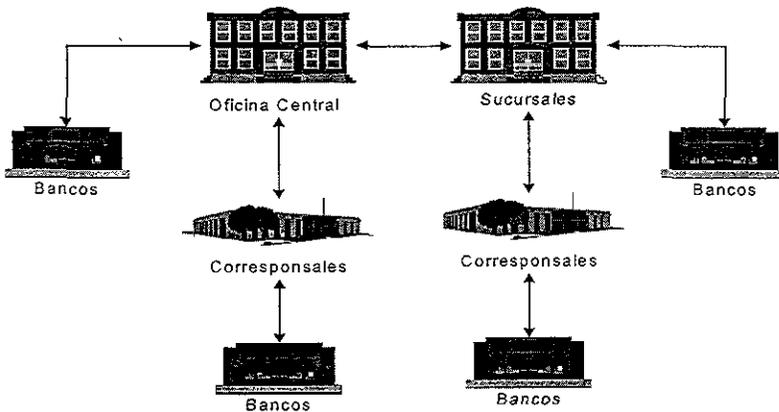


Figura 2-1. Infraestructura de los Bancos Centrales para distribuir efectivo.

Una necesidad imperante de los Bancos Centrales es la de conocer, en todo momento, donde están localizadas cada una de las piezas de su propiedad, es decir, cuántas piezas y de qué denominación están almacenadas en cada una de sus sucursales, en sus corresponsales y por supuesto en su oficina central.

Las piezas tienen por sí mismas un estado de desgaste, a lo que llamaremos en lo sucesivo estado físico, que podemos clasificar como nuevo, cuando las piezas nunca han salido de los Bancos Centrales a circulación; apto, cuando las piezas ya han estado en circulación pero su grado de desgaste permita que siga circulando y deteriorado, cuando por el grado de desgaste ya no sea conveniente que estas piezas sigan circulando.

Dado que en las diferentes localidades donde los Bancos Centrales tienen almacenadas piezas de su propiedad, a las que llamaremos en lo sucesivo sitios, se realizan varios tipos de operación que afectan la ubicación de las existencias, los Bancos Centrales cuentan con controles para determinar

cuántas piezas entran y cuántas salen, que es lo que se conoce como flujos de efectivo o flujos de caja.

Mientras las piezas se encuentren dentro del Banco Central ya sea en su oficina central, en sus sucursales o en sus corresponsales, son consideradas como propiedad del Banco Central y por tanto forman parte de sus existencias.

Como las piezas son objetos físicos, su traslado a las sucursales o corresponsales implican un movimiento físico de ellas; dependiendo de la distancia que separe al sitio origen del sitio destino será el tiempo que ocupe el traslado. Mientras las piezas estén viajando no es posible determinar el lugar físico donde se encuentran, así que, mientras las piezas no lleguen al sitio destino las piezas se consideran que están en "tránsito". Entonces, al salir las piezas del sitio origen dejan de formar parte de las existencias de este sitio para formar parte de las existencias en tránsito, al ingresar las piezas al sitio destino éstas se incluyen en las existencias de dicho sitio y dejan de formar parte de tránsito.

De esta forma es como se abastece a las sucursales y a los corresponsales para que ellos puedan responder a la demanda de los bancos.

Hasta este momento hemos visto como fluyen las piezas de un sitio a otro, pero, dentro de cada sitio, también se realizan un gran número de movimientos. Principalmente trasposos de piezas de una bóveda a otra o de una bóveda a una ventanilla y viceversa. Dado que algunas ventanillas son de proceso, es decir, ahí se realizan recuentos, clasificaciones, ordenamiento de las piezas, etc., es importante conocer también en donde están las piezas dentro de un sitio.

Una operación muy particular es la clasificación de piezas de la misma denominación pero con diferentes características. Como mencionamos al inicio de este apartado, en la emisión de billete y de moneda se especifican las características de las piezas, así desde la emisión hasta el almacenamiento se tiene control sobre las piezas según sus características, sin embargo, una vez que las piezas comienzan a circular ya es muy difícil controlar las existencias a este nivel. Por ejemplo, recordemos el caso de México en 1993 cuando se emitieron billetes y monedas con la leyenda "Nuevos Pesos", después de un tiempo se emitieron nuevas familias de billetes y monedas con las mismas características que los anteriores exceptuando la leyenda que ahora decía "Pesos", durante el periodo de transición los bancos depositaban indistintamente billetes y monedas con estas leyendas, clasificar las piezas según su leyenda hubiera resultado una labor titánica, tanto para los bancos como para el Banco Central realizarla en el momento del depósito. Para evitar este tipo de conflictos las piezas con la misma denominación y mismo valor real, son manejadas como de un solo tipo sin importar sus características, sin embargo, al momento de destruir, sí es importante considerarlas.

Con el avance de la tecnología los Bancos Centrales han implantado sistemas que les permiten, hasta cierto grado, controlar las existencias, sin embargo, no en todos los casos estos sistemas satisfacen las necesidades de las áreas encargadas del control al nivel de detalle requerido.

2.2 Problemática

Un balance simplificado de un Banco Central incluye como activos a la reserva internacional y el crédito interno bruto, que es el financiamiento que los Bancos Centrales hacen a las instituciones bancarias del país; y como pasivos, los depósitos bancarios, los cuáles sirven para determinar los sobregiros permitidos en su cuenta, y los billetes y monedas ya que son compromisos que se tienen con la sociedad.

El requerimiento contable ocasiona que el control de billetes y monedas tenga que estar incluido en la contabilidad. Al ser el aspecto contable de alta importancia, es común que los sistemas de los Bancos Centrales en materia de emisión y control de los flujos de efectivo estén orientados a este fin, relegando las necesidades de las áreas involucradas directamente a los flujos.

Por otro lado, en la contabilidad el efectivo debe representarse en el monto equivalente y no en piezas, si consideramos que los encargados de controlar los flujos de efectivo necesitan conocer el número de piezas para contemplar los espacios físicos necesarios para su almacenamiento y traslado, el control únicamente contable puede resultar inconveniente para las áreas operativas.

Además, operaciones de interés interno, como los traspasos pudieran no tener cabida en la contabilidad, lo que ocasiona que el control de flujos de efectivo esté fragmentado, complicando la explotación de la información, lo que puede generar retrasos en la obtención de información, ya sea de control o para toma de decisiones. Además, por la confidencialidad de la información, el compartir los datos entre subsistemas puede resultar un proceso no muy transparente.

Un inconveniente adicional de los sistemas para el control de los flujos de efectivo con naturaleza contable, es la falta de información complementaria, es comprensible que los sistemas contables no requieran toda la información asociada a una operación, la función de este tipo de sistemas es efectuar los cargos y abonos a las cuentas correspondientes, llenarlos con información innecesaria para este fin sería complicar la administración de los mismos, considerando que la contabilidad de un Banco Central contiene la operación completa de la institución, visto como Banco Central y como empresa.

En los sistemas contables, no siempre es requerido un control por denominación para cada sitio, sin embargo, para el control es muy importante conocer este detalle. Otro aspecto que es necesario considerar, es conocer cuántas piezas

están realmente disponibles, como se mencionó anteriormente, dentro de los Bancos Centrales hay bóvedas de proceso, mientras las piezas estén ahí, realmente no están disponibles y no pueden salir a la circulación; un control adecuado de la ubicación interna y del estado de proceso en que se encuentran las piezas permitirá una mejor atención a los usuarios.

Para las áreas encargadas del análisis de necesidades de efectivo, la información histórica es esencial, conocer la demanda por denominación en las distintas plazas, son elementos fundamentales para la toma de decisiones, aunadas con los eventos propios del país, por ejemplo períodos vacacionales, fiestas de tipo religioso como la Navidad o el pago de gratificaciones, días de pago, etc. La información histórica debe considerar un adecuado manejo de fechas para evitar los problemas asociados al cambio de milenio.

Un problema común en varias corporaciones con un grado de sistematización elevado es la falta de estandarización de los catálogos comunes, por ejemplo, el catálogo de las instituciones con las cuáles operan los bancos centrales; el problema radica en que mientras en un sistema se le asocia una clave determinada, en otro sistema se le asocia otra clave completamente distinta, entonces, además de duplicar la función de mantenimiento se generan incompatibilidades de los sistemas dentro de la misma corporación, ocasionando mayor complejidad para extraer información complementaria de varios sistemas.

Una situación común en los Bancos Centrales, es que aprovechen la infraestructura construida para almacenar también otras divisas propiedad de él, aunque éstas no son emitidas por el Banco Central, son a fin de cuentas piezas con un atributo adicional que es el nombre de la divisa y también pueden estar sujetas a procesos de recuento y verificación. Es decir, las piezas en otras divisas también forman parte de las existencias de un sitio y por tanto, son controladas por el mismo Banco Central.

Por estas razones, la necesidad de un sistema cuya finalidad sea el control de flujos de efectivo es inminente, un sistema que controle las existencias en piezas y que a su vez pueda expresarlas en su equivalente en moneda nacional o en la moneda del país a la que corresponda al nivel deseado, que contenga la información útil completa para la contabilidad, la parte operativa y toma de decisiones, en forma diaria e histórica y permita una administración simple sin que por ello se descuide la seguridad de la información.

En resumen, el controlar las piezas y en general los flujos de efectivo, desde sólo un punto de vista contable ocasiona una pérdida en el nivel de detalle de las operaciones, que por consiguiente, acarrea los problemas de integración, oportunidad y consistencia de la información.

2.3 Control de existencias

Desde la autorización de emisión hasta justo antes de la entrega, las piezas están en un grupo de existencias "no materializadas", su existencia es únicamente en cuentas contables y registros de control. A través de estas cuentas se sabe cuántas piezas han sido autorizadas para cada tipo de pieza, cuántas piezas se han mandado fabricar y no se han entregado y por tanto, cuántas piezas se han mandado fabricar y ya fueron entregadas. Como se comentó en su oportunidad, una pieza de la misma denominación y mismo valor real puede tener características distintas a otra, el control en este grupo de existencias debe considerar este nivel de detalle. Como las piezas no existen físicamente, no pueden ser asociadas a un sitio específico.

En la entrega de efectivo, por parte del fabricante hacia el Banco Central, las piezas dejan de pertenecer al grupo de existencias "no materializadas" para formar parte de las existencias físicas del Banco Central. Las existencias físicas son asociadas a las bóvedas del sitio que las posea. Las piezas físicas pueden ser distinguidas según sus características o material, o bien, manejarse indistintamente, clasificadas en todo momento por la denominación y valor real de cada pieza. Además, se debe poder identificar cuántas piezas nuevas, aptas y deterioradas existen en las bóvedas de cada sitio.

Cuando las piezas salen de un sitio rumbo a otro, se desconoce la ubicación exacta de las piezas, como se explicó anteriormente, están en tránsito. Mientras las piezas se encuentren viajando, debe conocerse la cantidad de piezas por denominación y valor real que están involucradas, además del estado físico que guardan y cuál es el medio de transporte. En algunos casos, también es necesario conocer cuál es el sitio origen y cuál el sitio destino.

Las piezas deterioradas tienen que regresar al Banco Central para ser destruidas, por lo que los corresponsales concentran estas piezas en la sucursal del Banco Central que les corresponde. Cuando el Banco Central las destruye, las piezas dejan de existir físicamente, sin embargo, se necesita un control de éstas piezas. Antes de ser destruidas se clasifican no solamente por su denominación y valor real, también se clasifican por sus características. A pesar de ser importante conocer a qué emisión corresponden las piezas destruidas, la clasificación a este nivel es un proceso agobiante, así que el control se realiza considerando quién las fabricó, de esta forma se puede saber a qué grupo de emisiones pertenecen las piezas, además de la fecha de su destrucción y el método utilizado.

El ideal sería que todas las piezas autorizadas que fueron fabricadas sean destruidas con lo que concluiría el ciclo de vida de esa emisión; en la práctica es difícil conseguirlo ya que el público conserva o en algunos casos destruye las piezas. Una forma de reducir los controles sobre emisiones muy antiguas cuyas características han cambiado sustancialmente y el valor real ya no corresponde

con el valor facial, es la desmonetización. La desmonetización es dejar sin valor un billete o moneda, de tal forma que deja de ser moneda de curso legal, por lo que ya no es un medio válido de pago.

Mientras las piezas están en circulación, no pertenecen a las existencias del Banco Central por lo que es imposible conocer la ubicación exacta de las piezas, con la información de las emisiones, las existencias, las piezas en tránsito y las piezas destruidas se puede determinar por inferencia cuántas piezas y de qué denominación están circulando.

Como podemos observar, las piezas que son propiedad de los Bancos Centrales pueden estar agrupadas en cuatro “familias” de existencias. Las existencias no materializadas, las existencias físicas, las existencias en tránsito y las piezas destruidas. Para llevar un correcto control de las existencias, se debe reflejar cualquier movimiento que tengan en el mundo real en estos controles, en otras palabras, si cada movimiento en el mundo real lo asociamos con una operación, cada operación debe afectar el control de las existencias que le corresponda. Entre mejor control sobre las distintas familias de existencias se tengan, mayor será la certidumbre de las piezas en circulación.

2.4 Control de operaciones

Cualquier movimiento de efectivo que sucede en el mundo real, es susceptible a ser conceptualizado como una operación. En los flujos de efectivo encontramos un grupo de operaciones que son comunes a la operativa de los distintos Bancos Centrales. Comenzamos con la autorización de emisión de billete o moneda. En estas operaciones es importante controlar la fecha de la autorización, el número de piezas autorizadas, la denominación, valor real, así como las características de las piezas considerando, en los casos pertinentes, el o los materiales que las componen. Normalmente los materiales están asociados a las monedas únicamente, ya que en este tipo de efectivo es común encontrar piezas con distintas aleaciones.

Como etapa siguiente a la autorización tenemos la orden de impresión de billetes o de acuñación de monedas, en estas operaciones es necesario conocer a que autorización corresponde la orden, la denominación y valor real, el número de piezas que se ordenaron fabricar y el fabricante.

Cuando el proceso de fabricación ha concluido, los fabricantes entregan las piezas a los Bancos Centrales, cuando se trata de billete se le conoce a la operación como entrega y cuando se trata de moneda se conoce como libranza. En las entregas y libranzas es necesario conocer a qué autorización y a qué orden de fabricación corresponde la entrega, cuál es la cantidad de piezas que

se entregan y de que denominación, además se necesita saber en qué sitio y en qué bóveda son entregadas. En el caso de los billetes se incluye también la información de los folios utilizados en las piezas fabricadas.

Un proceso común dentro de los Bancos Centrales son los recuentos y clasificaciones, así que a lo largo del día operativo se realizan traspasos de una bóveda a otra, de una bóveda a una ventanilla, de una ventanilla a otra y de una ventanilla a una bóveda. Estas operaciones suceden siempre dentro de un mismo sitio y pueden incluir más de una denominación. En estas operaciones interesa conocer el sitio donde se está efectuando el movimiento, el número de piezas y las denominaciones involucradas, la bóveda origen y la bóveda destino. En los procesos que se realizan en las ventanillas pueden detectarse piezas que pertenecen a otro estado físico, por ejemplo al momento de clasificar pueden detectar piezas deterioradas en lo que se suponía eran piezas aptas. Así que el estado físico origen y el estado físico destino también son información necesaria en estas operaciones. Como las ventanillas al cierre del día operativo no deben tener piezas en su poder, todas las operaciones se procesan el mismo día.

Como se ha comentado a lo largo de este documento, las instituciones bancarias acuden al Banco Central, ya sea a su oficina central, alguna de sus sucursales o a algún corresponsal a depositar y/o retirar efectivo.

Los retiros son operaciones mediante los cuales, los bancos con cargo a su cuenta, retiran una determinada cantidad de piezas de cada denominación, estos retiros pueden ser de dos formas básicamente, preavisados o "mismo día". Los retiros preavisados son aquellos que son programados al menos un día hábil antes de la fecha del retiro, esto permite a los bancos contemplarlos dentro de sus operaciones para contar con los fondos necesarios antes de efectuar el retiro, podemos decir que son operaciones con fecha valor² futura. Los retiros mismo día, son operaciones con fecha valor mismo día, es decir son concertadas y efectuadas el mismo día operativo, estas operaciones involucran mayor tiempo operativo porque el mismo día tienen que prepararse, es decir, realizar los traspasos necesarios para contar en la ventanilla con las piezas requeridas de la denominación solicitada, además, involucran un riesgo de sobregiro por parte de las instituciones que retiran por ser un cargo que no estaba contemplado en la tesorería del banco. En un retiro jamás se entrega billetes o monedas deterioradas, sin embargo, pueden entregarse piezas aptas o nuevas.

Los depósitos son operaciones mediante los cuales, los bancos con abono a su cuenta, depositan una determinada cantidad de piezas por cada denominación. Los depósitos siempre son con fecha valor mismo día y pueden ser de piezas con estado físico apto³ o con estado físico deteriorado. Es importante esta

² La fecha valor es la fecha en la cuál se realiza el movimiento, es independiente a la fecha de la concertación.

³ Nunca con estado físico nuevo porque se considera que ya estuvieron en circulación.

información ya que las piezas se trasladan a bóvedas distintas según el estado físico.

En los depósitos como en los retiros, la información que se necesita conocer está formada por el sitio donde se efectúa el movimiento, la ventanilla de atención, el estado físico con el cuál entra o sale, la institución que efectúa el movimiento y el número de piezas por cada denominación.

El tiempo requerido para recontar todas las piezas depositadas en un día puede ser considerable y si el sitio donde se efectúa el movimiento no cuenta con los recursos necesarios para realizarlo de una forma rápida puede ocasionar un retraso en la operación diaria. Por esta razón, en los sitios donde no se cuenta con los recursos suficientes para un recuento al momento del depósito, en primera instancia, se considera como real la cantidad que reporta el banco. Existen algunas convenciones que permiten hacer un conteo rápido de las piezas, por ejemplo una "fajilla" es un grupo de cien billetes de la misma denominación sujetos en la parte central por una cinta de papel que se le conoce como "cintilla".

Dentro de las bóvedas de recuento, verifican que la cantidad de piezas por denominación sea igual a la reportada. Si se detecta una diferencia, ya sea un sobrante o un faltante, se le notifica al banco esta situación y se procede a realizar el ajuste correspondiente en la cuenta del banco, ya sea al cargo o al abono. Pueden aplicar sanciones pero esto es decisión de cada Banco Central en sus políticas de operación y no influye en el control de flujos de efectivo. En estos casos se requiere conocer el sitio y la bóveda donde se detectó la diferencia, el estado físico de las piezas, la denominación y el número de piezas que faltan o sobran, además del banco al cuál se le detectó la diferencia.

Una operación un tanto compleja, es la operación de envío de billetes o monedas de un sitio a otro, esta operación es conocida como remesa. Las remesas pueden ser vistas como una operación de dos etapas, la primera etapa consiste en retirar de las existencias del sitio origen una determinada cantidad de piezas de cada una de las denominaciones que van a ser trasladadas al sitio destino y colocarlas en existencias en tránsito. En tránsito permanecerán hasta que lleguen a su destino, al momento de recibir estas piezas en el sitio destino, se retiran de las existencias en tránsito y forman parte de las existencias del destino. La complejidad de esta operación consiste en que el envío es una operación que puede tener fecha valor mismo día o valor futuro y la recepción es siempre valor mismo día, sin embargo, el día de envío puede ser distinto al día de recepción, además son operaciones íntimamente ligadas, por ejemplo es imposible cancelar o, lo que es lo mismo, decir que una remesa no fue enviada cuando el sitio destino ya la recibió, de igual forma, no es posible recibir una remesa que no fue enviada. Por otro lado, como la remesa es un movimiento físico de piezas de un lugar a otro fuera del sitio, involucra gastos inherentes, por lo que es posible que en una misma remesa estén involucradas piezas con distintos estados físicos. Además, según la infraestructura de los sitios puede

sucedir que mientras en el origen se contemplen varios estados físicos en el destino se reciban con un número distinto de estados físicos, un ejemplo puede ser que un corresponsal que envía una remesa con billetes cuyo estado físico sea deteriorado, en el destino los billetes deteriorados puedan indicar algún tipo de información que permita conocer más del tipo de deterioro, como billete parchado, roto, desgastado, etc.

La remesa es la que mayor información requiere, se necesita conocer el sitio y bóveda origen, el sitio y bóveda destino, además del número de piezas por denominación y estado físico tanto origen como destino, la fecha de envío y la fecha de recepción, la cuál solo se sabrá hasta que el sitio destino confirme el ingreso de las piezas en sus existencias.

La última operación esencial para considerar el ciclo completo es la destrucción. En esta operación está involucrado el último sitio y bóveda donde estaban las piezas antes de ser destruidas, el estado físico que guardaban, el fabricante y el número de piezas de cada denominación involucrada.

Con estas operaciones podemos representar el flujo de efectivo de cualquier Banco Central, sin embargo, cada Banco Central puede contemplar algunas operaciones adicionales originadas de las políticas que manejen.

2.5 Control de seguridad

La seguridad de la información es un aspecto a cuidar en cualquier organización, un manejo inadecuado puede provocar situaciones comprometedoras, o incluso de riesgo, ya sea para la integridad de la organización o de los participantes. Algunos autores han definido a las políticas de seguridad como la solución técnica a un problema no técnico.

La información necesita estar protegida a varios niveles. Al tratar el tema de seguridad se deben considerar múltiples aspectos, que van desde el acceso a los servidores, a las redes, a la información, hasta la autenticación⁴ de los mensajes. Sin embargo, también podemos ver la seguridad como un problema institucional por lo cual, la misma organización en su conjunto debe dictar las políticas en cuanto seguridad concierne. En este punto vamos a describir, únicamente, las necesidades de seguridad propias del problema de los flujos de efectivo, asumiendo que las políticas generales de acceso a los servidores y protección de las redes ya están solucionadas.

⁴ La autenticación es el procedimiento mediante el cuál se confirma tanto que el usuario es quién dice ser como la información proviene de donde dice provenir

Existen varios niveles de seguridad que es importante contemplar, sin embargo, los puntos que deben considerarse en todo momento son.

- **Confidencialidad:** Proteger el acceso a la información, para que solamente las personas autorizadas puedan tener acceso a ella.
- **Integridad de los Datos:** Proteger la información para que solamente las personas autorizadas puedan agregar, modificar o eliminar.
- **Control:** Regular el acceso al sistema.
- **Auditar:** Determinar quién y qué han hecho los usuarios en el sistema.

En los sistemas de flujos de efectivo estos puntos son perfectamente válidos, determinar quién puede ver la información y a qué información tiene acceso es una necesidad. Hay personal que tiene acceso a la información en forma parcial, los directivos tienen acceso a otro tipo de información.

Como se explicó en los requerimientos para las operaciones, la integridad de los datos es muy importante para proporcionar información veraz. En los Bancos Centrales, de hecho en cualquier organización, un objetivo institucional lo divide en tareas más simples, con la finalidad de prestar más atención a los detalles inherentes a cada tarea, de esta forma, el objetivo primario puede ser cumplido de una forma eficiente y satisfactoria. Dentro del Banco Central, tratando los flujos de efectivo, cada departamento se ocupa de alguna o algunas partes del ciclo de vida de las piezas, por ejemplo, hay departamentos que controlan los depósitos y retiros, otros de las remesas, etc., la división obedece directamente a las políticas internas de cada Banco Central, sin embargo el fin es el mismo, identificar que persona o personas pueden ingresar, modificar o eliminar información perteneciente a las funciones de su departamento. Imaginemos el efecto que tendría, por ejemplo, que un corresponsal pudiera manejar la información relativa a las emisiones; como cualquier organización, los Bancos Centrales se preocupan de la integridad de la información.

El acceso a la información debe estar controlado, se debe saber quiénes pueden tener acceso, y qué información pueden ver o en su caso, modificar. Una práctica normal sobre todo cuando hay interacción de información es el establecimiento de horarios, los sistemas contables de los Bancos Centrales están sujetos a horarios, para que los usuarios puedan conocer su situación al cierre de operaciones; como un gran número de Bancos Centrales no permiten cerrar el día operativo con sobregiros o les cobran algún tipo de interés, los bancos al cierre de las operaciones diurnas consultan su saldo y, si es necesario, solicitan algunos préstamos a otros bancos que les permita cubrir el sobregiro a un costo menor. Los flujos de efectivo tienen un reflejo contable, por

tanto sus horarios de operación deben acoplarse a los horarios de los sistemas contables.

Por otra parte, los responsables de la operación diaria, necesitan saber quién genera la información, y en el caso de un sistema, quién la ingresa, desde dónde la ingresa y a qué hora la ingresa.

Cuando existen varios departamentos integrando información para un fin común, se requiere de un coordinador que regule la forma de integración, si consideramos como el objetivo, controlar los flujos de efectivo, y varios departamentos están encargados de una parte del todo, este coordinador debe vigilar que toda la información generada en el día sea integrada dentro de los horarios establecidos. Dada la magnitud de esta tarea, es común encontrar más que un coordinador, a una área coordinadora que es la responsable de la operación. Esta área coordinadora regula, basándose en las políticas marcadas por el Banco Central, el tipo de operaciones que puede integrar cada departamento. En caso de contingencia, esta área puede realizar las funciones de integración de la información correspondiente a un departamento que tenga dificultades para hacerlo, por ejemplo, que no tuviera comunicación con un servidor.

2.6 Validaciones generales

Para controlar los flujos de efectivo, cada movimiento es representado como una operación, hay operaciones para cada una de las fases del ciclo de vida de las piezas, sin embargo las combinaciones posibles que puede tener un mismo tipo de operación pueden ser considerables, sin embargo, es necesario determinar cuáles son los movimientos válidos, un movimiento inválido sería, por ejemplo, que billetes deteriorados se almacenen en bóvedas de billete nuevo. Los movimientos válidos están definidos por los Bancos Centrales en sus políticas de operación, y cada tipo de operación debe contemplar el conjunto de posibilidades que le fueron otorgadas.

Las combinaciones para cada tipo de movimiento dependen, por una parte, de la infraestructura de cada Banco Central, entre más sitios y bóvedas contemple mayor será el número de posibilidades de movimientos. Por otra parte, de la diversidad de operaciones que maneje. En este documento hemos mencionado las operaciones básicas, más esto no significa que sean las únicas, cada Banco Central, según sus necesidades y políticas, podrá manejar un conjunto mayor de tipos de operación, lo que ocasiona un incremento en el número de posibles movimientos.

Retomando la parte de la infraestructura de los Bancos Centrales, no podemos considerar que todos los sitios tienen las mismas bóvedas, cada sitio o, en el mejor de los casos, cada grupo de sitios tiene un número de bóvedas distinto a otro grupo. Por tanto una validación importante en cada operación es la existencia de los sitios involucrados.

A pesar de que las piezas, ya sean billetes o monedas, son dinero en efectivo, para los flujos de efectivo es importante distinguir las operaciones con billetes de las operaciones con monedas, dado que el manejo físico y contable es distinto. Esta distinción se ve reflejada en las operaciones, por tanto no se mezclan billetes y monedas en una misma operación aun cuando sea el tipo de operación sea el mismo.

Las operaciones tienen como objetivo representar un movimiento en el mundo real, y en el mundo real pueden ocurrir situaciones no previstas o eventos fortuitos, en estos casos, se pueden realizar movimientos que normalmente no son válidos, pero que atendiendo una situación especial es necesario hacer. Si la finalidad de integrar la información es controlar los flujos de efectivo y conocer la ubicación de las piezas en todo momento, sería incongruente que por validaciones no se pudiera representar este movimiento. El área coordinadora con sus atribuciones de regular la integración de la información debe poder autorizar que estos movimientos puedan ser representados como una operación y hacer la excepción correspondiente, suponiendo que se trata de un imprevisto y no es una práctica común, ya que si fuera este el caso debería formar parte de las políticas de movimientos.

Validaciones comunes, como la existencia de una institución, de una denominación, de una divisa, etc., son necesarias para mantener consistente la información, no son menos importantes que las otras validaciones. Aceptar información que incluya sitios, bóvedas, estados físicos, denominaciones, etc., inexistentes crearía incertidumbre y restaría confiabilidad a la información.

La contabilidad nos dice que una póliza debe estar "cuadrada", es decir la suma de los cargos debe ser igual a la suma de los abonos; en las operaciones de los flujos de efectivo, la suma de los importes resultantes de multiplicar el número de piezas por su valor real debe ser igual al importe de la operación.

Otro punto a considerar en las validaciones es la fecha valor. La fecha valor de una operación debe ser igual a la fecha en la que el movimiento sucede, tenemos operaciones con fecha valor mismo día y operaciones con fecha valor futuro, bajo esta afirmación no sería aceptable tener operaciones con fecha valor anterior a hoy, porque esto representaría que un movimiento no fue incluido y por tanto la información no es confiable. Los encargados de las bóvedas antes del cierre verifican que las existencias en su bóveda sean las reportadas en los informes, si detectan alguna diferencia revisan los movimientos que realizaron contra las operaciones integradas hasta encontrar las diferencias. En las

operaciones con fecha valor futuro es necesario validar que dicha fecha sea un día hábil.

2.7 Disponibilidad

La información constituye uno de los elementos más importantes para la toma de decisiones, tanto en lo que concierne a los esfuerzos de desarrollo y de administración pública como a los negocios privados. La efectividad de las decisiones implementadas depende directamente de la calidad, pertinencia y oportunidad de la información que les dio fundamento. En ese sentido, podríamos decir que la información es un recurso estratégico para alcanzar nuestras metas de desarrollo.

La oportunidad de la información de los flujos de efectivo permiten conocer a los Bancos Centrales los recursos con que cuentan en todo momento. Dado que los billetes y monedas forman parte del balance, es necesario que la información esté disponible en todo momento.

Para que los Bancos Centrales puedan ofrecer un servicio eficiente requieren, como toda organización, una alta disponibilidad de la información, por consecuente, las plataformas donde se concentra deben estar disponibles en todo momento.

No es suficiente que la información de los movimientos (operaciones) se tenga con oportunidad, el concentrador de estas operaciones debe estar preparado para ingresarlas y poder mostrar la información a la persona autorizada que la solicite. Cuando el concentrador es un equipo de cómputo, un servidor, hay que considerar que los equipos pueden presentar fallas, ya sea del *hardware*, del *software* o bien en las comunicaciones.

Si un sitio no puede transmitir sus operaciones, hemos visto que el coordinador debe ser facultado para simular un sitio distinto al suyo e integrar la información del sitio "aislado", sin embargo, si el repositorio de datos no está disponible, será necesario migrarlo a otro lugar donde se pueda continuar la operación, evitando al máximo el reinicio del día operativo, ya que esto implicaría que todo el trabajo desde el inicio del día hasta el momento de la falla tendría que repetirse aumentando el riesgo de un error humano.

2.8 Información diaria e histórica

En el punto anterior veíamos la importancia de la información y de su disponibilidad. En los flujos de efectivo la información puede ser analizada de dos puntos de vista, la parte operativa y la de análisis de comportamientos.

En la parte operativa la información diaria es vital para sus funciones, la afectación contable, los movimientos realizados en el día, las existencias al cierre del día, son elementos sujetos a diversos controles. Todas las operaciones que tienen una afectación contable deben estar reflejadas en el sistema contable, todas los movimientos deben corresponder a una operación y las existencias de cierre deben concordar con las existencias reales de cada bóveda de cada sitio.

Sin embargo, en caso de controversia o duda en alguna operación ocurrida en el pasado, se necesita obtener la información relativa a esta operación. Dado que cada operación representa los movimientos efectuados en un día, se debe almacenar toda la información del día en un histórico, para así poder reproducir o consultar un día determinado.

Para el análisis de comportamientos, en varios casos, no se requiere la información al nivel de detalle de una operación, pero como éstas son el generador de los cambios en las existencias, esta información será la fuente de alimentación para el personal encargado del análisis.

La información histórica de las operaciones no necesariamente tiene que estar "en línea" en todo momento, basta con tener un determinado número de días para consultas inmediatas, la información anterior a estos días podrá ser almacenada en otro repositorio, en el cual, en caso de ser necesario se consultaría.

Una vez especificadas las necesidades de los Bancos Centrales para conseguir un sistema de flujos de efectivo adecuado, comenzaremos con la siguiente fase del desarrollo.

3 Análisis

En este capítulo analizaremos la problemática ya planteada en el capítulo anterior con el fin de obtener un modelo que satisfaga los requerimientos que se mencionaron con anterioridad.

3.1 Análisis del flujo de efectivo

Antes de comenzar el análisis de los flujos de efectivo, es necesario recordar que una de las funciones de los bancos centrales es la de abastecer de efectivo a la economía del país. La sociedad, para poder realizar sus transacciones, particularmente las más sencillas, requiere de dinero en efectivo y lo solicitan a los Bancos Centrales (ver Figura 3-1).

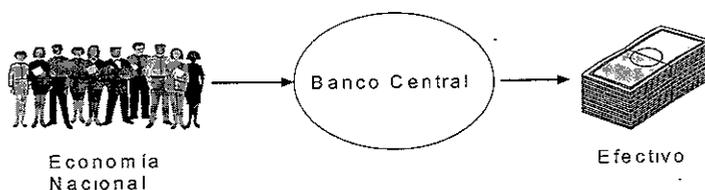


Figura 3-1. Función del Banco Central en materia de abastecimiento de efectivo

La economía del país realmente está convirtiendo a dinero físico el dinero en libros o electrónico. En este análisis no trataremos el origen del dinero y asumiremos que existe en cualquiera de sus formas. Así, los elementos que analizaremos en este documento serán todos aquellos movimientos que se realizan con dinero propiedad del Banco Central. En primera instancia definiremos el significado del dinero en libros y del dinero físico.

Entenderemos como dinero físico a los billetes y monedas de curso legal emitidas por un Banco Central. Estas piezas existen físicamente y son ampliamente utilizadas en la vida diaria de la sociedad. Normalmente son un medio de pago muy recurrido para transacciones de bajo valor.

El dinero en libros o electrónico¹ es aquel que tiene valor pero carece de una forma tangible, por ejemplo el saldo de una cuenta de cheques o el saldo de una cuenta de nómina. Existen medios físicos que afectan el saldo de estas cuentas, como el cheque o la tarjeta de débito, sin embargo, el dinero no está ni en el cheque ni en la tarjeta sino en un registro.

Así como se realizan transacciones con dinero físico, también se realizan con dinero electrónico y normalmente son utilizadas para pagos de alto valor.

Un proceso común es convertir dinero electrónico en dinero físico y viceversa, para esto existen algunos instrumentos y procedimientos establecidos. Uno de los más comunes es acudir a una ventanilla de un banco y “cobrar” un cheque. El cobrar un cheque implica una disminución de la cantidad de dinero electrónico que tiene asociado el número de cuenta de la persona que expidió el cheque y un aumento en la cantidad de dinero físico que tiene el beneficiario en su poder, ya que el cajero le proporciona el equivalente del monto del cheque en billetes y/o monedas. A través del número de cuenta se identifica en forma única un registro y por tanto se conoce su saldo.

De igual forma convertir dinero físico en dinero electrónico es una necesidad de la sociedad.

En la operación con los bancos centrales este fenómeno es idéntico, en la Figura 3-2 ejemplificamos la situación descrita anteriormente.

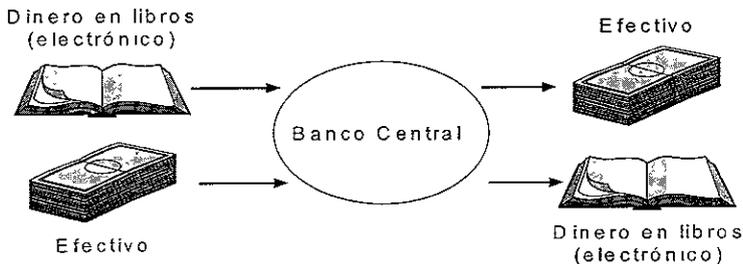


Figura 3-2. Ejemplificación de la conversión de dinero físico en electrónico y viceversa.

¹ En lo sucesivo haremos referencia al dinero en libros como dinero electrónico

Para abastecer de suficiente efectivo a la sociedad, es común encontrar que los Bancos Centrales se auxilian de la banca comercial. La banca comercial ofrece diversos servicios a sus usuarios con el fin de facilitar esta conversión de dinero físico a dinero electrónico y viceversa, por ejemplo ventanillas en sucursales y cajeros automáticos.

Los bancos comerciales reciben el dinero de la sociedad convirtiendo el dinero físico en electrónico. Los bancos comerciales se comprometen con sus usuarios a permitirles la conversión de dinero electrónico en físico cuando ellos lo requieran.

De igual forma, los bancos acuden al Banco Central a realizar estos movimientos utilizando la cuenta que tienen en el Banco Central.

En la Figura 3-3 podemos observar como la sociedad acude a los bancos para realizar las conversiones de dinero. Una vez que el dinero físico abandona las instalaciones del Banco Central consideramos que se encuentra en circulación, dado que la banca forma parte de las empresas del país.

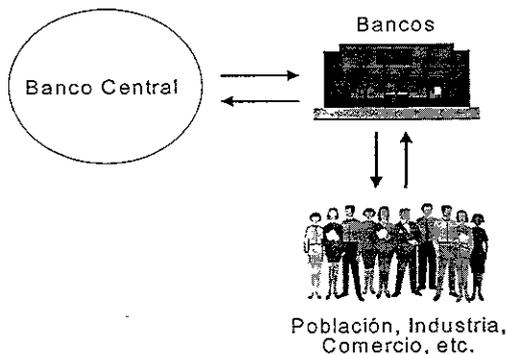


Figura 3-3. Esquema de abastecimiento de efectivo a la sociedad.

En el control de flujos de efectivo únicamente interesan los movimientos en los que interviene dinero propiedad del Banco Central, incluso los movimientos en los cuáles la banca comercial acude a depositar o retirar efectivo, ya que son piezas que pasan o dejan de formar parte de las existencias del Banco Central.

Para poder abastecer de efectivo a todas las instituciones bancarias, anteriormente habíamos comentado que algunos Bancos Centrales se auxilian de sucursales en lugares estratégicos e incluso de corresponsales, como lo podemos observar en la Figura 3-4.

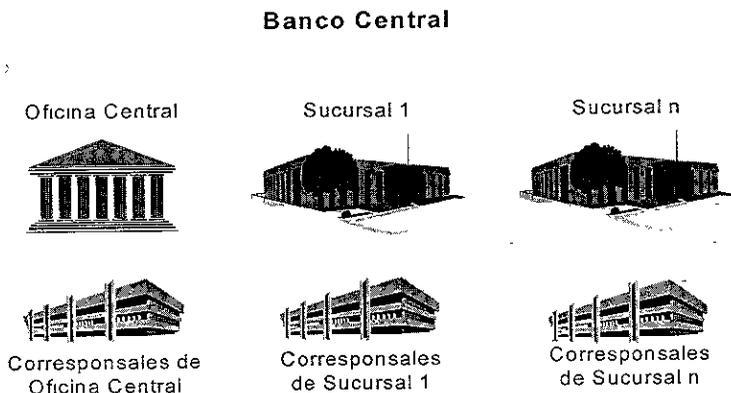


Figura 3-4. Estructura de un Banco Central para funciones de abastecimiento de efectivo

Una práctica común es que la Oficina Central de los Bancos Centrales sea quién distribuya en primera instancia el efectivo a sus Sucursales y a sus Corresponsales, a su vez, cada Sucursal se encarga de abastecer de efectivo a sus Corresponsales. Es factible que por instrucción de la Oficina Central o de alguna Sucursal, un Corresponsal le envíe efectivo a otro Corresponsal de la misma jurisdicción, y esporádicamente, puede un Corresponsal de una Sucursal enviarle efectivo a un Corresponsal atendido por otra Sucursal. Ver Figura 3-5.

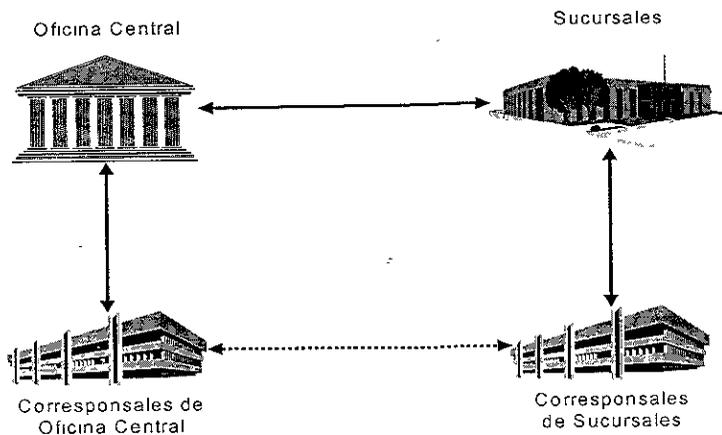


Figura 3-5. Flujos de efectivos entre distintos sitios.

Tanto la Oficina Central como cada una de las Sucursales y cada uno de los Corresponsales son identificados como un Sitio. Los sitios se conforman con la identificación de la institución y la identificación de su ubicación, por ejemplo un sitio sería Banco Central en la localidad "X", o Banco "Y" en la localidad "Z", donde la localidad "X" y "Z" son alguna población dentro del territorio nacional, y el Banco "Y" es el nombre de algún banco comercial.

Con esta estructura se facilita a la banca comercial la conversión de dinero electrónico en físico y viceversa, ya que los Corresponsales también se encuentran en lugares estratégicos que acortan las distancias que tienen que recorrer los bancos para abastecerse de efectivo. En la Figura 3-6 está ilustrado la interacción de los bancos con los distintos sitios del Banco Central.

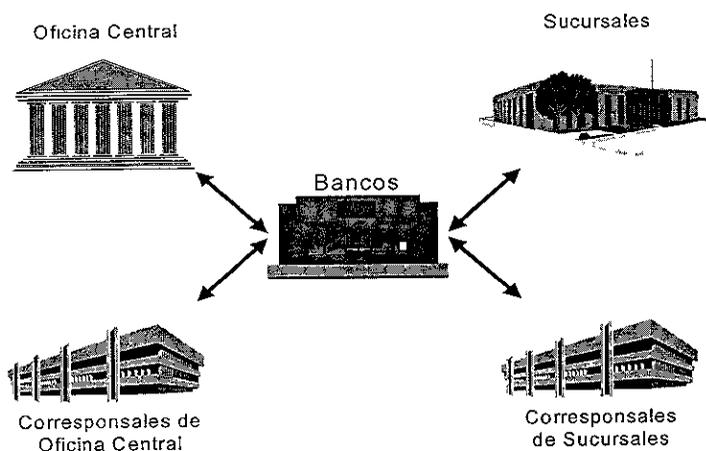


Figura 3-6. Aplicación de la estructura de los bancos centrales para abastecer efectivo.

Además de los movimientos de efectivo que se dan entre sitios y aquellos en los que interviene la banca, internamente es factible encontrar movimientos dentro de cada sitio. Los sitios, al menos, tienen una bóveda y/o una ventanilla, hay que recordar que en las bóvedas se encuentra el dinero almacenado y en las ventanillas se da atención a usuarios o se realizan procesos. Un sitio típico está ejemplificado en la Figura 3-7. Dada la variedad de las bóvedas y ventanillas existe la necesidad de realizar los mismos procesos en varios sitios. Por lo tanto podemos clasificar las bóvedas en tipos de bóvedas, tomemos como ejemplo la ventanilla de atención a usuarios, está ventanilla prácticamente existe en cada uno de los sitios, entonces podemos decir que tales sitios cuentan con un tipo de ventanilla de atención a usuarios en lugar de considerarla como un ente único como sería la ventanilla de atención a usuarios del sitio "X", la ventanilla de atención a usuarios del sitio "Y", etc. Bajo este supuesto para conocer las

existencias de una ventanilla o bóveda² necesitamos adicionalmente el sitio donde se tiene este tipo de bóveda. A pesar de requerir información adicional para identificar una bóveda, este esquema permite mayor flexibilidad al fijar las reglas del negocio.

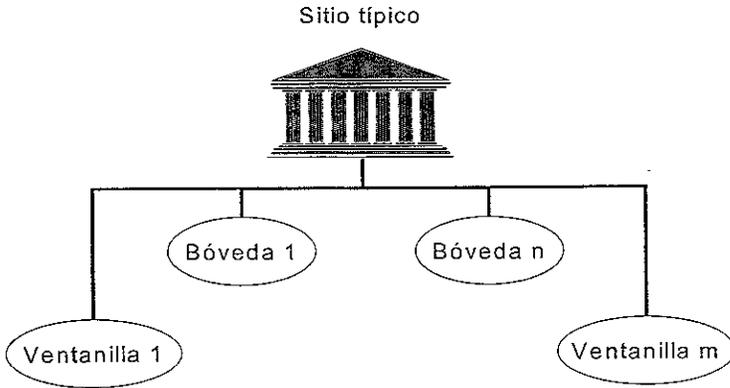


Figura 3-7. Estructura de un sitio típico.

En el caso de los corresponsales, es probable que utilicen una parte de alguna de sus bóvedas donde almacenan su efectivo, para almacenar las piezas del Banco Central, una alternativa que puede formar parte de las reglas del negocio, sería crear un tipo de bóveda, "Bóveda en Corresponsal", y con esto evitar conflictos en la forma en la cual se nombrarán estas bóvedas, aunque, como hemos mencionado, esto dependerá exclusivamente de las reglas que fije cada Banco Central.

Retomando los movimientos dentro de un sitio, es claro que se requiere mover efectivo de una bóveda a otra, ya sea para su recuento, clasificación, atención a usuarios o simplemente para almacenar el efectivo. El objetivo de los sistemas de control de flujos de caja es conocer la ubicación de las existencias del Banco Central en todo momento. Por esta razón es importante considerar todos los movimientos que se susciten dentro de cada sitio.

En la Figura 3-8 podemos observar que también es posible efectuar movimientos dentro de una misma bóveda, esto obedece a la naturaleza del proceso, si una bóveda, al verificar sus existencias, detecta que algunas piezas están mal clasificadas pueden realizar un movimiento que corrija el error y esto tiene que reflejarse en el sistema de control para que las existencias reportadas concuerden con la realidad.

² En lo sucesivo el término bóveda se utilizará indistintamente para ventanillas y bóvedas

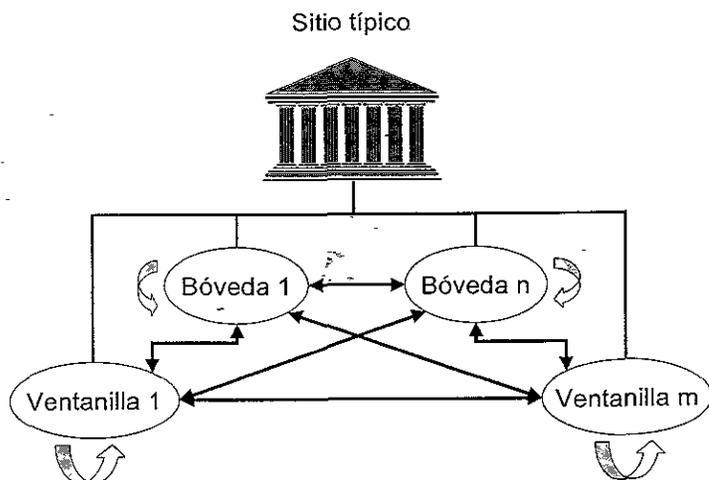


Figura 3-8. Movimientos de efectivo realizados en el interior de un sitio.

Hasta este momento hemos contemplado los movimientos que se realizan con piezas físicas, sin embargo, es necesario analizar el proceso anterior a la existencia de las piezas, que está ilustrado en la Figura 3-9. Los Bancos Centrales son los encargados de la emisión del dinero, no obstante requieren de una autorización que puede ser una atribución del propio Banco Central, o bien, provenir de alguna entidad ajena a él, por ejemplo el Congreso. En la autorización se detalla la cantidad de piezas que se autoriza emitir, describiendo las características de éstas. Por tanto es importante conocer cuántas piezas son autorizadas y cuántas se han emitido para evitar una producción que sobrepase el límite establecido.

Una vez autorizada la emisión, el Banco Central, a través del departamento que le compete, genera las órdenes de fabricación a la entidad correspondiente; como se mencionó con anterioridad, el propio Banco Central puede encargarse de la fabricación o bien recurrir a un tercero que se encargue de esta fase.

El Banco Central controla cuántas piezas han mandado fabricar y a quién se le ha confiado esta tarea.

Cuando concluye el proceso de fabricación, la entidad responsable realiza una entrega de piezas³, con lo cual las piezas dejan de pertenecer a las piezas no materializadas para convertirse en existencias en un sitio. En la Figura 3-9 podemos observar las fases por las cuáles tienen que pasar las piezas antes de ser materializadas y constituir las existencias del Banco Central.

³ Cuando se trata de billete se le llama "entrega", cuando es moneda se le llama "libranza"

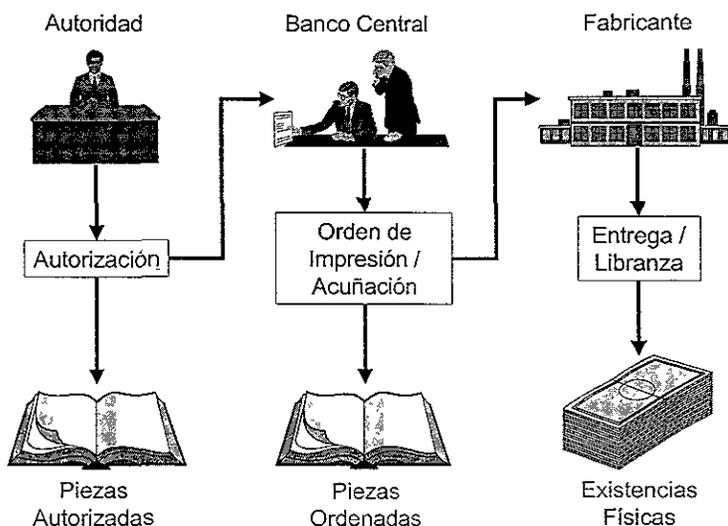


Figura 3-9. Fases de las piezas no materializadas.

La última fase en la cual pueden estar las piezas es la destrucción. Las piezas, por su uso, van deteriorándose hasta llegar al punto en el que no es conveniente que sigan en circulación. El ciclo vida de cada pieza termina cuando se destruyen; los analistas de los Bancos Centrales estiman la vida útil de las piezas cuando elaboran las propuestas de autorización, ya que las piezas destruidas no son sustituidas por nuevas de forma sistemática. Para evitar el desabasto, solicitan a la autoridad competente considere la cantidad de piezas que será necesario fabricar para sustituir a las destruidas. Como se mencionó en la autorización, no es factible fabricar más piezas de las autorizadas, el fabricar nuevas piezas para sustituir las destruidas implica un sobregiro en la cantidad autorizada. Por ejemplo, supongamos que se da una autorización a un Banco Central para emitir 1,000,000 de piezas de la denominación X y todas las piezas autorizadas son fabricadas, si al destruir 1,000 piezas por deterioro, sin otra autorización se fabrican las 1,000 piezas para sustituir las destruidas, la fabricación habría sido de 1,001,000 cuando solamente se había autorizado 1,000,000, esto representa un sobregiro en la fabricación de 1,000 piezas.

Por esta razón es importante controlar el número de piezas de cada denominación, distinguiéndolas por sus características de fabricación. Así, el Banco Central genera instrucciones a la ventanilla de destrucción para que proceda a destruir una determinada cantidad de piezas, como se ilustra en la Figura 3-10.

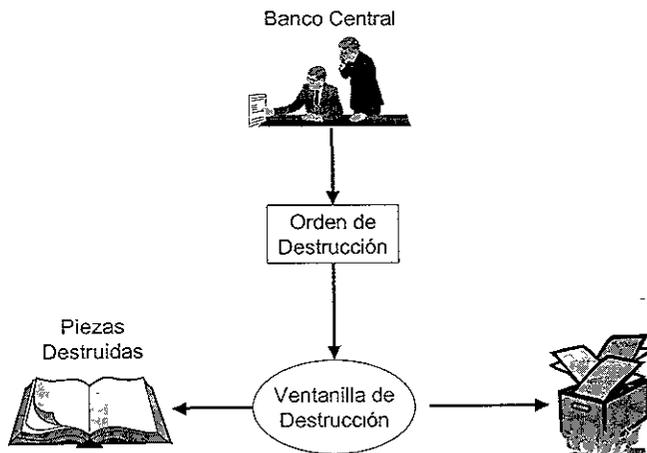


Figura 3-10. Proceso de destrucción de efectivo.

En resumen, los flujos de efectivo deben considerar todas y cada una de las fases por las cuáles atraviesan las piezas. El ciclo comienza con las piezas no materializadas, esta parte la componen la autorización y la orden de fabricación, el proceso de fabricación compete a un proceso industrial por lo que no se considera en los flujos de efectivo. Esta parte del ciclo se caracteriza por la inexistencia física de las piezas.

La segunda parte del ciclo la componen todos los movimientos que se realizan con piezas físicas, aquellos que se efectúan dentro de una bóveda o de un sitio, los movimientos entre sitios y los movimientos con la banca comercial. En esta parte del ciclo podemos observar que siempre están involucradas piezas físicas.

La tercera parte del ciclo la componen las piezas que han sido destruidas; a pesar de no existir las piezas físicamente, podemos diferenciarla de las otras partes por ser la fase terminal, aquí no podemos decir que son piezas no materializadas, porque implicaría que nunca han existido y por el contrario, existieron y fueron destruidas.

Existen tres elementos que permiten la unión de las partes antes mencionadas: la entrega de piezas, las piezas en tránsito y el proceso de destrucción. Cada uno de estos elementos tiene características propias. La entrega de billetes o libranza de monedas, sucede cuando el fabricante ha concluido su proceso, él es el encargado de materializar las piezas, y al realizar una entrega de piezas, implícitamente está disminuyendo la cantidad de piezas no materializadas y está incrementando las existencias del sitio donde se realice la entrega. La

destrucción implica disminuir existencias del Banco Central y aumentar la cantidad de piezas destruidas. Estos dos elementos afectan “mundos” distintos, sirven de enlace entre un estado y otro, a diferencia de las piezas en tránsito que no afectan varios “mundos”, ya que las piezas siguen existiendo en forma física.

Como recordaremos, las piezas son almacenadas en bóvedas dentro de sitios, los sistemas de control de flujos de efectivo, proporcionan información de la ubicación de las piezas en todo momento, sin embargo, cuando un sitio envía efectivo a otro sitio, es imposible determinar dónde están localizadas en un momento del tiempo. En este tipo de movimiento se están trasladando piezas físicas, lo que consume tiempo antes que las piezas lleguen a su destino, mientras las piezas estén viajando se considera que las piezas están en tránsito, es decir, no están en un lugar físico estático.

En la Figura 3-11 se ilustran cada uno de los movimientos que conforman el ciclo de vida del efectivo, desde su autorización hasta su destrucción.

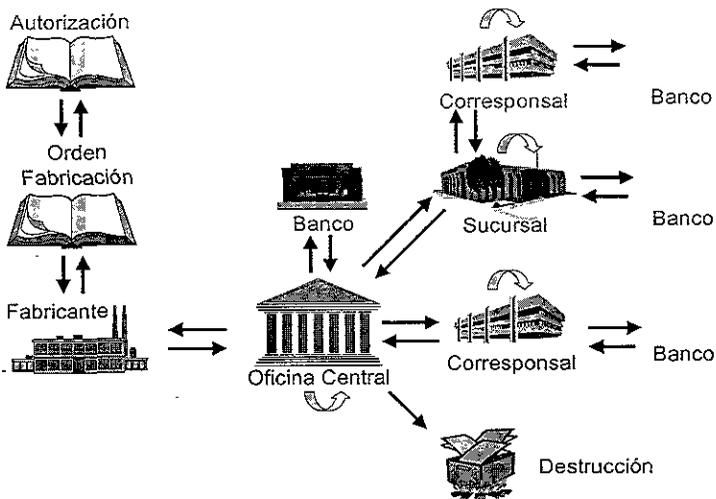


Figura 3-11. Ciclo de vida del efectivo.

Un aspecto importante de analizar es el estado que conservan las piezas, las piezas pueden tener estados lógicos de proceso o no físicos, como son: “Autorizado”, “Ordenado”, “Destruído” o cualquier otro que contribuya a mejorar el control que se tiene sobre piezas no físicas, por ejemplo, pudiera existir un estado que indicará cuántas piezas no han sido fabricadas, que, aunque pudiera determinarse considerando los demás estados, las políticas de cada Banco Central las fijan ellos y se debe dar la flexibilidad necesaria para que ellos operen de acuerdo a sus necesidades.

Por otro lado, existen los estados físicos, que en una primera instancia pueden ser: "Nuevo", "Apto" y "Deteriorado". Al igual que en el caso anterior debe permitirse cualquier otro estado físico que permita a los Bancos Centrales tener mayor control, por ejemplo, algún Banco Central podría estar interesado en catalogar el deterioro en la causa de éste, como parchado, roto, incompleto, etc. En la Figura 3-12 encontramos esta clasificación básica con la posibilidad de ampliar los estados no físicos y los físicos de acuerdo a las necesidades de cada Banco Central.

Por tanto, la definición de los estados no físicos y físicos deben de acordarse en las reglas del negocio y únicamente considerar que pueden existir diferentes tipos de ellos. También es importante considerar que dentro de las bóvedas pueden almacenarse diferentes estados físicos, pero esto también depende de las reglas con las que opere cada Banco Central.



Figura 3-12. Clasificación de los estados que conservan las piezas.

Otro punto que se ha mencionado es el control de las piezas por su denominación y características de fabricación. Adicionalmente hay que considerar que el valor facial de una pieza puede ser distinto al valor de real de ella, en el capítulo 2 se mencionó el caso de los nuevos pesos y las piezas anteriores a éstos. De ahora en adelante vamos a referirnos a una pieza con una denominación X y valor real Y como espécimen. Un espécimen es un billete sin valor, sin embargo, para fines de este documento lo consideraremos como una forma de distinguir e identificar una pieza por su valor y otras propiedades.

Si consideramos que un espécimen tiene características relacionadas con la denominación o valor facial, valor real y tipo de efectivo, y otras relacionadas con la fabricación, tendremos entonces dos formas de identificar las piezas: por su denominación y valor real y otra por sus características de fabricación. Dado que después de que las piezas han estado en circulación no tiene ningún valor agregado, el saber cuáles son las características de las piezas cuando regresan al Banco Central siendo aún aptas para circular, la utilización de un espécimen distinto permite un manejo ágil de las existencias. Veamos el siguiente ejemplo, tenemos el espécimen X con ciertas características, por otro lado tenemos el espécimen Y con otras características pero con la misma denominación y valor real, cuando estos especímenes regresan al Banco Central pueden manejarse como espécimen Z que no tendría asociada ninguna característica de fabricación, si estas piezas van a ser destruidas, en la ventanilla de clasificación se convertiría el espécimen Z en el espécimen X o Y según sea el caso.

En la Figura 3-13 encontramos el esquema de control de especímenes dentro de cada "mundo".

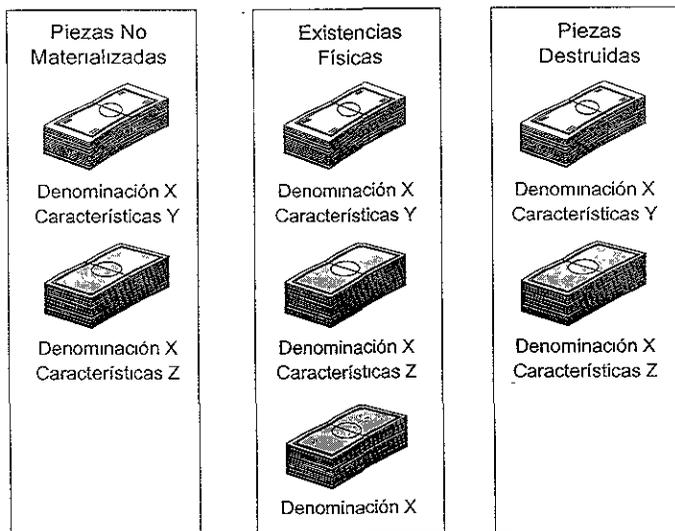


Figura 3-13. Esquema de manejo de los especímenes.

Este esquema de manejo de especímenes aplica perfectamente para cualquier tipo de existencias.

3.2 Control de existencias

El ciclo de vida de las piezas podemos dividirlo en tres grandes grupos, el grupo de las piezas no materializadas, el grupo de las piezas físicas y el grupo de piezas destruidas, como se muestra en la Figura 3-13 del apartado anterior.

En las piezas no materializadas encontramos aquellas que tienen un valor real pero carecen de un cuerpo físico. Básicamente son las piezas autorizadas y las piezas ordenadas para su fabricación. Los Bancos Centrales como mencionábamos pueden "crear" otros estados lógicos que mejoren o faciliten el control de las piezas en este grupo, por ejemplo piezas pendientes de ordenar, con esto sabrían de una forma fácil la diferencia entre el total autorizado menos las piezas ordenadas. Las piezas pertenecientes a este grupo deben identificarse por su denominación, valor real y características de fabricación, así como el estado lógico que guardan guardan (autorizadas, ordenadas o destruidas, por ejemplo).

El grupo de piezas destruidas la conforman aquellas piezas que dejaron de existir, tuvieron una vida útil y el deterioro por su uso las quitó de la circulación. Dado que todas las piezas están destruidas no tiene caso conservar el estado lógico actual, ya que en este grupo no hay otro estado lógico. Sin embargo, para los analistas de los Bancos Centrales es importante considerar las causas de deterioro más comunes y así tomar nuevas consideraciones para mejorar el control de calidad requerido a los fabricantes y materiales usados. Por tanto las piezas de este grupo deben de identificarse por su denominación, valor real, características de fabricación y el estado físico que conservaban hasta antes de la destrucción.

El grupo de piezas físicas es el más complejo de los tres, aquí encontramos todas las piezas que existen físicamente, sin importar donde estén. Para los flujos de efectivo este grupo puede ser visto como dos subgrupos, existencias del Banco Central y piezas en circulación.

Las existencias del Banco Central pueden dividirse a su vez en dos elementos: las piezas almacenadas en un sitio específico y las piezas en tránsito. Las piezas almacenadas en un sitio deben de identificarse por el sitio y bóveda donde se encuentran, el estado físico que conservan y la denominación, valor real y cuando aplique las características de fabricación. De esta forma se sabe cuántas piezas de cada espécimen y estado físico hay en Oficina Central, en cada Sucursal y en cada Corresponsal.

Las piezas en tránsito, al no poder ser asociadas a un sitio específico, deben controlarse por el tipo de transporte⁴, la denominación, valor real y el estado físico que conservan las piezas. La razón por la cual se utiliza el tipo de transporte en lugar del transportista es porque el detalle de cada movimiento

⁴ Terrestre, marítimo, avión, etc.

está contenido en la operación respectiva, para fines de control de piezas en tránsito basta con conocer por cuál medio se están transportando.

Dentro de las funciones de los Bancos Centrales encontramos la regulación del sistema financiero y esto abarca también las piezas en circulación. La determinación de cuántas piezas de cada denominación y valor real están circulando es responsabilidad de los sistemas de flujos de caja ya que ellos controlan todo el ciclo de vida de los billetes y monedas. Para contar con información veraz de las piezas en circulación se requiere un estricto control del ciclo de vida. El circulante lo podemos calcular como el total de piezas autorizadas por denominación y valor real menos las piezas no ordenadas menos las piezas ordenadas menos las existencias en el Banco Central menos las piezas en tránsito menos las piezas destruidas.

En la Figura 3-14 encontramos esquemáticamente los grupos de existencias.

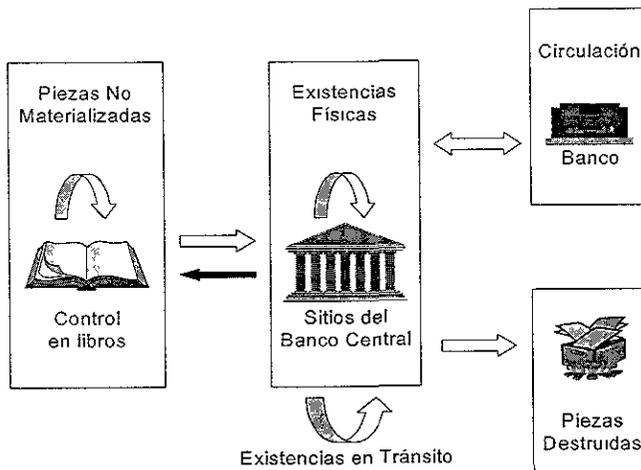


Figura 3-14. Grupos de existencias y los movimientos permitidos.

Tal y como sucede en el mundo real, lo que origina el cambio de una piezas dentro de los grupos o el paso de un grupo a otro son los movimientos que se realicen.

3.3 Control de operaciones

Los movimientos en mundo real de los flujos de caja son simples, salvo la autorización, todos los demás movimientos tienen un origen y un destino. Para poder sistematizar los movimientos es necesario asociarlos a una operación.

En el mundo real cuando se traslada una cierta cantidad de piezas de una bóveda a otra, disminuyen las existencias de la bóveda origen y se incrementan las existencias de la bóveda destino. Al sistematizar este movimiento se tiene que contar con un control en libros o electrónico de cuantas piezas hay en cada una de las bóvedas y éste es el objetivo de las operaciones, mantener información del movimiento y reflejarlo en los controles electrónicos que se posean. Ver Figura 3-15.

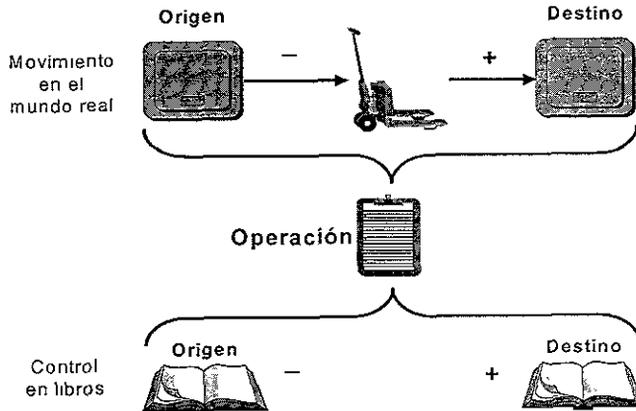


Figura 3-15. Sistematización de un movimiento

Podemos identificar varios tipos de operación que representan el ciclo de vida básico del efectivo. Un tipo de operación se convierte efectivamente en operación cuando tiene la información del movimiento.

Pueden existir varios tipos de operación que requieren los mismos datos pero que, por las reglas del negocio deben existir. Por ejemplo el movimiento de piezas de un sitio a otro puede ser catalogado simplemente como remesa, sin embargo, pueden quererse distinguir según los tipos de sitios involucrados, así el tipo de operación remesa se convertiría en un conjunto de tipos de operación, por ejemplo, remesa de Oficina Central a Sucursal, remesa de Sucursal a Corresponsal, remesa de Corresponsal a Corresponsal de la misma Sucursal, etc., esto puede facilitar el control de las operaciones y mejorar las validaciones, simplificando la implantación de las reglas del negocio. Entonces, además de tipos de operaciones tenemos también familias de operaciones.

Dada la naturaleza de las operaciones y los sitios involucrados, podemos prever que se pueden intentar clasificar las familias en grupos de operaciones, siguiendo con las remesas, podemos clasificar los tipos de operación que indican que salen de Oficina Central o Sucursales hacia un Corresponsal en un grupo que sea "Envíos" y cuando las remesas tengan como destino Oficina

Central o alguna Sucursal y provengan de un Corresponsal en un grupo que sea "Concentraciones".

Así, los movimientos son representados como operaciones, una operación es una instancia de un tipo de operación, los tipos de operación pertenecen a grupos de operaciones y los grupos pertenecen a familias de operaciones.

Las familias de operaciones básicas son:

- Autorización de emisión
- Orden de fabricación
- Entrega de billete o libranza de moneda
- Traspaso
- Remesa
- Depósitos de efectivo
- Retiros de efectivo
- Destrucción

Ahora analicemos cada una de estas familias.

En lo que respecta a las piezas no materializadas tenemos que existen las operaciones de Autorización y Orden de Fabricación. Estas operaciones solamente afectan piezas que están en este "mundo" de existencias no físicas. Existe la posibilidad de una cancelación que puede ser total o parcial. En la Figura 3-16 podemos observar las familias de operación de este tipo.

La Autorización es el permiso expreso de emitir hasta la cantidad de piezas que se indica de la denominación, valor real y características aquí indicadas. La autoridad competente basándose a las políticas monetarias, las situaciones económicas de su país y los estimados de piezas que habrá que fabricar para sustituir las piezas deterioradas, determinan la cantidad de piezas, esta operación deberá de incluir en el concepto piezas autorizadas de la denominación a la que se haga referencia el total de piezas.

Una vez autorizada la emisión, el Banco Central determina los planes de producción y genera la orden de fabricación, que puede ser una orden de impresión de billete o una de acuñación de moneda.

Podríamos hacer independientes en las operaciones las piezas autorizadas y las piezas ordenadas, llevando el control independiente de estas piezas, sin embargo no hay que descuidar los cálculos del circulante. A pesar de ser factible este manejo es preferible utilizar un esquema de "cajas" en el cuál, el contenido de una caja debe verse en la siguiente, así lo autorizado debe fluir a lo ordenado y lo ordenado a existencias. Bajo este esquema las piezas ordenadas deben descontarse de lo autorizado.

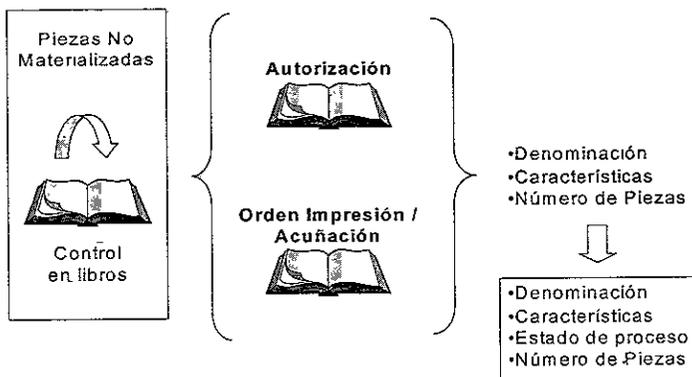


Figura 3-16. Familias de operaciones para piezas no materializadas.

Dado que una necesidad del circulante es conocer el total autorizado, en la operación de autorización se incrementará por un lado las piezas autorizadas y por otro lado las piezas no ordenadas. En la operación de Orden de Impresión se incrementarán las piezas ordenadas y se decrementarán las piezas no ordenadas.

En la Figura 3-17 se muestran gráficamente estas operaciones. La fabricación como ya se mencionó anteriormente no forma parte de los flujos de caja por tanto lo único que nos interesa es saber cuántas piezas están en este proceso.

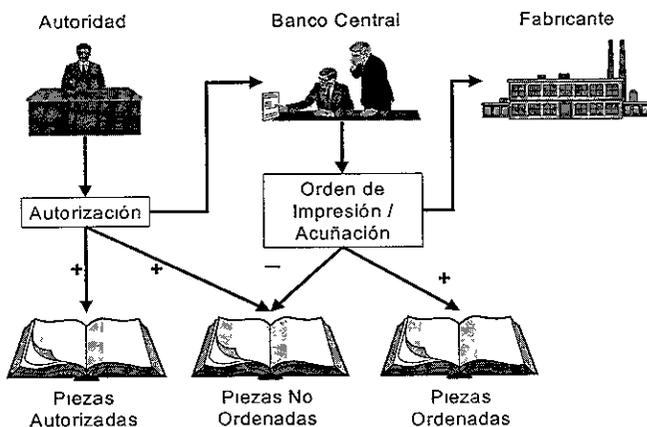


Figura 3-17. Esquemática de las de operaciones con piezas no materializadas.

El fabricante, en este punto del ciclo, es el encargado de convertir el dinero en físico, el movimiento de entrega de piezas es el enlace entre dos "mundos" distintos, y sólo a través de estas operaciones puede materializarse el dinero, como se muestra en la Figura 3-18.

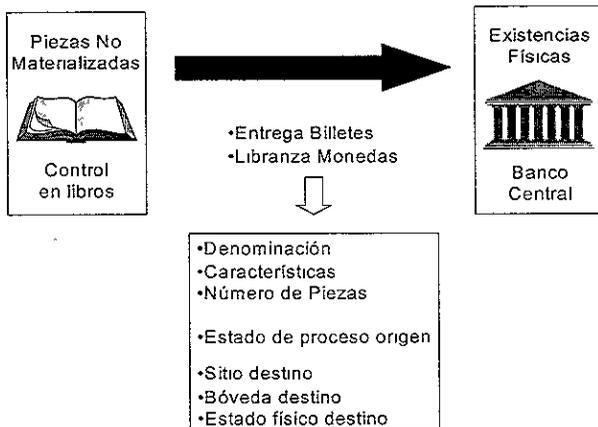


Figura 3-18. Entrega de Efectivo.

El fabricante informa al Banco Central la terminación del proceso de fabricación y se acuerda una fecha de entrega, al realizarse la entrega, se verifica que la cantidad entregada corresponda con lo especificado y se procede a descontar de las piezas ordenadas para formar parte de las existencias del sitio donde se efectúe la entrega. Ver Figura 3-19.

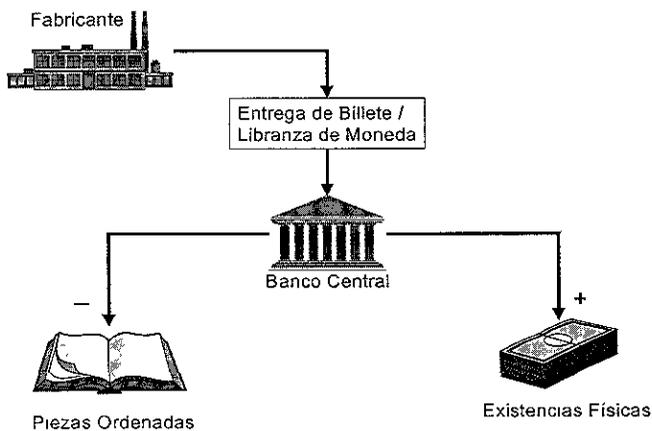


Figura 3-19. Esquematación de la entrega de efectivo.

Cuando las piezas entran al Banco Central forman parte de las Existencias, por tanto el modo de controlarlas difiere, ahora se requiere conocer la ubicación exacta de todas y cada una de las piezas. Es en esta parte es donde se tiene que considerar el sitio y la bóveda en que pueden estar las piezas y el estado físico que irán conservando.

Las piezas inicialmente son recibidas en una ventanilla, pero tienen forzosamente que almacenarse en bóvedas, después se tendrán que mover a otras bóvedas o ventanillas tanto para enviarlas a otros sitios como para proporcionar el servicio de conversión de dinero electrónico en dinero físico. A estas operaciones se les llaman traspasos. En la Figura 3-20 se ilustra el traspaso y la información requerida.

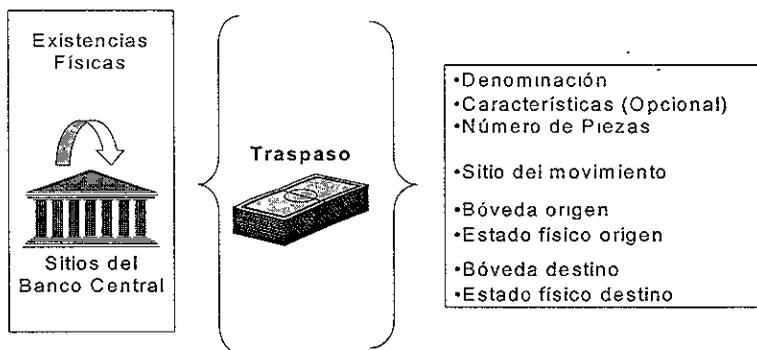


Figura 3-20. Traspaso de efectivo.

Los traspasos son un movimiento muy recurrente en el interior de los sitios, ya sea para empacar las piezas, almacenarlas o para someterlas a varios procesos. El origen de los traspasos puede ser muy diverso, desde la necesidad de abastecer una bóveda hasta cuestiones estratégicas o políticas del Banco Central.

La bóveda o ventanilla que requiera efectivo para cumplir una tarea específica solicita a la bóveda correspondiente que le dote de efectivo, la bóveda traslada las piezas a la bóveda destino y termina el movimiento, como se muestra en la Figura 3-21. Al cierre operativo, todas las ventanillas no deben tener piezas en su poder así que, sistemáticamente traspasan las piezas que tengan a la bóveda correspondiente. Esta operación se efectúa dentro de un sitio, así que se necesita conocer la bóveda y estado físico origen y la bóveda y estado físico destino. En estas operaciones pueden involucrarse más de un espécimen.

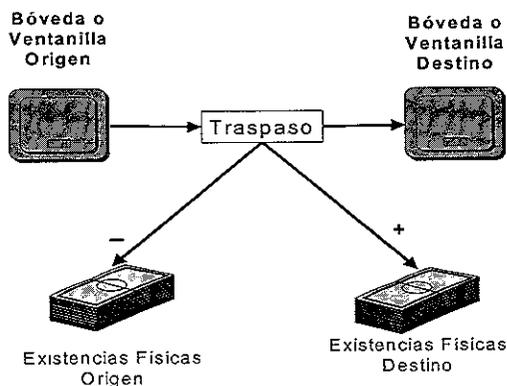


Figura 3-21. Esquematación de un traspaso.

Además de abastecer las bóvedas internas es necesario abastecer a las Sucursales y Corresponsales, para esto se utilizan las operaciones de remesa. Estas operaciones pueden ser vistas como dos operaciones que ocurren en momentos distintos pero que están íntimamente ligadas.

Una remesa, como se muestra en la Figura 3-22, es un movimiento de piezas de un sitio a otro, por tanto, puede pasar más de un día operativo para que las piezas lleguen al destino. A diferencia de los traspasos que a pesar de involucrar un movimiento de piezas, puede asumirse que el movimiento se da en el mismo instante, en las remesas esto no es válido ya que mientras las piezas están viajando no están ni en el origen ni en el destino.

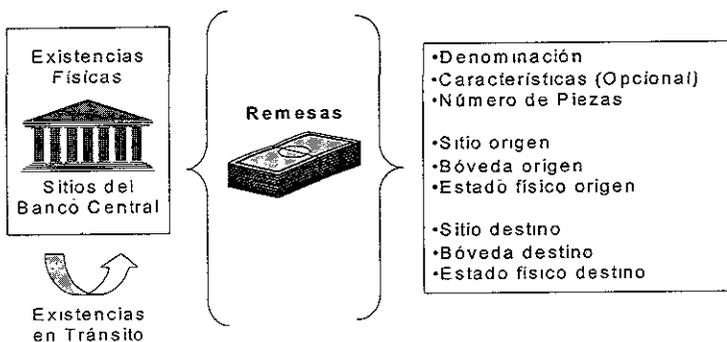


Figura 3-22. Remesas de efectivo.

Las remesas como movimiento requieren especificar: sitio, bóveda y estado físico origen; el sitio, bóveda y estado físico destino. Además del número de piezas y el espécimen que se está trasladando. Dado que una remesa involucra transporte se puede utilizar el mismo vehículo para trasladar piezas con diferentes estados físicos y en el destino estos estados físicos pueden ser los mismos que se utilizaron en el origen o bien otros dependiendo de las reglas del negocio.

El movimiento de remesa involucra descontar piezas en la bóveda del sitio origen e incrementar las existencias en tránsito, en este estado continuarán hasta que lleguen al destino en donde se disminuirán las existencias en tránsito y se incrementarán las existencias en el sitio destino, como se ilustra en la Figura 3-23.

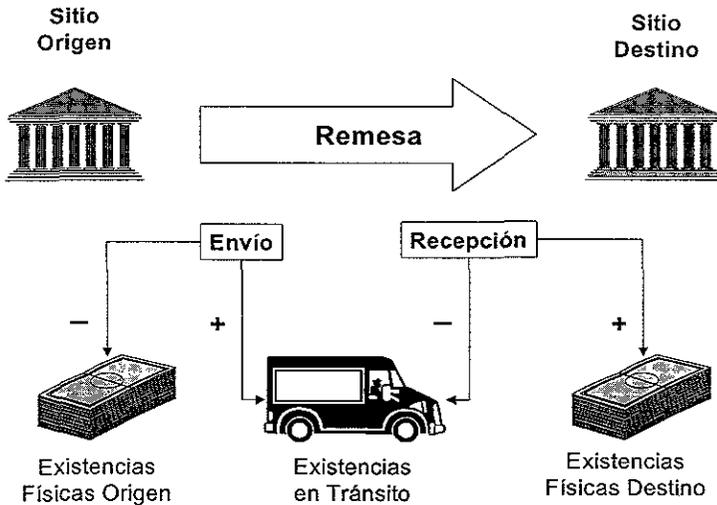


Figura 3-23. Esquematización de las remesas.

Las remesas como se mencionaba anteriormente, involucran dos operaciones distintas, el envío y la recepción (ver Figura 3-24). El envío sucede cuando el sitio origen embarca las piezas en el medio de transporte convenido, la operación decrementa las existencias en la bóveda del sitio origen e incrementa las piezas en tránsito. La recepción ocurre cuando las piezas llegan al sitio destino y son aceptadas por él, esta operación descuenta las piezas involucradas de piezas en tránsito y las agrega a las existencias de la bóveda del sitio destino.

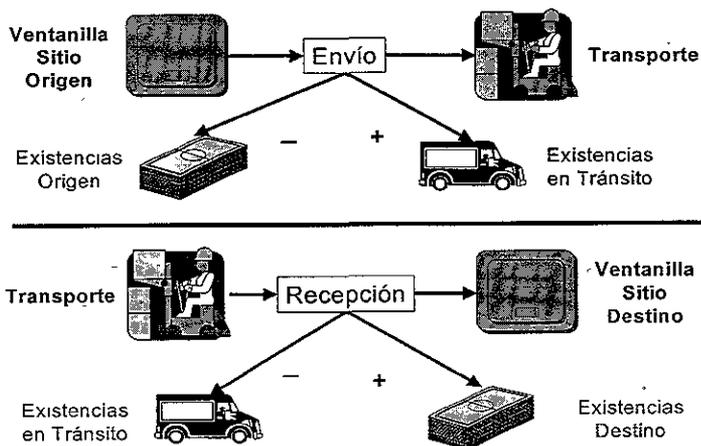


Figura 3-24. Envío y recepción de remesas.

Una vez analizado el abastecimiento de efectivo a los sitios vemos la conversión de dinero electrónico en físico. En la Figura 3-25 encontramos ilustrado como los bancos realizan depósitos y retiros en los Bancos Centrales.

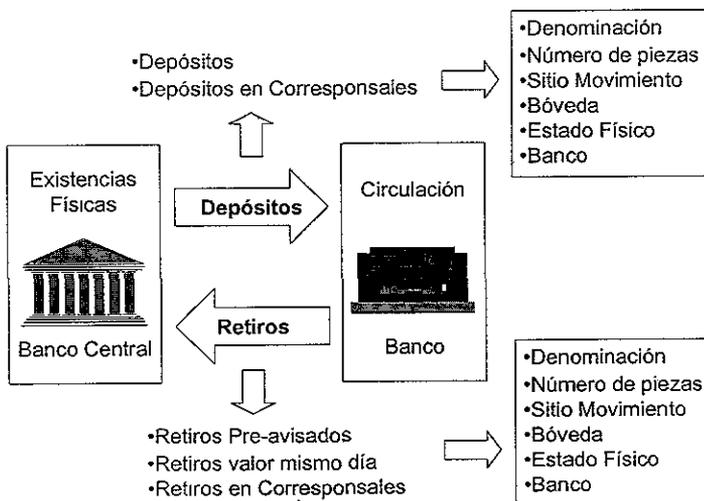


Figura 3-25. Depósitos y retiros de efectivo.

Los bancos acuden a los sitios del Banco Central a realizar depósitos y retiros. Los depósitos convierten dinero físico en dinero electrónico, y los retiros convierten dinero electrónico en dinero físico. Dado que la cuenta de los bancos es afectada con cada uno de estos movimientos, las tesorerías de los bancos requieren saber de antemano cuánto dinero necesitan en su cuenta para que se abastezcan sus sucursales de efectivo, ya que no todos los Bancos Centrales permiten sobregiros y aquellos que lo permiten pueden cobrar una penalización.

En los depósitos las tesorerías no tienen problemas de sobregiro porque es dinero que está ingresando a su cuenta, por tanto con consultarla bastará para saber cuanto tienen y poder hacer frente a sus compromisos o inversiones.

Los depósitos que realizan los bancos en los sitios del Banco Central originan un aumento de las existencias del propio sitio y un decremento de las piezas en circulación. Dado que el circulante puede calcularse, podemos omitir la afectación de las piezas en circulación y solamente afectar las existencias del sitio dónde se realiza el movimiento.

En los retiros, al igual que en los depósitos, podemos afectar solamente las existencias del sitio donde se realiza el movimiento. En esta operación se descuentan las piezas retiradas de las existencias del sitio.

En la Figura 3-26 se ilustran los depósitos y los retiros indicando el tipo de afectación que origina.

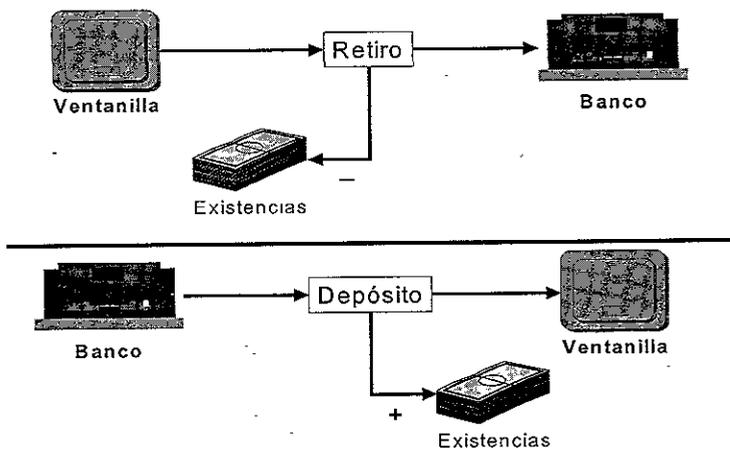


Figura 3-26. Esquematización de los depósitos y retiros.

Obedeciendo las reglas del negocio, pueden existir diferentes tipos de depósitos, por ejemplo: Depósitos en Sucursal y Depósitos en Corresponsales, y también

pueden existir diferentes tipos de retiros. En los retiros encontramos aquellos con fecha valor mismo día y con fecha valor día siguiente. Los retiros con fecha valor día siguiente o valor futuro se aplicarán en la fecha valor, lo que permite a los Bancos Centrales preparar las piezas para otorgar un mejor servicio y, a las tesorerías, prever estos movimientos y procurar tener el saldo necesario en su cuenta.

Después de circular el dinero, hemos visto que sufre deterioro hasta que prácticamente ya es inservible. Al llegar a este punto es necesario destruir las piezas para evitar bodegas inmensas almacenando dinero que no puede circular, además de constituir un riesgo para el propio Banco Central.

El proceso de destrucción nace de una orden de destrucción que genera el propio Banco Central. Él notifica a la bóveda correspondiente que traspase una determinada cantidad de piezas a la ventanilla de destrucción, en la ventanilla se procede a destruir las piezas. Este movimiento final implica un decremento en las existencias físicas del sitio y un incremento en las piezas destruidas.

En la Figura 3-27 se muestra como a través de la destrucción las piezas dejan de pertenecer a las existencias físicas, además se muestra la información mínima necesaria que conforma el movimiento.

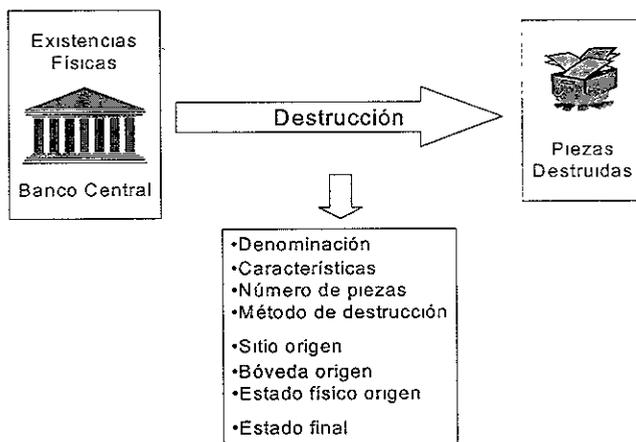


Figura 3-27. Destrucción de piezas.

En la Figura 3-28 se puede observar como las piezas dejan de pertenecer a una bóveda para formar parte de un control en libros.

Con esta operación de destrucción concluye el ciclo de vida del efectivo, pueden originarse más familias de operaciones dependiendo del Banco Central de que se trate, sin embargo éstas son las más representativas.

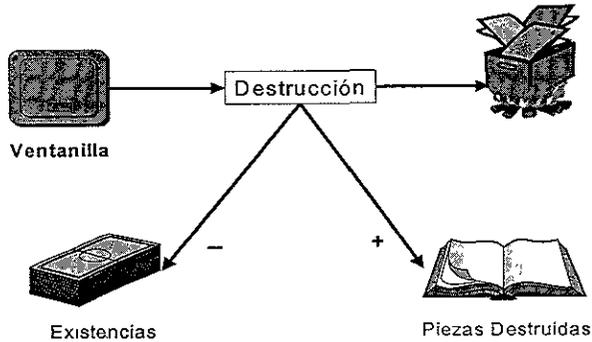


Figura 3-28. Esquematzación del proceso de destrucción.

Diagrama de Flujo de datos

Un Diagrama de Flujo de Datos (DFD) representa en forma gráfica el flujo que siguen los datos dentro de la organización, esto es: dónde se producen, hacia dónde se dirigen, qué transformaciones sufren y dónde son almacenados.

Los DFDs no solamente se utilizan para modelar el proceso de información, sino también para modelar la organización por completo o para realizar planeación estratégica y de negocios.

Algunos de los beneficios que se obtienen al utilizar los DFD's incluyen:

- Representación del sistema como una red de procesos.
- Identificación de los procesos y subprocesos del negocio, así como de las fuentes de datos externas y de los destinos de los datos.
- Entendimiento de los procesos diagramados.
- Representación de las necesidades del negocio a resolver.

Simbología utilizada en el Diagrama de Flujo de Datos

La simbología que utilizamos para describir el flujo de datos en los procesos de los flujos de caja está ilustrada en la Figura 3-29.

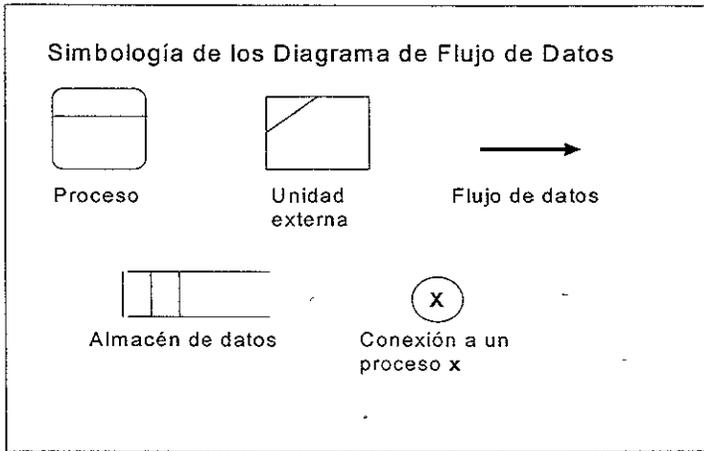


Figura 3-29. Simbología utilizada para los Diagramas de Flujo de Datos

Cada uno de los elementos de un DFD se representa a través de un símbolo. La notación utilizada en este documento está representada en la Figura 3-29 y explicada en la Tabla 3-1.

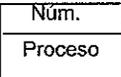
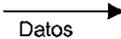
<p>Proceso</p> 	<p>Un proceso muestra la parte del sistema en la cual se realiza una transformación de datos. Se compone de una o varias entradas de datos, un proceso (función de transformación) y una o varias salidas de datos.</p>
<p>Flujo</p> 	<p>Un flujo de datos, describe el movimiento de datos a través del sistema. Se representa con una flecha que entra a o sale de un proceso.</p>
<p>Almacenamiento de datos</p> 	<p>Indica el momento en el que se requiere almacenar los datos.</p>
<p>Entidad externa</p> 	<p>Una Entidad Externa representa un sistema externo con el cual se requiere comunicación. Los sistemas externos pueden actuar como proveedores y/o usuarios de datos.</p>
<p>Conexión con un proceso X</p> 	<p>Permite conectar un flujo con algún proceso.</p>

Tabla 3-1. Simbología utilizada para los DFDs

Análisis del Flujo de datos

Para realizar el análisis del flujo de datos para los sistemas de flujos de caja nos auxiliamos de los DFDs previamente definidos.

En la Figura 3-30 se muestra el diagrama general de flujos de datos que le llamaremos diagrama cero, en él se pueden observar todas las funciones referentes al flujo de dinero en el Banco Central, que van desde la autorización de fabricación hasta la destrucción incluyendo la distribución.

En el proceso uno ilustrado en este diagrama, observamos que después de haberse otorgado la autorización de emisión, hay una aceptación de la ejecución, esto involucra registrar las piezas autorizadas en el registro correspondiente, con dicho procedimiento fluyen la información de las piezas autorizadas para generar la orden de impresión de billete y/o acuñación de moneda.

En el proceso dos, ya obtenida la orden de impresión o acuñación, se válida la orden de impresión y/o acuñación, de tal forma que con dicho proceso se obtiene la petición de piezas para el fabricante, dichas piezas serán controladas en libros electrónicos, tanto en piezas autorizadas y piezas ordenadas, todo esto contemplado en el proceso tres, cuatro y cinco.

En el proceso seis, al ser obtenidas las piezas ordenadas se mantiene el control de las existencias, este proceso se encarga de la distribución del dinero en los diferentes sitios, también atiende a las peticiones de los bancos, recibe las piezas fabricadas y, cuando las piezas ya están deterioradas, se las envía al centro de destrucción.

En la Figura 3-31 encontramos detallado el proceso seis en lo referente a atención a bancos. En los procesos 6.1, 6.3, 6.7 y 6.8, se pueden observar las validaciones para los depósitos y retiros de dinero a solicitud de los bancos comerciales ya que este proceso forma parte del control de existencias.

En el proceso 6.4, después de haber depositado las piezas el banco comercial, se cuentan y se clasifican en piezas aptas y piezas deterioradas.

Para las piezas aptas se lleva un control en las bóvedas, por lo que las piezas son verificadas antes de ser llevadas a ellas. Tal como se observa en el proceso 6.5.

En cuanto a las piezas deterioradas se tiene que llevar un control de la clasificación por denominación y características de las piezas, para ser almacenadas en las bóvedas. Esto se lleva a cabo en el proceso 6.6.

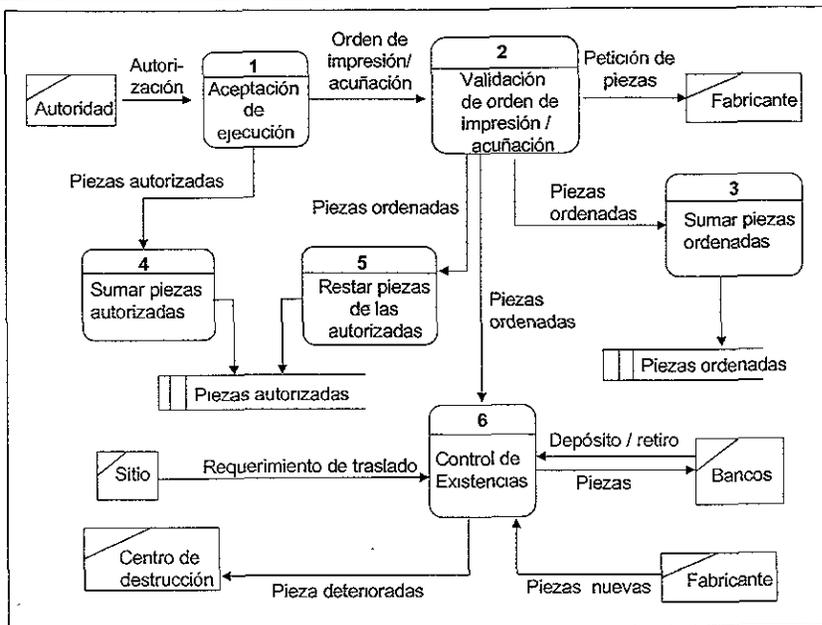


Figura 3-30. Diagrama de flujos de datos general.

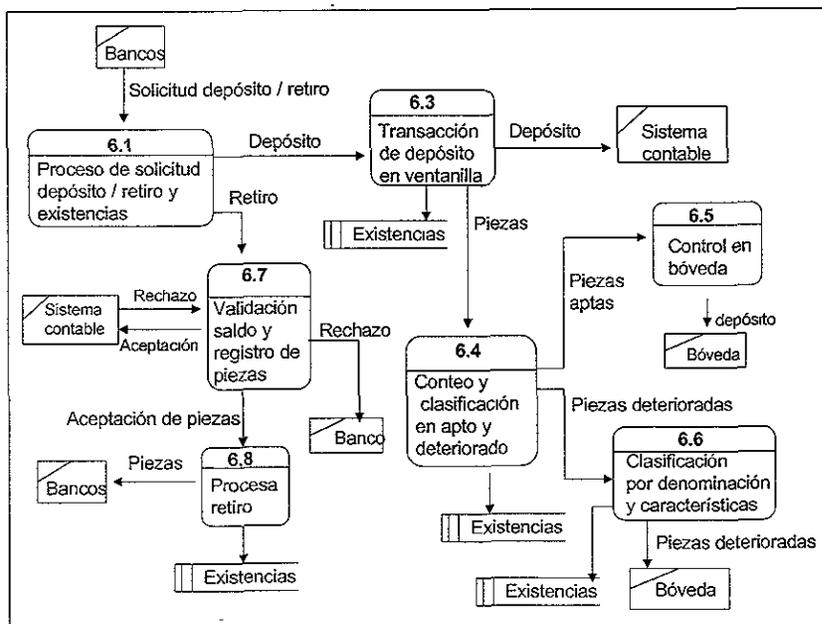


Figura 3-34. Diagrama del proceso 6.

En la Figura 3-32 se detalla el proceso 6.7, perteneciente al proceso 6 descrito en el diagrama anterior (ver Figura 3-31). Denota la validación de movimientos necesarios para que se efectúen los retiros en el Banco Central.

En este diagrama se lleva a cabo un control de las entradas y salidas de las piezas desde la bodega hasta la ventanilla de atención a usuarios, como se muestra en los procesos 6.7.1, 6.7.2 y 6.7.3 de este diagrama.

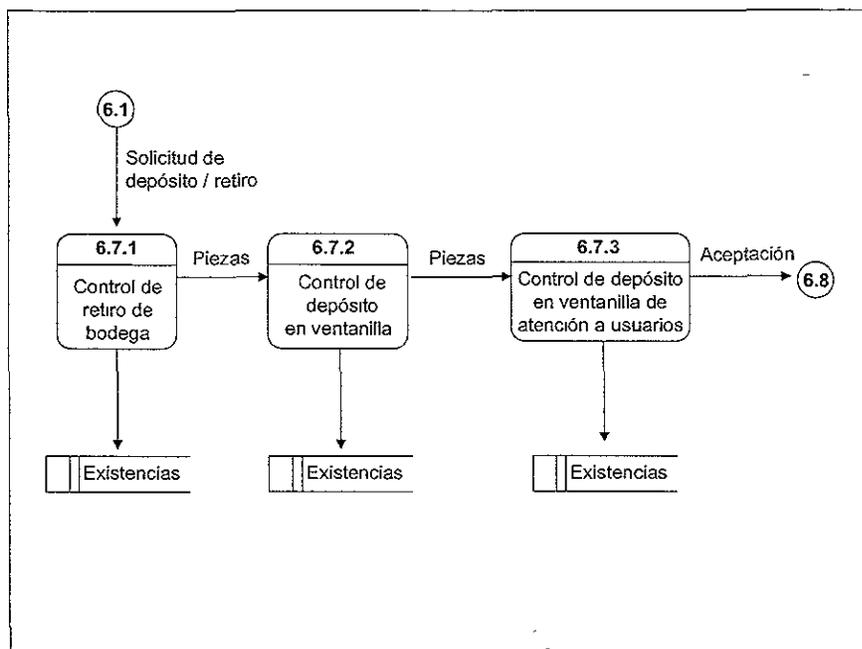


Figura 3-32. Diagrama del proceso 6.7.

En la Figura 3-33 encontramos el detalle del proceso 6.9, referente a la entrega de efectivo. En el proceso 6.9.1, se lleva a cabo un control de las piezas fabricadas para que cumpla con las características marcadas en los libros electrónicos, si dichas piezas no cumplen con las características fijadas se le devuelven al fabricante.

En el proceso 6.9.2, se mantiene el control de los depósitos en la ventanilla de las piezas que entran a ella.

En el proceso 6.9.3, después de haberse obtenido el control de las piezas en la ventanilla, se continúa con el control de los depósitos en la bóveda.

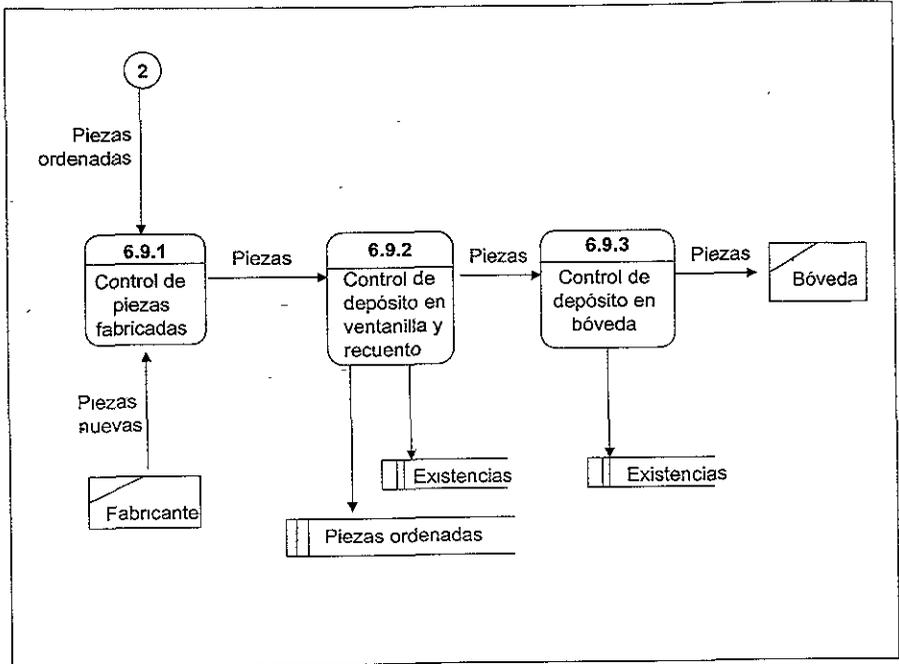


Figura 3-33. Diagrama del proceso 6.9.

En la Figura 3-34 y en la Figura 3-35 encontramos los diagramas de flujos de datos pertenecientes al envío y recepción de las remesas respectivamente.

En la Figura 3-34 encontramos los procesos que se llevan a cabo dentro del sitio origen, lo que incluye los movimientos de las piezas dentro de las bóvedas hasta que son confiados al transportista, ya que es importante el mantener el control de las piezas en tránsito por ser dinero de los Bancos Centrales.

En la Figura 3-35 encontramos los procesos involucrados en la recepción de las piezas en el sitio destino. Una vez que las piezas son entregadas por el transportista, se procede a contar las piezas antes de ser almacenadas en la bóveda que les corresponda, tal como se observa en los procesos 6.11.1 y 6.11.2.

El último paso del ciclo de vida del efectivo se encuentra ilustrado en la Figura 3-36, aquí se muestran los procesos involucrados en la destrucción de las piezas deterioradas, y por la importancia de este rubro es necesario contar con un control de las piezas destruidas.

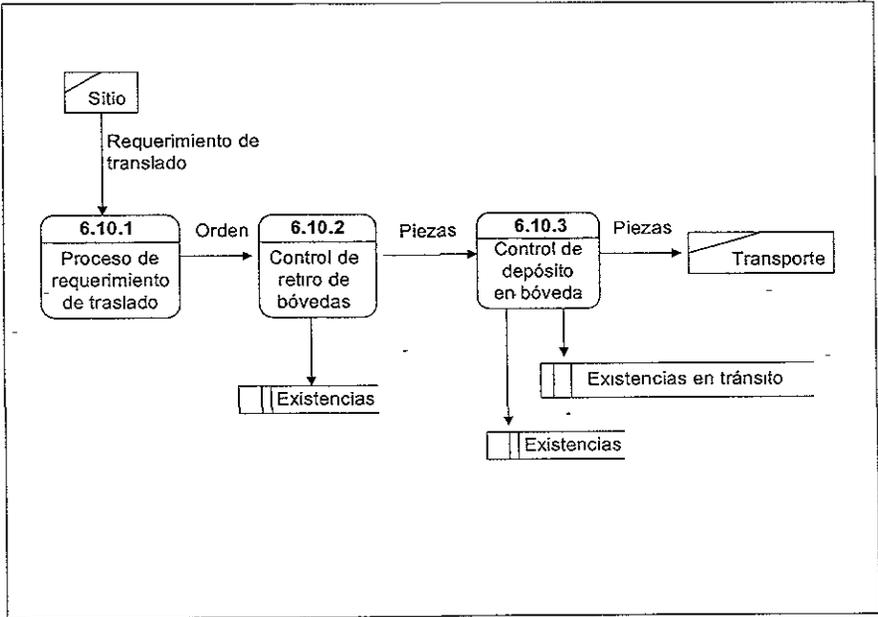


Figura 3-34. Diagrama del proceso 6.10.

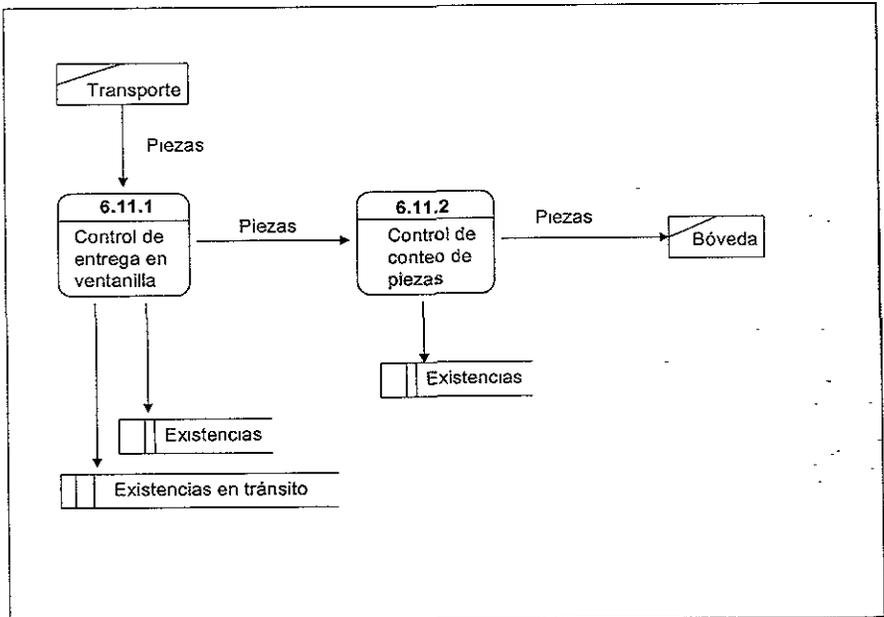


Figura 3-35. Diagrama del proceso 6.11.

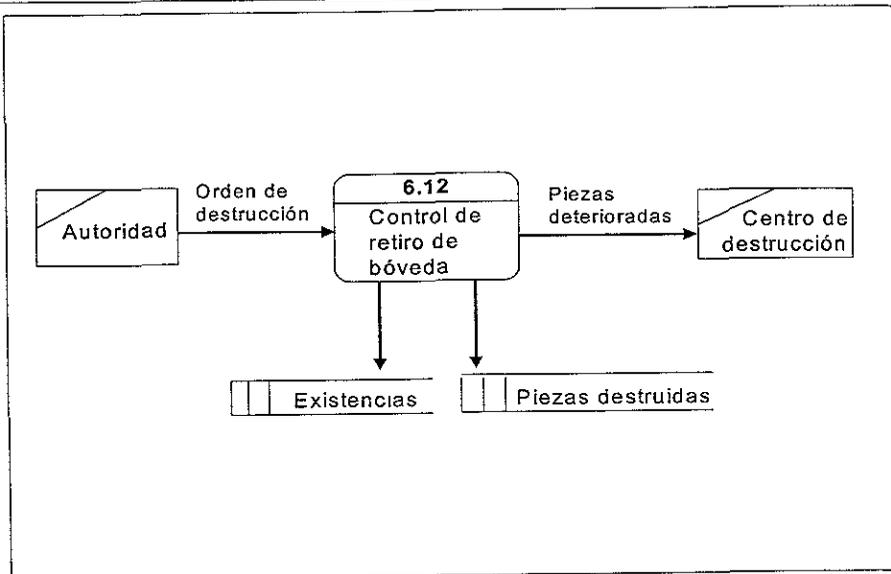


Figura 3-36. Diagrama del proceso 6.12.

3.4 Esquema de seguridad

La seguridad de la información es de suma importancia para cualquier sistema, podemos dividir la seguridad en diversos puntos o frentes, sin embargo, cada Banco Central tiene sus políticas generales en esta materia.

Para el análisis que estamos haciendo vamos a tocar solamente dos frentes que están directamente relacionados con los flujos de caja, para esto asumiremos que se utilizará un equipo de cómputo como plataforma.

El primer frente es a nivel plataforma, es decir, el esquema de seguridad mínimo que se requiere en la plataforma a utilizar y el segundo frente es a nivel aplicación.

Frente de seguridad en la plataforma

Cada usuario del sistema de flujos de caja debe identificarse en forma única. Dada la confidencialidad de la información es necesario controlar quién puede entrar al sistema, lo que nos lleva a la conclusión que cada usuario debe contar con una clave de acceso (*login*) y una contraseña (*password*). La contraseña puede ser asignada por el administrador de la plataforma, pero la plataforma deberá de permitir que el usuario la cambie ya que todo lo que realice en el sistema será responsabilidad del dueño de la clave de acceso. Así mismo, es

recomendable que se cuente con un esquema de vigencia de la contraseña, lo que obligará a los usuarios a cambiarla con la periodicidad que cada Banco Central determine.

Es común que un equipo de cómputo se utilice para operar más de una aplicación, por tanto no basta con la asociación de una clave de acceso y una contraseña para el esquema de seguridad, es necesario que se pueda restringir el acceso al lugar dónde resida esta aplicación. Si la plataforma cumple con este requerimiento se podrá permitir el acceso solamente a aquellas claves de acceso que sean usuarias.

Frente de seguridad en al aplicación

Una vez restringido el acceso a la plataforma y particularmente al lugar donde resida la aplicación es necesario analizar el esquema a utilizar dentro de la aplicación.

Por la distribución del trabajo, los Bancos Centrales utilizan diversos departamentos para controlar las operaciones. A su vez cada departamento distribuye el trabajo entre sus empleados. De esta forma controla de mejor manera el detalle de cada operación. En la Figura 3-37 ilustramos esta distribución.

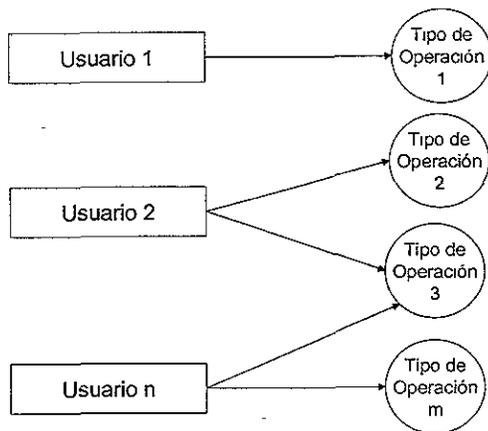


Figura 3-37. Esquema de distribución de las operaciones por usuario.

Como se mencionó en el capítulo 2, no solamente la integración de la información es importante, también lo es la obtención de la misma. Los usuarios que integran operaciones en la aplicación consultan lo que ellos mismo integraron, a su vez, el responsable de cada departamento puede consultar lo que integró el personal a su mando. Los directivos o mandos altos del Banco

Central obtienen otro tipo de información que engloba la situación de los flujos de efectivo.

Dada la estructura de los Bancos Centrales para distribuir el efectivo en el territorio nacional, encontramos que varias personas en distintas localidades realizan funciones semejantes, por ejemplo los encargados de atender depósitos y retiros de la banca, controlan los mismos tipo de operación, sin embargo solamente pueden obtener información relativa a lo acontecido en su sitio. Si el sitio en cuestión es una Sucursal también pueden obtener información de lo acontecido en los Corresponsales que están en su jurisdicción.

En resumen, la integración y consulta de información relativa a los movimientos está restringida en varios niveles. En cuanto a la integración de información, cada usuario integra solamente información de su sitio y en caso de laborar en el Banco Central integra también información de cualquiera de sus Corresponsales.

Para atender cualquier problema para integrar información existe un área que funge como coordinadora que puede integrar información de cualquier sitio.

Nivel de seguridad

Para restringir el acceso a la información existen varios niveles de seguridad, la información integrada en el sitio, en un sitio y sus Corresponsales y toda la información integrada. Anticipando un esquema más completo es conveniente considerar algunos niveles adicionales como la información integrada por el usuario, la información integrada desde un punto o terminal, lo que ofrecerá mayor confidencialidad de la información.

La información que se obtiene, principalmente para los mandos altos, no consiste únicamente en el detalle de integración, se obtiene información consolidada o, como en el circulante, información procesada de alguna forma.

Ahora bien, en lo que respecta a la integración, se explicó con anterioridad la necesidad de reflejar los flujos de caja en el sistema contable, estos sistemas normalmente fijan horarios de operación, lo que obliga a los encargados de integrar información a realizar su trabajo dentro del horario permitido, el cual es establecido por el área normativa.

A pesar de necesitar información oportuna, hay información que no debe ser obtenida hasta que se concluya la integración, como pueden ser las existencias de cierre de una bóveda; se pueden obtener las existencias en el momento pero la de cierre sólo hasta que se hayan integrado todos los movimientos en los que se involucró la bóveda.

Otro aspecto importante es la relación entre los sistemas contables y los flujos de caja. Los movimientos deben ingresar al sistema contable dentro de los horarios permitidos, sin embargo, no están exentos de alguna contingencia que

ESTA TESIS NO SALE DE LA BIBLIOTECA

CAPÍTULO 3. Análisis

ocasione un retraso en la integración, por tanto el área coordinadora de los flujos de efectivo notifica al sistema contable cuando existe algún problema, o bien cuando es necesario ampliar el horario. Así mismo, una vez que el coordinador ha determinado que todos los movimientos fueron integrados, impide la integración de más información y procede a realizar los respaldos necesarios.

Resumiendo, el esquema de seguridad considera dos frentes, el de la plataforma y el de la aplicación. El frente de la plataforma requiere control de acceso al equipo y el control de acceso al lugar donde se encuentre la aplicación. En el frente de la aplicación se requiere una identificación de los privilegios que tiene cada usuario y el horario para realizarlos.

Servicios

Dado que la integración y la obtención de información pueden estar sujetas a un horario, y los permisos que tiene un usuario son variados, podemos recurrir al concepto de servicio. En la arquitectura cliente – servidor, el servidor es el encargado de atender las peticiones del cliente, las peticiones que podrá atender corresponden a los servicios que proporciona, por ejemplo si tenemos un servidor de archivos y el cliente solicita ejecutar una consulta en una base de datos, el servidor no podrá atender esta petición por carecer de la base de datos. De esta forma los servicios que proporciona el sistema de flujos de caja son:

- Servicios de captura
- Servicios de consulta de captura
- Servicios de extracción de reportes
- Servicios de administración
- Servicios generales

Los servicios de captura corresponden a la integración de la información, cada tipo de operación puede ser catalogada y asociarse una clave que servirá para distinguirla de las demás.

Los servicios de consulta de captura corresponden a la obtención del detalle de la información integrada, a pesar de obtener la misma información cada servicio de esta naturaleza restringe los datos devueltos, según el nivel de consulta que tenga el usuario, por ejemplo, sólo se podrá consultar lo integrado en su sitio o lo integrado en su sitio y en sus Corresponsales.

Los servicios de extracción de reportes corresponden a la obtención de la información consolidada y/o procesada.

Los servicios de administración son aquellos a través de los cuáles se otorgan permisos sobre otros servicios, se determinan los horarios, se realizan los respaldos, etc.

Los servicios generales son aquellos que todo usuario debe tener, por ejemplo, el cambio de contraseña.

Al tipificar todas las actividades realizables en los sistemas de flujos de caja como servicios, podemos pensar en un esquema en el cual, a cada usuario se asocien los permisos correspondientes.

Grupos de usuarios

Dada la distribución del trabajo y la estructura de los Bancos Centrales, podemos ver que existen grupos de usuarios que tienen prácticamente los mismos permisos, podemos ejemplificar nuevamente a los usuarios encargados de la integración de información de depósitos y retiros. En varias empresas se manejan niveles de seguridad, en los cuáles los usuarios de un nivel básico tienen un conjunto de permisos, el nivel siguiente tiene los mismos permisos que el anterior más algunos otros, y así sucesivamente. En los flujos de efectivo este esquema no aplica, ya que el nivel de captura no involucra poder integrar todos los tipos de operación, de hecho, los grupos de permisos de usuario pueden no ser mutuamente exclusivos. Por tanto el esquema de asignación de permisos tiene que ser lo suficientemente flexible para soportar esta necesidad.

Para resolver este requerimiento podemos crear grupos de usuarios a los cuales se les asignarán los permisos correspondientes. De esta forma podemos tener tantos grupos como requiera cada Banco Central. Como una persona, ya sea temporal o definitivamente puede tener los derechos de más de un grupo, por ejemplo los administradores, los permisos que tenga un usuario será la unión de los servicios de todos los grupos a los que pertenezca.

Para no dejar cabos sueltos, podemos imaginar el caso hipotético de que un usuario tenga todos los permisos de grupo excepto uno, este caso lo podemos aunar con el hecho de que en algunas ocasiones un usuario tiene uno o más permisos adicionales a su grupo. Para evitar una creación exagerada de grupos que tengan sólo a un usuario, podemos agregar servicios de forma individual o bien restringir permisos a un usuario aun cuando el grupo al que pertenezca los tenga.

Horarios

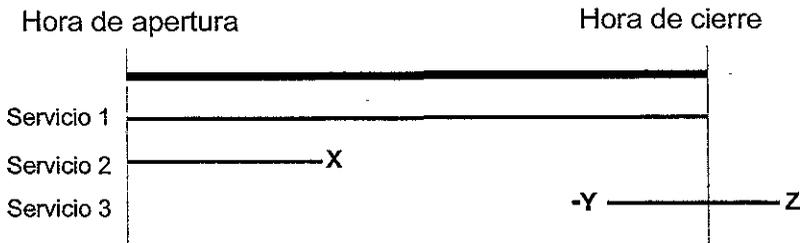
Dada la variedad en los horarios de los servicios, podemos encontrar que el horario de un servicio es distinto a los demás servicios del conjunto al que pertenece. Por esta razón es conveniente que cada servicio tenga un horario en el cual sea válido, por ejemplo, en los servicios de extracción de reportes, algunos pueden obtenerse a cualquier hora, otros solamente cuando el horario de captura haya concluido. Este esquema ofrece independencia de horario a cada servicio, lo cual, en casos de excepción de horario de una operación permitirá mantener el control sobre las operaciones que pueden capturarse fuera de su horario normal.

No hay que perder de vista que en caso de una contingencia, la operación de un día puede concluir hasta el día natural siguiente. En esta situación todo lo que se registre en el sistema de flujos de caja, sin importar el día natural que sea, pertenecen a un día operativo.

Si imaginamos la cantidad de servicios que puede ofrecer un sistema de flujos de caja, considerando que los Bancos Centrales pueden crear tantos tipos de operación como deseen, las consultas asociadas, los reportes y las actividades de control; el número de servicios sería considerable y mover el horario de una forma individual a cada uno de ellos en caso de una contingencia requeriría una considerable cantidad de tiempo. Para evitar este tipo de conflictos, podemos parametrizar los horarios de los servicios a horas de referencia.

Como la operación es diaria, es decir, cada día operativo tiene un inicio y un fin, aún cuando termine el día natural siguiente se debe realizar un corte. Por tanto podemos asumir que tiene un horario de apertura y uno de cierre y ambos son establecidos por cada Banco, estas horas podemos considerarlas como nuestras horas de referencia.

Para determinar el horario válido de un servicio es necesario fijar dos puntos en tiempo, y para hacerlo flexible, nos podemos auxiliar en las horas de referencia. Tanto la hora de inicio como la de fin del horario válido de un servicio, podrán fijarse a la apertura o al cierre, indicando cuánto tiempo antes o después de la hora de referencia toma efecto. Supongamos que la operación normal ocurre en el horario de 9:00 A.M. a 6:00 P.M., la hora de apertura sería las 9:00 A.M. y la hora de cierre las 6:00 P.M., ahora supongamos que el horario de un servicio es de las 9:30 A.M. hasta las 5:00 P.M. Siguiendo el esquema propuesto, se tendrían que fijar dos puntos en el tiempo de tal forma que el horario cumpla con lo estipulado. El punto de inicio sería configurado tomando como referencia la hora de apertura y sumándole 30 minutos y el punto de fin del horario se calcularía tomando como referencia la hora de cierre y restándole 1 hora. En caso de algún retraso con el simple hecho de recorrer la hora de referencia de cierre se ajustaría automáticamente el horario del servicio. Sólo en aquellos casos donde un conjunto reducido de servicios deban mover su horario se modificarían los parámetros que delimitan el servicio, ajustando ya sea el tiempo antes o después de la hora de referencia o la hora que se toma como referencia (apertura o cierre). En la Figura 3-38 esta ejemplificado el esquema de horarios.



Servicio 1.

Referencia inicio: **Apertura** Tiempo a agregar: **0**
 Referencia fin: **Cierre** Tiempo a agregar: **0**

Servicio 2.

Referencia inicio: **Apertura** Tiempo a agregar: **0**
 Referencia fin: **Apertura** Tiempo a agregar: **X**

Servicio 3.

Referencia inicio: **Cierre** Tiempo a agregar: **-Y**
 Referencia fin: **Cierre** Tiempo a agregar: **Z**

Figura 3-38. Configuración de los horarios.

3.5 Esquema de validaciones

Las reglas del negocio varían entre los Bancos Centrales, a pesar de existir prácticamente los mismos escenarios, los movimientos que permiten dependen de las estructuras con que cuentan y de las políticas internas con las cuales se rigen. Cada Banco Central determina que movimientos internos o externos son válidos, así como la cantidad de tipos de bóvedas, estados físicos y más aún especímenes.

Por tanto necesitamos un esquema que permita validar los movimientos, que sea flexible y se acople con facilidad a cualquier cambio que pueda existir en las reglas del negocio.

Partamos entonces de los distintos puntos donde se pueden localizar las piezas, a estos puntos los llamaremos en lo sucesivo "nodos". Recordando, las piezas en producción son aquellas que están autorizadas o en proceso de fabricación, por lo que no pueden ser asignadas a un sitio o a una bóveda. Las piezas físicas propiedad del Banco Central pueden estar en su Oficina Central, en alguna de sus Sucursales o en algún Corresponsal, aunque también pueden estar en tránsito. Cuando las piezas pasan a poder de la banca se consideran que están en circulación, sin importar si las conserva el banco o las tienen los usuarios.

Finalmente, cuando termina la vida útil de las piezas se registran en piezas destruidas. Estos puntos están ilustrados en la Figura 3-39

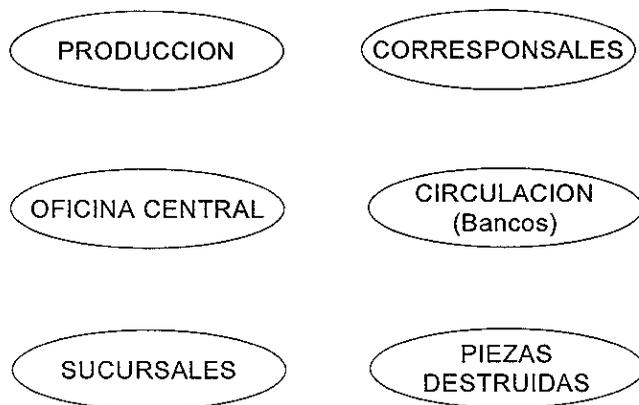


Figura 3-39. Puntos o "nodo" donde se localizan las piezas.

Las piezas tienen distintos movimientos válidos dentro de cada nodo, aun cuando no se conozca con exactitud todas las reglas del negocio de todos los Bancos Centrales, sí podemos generalizar por la información necesaria para realizar un movimiento.

En la Figura 3-40 podemos visualizar los movimientos en cada nodo. En el nodo PRODUCCION, los movimientos sólo pueden ser de un estado lógico a otro, ya que no tienen una ubicación física. En el nodo que representa a la Oficina Central, Sucursales y Corresponsales, los movimientos son de una bóveda – estado físico origen a otra bóveda – estado físico destino. En el nodo de CIRCULACION hay diversos movimientos, sin embargo para el objetivo de los sistemas de flujo de caja, no se tiene control sobre ellos. Por último, en el nodo PIEZA DESTRUIDA no se efectúa ningún movimiento.

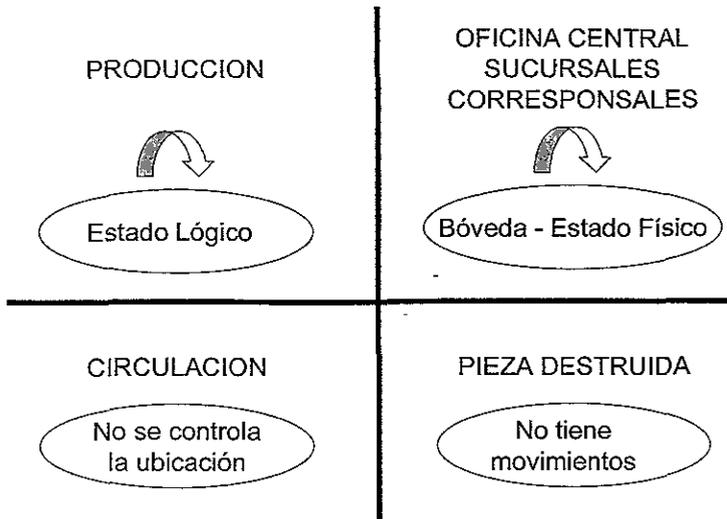


Figura 3-40. Movimientos en los nodos.

Una vez identificados los movimientos posibles dentro de cada nodo veamos los movimientos en nodos. En la Figura 3-41 vemos los movimientos posibles entre los nodos. Para que las piezas dejen de pertenecer al nodo PRODUCCION, se requiere de un movimiento originado por la entrega de billete o moneda, lo que ocasiona que las piezas ahora formen parte de las existencias del nodo Oficina Central o bien del nodo Sucursal. Si las piezas están defectuosas podrán ser devueltas al fabricante, lo que origina un regreso al nodo PRODUCCION. Otros movimientos posibles se generan con el abastecimiento a Sucursales o Corresponsales, que están agrupados bajo la operación remesa. Con esta operación las piezas dejan de pertenecer al nodo origen para formar parte de las existencias del nodo destino. Para llevar el dinero a la población, hemos visto que los Bancos Centrales se auxilian de la banca, así, los bancos a través de depósitos y retiros mueven piezas del nodo Oficina Central, Sucursal o Corresponsal a la circulación. Y finalmente, cuando las piezas son destruidas, dejan de formar parte del nodo Oficina Central para formar parte del nodo PIEZAS DESTRUIDAS.

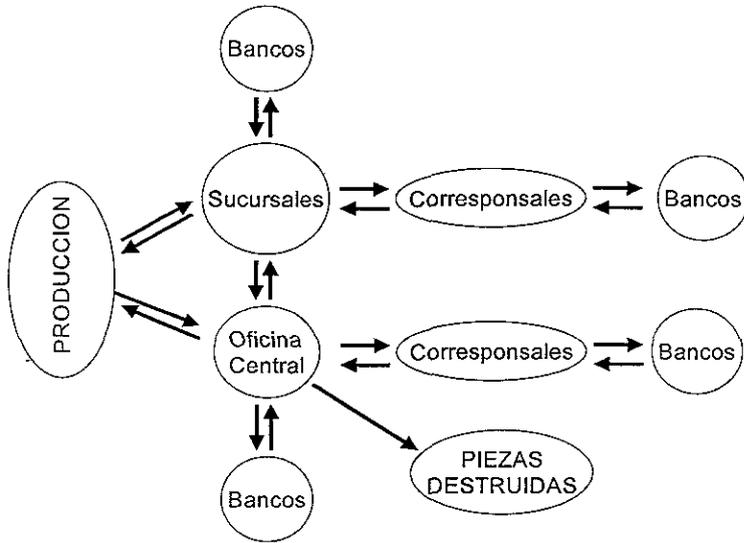


Figura 3-41. Movimientos entre nodos.

Los nodos Oficina Central y Sucursales forman parte de la estructura interna de un Banco Central, entonces para simplificar el modelo podemos agruparlos como un mismo nodo, ya que los movimientos internos son los mismos y los externos (de un sitio a otro) los podemos representar como una arista circular. En la Figura 3-42 encontramos el resultado de la simplificación y podemos observar que los flujos se mantienen.

Dado que los Corresponsales, a pesar de sucursales bancarias, funcionan como una parte de la Caja de los Bancos Centrales, podemos resumir aún más el modelo y considerarlo como un solo nodo Banco Central, el cuál, en su interior contiene los distintos sitios que lo conforman. En la Figura 3-43 encontramos el modelo final.

Recordemos que lo que origina un cambio de ubicación física de las piezas o un cambio en su estado lógico es el movimiento que se realiza. Por tanto podemos decir que dependiendo del tipo de operación será la regla del negocio que se debe contemplar.

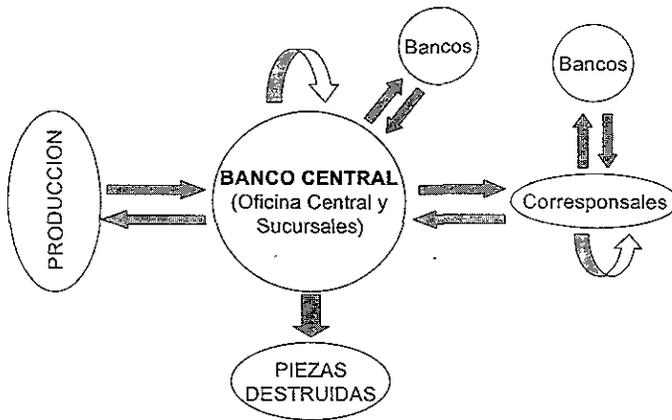


Figura 3-42. Modelo simplificado de los movimientos entre nodos

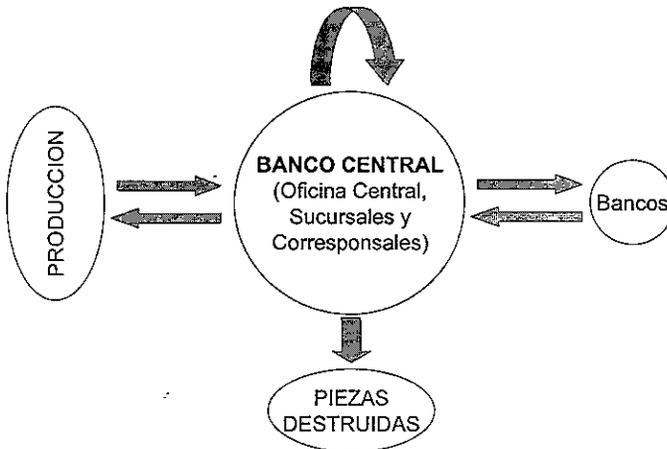


Figura 3-43. Modelo final de movimientos entre nodos.

Considerando la información ilustrada en la Figura 3-40 para cada nodo representado en la Figura 3-43, encontramos que las reglas del negocio para cada nodo las podemos describir como se muestra en la Tabla 3-2.

NODO	Elementos requeridos
PRODUCCION	T – EL origen – EL destino
BANCO CENTRAL	T – (B – EF) origen – (B – EF) destino

Tabla 3-2. Elementos requeridos para realizar movimientos en cada nodo.

En esta tabla **T** representa el tipo de operación, **EL** el Estado Lógico, **B** el Tipo de Bóveda y **EF** el Estado Físico.

Siguiendo con la misma nomenclatura y agregando **S** como Sitio y **U** como banco usuario, podemos describir las reglas del negocio relativas a las aristas o enlaces entre nodos como se muestra en la Tabla 3-3.

ARISTA	Elementos Requeridos
PRODUCCION – BANCO CENTRAL	T – EL origen – (S – B – EF) destino
BANCO CENTRAL – PRODUCCION	T – (S – B – EF) origen – EL destino
PRODUCCION – BANCO CENTRAL	T – (S – B – EF) origen – (S – B – EF) destino
BANCO CENTRAL – CIRCULACION	T – (S – B – EF) origen – U destino
CIRCULACION – BANCO CENTRAL	T – U origen – (S – B – EF) destino
BANCO CENTRAL – BANCO CENTRAL	T – (S – B – EF) origen – (S – B – EF) destino
BANCO CENTRAL – PIEZA DESTRUIDA	T – (S – B – EF) origen – EF destino

Tabla 3-3. Elementos requeridos para realizar los movimientos representados por aristas.

Considerando las dos tablas anteriores, podemos representar todas las reglas del negocio de un Banco Central. En la Tabla 3-4 se muestran los distintos requerimientos para validar un movimiento, ya sea en un nodo o en una arista. Siendo el estímulo, para cualquier caso, el tipo de operación. Para mayor claridad en esta tabla ya se muestra dividida la información requerida en el origen y en el destino.

⁵ En este momento no consideraremos el espécimen ni la cantidad de piezas involucradas

NODO o ARISTA	Estímulo	Origen	Destino
PRODUCCION	T	EL	EL
BANCO CENTRAL	T	B – EF	B – EF
PRODUCCION – BANCO CENTRAL	T	EL	S – B – EF
BANCO CENTRAL - PRODUCCION	T	S – B – EF	EL
BANCO CENTRAL – BANCO CENTRAL	T	S – B – EF	S – B – EF
BANCO CENTRAL - CIRCULACION	T	S – B – EF	U
CIRCULACION – BANCO CENTRAL	T	U	S – B – EF
BANCO CENTRAL – PIEZA DESTRUIDA	T	S – B – EF	EF

Tabla 3-4. Elementos requeridos para cualquier movimiento.

Considerando que un sitio realmente es una institución en una plaza, y que en varios sitios se pueden realizar los mismos movimientos y además varios sitios pueden tener la misma estructura interna⁶, podemos definir un tipo de sitio. Este tipo de sitio servirá para agrupar sitios con las mismas características, los tipos de sitio básicos son: Oficina Central, Sucursal del Banco Central y Corresponsal. Aunque para facilitar el control de las reglas del negocio podemos considerar también como tipo de sitio, aunque no lo sean propiamente, a los fabricantes de billete, fabricantes de moneda y bancos usuarios. La importancia de este concepto es la asignación de reglas a varios sitios sin tener que definir las en forma individual. Veamos el siguiente ejemplo, supongamos que en todas las sucursales hay una ventanilla de atención a usuarios y una bóveda de billete apto, el tipo de operación traspaso tiene una regla que permite mover piezas de esta bóveda a la ventanilla, si no usamos un tipo de sitio para definir la regla se tendrían que definir tantas reglas como Sucursales tenga el Banco Central, cambiando solamente el sitio; si por el contrario, definimos esta regla para el tipo de sitio SUCURSAL, la regla se tendrá que definir una sola vez.

A pesar de que el tipo de sitio facilita el establecimiento de las reglas del negocio, es importante considerar que pueden existir excepciones, tanto movimientos que sólo puede realizar un sitio como movimientos que todos los sitios de un mismo tipo pueden realizar excepto alguno. Por tanto es conveniente crear tipos de regla que sean Generales (aplican a todos los tipos de sitio), Adiciones (permisos a algún sitio de realizar un movimiento) o Restricciones (que indiquen a quién no se le permite el movimiento).

⁶ Para evitar conflictos con la forma de escoger el tipo de bóveda en los corresponsales, es factible que a cada Corresponsal se le asigne un mismo tipo de bóveda y manejan prácticamente el mismo conjunto de estados físicos. Este mismo caso puede aplicar en las Sucursales del Banco Central.

El esquema resultante está ilustrado en la Tabla 3-5, donde **EL** representa el estado lógico, **Ts** el tipo de sitio, **I** la institución (que puede ser un banco o el Banco Central), **P** la plaza o localidad, **B** la Bóveda, **EF** el estado físico y **U** el banco usuario, aunque pudiéramos representarlo como institución ya que los bancos forman parte de las instituciones con las que operan los Bancos Centrales; por claridad los manejaremos con distinta identificación.

NODO o ARISTA	Estímulo	Origen	Destino
PRODUCCION	T	EL	EL
BANCO CENTRAL	T	B - EF	B - EF
PRODUCCION - BANCO CENTRAL	T	EL	(Ts ó I-P) - B- EF
BANCO CENTRAL - PRODUCCION	T	(Ts ó I-P) - S - B - EF	EL
BANCO CENTRAL - BANCO CENTRAL	T	(Ts ó I-P) - B - EF	(Ts ó I-P) - B - EF
BANCO CENTRAL - CIRCULACION	T	(Ts ó I-P) - B - EF	U
CIRCULACION - BANCO CENTRAL	T	U	(Ts ó I-P) - B - EF
BANCO CENTRAL - PIEZA DESTRUIDA	T	(Ts ó I-P) - B - EF	EF

Tabla 3-5. Esquema completo de requerimientos para cada movimiento.

El esquema anterior nos ayuda a controlar parcialmente las reglas del negocio, con esto podemos restringir que los movimientos sean validos en cuanto a origen y destino, sin embargo es necesario controlar también la estructura de donde se controlarán las existencias.

Como las existencias en producción, las existencias en tránsito y las piezas destruidas no tienen un sitio donde se ubiquen, el mismo esquema utilizado para validar los movimientos ayudan a controlar el flujo. En el caso de las existencias físicas es importante contar con un modelo que permita conocer la estructura.

Antes de analizar este control sobre la estructura de los sitios hay que considerar que los Bancos Centrales están en libertad de manejar tantos especímenes como lo deseen, por tanto el proceso de agregar en todos los estados físicos que son almacenados en todas las bóvedas de todos los sitios un nuevo espécimen es agobiante. Un esquema más flexible es que los registros de

control se agreguen automáticamente conforme se vayan necesitando. Por esto un esquema que ofrezca el control de la estructura de cada sitio y además permita la adición de nuevos especímenes constaría de la identificación del sitio, los tipos de bóvedas con que se cuenta, los estados físicos que puede almacenar y el tipo de efectivo para la cuál está destinada, de esta forma se fijarían las reglas del negocio en materia de almacenamiento de una forma dinámica. Por ejemplo si se desea agregar un nuevo registro en el controlador de existencias se tendría que validar que el sitio, bóveda y estado físico existan en la estructura y que además el tipo de efectivo del espécimen que se va agregar sea permitido en esta combinación de datos. Con esto se evita tener que agregar cada nuevo espécimen en el controlador de existencias en todas las combinaciones posibles.

3.6 Propuestas de Solución

En esta parte se describirán las diferentes propuestas de solución que nos permitirán resolver la problemática.

Considerando el análisis de los sistemas de flujo de caja, requerimos de una plataforma que permita almacenar datos en forma centralizada y que ofrezca además seguridad, confiabilidad, consistencia y continuidad. Para determinar nuestras alternativas plantearemos las posibilidades tanto en repositorio de datos como en plataforma.

Repositorio de datos

Encontramos como alternativas la utilización de un esquema de archivos o un manejador de bases de datos. Sin embargo, el utilizar un esquema de archivos tiene las siguientes desventajas:

- No se comparten recursos.
- Control de datos pobre.
- Redundancia en los datos.
- Capacidad de manipulación de datos pobre.
- Incapacidad de generar información de manera rápida.

Mientras que un manejador de bases de datos intrínsecamente está capacitado para:

- Realizar un manejo eficiente de grandes volúmenes de información.

- Compartir la información.
- Manejar transacciones.
- Restringir el acceso a la información a nivel usuario.
- Mantener la integridad referencial⁷.

En cuanto a los Manejadores de Bases de Datos, el más utilizado es el modelo relacional por tener las siguientes características:

- Se fundamenta en un base teórica (teoría de conjuntos).
- Independencia de datos. La estructura física de los datos es transparente para el usuario. En contraste con el modelo jerárquico, en el cual el usuario necesariamente debe conocerla.
- Reglas de integridad. Permite definir para cada caso la forma como se controlará la integridad referencial.
- Flexibilidad. Las relaciones entre entidades son parte de los datos y no forma parte de la estructura de los datos. Esta característica permite la creación o alteración de relaciones entre entidades de una forma flexible e independiente a la implantación física.

Otro modelo existente es el jerárquico, este modelo está basado en una estructura de árbol jerárquico, este modelo presenta problemas cuando los elementos no encajan, en forma natural, en una estructura de este tipo. Además, presenta inconvenientes para representar relaciones "muchos a muchos".

En resumen, el modelo que mejor se adapta para representar los flujos de efectivo, es el modelo relacional que esté basado en el estándar. En la Tabla 3-1 se muestra un comparativo de los principales RDBMS que hay en el mercado.

	ORACLE	SYBASE	DB2
ADMINISTRACIÓN			
Base de Datos multiusuario	SI	SI	SI
Compartiendo Datos ocultos	SI	SI	SI
Compartiendo SQL dinámico	SI	NO	NO

continúa ...

⁷ Entenderemos como integridad referencial al conjunto de reglas que permiten o restringen la actualización de datos en las tablas, con la finalidad de que todos los datos relacionados tengan consistencia. Por ejemplo, al escribir un registro, todos los elementos catalogados deben existir en el catálogo correspondiente y si se modifica un elemento de un catálogo deben modificarse en todos los lugares donde esté referenciado.

	ORACLE	SYBASE	DB2
DICCIONARIO DE DATOS			
Diccionario de Datos activo	SI	SI	SI
Referencia a Objetos Remotos	SI	SI	NO
Sinónimos	SI	SI	NO
TIPO DE DATOS			
Tipo de Datos "Arreglo"	SI	NO	NO
Carácter, longitud fija (CHAR)	SI	SI	SI
Carácter, longitud variable (VARCHAR)	SI	SI	SI
Fecha y/o tiempo	SI	SI	SI
Decimales	SI	SI	SI
Flotantes	SI	SI	SI
Enteros	SI	SI	SI
Definido por el usuario	NO	SI	SI
REPLICACIÓN			
Cuenta con algún producto para replicar información	SI	SI	SI
INDEXACIÓN			
Arbol Binario	SI	SI	SI
Indices <i>clustered</i> , tablas múltiples	SI	NO	SI
Indices <i>clustered</i> , tablas simples	NO	SI	SI
LIMITES			
Núm. Máximo de columnas por tabla	255	250	255
Núm. Máximo de columnas por índice	16	16	16
Núm. Máximo de índices por tabla	Sin límite	251	32767
Definición del tamaño de página	508 k	1962 bytes	4005 bytes
Longitud máxima de variables tipo carácter	2000	255	32700
Núm. máximo de tablas en un unión de consultas	Sin límite	16	Sin límite
BLOQUEO			
Bloqueo de índices en la entrada/llaves	SI	NO	NO
Bloqueo a nivel de renglón	SI	NO	SI
Bloqueo a nivel página	O	SI	N/D
Resolución de bloqueo " <i>dead lock</i> "	SI	SI	SI

continúa ...

	ORACLE	SYBASE	DB2
INTERCONEXIÓN EN REDES			
Soporte Cliente – Servidor	SI	SI	SI
Nombre del Servidor	SI	SI	NO
Interface propietaria	OCI	CT-Lib	-
OPTIMIZACIÓN DE CONSULTAS			
Optimiza costo – base	SI	SI	SI
Acceso solamente a índices	SI	SI	NO
Valor máximo/mínimo almacenado	SI	SI	SI
IMPLEMENTACIÓN SQL			
Asignación de valores en tablas a variables	SI	SI	NO
Cursor	SI	SI	SI
Búsqueda múltiple de renglones vía cursor	SI	SI	NO
Nulos	SI	SI	SI
<i>Outer Join</i> 1 lado (ver glosario)	SI	SI	NO
SEGURIDAD			
Asignación de privilegios para actualizar y consultar columnas	SI	SI	SI
Restricción de acceso a una DB (Data Base, Base de Datos)	SI	SI	SI
Expiración de contraseña	NO	SI	NO
Encriptación de contraseña dentro de la DB	SI	SI	NO
Encriptación de contraseña en la red	SI	SI	NO
PROCEDIMIENTOS ALMACENADOS			
Llamar a otros procedimientos	SI	SI	SI
Ejecutar privilegios con procedimientos almacenados	SI	SI	NO
Datos remotos pueden ser referenciados	SI	SI, Omnisql	SI
Llamadas a procedimientos remotos	SI	SI	NO
CONTROL TRANSACCIONAL			
<i>Rollback</i> (ver glosario)	SI	SI	SI
<i>Savepoint</i> (ver glosario) en subtransacciones	SI	SI	NO

continúa ...

CAPÍTULO 3. Análisis

	ORACLE	SYBASE	DB2
TRIGGER			
Acceso nuevo a valores viejos dentro del <i>trigger</i>	SI	SI	NO
Acceso remoto de datos desde el <i>trigger</i>	SI	SI	NO
<i>Triggers</i> en cascada	SI, 32 niveles	SI, 16 niveles	NO
El <i>trigger</i> puede llamar a SP	SI	SI	NO
PUERTOS DE SERVIDOR			
Microsoft Windows NT	SI	SI	NO
UNIX	SI	SI	NO

Tabla 3-6. Comparativo entre los principales RDBMS

Siendo ORACLE 7.3 y SYBASE (*Adaptive Server Enterprise* 11.5) los productos más destacados en el mercado y los que se adaptan mejor para soportar el modelo de los flujos de caja, por proveer más seguridad y facilidades para el manejo de transacciones, veamos el comparativo mostrado en la Tabla 3-7.

ORACLE	SYBASE
Recomienda el manejo de las transacciones dentro del código del cliente.	Recomienda el manejo de transacciones dentro de los procedimientos almacenados.
No permite anidar transacciones	Permite anidar transacciones.
Los procedimientos almacenados no regresan un código de ejecución (<i>RETURN STATUS</i>)	Los procedimientos almacenados regresan código de ejecución
No permite realizar actualizaciones (<i>update</i>) a través de <i>joins</i> , en su lugar se utilizan subconsultas.	La actualización a través de <i>joins</i> es permitida.

Tabla 3-7. Comparativo entre ORACLE y SYBASE.

ORACLE es un producto maduro con múltiples facilidades para desarrollar aplicaciones basadas en el modelo relacional, por sus características ha sido el preferido de varios autores y empresas.

SYBASE es un producto orientado al proceso transaccional, siendo éste uno de los aspectos más reconocidos, principalmente en sistemas OLTP (*On Line Transaction Processing*, Procesamiento transaccional en línea) y DSS (*Decision support system*, Sistemas de Toma de Decisiones).

Basándonos en la Tabla 3-6, en el comparativo entre ORACLE y SYBASE de la Tabla 3-7 y considerando que este sistema es altamente transaccional con una parte importante para toma de decisiones optamos por recomendar SYBASE - *Adaptive Server Enterprise* 11.5 como repositorio de datos. Esta decisión está apoyada en la idea de reducir el tiempo de contención en las tablas para que la inserción de operaciones sea eficiente, podemos considerar que seguramente habrá varios usuarios concurrentes y el tiempo de actualización puede ser un factor importante, sobre todo si consideramos que existe un horario.

El hecho que el cliente abra una transacción, después ejecute el código y al terminar, dependiendo del resultado, acepte o rechace la transacción contiene un cierto riesgo ya que, si el cliente pierde temporalmente la comunicación o simplemente el tráfico en la red ocasione que se trasmita más lento de lo normal ocasionaría que el tiempo de contención fuera elevado. ORACLE, como vimos en la tabla anterior, no recomienda la inclusión de las transacciones dentro de los procedimientos, ante esto podemos prever el riesgo de que la transacción dure más tiempo del deseable.

Por otro lado, la idea de agrupar todas las sentencias SQL en un procedimiento almacenado facilita la administración de la base de datos y el hecho de que regrese un código de error facilita el control de la ejecución.

Desde el punto de vista de desarrollo, el permitir actualizaciones (*update*) en *joins* facilita la lectura del código cuando sea necesario darle mantenimiento.

Plataforma

Se requiere de la plataforma cliente - servidor con soporte multiusuarios, además que el servidor cuente con una administración de usuarios sólida con restricción de acceso a los datos y soporte a SYBASE - *Adaptive Server Enterprise* 11.5. En este sentido, en la Tabla 3-8 se muestra un comparativo entre UNIX y WINDOWS NT que son los sistemas operativos que soportan al RDBMS seleccionado.

	UNIX	NT
Confiabilidad	Muy bueno	Regular
Seguridad	Bueno	Mala
Escalabilidad	Muy bueno	Muy mala
Disponibilidad	Bueno	Muy mala
Manejo	Muy bueno	Mala
Interperabilidad	Bueno	Bueno
Adaptabilidad	Regular	Regular
Precio	Muy bueno	Muy bueno

CAPÍTULO 3. Análisis

Facilidad de uso	Muy bueno	Bueno
	UNIX	NT
Confiabilidad	Sistema de recuperación de archivos robusto y de bloqueo de archivos.	Cuenta con un subsistema de archivos que "casi" nunca presenta corrupción de datos, realiza rutinas de revisión de la integridad de datos al momento de iniciar el sistema, deteniéndose automáticamente si detecta algún error. Presenta fallas de protección general originadas por la pérdidas ocasionales de las ligas con la memoria.

continúa ...

	UNIX	NT
Seguridad	Se ha constatado que es seguro en varios ambientes, proveyendo múltiples opciones de seguridad.	Razonablemente seguro, sólo por debajo de las versiones comerciales de UNIX. Esta seguridad se presenta sólo bajo un ambiente de Microsoft, e incluso ahí se han detectados algunos puntos de falla.
Escalabilidad	Soporte de múltiples procesadores con una arquitectura de 64 bits del CPU. Soporte de archivos extensos (terabyte).	Opera en equipos de 32 bits.
Disponibilidad	Alto nivel de tiempo de funcionamiento.	Cuando falla la memoria y de protección es necesario reinicializar el equipo.
Manejo	Fácil reparación de fallas remotamente sin necesidad de alterar los procesos alternos.	Fácil de manejar, para instalar nuevas herramientas, actualizaciones.
Interoperabilidad	Opera en un amplio rango de protocolos de redes y ODBC (<i>Open Database Connectivity</i> , Conectividad abierta con base de datos)	Soporta un amplio rango de dispositivos y tarjetas en ambiente Windows. Es débil para soportar ODBC.

Adaptabilidad	Tratamiento fuerte a las bases de datos.	Soporta amplio rango de clientes remotos, aunque en la mayoría de las aplicaciones no proveen de un interfaz textual tradicional.
Precio	Es precio no se incrementa por el número de usuarios, el precio varía entre los \$100 y \$200 USD.	Es a menudo más caro, mientras mas extensa sea la cantidad de usuarios (\$800 USD licencia para 5 usuarios, \$40 USD por cada usuario adicional)
Facilidad de uso	Provee de procedimientos de instalación y sistemas de administración GUI (<i>Graphical User Interface</i> , Interface Gráfica de Usuario) a demás de aceptar comandos en línea.	Es fácil la instalación y uso, provee una interface GUI, pero no es administrable remotamente.

Tabla 3-8. Comparativo entre sistema operativo UNIX y Windows NT.

3.6.1 Propuestas y factibilidad

A continuación procederemos a analizar las alternativas de solución.

Propuesta 1

Utilizar SYBASE - *Adaptive Server Enterprise* 11.5 sobre una plataforma Windows NT.

Estudio de factibilidad técnica

Los equipos NT son rápidos de instalar porque son más pequeños que los UNIX, permite la instalación de aplicaciones en forma estándar, es decir, en forma local o en forma de red. Acepta múltiples protocolos de red y son fáciles de administrar a través de su interfaz gráfica, que consume recursos tanto de disco como de memoria.

Sin embargo, también presenta los siguientes inconvenientes. Después de realizar cualquier cambio en la configuración es necesario reiniciar el equipo, suspendiendo temporalmente el servicio. En cuanto los protocolos de red, a pesar de soportar una amplia variedad carece de uno nativo.

Windows NT, a pesar de lo que dice Microsoft, que es el único proveedor, aún presenta inestabilidad en su funcionamiento, suspendiendo todos los servicios montados sobre él en caso de una falla en algún módulo aislado. En varias ocasiones la única solución es reiniciar el equipo. Por otra parte, Windows NT es susceptible a los virus informáticos.

Windows NT solamente funciona sobre plataformas *Intel* o *Alpha*.

Estudio de factibilidad económica

Basados en los resultados mostrados en la Tabla 3-8 podemos notar que la licencia de Microsoft Windows NT Server 4.X tiene un costo del orden de \$800 USD más \$40 USD por cada usuario.

Estudio de factibilidad legal

No existen ningún obstáculo de esta índole que impida a un Banco Central desarrollar una aplicación para controlar sus flujos de caja.

Propuesta 2

Utilizar un SYBASE *Adaptive Server Enterprise* sobre una plataforma UNIX

Estudio de factibilidad técnica

El sistema operativo UNIX es multiusuarios, escalable y realiza un manejo más eficiente de los recursos. Una característica importante es la administración, a pesar de no contar con una interfaz gráfica, permite cambios en la configuración sin tener que apagar el equipo, además permite administrar el equipo en forma remota.

UNIX ha demostrado ser un sistema operativo robusto que ofrece estabilidad. Varias aplicaciones críticas pueden ser ejecutadas simultáneamente maximizando el uso de los recursos del servidor. Permite al administrador asignar prioridades para mejorar el rendimiento de las aplicaciones con mayor importancia. En caso que una aplicación presente problemas, es factible suspender su ejecución sin reiniciar el equipo.

Los estándares UNIX son dictados por grupos ajenos a los proveedores.

Estudio de factibilidad económica

El sistema operativo UNIX en general es barato como lo muestra la Tabla 3-8, sin embargo la potencia de UNIX depende directamente del hardware con que se cuente. Esto nos lleva a la adquisición de un equipo con los suficientes recursos para atender al número de usuarios que vayan a utilizar la aplicación.

Estudio de factibilidad legal

No existen ningún obstáculo de esta índole que impida a un Banco Central desarrollar una aplicación para controlar sus flujos de caja.

3.6.2 Selección de alternativa

Basándonos en las características mencionadas en el punto anterior consideramos que la opción que mejor rendimiento ofrece es la propuesta de utilizar un SYBASE *Adaptive Server Enterprise* 11.5 sobre una plataforma UNIX.

La razón principal para esta elección está fundamentada en la seguridad de la información y continuidad del servicio, considerando la relevancia de la información manejada. A pesar de resultar más costosa la elección, en el contexto financiero es preferible mantener la continuidad de los servicios aun cuando la inversión sea mayor.

Dado que UNIX es escalable, permite adaptarse a las necesidades particulares de cada Banco Central, la elección del equipo es responsabilidad de ellos y dependerá del volumen de operaciones, el número de usuarios y la cantidad de aplicaciones que quiera ejecutar en este mismo equipo.

Esta propuesta de solución solamente abarca la parte del servidor, que es dónde se almacenarán las operaciones y se controlarán las existencias. El desarrollo del cliente dependerá de cada Banco Central ya que las políticas de desarrollo y plataformas que tengan establecidas varían en cada caso.

3.6.3 Plan de trabajo

El sistema será desarrollado por 5 analistas, que también se encargarán de codificar.

El plan de actividades es el siguiente:

1. Análisis de requerimientos.

- 1.1. Estructura
- 1.2. Operaciones
- 1.3. Existencias
- 1.4. Validaciones
- 1.5. Revisión General
2. Diseño del modelo Entidad – Relación.
 - 2.1. Catálogos
 - 2.2. Existencias
 - 2.3. Operaciones
 - 2.4. Validaciones
 - 2.5. Revisión General
3. Codificación.
 - 3.1. Definición y creación de la base de datos
 - 3.2. Creación de tablas
 - 3.3. *Stored procedures* de uso general
 - 3.4. *Triggers*
 - 3.5. *Stored procedures* para servicios
4. Pruebas.
5. Liberación
 - 5.1. Esquema de contingencia
 - 5.2. Actualización de catálogos
 - 5.3. Creación de cuentas
 - 5.4. Esquema de seguridad (BD)
 - 5.5. Creación de grupos de usuarios
 - 5.6. Asignación de permisos
 - 5.7. Capacitación
 - 5.8. Verificación de Información
6. Mantenimiento

Análisis de requerimiento. Se analizarán los requerimientos para obtener las partes funcionales y los elementos de control que garanticen un producto confiable.

Diseño del modelo Entidad – Relación. En base al análisis se diseñará el modelo Entidad – Relación que representa los flujos de caja y los distintos tipos de existencias sobre los cuales se tiene un estricto control.

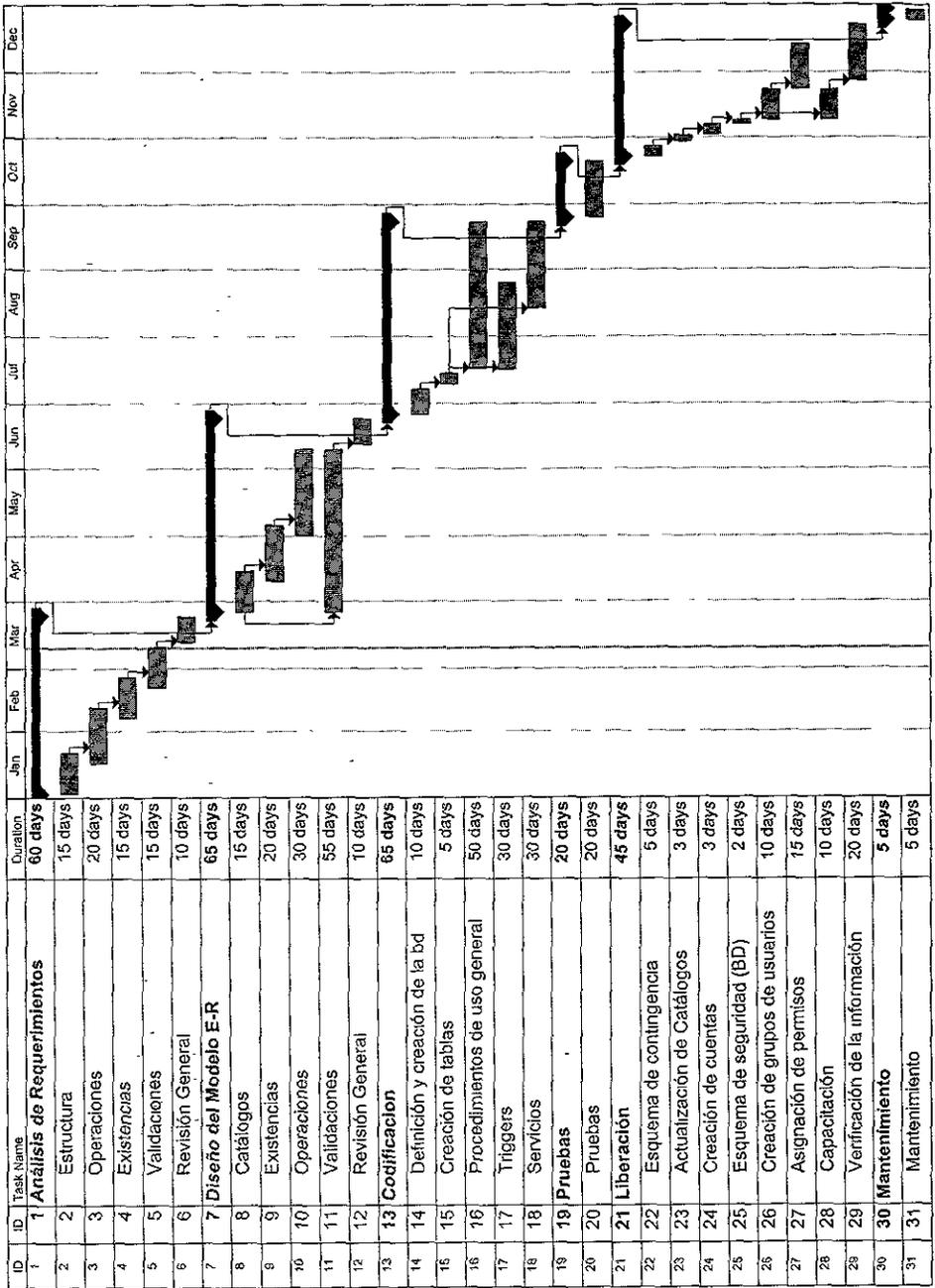
Codificación. Se crearán las tablas resultantes de la generación del modelo Entidad – Relación, así como las reglas de integridad y demás objetos necesarios para operar el sistema.

Pruebas. Se realizarán las distintas pruebas que garanticen que el sistema resultante *satisface con todos los requerimientos mencionados.*

Liberación y capacitación. Una vez probado el sistema y garantizando que cumple con los requerimientos establecidos se procederá a cargar los catálogos y comenzar su operación. Al mismo tiempo, se capacitará a los administradores y a los usuarios en cómo interactuar con el sistema.

En la Tabla 3 - 9 se muestra el diagrama de Gantt con la planeación de las actividades.

Una vez analizados los requerimientos y resuelto la plataforma de desarrollo, comenzaremos con el diseño del modelo Entidad – Relación respectivo. En el siguiente capítulo se detallará la obtención de dicho diagrama.



4 Diseño

En este capítulo se mencionan las principales características de un RDBMS (*Relational Database Management*, Sistema Manejador de Bases de Datos Relacional), se describe el modelo Entidad – Relación de los sistemas de flujos de caja, y se incluye la normalización y ajustes necesarios al modelo con el fin de mejorar el rendimiento de la aplicación.

4.1 Principales componentes de un RDBMS

Algunas de los elementos comunes a los RDBMS incluyen los *defaults* (valores por omisión), *check constraints* (verificación de restricción), *rules* (reglas), *stored procedures* (procedimientos almacenados), *triggers* (disparadores) e índices.

Defaults y check constraints

Los *defaults* y *check constraints* sirven para mantener la integridad de dominio. La integridad de dominio garantiza que los valores de una o varias columnas se encuentren dentro de un rango de valores establecido. Un *default* se aplica cuando se realiza una inserción sobre la tabla y no se proporcionan los valores de algunas columnas, en este caso se toma un valor definido como *default*. Existen varias características que debe cumplir un *default*:

- Un *default* se aplica a nivel columna.
- Una columna puede tener sólo un *default* asociado.
- El valor especificado en el *default* debe ser del mismo tipo ó convertible al tipo definido para la columna.

Por lo que se refiere a los *check constraint*, estos se utilizan durante una actualización o una inserción sobre la tabla y proporcionan una lista de valores, un rango o una condición que deben cumplir los datos a insertar o actualizar, de no cumplirse tales condiciones se rechaza la operación. Los *check constraints* pueden ser aplicados tanto a nivel tabla como a nivel columna.

Reglas

Una regla es una definición dentro de la base de datos que establece alguna restricción sobre una columna. Al igual que los *check constraints* sirven para implementar la integridad de dominio, sin embargo, se puede aplicar tanto a las columnas como a los tipos de datos definidos por el usuario.

Procedimientos almacenados

Un procedimiento almacenado es un conjunto de sentencias en SQL agrupadas y referenciadas bajo un nombre. Los procedimientos almacenados son guardados dentro de la Base de Datos y presentan las siguientes ventajas:

- Permiten que muchos usuarios tengan acceso al mismo código.
- Permiten una construcción centralizada y consistente de la lógica de integridad.
- Proporcionan seguridad a la Base de Datos bajo un esquema de permisos de ejecución a usuarios, no siendo necesario por tanto el proporcionar acceso directo a las tablas.
- Permiten modificarlos en forma centralizada sin tener que notificar al cliente.
- Facilitan que las transacciones sean atómicas al permitir abrir y cerrar una transacción dentro del mismo código, reduciendo el tiempo de contención en las tablas involucradas.
- El conjunto de sentencias se ejecutan más rápido que si fueran procesadas una a una de forma interactiva por un usuario.
- Reducen el tráfico en la red ya que sólo se envía la orden de ejecución.

Pueden contener código para:

- Validar que la operación se haya completado de manera satisfactoria asegurando la consistencia de la información.
- Proporcionar mensajes de error adecuados si es que los datos no son válidos.
- Realizar revisiones de integridad.
- Insertar información en la tabla apropiada.
- Ejecutar consultas definidas que se realizan con frecuencia.
- Implementar las reglas del negocio.

- Crear rutinas comunes a otros procedimientos almacenados, por ejemplo las rutinas de manejo de errores.

Una biblioteca de procedimientos almacenados proporciona la capacidad de reutilizar el código y promueve el desarrollo de forma modular con funciones bien identificadas y delimitadas.

Triggers

Un *trigger* es una variante de los procedimientos almacenados, se ejecutan de forma automática cuando se realiza sobre una tabla una operación de actualización, borrado o inserción. Los *triggers* tienen los siguientes propósitos:

- Realizar actualizaciones en cascada cuando en una tabla se actualiza o borra una llave primaria.
- Mantener datos duplicados. Frecuentemente, por razones de desempeño, se opta por tener datos duplicados en la Base de Datos; esta práctica hace necesaria la utilización de un *trigger* que actualice de manera automática las copias cuando el original cambie asegurando la consistencia de la información.
- Calcular automáticamente el contenido de columnas derivadas.
- Permitir la construcción de restricciones complejas.
- Registrar operaciones excepcionales durante el funcionamiento normal del sistema.

Índices

Un índice es una estructura de almacenamiento, separada de la base de datos, cuyo propósito es el de permitir el acceso directo a un renglón específico de la tabla. Los índices contienen apuntadores lógicos que agilizan la recuperación de los datos almacenados. Los índices se utilizan para mejorar el desempeño al realizar una búsqueda de renglones, relacionar datos de distintas tablas y ordenar los resultados de alguna consulta.

Existen dos tipos de índices, el índice *clustered* (agrupado) y el *non – clustered* (no agrupado). Los índices tipo *clustered* se caracterizan por ordenar físicamente el contenido de la tabla, mientras el índice *non – clustered* almacena en las tablas de índices el apuntador al renglón correspondiente. En las tablas de índices, el propio manejador almacena las columnas que conforman el índice y

un apuntador a la página de datos donde se encuentra almacenado el renglón completo.

Para mejorar el tiempo de respuesta, varios manejadores cuentan con una estructura *B – Tree* (*binary tree*, árbol binario) que les permite localizar un valor del índice en forma eficiente y por tanto, la página de datos que requiere.

Dado que los índices *clustered* ordenan físicamente los datos, permiten resolver con mayor velocidad las consultas, ya que solamente almacena en la tabla de índices el valor de las columnas índice que están en el inicio de la página de datos, esto reduce la cantidad de renglones que tiene que revisar. Sin embargo, su existencia consume mayor tiempo al momento de la escritura ya que tiene que reordenar las páginas de los datos y las páginas de índices.

Los índices *non – clustered* mantienen un apuntador por cada renglón, son útiles para localizar un conjunto reducido de datos, regularmente las consultas que utilizan este tipo de índices son más lentas que aquellas que utilizan un índice *clustered*, sin embargo permiten una escritura con mayor rapidez ya que escriben al final de la tabla y el único ordenamiento es en páginas de índices.

Para determinar las columnas que conforman cada índice y el tipo de índice, se deben considerar las consultas críticas, no necesariamente el índice *clustered* estará formado por la llave primaria, aunque en varios casos así es. Por las características de los índices *clustered* solamente se puede crear uno por tabla, sin embargo pueden crearse los índices *non clustered* que sean necesarios.

4.2 Diseño Entidad – Relación

Para realizar el diseño, nos basamos en lo descrito en los capítulos 2 y 3 de este documento. Antes de comenzar con el desarrollo de esta parte del diseño, recordemos que es una entidad, una relación, una llave primaria y una llave foránea, así como los principios de integridad referencial.

Una entidad representa un objeto (físico o no) en el mundo real, tiene atributos propios y puede ser identificado en forma única. Una relación establece los lazos que existen entre dos entidades o incluso entre la misma entidad, veamos el siguiente ejemplo, si quisiéramos modelar las personas y los centros de enseñanza, encontramos que una persona puede asistir a uno o varios centros de enseñanza, podemos observar dos entidades, personas y centros de enseñanza, y una relación entre ambas que es "asisten".

La entidad "persona" tiene atributos propios como el nombre, apellido paterno, apellido materno, edad, peso, estatura, etc., la entidad "centro de enseñanza" tiene sus atributos como nombre, tipo de educación, número de aulas, etc., en ambos casos requerimos de un conjunto de atributos que hagan único cada

elemento de la entidad. En el caso de las personas tal vez pudiéramos utilizar el nombre y sus apellidos, si esto no resultará suficiente podemos agregar atributos hasta conseguirlo, si a pesar de todo no lo conseguimos o determinamos que para hacer único cada elemento requerimos de una gran cantidad de atributos podemos crear un elemento artificial, que podría ser un número consecutivo, al que llamaremos "clave persona", y utilizarlo para diferenciar a cada elemento. A este conjunto de elementos se le conoce como llave primaria (PK, *primary key*).

Si una persona solamente pudiera asistir a un centro de enseñanza a la vez la relación sería simple, porque un centro de enseñanza podría tener uno o muchos alumnos pero un alumno solo podría asistir a un centro de enseñanza. Sin embargo esto no representa la realidad, en mundo real una persona puede asistir a uno o varios centros de enseñanza simultáneamente, pensemos en un estudiante de secundaria que también está inscrito en una escuela de inglés; así, una persona puede asistir a uno o varios centros de enseñanza y un centro de enseñanza puede tener uno o varios alumnos, este tipo de relaciones se identifican como "muchos a muchos". Estas relaciones pueden ser representadas en el modelo relacional, sin embargo, se resuelven al momento de normalizar con la creación de una entidad asociativa, siguiendo con nuestro ejemplo tendría una entidad "PERSONA – CENTRO DE ENSEÑANZA" que indicaría para cada persona en que centro estudia, existiendo un elemento por cada combinación existente.

Las entidades asociativas tienen como llave primaria a la llave primaria de cada una de las entidades que están involucradas en la relación, en nuestro ejemplo sería la clave de persona y la clave del centro. Cuando una llave primaria de una entidad forma parte de la llave primaria o de los atributos de otra entidad se le llama llave foránea (FK, *foreign key*), en nuestro ejemplo, clave de persona sería llave primaria y llave foránea en la entidad asociativa.

Normalmente las llaves foráneas nos ayudan a cuidar la integridad referencial. La integridad referencial está formada por un conjunto de reglas que impiden que un valor de un atributo no pueda ser entendido, si en nuestro ejemplo eliminamos todos los elementos de la entidad "centro de enseñanza" sin eliminar los elementos en la entidad asociativa, no podríamos saber en que centro están estudiando las personas. Para evitar esta situación se definen las reglas de integridad según sea el caso. Existen tres reglas de integridad que son:

- Cascada
- Restrictiva
- Ninguna

Las reglas en cascada permiten reflejar los cambios en todas las entidades que estén referenciadas a la entidad que cambia su llave primaria, por ejemplo si a un centro de enseñanza se le cambia el valor de la llave primaria de M a S, todos los elementos en la entidad asociativa que tengan valor M en clave de

centro tendrán ahora valor S, si en lugar de modificar se elimina, se podrían eliminar todos los elementos que tuvieran el valor del centro eliminado o bien asignarles un valor nulo si no forma parte de la llave primaria.

Las reglas restringidas permiten impedir la modificación o eliminación de alguna llave primaria si está siendo referenciada en otra entidad, también impide insertar valores en una llave foránea que no existan en la entidad de donde proviene.

En el caso de "ninguna", no podemos decir que sea una regla ya que más bien es la no - importancia de la integridad.

La nomenclatura utilizada para el modelo de la base de datos está ilustrada en la Figura 4-1.

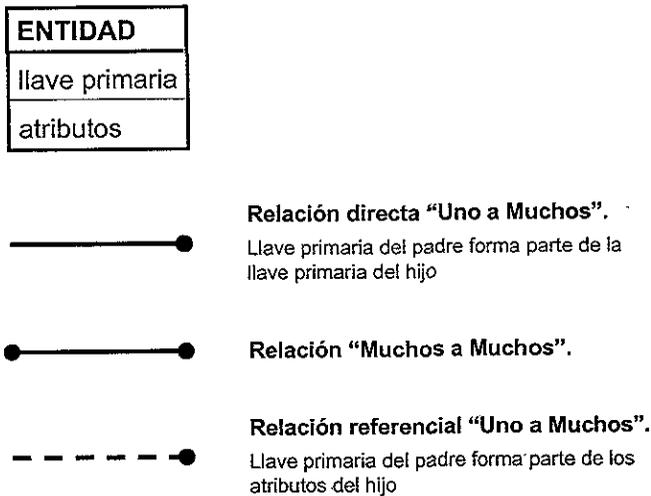


Figura 4-1. Nomenclatura utilizada en el diseño del modelo relacional.

Una vez definida la nomenclatura y recordado los principios básicos, comencemos con el diseño. Para explicar más claramente el diseño del modelo Entidad – Relación, dividiremos la problemática en varios frentes hasta conjuntar el modelo definitivo. Los puntos son:

- **Existencias** Se diseñarán las entidades en las cuáles se controlarán los diferentes tipos de existencias que han sido descritas.

- **Operaciones** Se diseñará el esquema con el cuál se almacenarán y procesarán las distintas operaciones.
- **Catálogos** Se detectarán los catálogos necesarios y se definirán sus atributos.
- **Permisos** Se planteará el esquema a utilizar y se definirán las entidades correspondientes.
- **Validaciones** Se diseñarán los esquemas necesarios para realizar las validaciones dictadas por las reglas del negocio.
- **Control y configuración** Se planteará el esquema de control del sistema en general.

Existencias

El ciclo de vida del efectivo, desde el punto de vista de los sistemas de flujos de caja, nos muestra cinco "mundos" donde las piezas pueden estar, que son: en las piezas no materializadas, las existencias estacionadas, las existencias en tránsito, las piezas en circulación y entre las piezas destruidas, con excepción de las piezas en circulación, estos son los "mundos" que a los Bancos Centrales les interesa controlar.

Las piezas no materializadas son aquellas que están en alguna de las fases de la producción. Este tipo de existencias se caracteriza por carecer de un cuerpo tangible, sin embargo tienen una denominación, valor real y características de fabricación, además de un estado de proceso. Por tanto la entidad que representa este tipo de existencias contiene el número de piezas que se tienen en cada estado de proceso por cada denominación y se llama EXISTENCIA PRODUCCIÓN, por la parte del ciclo que representa.

Las piezas físicas propiedad del Banco Central son las existencias, y están distribuidas en los diferentes sitios. Dentro de cada sitio las piezas se pueden encontrar en una bóveda y conservar un estado físico. A los Bancos Centrales les interesa conocer sus existencias por denominación y valor real, además del material que las compone cuando se tiene conocimiento de él, que prácticamente es cuando las piezas son nuevas o cuando van a ser destruidas. La entidad que las representa es EXISTENCIA y se controla cuántas piezas de cada denominación hay en cada estado físico, en cada bóveda y dentro de cada sitio.

Cuando las piezas están en tránsito no se tiene una ubicación fija de ellas, sin embargo si se conoce el tipo de transporte en el cuál se están moviendo. Aun cuando las piezas no estén estáticas, cada una de las piezas conserva su

estado físico así como su denominación y valor real. La entidad que almacena estas piezas es EXISTENCIA TRÁNSITO y contiene el número de piezas para cada espécimen con cierto estado físico que es transportada en un tipo de transporte. Los detalles del origen y destino están contenidos en las operaciones, así que desde el punto de vista del control de existencias basta con conocer el medio de transporte.

También se cuenta con un registro del total de piezas destruidas por denominación y características de fabricación. Una de las características es quién las fabricó, ya que esto ayuda a determinar el circulante de piezas por cada fabricante. Para fines estadísticos es necesario saber cuáles han sido los métodos de destrucción más utilizados y el estado físico que tenían las piezas antes de ser destruidas. La entidad que representa a éstas es PIEZA DESTRUIDA y contiene el número de piezas por cada denominación, estado físico, método de destrucción y fabricante.

En la Figura 4-2 se muestran las entidades resultantes para controlar las existencias, que son: EXISTENCIA PRODUCCIÓN, para las piezas autorizadas y las piezas que fueron ordenadas para su fabricación; EXISTENCIA, para almacenar las piezas estacionadas en un sitio específico; EXISTENCIA TRÁNSITO, para controlar las piezas que están viajando; y PIEZA DESTRUIDA, para controlar las piezas que dejaron de existir.

EXISTENCIA PRODUCCIÓN

estado físico (PK) espécimen (PK)
número piezas

EXISTENCIA

institución (PK) plaza (PK) bóveda (PK) estado físico (PK) espécimen (PK)
número piezas

PIEZA DESTRUIDA

espécimen (PK) fabricante (PK) método destrucción (PK) estado físico (PK)
número piezas

EXISTENCIA TRÁNSITO

transporte (PK) estado físico (PK) espécimen (PK)
número piezas

Figura 4-2. Entidades para controlar las existencias y las piezas destruidas

Operaciones

Todos los movimientos son representados por operaciones, éstas pueden ser agrupadas en grupos de operaciones para facilitar la aplicación de las reglas del negocio, y estos grupos pueden ser englobados en familias de operaciones. Auxiliándonos del diseño del control de existencias y de las características de las operaciones, encontramos una variedad de familias; sin embargo, todas las operaciones tienen información común, como el tipo de operación, la terminal desde dónde se envió, una fecha valor o de aplicación.

Por cuestiones de control es importante conocer el orden conforme fueron ingresando las operaciones, también es importante conocer la fecha en la cuál el servidor las recibió, por esta razón, es importante crear una llave artificial que consiste de la fecha operativa¹ y asignarle un folio consecutivo a cada operación.

Para manejar un esquema que facilite la creación de nuevas familias de operaciones podemos basarnos en un esquema de subentidades, donde la entidad padre contiene las características comunes y dentro de ella hay subdivisiones donde se detallan las particularidades de cada división.

Tomemos como ejemplo los vehículos, si tuviéramos que modelarlos podemos partir del hecho que todos los vehículos tienen características comunes como un número de asientos, el tipo de combustible o energía que utilizan para moverse, el peso del vehículo, etc., y dentro de la entidad podemos clasificarlos según sus particularidades en aviones, camiones de carga, automóviles, etc., donde cada subclase tiene características muy diferentes a las demás, esto lo vemos ilustrado en la Figura 4-3. En este ejemplo podemos notar que un vehículo pertenece a una y sólo a una subclase, lo que nos lleva a decir que a pesar de que todos los elementos de la entidad son vehículos, las subentidades son mutuamente excluyentes.

Al ser excluyentes se tiene que determinar cuáles son los atributos que determinan en qué subentidad debe estar el detalle del vehículo, en este caso el tipo de vehículo lo determina.

Para representar físicamente este tipo de subentidades, la llave primaria de la entidad padre forma la llave primaria de cada una de las entidades hijas. Las entidades hijas pueden tener a su vez entidades que dependan de ellas, o incluso puede repetirse este esquema una vez más.

¹ La fecha operativa puede ser distinta a la fecha natural. La fecha operativa es aquella en la que el servidor recibe la operación y en lo sucesivo la llamaremos fecha registro.

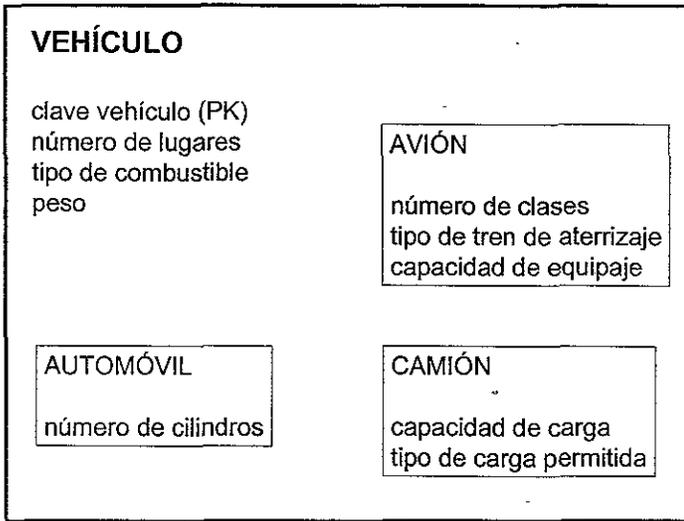


Figura 4-3. Ejemplo de una entidad con subentidades.

Utilizando esta forma de conceptualizar la realidad encontramos que todos los movimientos, sin importar del tipo, pueden expresarse en una entidad OPERACIÓN y el detalle de cada movimiento en la subentidad que le corresponda. Tomando las familias de operaciones descritas en el capítulo 3, tenemos las siguientes subentidades:

- Autorización de emisión
- Orden de fabricación
- Entrega de efectivo (billete o moneda)
- Traspaso
- Remesa
- Depósitos de efectivo
- Retiros de efectivo
- Destrucción

La información que se requiere para cada una de estas familias básicas es:

Autorización de emisión Fecha de emisión, estado lógico autorizado, estado lógico dónde se controle el saldo de lo autorizado, el espécimen y el número de piezas.

Orden de fabricación Número de orden, fecha de la emisión afectada, fecha de la orden, fabricante, estado lógico origen, estado lógico destino, espécimen y número de piezas.

Entrega de efectivo	Número de acta en la cuál se exprese a que ordenes de fabricación corresponde la entrega, fecha de la emisión afectada, fecha de entrega, fabricante, estado lógico origen, sitio, bóveda y estado físico destino, espécimen y número de piezas.
Traspaso	Sitio del movimiento, bóveda y estado físico origen, bóveda y estado físico destino, lista de especímenes involucrados y número de piezas para cada caso.
Remesa	Número de orden de remesa, fecha de embarque, sitio, bóveda y estados físicos origen, sitio, bóveda y estados físicos destino, lista de especímenes involucrados y número de piezas para cada caso.
Depósito de efectivo	Banco usuario, sitio, bóveda y estado físico destino, lista de especímenes involucrados y número de piezas para cada caso.
Retiro de efectivo	Sitio, bóveda y estado físico origen, banco usuario, lista de especímenes involucrados y número de piezas para cada caso.
Destrucción	Fecha de destrucción, método empleado, sitio, bóveda y estados físicos origen, lista de fabricantes, especímenes y el número de piezas para cada caso.

Ahora analicemos cada operación que tipo de existencias afecta.

- Autorización de emisión EXISTENCIA PRODUCCIÓN
- Orden de fabricación EXISTENCIA PRODUCCIÓN
- Entrega de efectivo EXISTENCIA PRODUCCIÓN y EXISTENCIA
- Traspaso EXISTENCIA
- Remesa EXISTENCIA y EXISTENCIA TRÁNSITO
- Depósitos de efectivo EXISTENCIA
- Retiros de efectivo EXISTENCIA
- Destrucción EXISTENCIA y PIEZA DESTRUIDA

Ahora bien, podemos observar que los depósitos y retiros requieren la misma información y afectan el mismo tipo de existencias de formas muy similares, podemos afirmar que la única diferencia es que el depósito aumenta las existencias del sitio del movimiento y en el retiro se disminuyen, por tanto podemos agruparlas bajo una misma entidad llamada DEPÓSITO RETIRO.

En el caso de las remesas sabemos que es una operación que involucra dos movimientos en diferentes puntos del tiempo, el envío y la recepción. En el envío es necesario detallar la información que describe el movimiento, por ejemplo, de dónde sale, a dónde va, cuántas piezas de cada espécimen, etc., y en la recepción basta con definir la fecha en la cuál las piezas ingresan al sitio destino. Entonces podemos definir una familia de operaciones adicional que es la confirmación de recepción de la remesa, esta entidad lleva por nombre CONFIRMACIÓN REMESA. Considerando que mientras las piezas están viajando pudiera ocurrir alguna situación que impida recibir las piezas en la bóveda destino y tuvieran que recibirlas en otra bóveda o bien por cuestiones estratégicas el sitio destino así lo decida, en la confirmación de recepción se indicará en que bóveda se están recibiendo las piezas. Así, los atributos de esta entidad son: la fecha en la que se registro el envío de la remesa y el folio que le fue asignado y la bóveda destino. Los demás datos no tienen porque volverse a capturar dado que el estado físico no cambia en el trayecto ni la cantidad de piezas por cada espécimen. Esta entidad afecta EXISTENCIA TRÁNSITO y EXISTENCIA.

Una vez definido el esquema e identificados los atributos de cada subentidad, retomemos el diseño de la entidad OPERACIÓN. Para facilitar la sincronización con el cliente, es recomendable que él asigne un folio local a cada operación, así, en caso de contingencia el servidor podrá proporcionar una lista de todos los folios de una terminal, el cliente revisará de esta manera que todas operaciones enviadas existan en el servidor, en caso contrario reenviará aquellas que no tenga el servidor. Otros atributos de esta entidad son: la clave del usuario que envió cada operación, el tipo de movimiento², el tipo de efectivo, la fecha de aplicación o fecha valor, la fecha del servidor en la cuál la operación fue procesada y el estado de proceso. Este último atributo nos indica si la operación ya fue procesada, en el caso de los retiros pre – avisados, indica que solamente fue insertada pero aún no ha sido procesada por no haber llegado aún su fecha valor; en el caso de las remesas permite controlar de una forma fácil la relación con la confirmación, dado que el movimiento se compone de dos operaciones, al procesar el envío este atributo indica que está enviada, al confirmar la remesa se marca como procesada. Este esquema facilita también la iniciación de las entidades para el siguiente día operativo, ya que, previo respaldo de la información, se eliminan todas las operaciones procesadas permaneciendo solamente aquellas cuyo estado de proceso no indique que ya fue procesada. Para que este esquema funcione al cancelar una operación se tiene que marcar como procesada.

En resumen, la entidad OPERACIÓN tiene los siguientes atributos: fecha registro, folio del sistema, número de terminal, clave de acceso o *login* del usuario que insertó la operación, tipo de movimiento, tipo de efectivo, tipo de operación, fecha de aplicación, fecha de proceso y estado de proceso. El detalle de las operaciones es almacenado en subentidades en las que contiene la

² Los tipos de movimiento que consideramos son: Altas, Cancelaciones y Modificaciones.

información particular para cada familia de operaciones. En la Figura 4-4 se ilustra el modelo de las operaciones.



Figura 4-4. Entidad OPERACIÓN y subentidades para detallar los movimientos.

Catálogos

En los capítulos 2 y 3 se mencionaron distintos aspectos en la organización que nos muestra la necesidad de catalogar información, y como se pudo observar tanto en el diseño de las existencias como en el de las operaciones, necesitamos entidades que permitan mantener la integridad de la información.

Comencemos por la infraestructura, los bancos centrales tienen una Oficina Central, Sucursales y Corresponsales dentro del territorio nacional, en estos sitios es donde se da atención a los bancos. Un sitio está conformado por la institución y la plaza donde se ubica. Esto nos lleva a la necesidad de un catálogo de instituciones, uno de plazas y uno de sitios. Donde un sitio es la combinación válida de una institución en una plaza.

También se describió que tanto la Oficina Central como cada una de las Sucursales es responsable de su conjunto de Corresponsales, si consideramos a Oficina Central con una sucursal más, podemos obtener un conjunto de sucursales y esto nos lleva a la entidad SUCURSAL, con esta entidad podemos identificar a qué Sucursal pertenece cada sitio.

Dentro de cada sitio sabemos que hay bóvedas o tipos de bóvedas donde se almacena el efectivo. Cada pieza dentro de una bóveda presenta un estado físico. De aquí obtenemos las entidades BÓVEDA y ESTADO FÍSICO. En la entidad EXISTENCIA PRODUCCION encontramos un estado lógico, que en el contexto de este tipo de existencias indican un proceso, sin embargo podemos verlos como un estado que conservan las piezas en ese momento, así que podemos unificarlas con los estados físicos que representan el estado que conservan las piezas e incluso, según el Banco Central, puede representar el proceso en que se encuentra.

El efectivo puede ser billete o moneda y cada pieza tiene un valor facial o denominación y un valor real, además se tienen características de fabricación, principalmente en las monedas en las cuáles se utilizan diferentes aleaciones incluso para monedas de igual denominación. Otro aspecto que hay que considerar es el hecho que hasta cierto punto del ciclo de vida de las piezas se tiene un control de las existencias según sus características de fabricación, después de ese punto se pierde y se manejan indistintamente todas las piezas de igual denominación y valor real, siendo hasta los últimos movimientos antes de la destrucción donde se reclasifican y se vuelve a tener control de las piezas según sus características.

Si aumentamos el dominio de la aplicación y consideramos que dentro de algunos sitios pueden existir billetes o monedas de otros países, que al igual que las emitidas por el Banco Central son sujetas a varios procesos, es conveniente considerar un catálogo de divisas.

El efectivo puede ser catalogado bajo la entidad ESPÉCIMEN, que además es el término usado en este documento. Podrán existir tantos especímenes como sea necesario, asignando una clave única para distinguirlos. Cada espécimen tendrá atribuciones propias como su valor facial o denominación y su valor real, además de estar asociado, cuando sea necesario; a un catálogo de materiales. Es común encontrar que las mismas aleaciones son usadas para familias de monedas, por esto es conveniente considerar las aleaciones como una entidad independiente. Este esquema permite que se puedan diferenciar las piezas según sus características de fabricación o bajo una clave genérica independiente al material utilizado, ya que podrán utilizar la clave del espécimen asociado a un material cuando sea necesario o bien utilizar otra clave que englobe todas las piezas con la misma denominación y tipo de efectivo.

Cada Banco Central tiene sus propios métodos de destrucción, y sin importar cuáles sean, podemos definirlos como una entidad en la cuál estén todos los métodos que utilicen. También tienen sus propios conjuntos de fabricantes, los cuales pueden ser representados como una entidad. En esta misma situación encontramos a los tipos de transporte utilizados en las remesas.

Revisando la descripción de las operaciones encontramos que cada uno de los movimientos puede ser representado como un tipo de operación. Hay varios

tipos de operación que son catalogados con claves distintas con la finalidad de facilitar el establecimiento de las reglas del negocio, sin embargo, desde el punto de vista de los sistemas hacen exactamente lo mismo, la diferencia radica en el sentido de las afectaciones.

Cada tipo de operación pertenece a un grupo de operaciones que a su vez pertenece a una familia de operaciones. El tipo de operación debe indicar por sí mismo el tipo de efectivo para el cuál fue concebido, existiendo la posibilidad de que un tipo de operación sea válido tanto para billete como para moneda; también debe indicar cuál es el conjunto de fechas de aplicación válidas, por ejemplo sólo valor mismo día, sólo valor futuro, valor mismo día y valor futuro, etc.

Como se mencionó anteriormente, existe la posibilidad de que varias operaciones estén en la misma entidad y la diferencia sea la afectación, con la intención de que propio modelo resuelva esta característica y considerando que en la mayoría de los casos se afecta un origen y un destino, es factible que uno de los atributos de esta entidad sea la forma de afectar o tipo de afectación, de esta forma el administrador podrá configurar las operaciones según indiquen las reglas del negocio.

Como existen operaciones con fecha valor futuro, es necesario contar con los días inhábiles, para así poder determinar si la fecha de afectación es correcta.

Dado que el envío de operaciones, las consultas, reportes y en general, cualquier actividad que se realice con el servidor puede ser visto como un servicio, encontramos otra entidad, SERVICIO. Un mismo servicio puede resolver las necesidades de un usuario a diferentes niveles, por ejemplo, la consulta de operaciones, ésta puede ser a nivel usuario, terminal, sitio, sucursal o general, mostrando un conjunto de operaciones distintas según lo determine el tipo de servicio.

Cada combinación "servicio – tipo de servicio" indica por sí misma el horario en el cuál es válida, si es necesario verificar permisos de ejecución o es un servicio general y también si está disponible en el momento que se solicita.

Una combinación de "servicio – tipo de servicio" puede tener un horario distinto al normal si hubiere un imprevisto, sin embargo al día operativo siguiente debe retomarse el horario definido. Esto nos lleva a contar con una entidad que mantenga los horarios para cada servicio – tipo de servicio. Si pensamos que por cada servicio se tiene que consultar SERVICIO y además HORARIO, podemos concebirlo de una forma alterna, en SERVICIO mantenemos el horario del día operativo y en HORARIO mantenemos el horario normal, así al final de cada día operativo se reemplazan los valores concernientes al horario de SERVICIO con los almacenados en HORARIO que para mayor claridad llamaremos HORARIO PERMANENTE.

Siguiendo con los servicios, recordemos que se desea una administración flexible, por tanto es necesario que existan grupos de usuarios a los cuales se les otorgaran permisos sobre los servicios, así que encontramos dos entidades más, USUARIO y GRUPO USUARIO.

Algunas de las entidades las utilizamos como catálogos del sistema, y encontramos que algunas están relacionadas. El conjunto de entidades "catálogo" que existe esta ilustrado en la Figura 4-5.

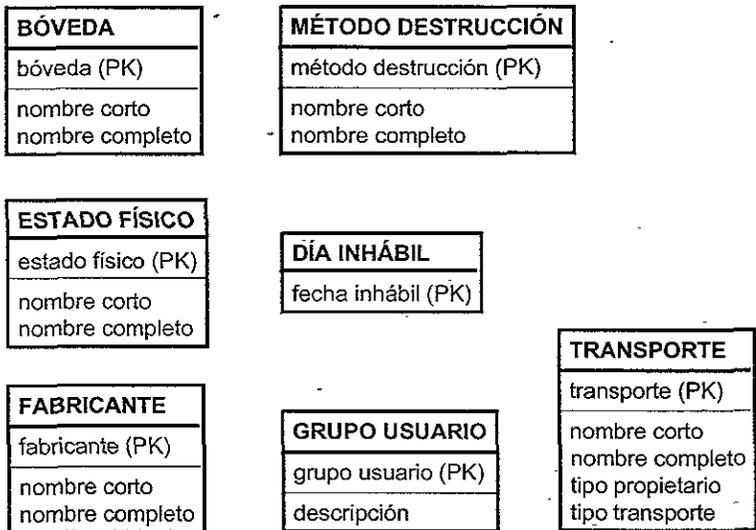


Figura 4-5. Algunas de las entidades que son utilizadas como catálogos.

Describamos ahora las entidades "catálogo" que están relacionadas con otras entidades. Recordemos que para encontrar las relaciones habitualmente se identifica un verbo que las relacione, así, con ayuda de algunos "verbos" definamos la parte faltante de los "catálogos".

En primera instancia encontramos que un espécimen está expresado en una divisa. También un espécimen puede estar asociado a un conjunto de materiales. Esta última frase nos indica una relación muchos a muchos, sin embargo en el mundo real es común encontrar la aleación en lugar de la lista de materiales en forma individual, por lo que podemos conceptualizar el material como un solo elemento, el cuál utilizamos como clave de material. Tomando estas consideraciones, obtenemos el modelo de la entidad ESPÉCIMEN ilustrado en la Figura 4-6.

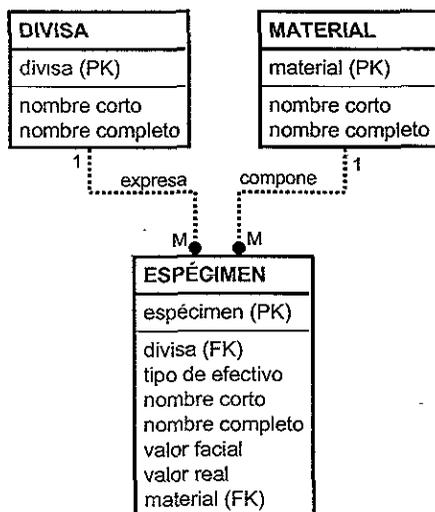


Figura 4-6. Modelo de la entidad ESPÉCIMEN con sus relaciones.

Cambiando al ámbito de los tipos de operación encontramos que una familia de operaciones tiene grupos de operaciones, y cada uno de ellos agrupa a un conjunto de tipos de operación. El modelo de la entidad TIPO OPERACIÓN lo encontramos en la Figura 4-7.

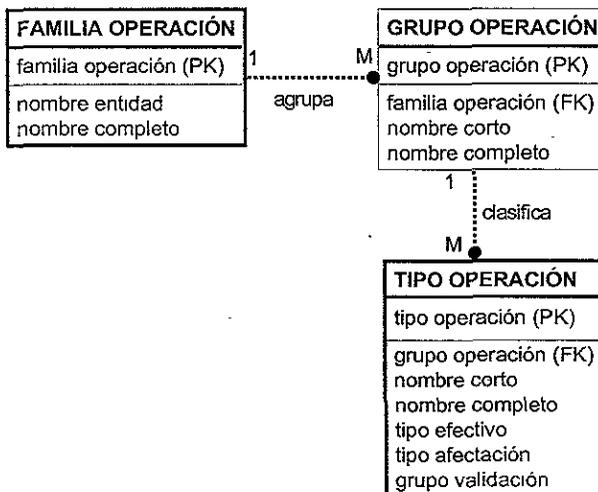


Figura 4-7. Modelo de la entidad TIPO OPERACIÓN y sus relaciones.

En cuanto a estructura, una institución y una plaza forman un sitio y cada sucursal controla a un conjunto de sitios.

Por otra parte, cada sitio tiene asociado un conjunto de terminales desde las cuáles se interactúa con el sistema.

Una entidad importante en la estructura funcional son los usuarios, que son los que llevan a cabo los movimientos, dada la diversidad de instituciones participantes, podemos identificar que un usuario labora para una institución.

El modelo de la estructura está ilustrado en la Figura 4-8, donde se observan las relaciones entre las distintas entidades. Si en un futuro fuera necesario crear un árbol de dependencias se podría agregar una relación circular entre sitio que permitiera que un sitio pudiera tener hijos, aunque en la operación normal esto no sucede.

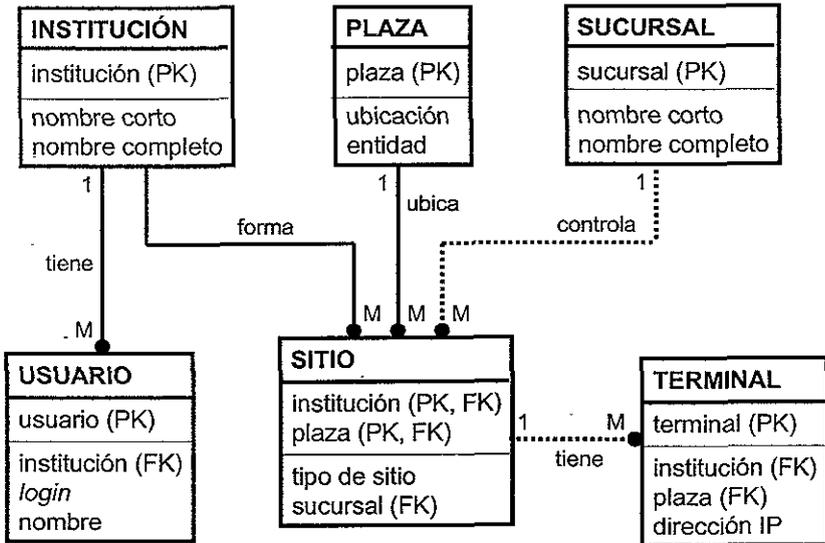


Figura 4-8. Modelo de las entidades que definen la estructura del Banco Central.

Por último encontramos a los servicios, cada servicio tiene un horario normal de operación, como se comentó anteriormente, los horarios definitivos o permanentes son copiados a los horarios del día almacenados en la entidad SERVICIO. El modelo de los servicios y sus horarios permanentes está ilustrado en la Figura 4-9.

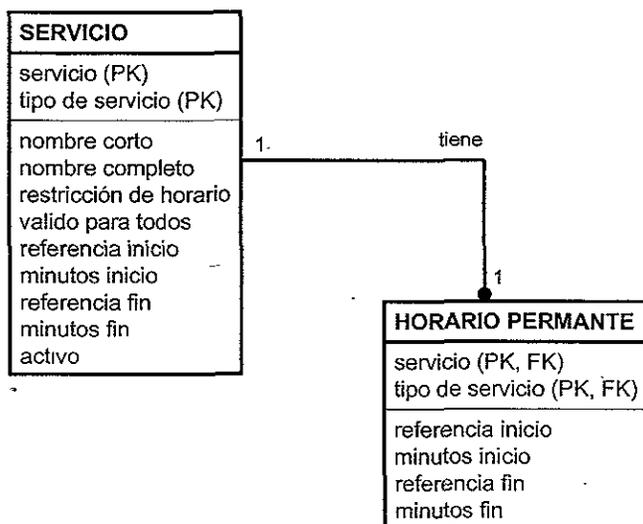


Figura 4-9. Modelo de la entidad SERVICIO y su horario normal.

Permisos

En cuanto a los permisos, encontramos que la entidad a la cual se le asignan o revocan permisos es a USUARIO. En el mundo real es a las personas a quienes se les señalan sus responsabilidades, por tanto nuestro modelo refleja esta situación. Dado que todas las actividades que se pueden realizar en el servidor están clasificadas como servicios, encontramos una relación natural entre usuarios y servicios, sin embargo, en este documento hemos detectado que es factible agrupar a los usuarios por los servicios que tienen derecho a ejecutar, también hemos visto que para conseguir un esquema flexible, un usuario puede pertenecer a más de un grupo de usuarios. De esta forma un usuario puede tener derecho sobre los servicios que tengan los grupos a los que pertenezca o solamente aquellos generales a los cuáles todos los usuarios tengan permiso de ejecutar si no pertenece a ningún grupo o el grupo en que está no tiene ningún servicio asociado.

Existe la posibilidad de que un usuario tenga todos los derechos de un grupo y algunos adicionales o bien, que tenga todos los derechos de un grupo excepto algunos. En este caso tenemos dos caminos, el primero consiste en crear un grupo distinto con los permisos que tiene este usuario, el segundo camino es modelar un esquema de excepciones. La primera opción tiene el inconveniente operativo de complicar la administración; supongamos que el grupo al que

pertenece el usuario con privilegios “diferentes” al resto de los integrantes, tiene permisos sobre una gran cantidad de servicios y este usuario necesita una pequeña cantidad de servicios adicionales, en este caso se tendría que crear un nuevo grupo y asociarle todos y cada uno de los servicios del otro grupo y mantener dos grupos con características similares, si posteriormente surge otro caso similar entonces el administrador tendría que mantener tres grupos y así sucesivamente, teóricamente es correcto, pero operativamente es complejo de mantener y aumentan las probabilidades de perder el control del sistema.

La segunda opción planteada, facilita el manejo de excepciones sin tener que mantener varios grupos, el riesgo de este esquema se reduce prácticamente a la facilidad que se le ofrece al usuario de no manejar grupos y mantener todos los permisos a nivel individual.

Evaluando ambas opciones encontramos que la primera opción complica la administración y la segunda la facilita, corriendo el riesgo de no manejar grupos. Si tomamos como un objetivo el diseño de un sistema flexible, fácil de mantener delegando las responsabilidades de administración al área coordinadora, optamos por la segunda opción, ofreciendo al usuario facilidad de administración de permisos con la posibilidad de hacer grupos de usuarios que simplifiquen el control. En la Figura 4-10 encontramos el modelo resultante, en él podemos observar las entidades USUARIO, GRUPO_USUARIO y SERVICIOS que están relacionadas y en las relaciones están los permisos que se tienen. Sin embargo detectamos que las relaciones son “muchos a muchos”, es decir, un usuario puede estar en ninguno, uno o varios grupos y un grupo tiene ninguno, uno o muchos usuarios, lo mismo sucede con los grupos y los servicios y entre los usuarios y los servicios.

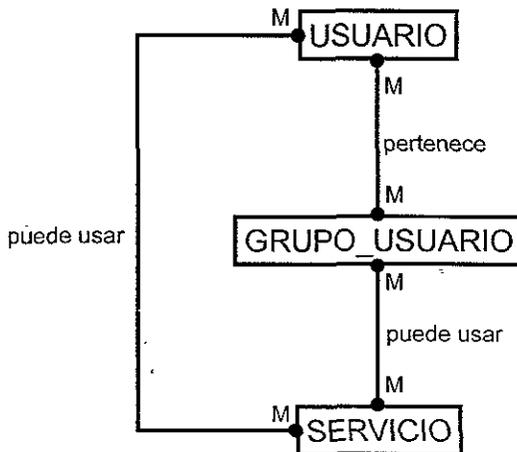


Figura 4-10. Modelo para controlar los permisos en el sistema.

Podemos anticiparnos a la normalización y “romper” estas relaciones, definiendo las entidades asociativas pertinentes, dado que en el modelo relacional esta práctica es necesaria, de no hacerlo se tendría una redundancia innecesaria, imposible de justificar³. En la Figura 4-11 podemos observar el modelo resultante final, las relaciones “muchos a muchos” son sustituidas por entidades asociativas donde la llave primaria de las entidades forma parte de la llave primaria de las entidades asociativas. Las relaciones INTEGRANTE GRUPO USUARIO y DERECHO GRUPO USUARIO no contienen atributos y se utilizan para asociar usuarios a grupos y para determinar los permisos de cada grupo respectivamente, la relación DERECHO USUARIO sí tiene un atributo que indica si el servicio indicado para el usuario es una adición o una excepción.

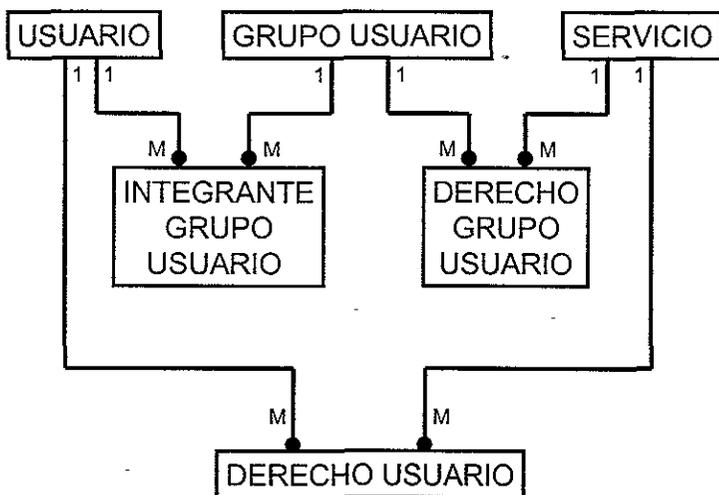


Figura 4-11. Modelo normalizado para controlar los permisos en el sistema

Validaciones

En lo referente a validaciones tenemos diferentes aspectos que son necesarios contemplar, por una parte tenemos los permisos de ejecución en los servicios, este problema está resuelto con el esquema de permisos, por otra parte tenemos las reglas del negocio y la tercera consiste en la integridad de los datos.

La integridad de los datos la resuelve el propio modelo relacional, ya que contamos con un conjunto de entidades que son utilizadas como catálogos (ver

³ En algunas ocasiones se permite la redundancia de información para mejorar el rendimiento de las aplicaciones al mejorar los tiempos de respuesta.

Figura 4-5, Figura 4-6, Figura 4-7, Figura 4-8 y Figura 4-9), basta con relacionarlas con las entidades respectivas, por ejemplo, tenemos la entidad EXISTENCIA PRODUCCIÓN que tiene entre sus atributos el espécimen y el estado físico, por otra parte tenemos una entidad ESPÉCIMEN y una entidad ESTADO FÍSICO, las llaves primarias de estas entidades forman parte de la llave primaria de EXISTENCIA como lo podemos observar en la Figura 4-12. En dicha figura podemos observar que la llave primaria de EXISTENCIA PRODUCCIÓN se conforma por las llaves primarias de las entidades ESPÉCIMEN y ESTADO FÍSICO, indicando que también son llaves foráneas, con esto delimitamos a que los valores permitidos sean los valores que estén en la entidad de dónde proviene.

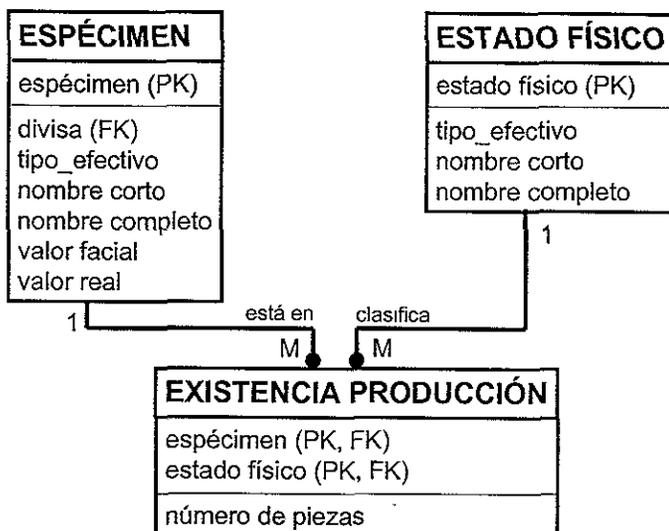


Figura 4-12. Modelo entidad - relación para el caso de existencias en producción

Relacionando las entidades con aquellas que consideramos catálogos podemos mantener la integridad de los datos.

En nuestro diseño utilizamos las reglas de cascada y restringida para mantener la integridad referencial, en los padres utilizamos cascada para el caso de las modificaciones y restringida para el caso de la eliminación. La razón para utilizar cascada en las modificaciones es para que todos los cambios en los catálogos estén reflejados en las diferentes entidades, principalmente en las existencias que son entidades cuyos valores permanecen en el tiempo, mientras que en las operaciones una vez terminado el día operativo son respaldadas junto a los catálogos que le dan validez. La eliminación de elementos está restringida porque no es posible eliminar una operación ni asignarle un valor nulo en el día operativo, las operaciones son la justificación de los cambios en las existencias

por lo que no es posible perder ningún movimiento. Por otro lado, no se pueden eliminar renglones de las existencias mientras tengan un saldo, así que si ya no se desea utilizar un elemento primero se debe garantizar que dicho elemento no esté referenciado.

En los hijos únicamente utilizamos la regla restringida, de esta forma no es posible insertar valores no catalogados.

Una vez definidas las reglas de integridad referencial, veamos lo concerniente a las reglas del negocio. En el capítulo 3 se mencionó un esquema que permite al usuario definir las reglas del negocio tanto a nivel registros de existencia como a nivel operación.

Las reglas del negocio relacionadas con los registros de existencias deben ser las primeras en definirse, ya que éstas marcan los sitios disponibles y la estructura interna, además definen que estados físicos son aceptados en cada bóveda. Podemos decir que los sitios, bóvedas y estados físicos en su conjunto presentan pocos movimientos, es decir, hay cambios de forma eventual, sin embargo no podemos decir lo mismo de los especímenes, ya que es más probable que existan nuevos ejemplares o que en las bóvedas se comiencen a almacenar. Por tanto es preferible delimitar que tipo de efectivo acepta cada combinación de sitio, bóveda y estado físico, y agregar a las existencias los renglones correspondientes cuando se involucre el espécimen. Veamos el siguiente ejemplo, supongamos que el sitio S1, en su bóveda B1, acepta los estados físicos EF3 y EF4, y actualmente almacena los billetes identificados con la clave de espécimen D1 y D2, y por cualquier motivo tiene ahora que almacenar el billete con la clave de espécimen D3 con el estado físico EF3, si se configurara cada uno de los especímenes que acepta la bóveda se tendría que agregar el renglón correspondiente antes de capturar el movimiento, sin embargo, si el capturista no se percató que dicho renglón no existía la operación será rechazada. Por el contrario, si se configura que el sitio S1 en su bóveda B1 que acepta los estados físicos EF3 y EF4 se puede almacenar billete, al momento de procesar se valida, si el renglón no existe⁴, entendiéndose por el renglón al sitio, bóveda, estado físico y clave de espécimen⁵; se insertaría el renglón después de verificar que la combinación sitio, bóveda y estado físico exista en las reglas del negocio y que además el tipo de efectivo del aceptado sea el mismo del espécimen.

La entidad resultante para establecer las reglas del negocio relativas a las existencias se llama **RESTRICCIÓN EXISTENCIA**, la cual contiene las combinaciones válidas de sitio, bóveda y estado físico así como el tipo de efectivo que acepta. El modelo puede observarse en la Figura 4-13.

⁴ Por las reglas de integridad referencial, asumimos que si existe el renglón es porque el registro de existencia es válido, de lo contrario no podría estar.

⁵ El número de piezas por principio es cero.

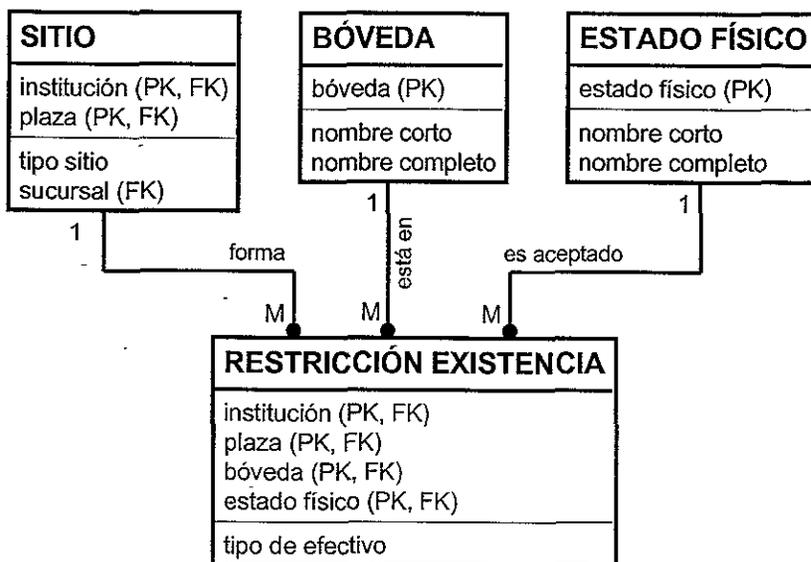


Figura 4-13. Modelo para definir reglas del negocio relativas a las existencias

Siguiendo con las reglas del negocio, ahora con las referentes a las operaciones, encontramos en el capítulo 3 un análisis de que información es importante para establecer las reglas según el tipo de operación. En dicho análisis se muestran los distintos escenarios, si el tipo de operación es relativo a la autorización o a las ordenes de fabricación existe un modelo que permite establecer los movimientos válidos, si las operaciones involucran movimientos internos dentro de un sitio hay otro modelo, y así sucesivamente, en total encontramos ocho modelos diferentes que engloban todas las reglas del negocio.

El modelo con mayor número de variables involucradas es el de las remesas o "Banco Central – Banco Central" que incluye tanto para el origen como para el destino el tipo de sitio o la institución y la plaza, la bóveda y el estado físico; el modelo más sencillo es el de producción que solamente requiere un estado físico para el origen y otro para el destino.

Podemos diseñar cada uno de estos modelos fragmentando las reglas según el tipo de operación o utilizar una sola entidad que contenga todas las reglas. Apegándonos a las formas normales nuevamente, encontramos que la primera opción es la más adecuada ya que disminuye la porosidad⁶ de la entidad. Sin

⁶ Una entidad es porosa cuando en los renglones se almacenan para algunos atributos valores nulos.

embargo, esto complica al administrador la definición de las reglas ya que tiene que conocer a que tipo de regla corresponde para definirla en el lugar adecuado.

La segunda opción obviamente está en contra de las formas normales por la alta porosidad que acepta, sin embargo, también facilita la administración de las reglas por estar concentradas en un solo punto.

Conociendo el costo de tener de una entidad altamente porosa, pero justificando su existencia por la facilidad de administración y claridad de la información optamos por concentrar todas las reglas del negocio bajo una sola entidad. Ante esta decisión nos encontramos ahora que las reglas del negocio pudieran ser mal definidas si el encargado de alimentar la entidad no ha entendido la operación a la cuál le esté asignando la regla, así que utilizaremos un esquema alterno que facilite la definición.

Partiendo del hecho que las operaciones están clasificadas en familias de operaciones es factible asociar un tipo de asignación de reglas, de tal forma que al definir una regla del negocio el encargado solamente inserte los valores a los atributos correspondientes. Por ejemplo, para una operación relativa a una orden de fabricación, las reglas del negocio únicamente pueden estar en función de los estados físicos, por tanto los demás atributos deben ser nulos, esta regla para definir reglas la mantendremos en otra entidad, de tal forma que al ingresar una regla para un tipo de operación se aceptará siempre y cuando cumpla con la regla de definición.

Dado que las reglas para un mismo tipo de operación pueden ser diversas es conveniente utilizar como llave primaria el tipo de operación y un consecutivo, principalmente porque los demás atributos pueden ser nulos y esto no es aceptado para definir una llave primaria.

Al igual que los permisos, las reglas del negocio pueden tener excepciones, en este caso es conveniente distinguir las según el tipo de regla, funcionalmente basta con distinguir si una regla es general o si es una restricción, sin embargo, para una mayor claridad al usuario vamos a distinguir las reglas en generales, adiciones o restricciones.

Una regla general es la regla fundamental del negocio, una adición es una regla que habilita a un sitio a realizar un movimiento en forma exclusiva, es decir ningún otro sitio podrá hacerlo, y una restricción es limitar a un sitio de realizar algún tipo de movimiento. En la Figura 4-14 podemos observar el modelo resultante.

A lo largo de este documento se planteó la necesidad de manejar un esquema que permita realizar algunos movimientos, aun cuando no estén contemplados en las reglas del negocio, principalmente para casos de contingencia, en los cuáles se tengan que almacenar piezas en lugares no destinados para ellas. Partiendo del supuesto que son casos excepcionales, de lo contrario deberían

formar parte de las reglas del negocio, manejamos un esquema que indique que a una operación específica no se le validen las reglas del negocio.

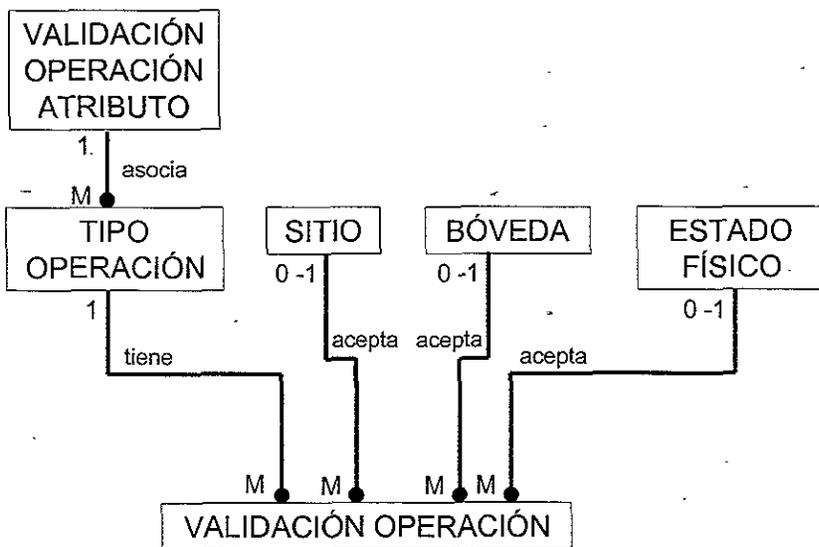


Figura 4-14. Modelo utilizado para almacenar las reglas del negocio.

Tomando como base la autorización que solicitan los establecimientos cuando se realiza un pago con una tarjeta de crédito, este esquema consiste en que el usuario que se encuentre ante una situación de esta naturaleza solicite la autorización al administrador del sistema para que su operación sea aceptada, indicando el folio de la terminal para esa operación, el tipo de operación, el importe del movimiento y el motivo por el cuál se solicita, esta información forma parte de la entidad AUTORIZACIÓN OPERACIÓN, que tendrá además el usuario que la solicitó y el usuario que la otorgó. Como llave primaria tendrá un folio asociado por el sistema.

Para simplificar el uso de la autorización el administrador registra la autorización y posteriormente el usuario envía la operación, al detectar la operación en el servidor se le asocia la fecha de registro y el folio del sistema para que no pueda volverse a utilizar. De esta manera cuando se procese la operación no se le aplicarán las validaciones relativas a las reglas del negocio. En la Figura 4-15 podemos observar el modelo resultante.

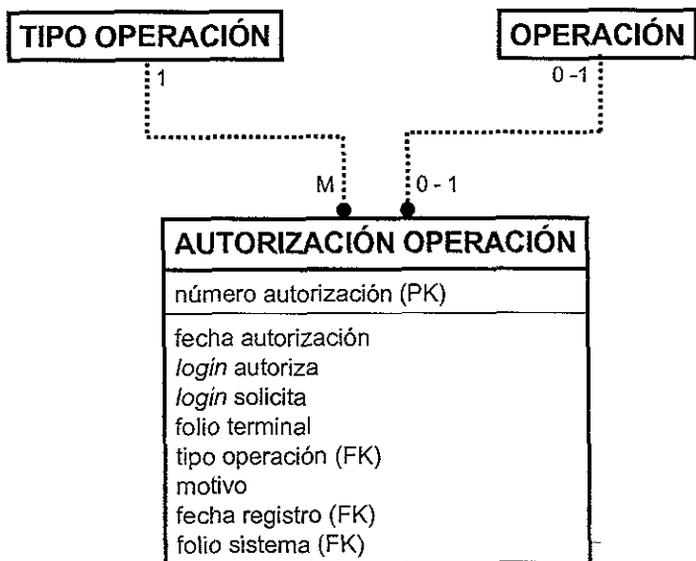


Figura 4-15. Modelo para autorizar operaciones que no cumplen con las reglas del negocio.

Control y Configuración

Uno de los requerimientos planteados es el control de las actividades dentro del sistema. Podemos dividir el control en tres aspectos, el primero es relativo al establecimiento del horario de operación, el segundo corresponde a la identificación de los servicios utilizados y el tercero al control de las actividades.

Para el establecimiento de los horarios hemos definido dentro de la entidad SERVICIO algunos atributos que permiten mantener un horario flexible y configurable por el administrador, sin embargo, falta definir las referencias, en los servicios se determina cuántos minutos antes o después de la hora cierre o de apertura comienza su horario y de la misma forma se delimita el fin del horario, esto nos lleva a contar con una entidad donde se almacenen estas referencias. En la parte del diseño de las operaciones se determinó la necesidad de una llave artificial formada en el servidor, uno de los atributos de esta llave es la fecha operativa, esto nos lleva a la necesidad de una entidad en la cual se almacene la fecha operativa actual, dado que el horario general del sistema también forma parte del día operativo encontramos la entidad FECHA REGISTRO, que tiene la fecha operativa y las fechas y horas de referencia que permiten la determinación del horario de cada servicio.

En la Figura 4-16 encontramos la definición de esta entidad, por cuestiones prácticas y para facilitar la validación de la fecha valor, asumiendo que la mayor parte de las operaciones con fecha valor futuro utilizan el día hábil siguiente, agregamos esta información a la entidad.

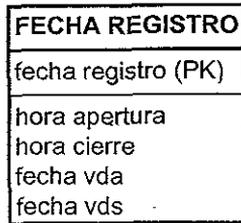


Figura 4-16. Modelo de la entidad FECHA REGISTRO

Para conocer cuáles servicios fueron utilizados en el día y por quién, en este sistema manejamos una bitácora. En ésta almacenamos por cada usuario, qué servicios utilizó, sin embargo, como un usuario puede interactuar con el servidor desde más de una terminal, es conveniente agregar de qué terminal ejecutó cada servicio, por otro lado, un usuario seguramente utiliza más de un servicio en un día, por tanto es importante almacenar la fecha y hora del servidor en la cuál se ejecutó el servicio, además, este esquema permite estudiar las horas de mayor carga en el servidor.

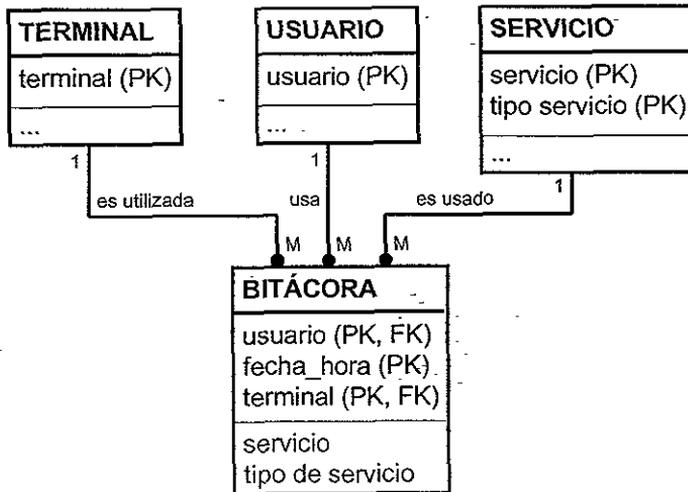


Figura 4-17. Modelo de la entidad asociativa BITÁCORA.

Por último veamos la parte del control de actividades.

En cuanto configuración, contamos con dos tablas básicas, una para las constantes del sistema y otra para las reglas. Las constantes son aquellos valores que se requieren para identificar o procesar la información, por ejemplo, dada la estructura de los sitios, es importante identificar quién es el Banco Central, así, en lugar de codificar la clave de dicha institución, se define como una constante en una entidad, permitiendo cambiar la clave sin modificar código.

En el caso de las reglas, pudiéramos utilizar el esquema que proporcionan los distintos manejadores como las *rules* y los *check constraint*, sin embargo, en caso de que se inserte un valor que no cumpla con las reglas el mensaje de error que devuelven los manejadores es poco amigable; tomando como base que es deseable que el propio usuario identifique el motivo del rechazo, utilizamos una entidad que contenga los valores aceptables para cada regla, por ejemplo, en varias entidades definimos el atributo tipo de efectivo, este atributo puede ser definido como regla donde los valores válidos son: billete y moneda. Así en caso de incumplimiento se puede notificar que está insertando un tipo de efectivo erróneo y el propio usuario puede detectar el motivo y arreglarlo.

Todos los movimientos de los sistemas de flujos de caja tienen que registrarse en el día operativo, esto nos dice que hay un inicio y un fin de día. En el inicio de día se tienen que realizar algunos procesos importantes para que el sistema funcione correctamente, por ejemplo obtener una copia de los catálogos institucionales para mantener actualizado el sistema, esto siempre y cuando el Banco Central maneje un esquema centralizado de catálogos.

En general podemos decir que antes de permitir el uso del sistema se requiere establecer la fecha operativa, los horarios de referencia, obtener los catálogos institucionales, si los hay, y en algunos casos pueden procesarse las operaciones cuya fecha valor coincida con la fecha operativa. En este sentido el Banco Central determina la forma de manejar las operaciones con fecha valor futuro, algunos Bancos las procesan antes de abrir el sistema, otros hasta que se presenta el movimiento.

Una vez realizados los procesos del inicio, el administrador del sistema lo tiene que habilitar, en este sentido tenemos dos puntos que cuidar, las consultas y el envío de operaciones, el sistema puede estar habilitado para consultas y deshabilitado para el envío, lo que no es prudente es permitir el envío sin permitir las consultas, sin embargo esto también es decisión de los Bancos Centrales.

Mientras el sistema esté habilitado los usuarios envían, consultan, obtienen reportes, etc., cuando ya se han capturado y enviado todas las operaciones y el administrador decide que puede cerrar el sistema lo deshabilita y procede a realizar los procesos de fin de día, que comprenden al menos el respaldo de la

información del día en un histórico y en un dispositivo alterno, y la depuración de las entidades de operaciones.

Para mantener en la historia algunos días en línea, el respaldo a histórico consiste en insertar todo el contenido de las entidades en entidades históricas, éstas tienen la misma estructura que las entidades de operación con un atributo más en la llave primaria que es la fecha de respaldo. Con este esquema se tiene una "foto" de cómo cerró el sistema.

Todos los procesos, tanto del inicio como del fin de día, tienen un orden lógico que es necesario cuidar, sería ilógico eliminar las operaciones procesadas sin antes haber sido respaldadas. Por tanto se requiere un control de procesos que impida al administrador realizar un proceso sin haber ejecutado el proceso precedente.

Una lista de procesos de inicio de día en orden sería: Establecer la fecha operativa y los horarios de referencia, actualización de catálogos institucionales (si los hay), procesamiento de operaciones capturadas en días anteriores cuya fecha de proceso sea la misma que la fecha operativa⁷, habilitar el sistema y habilitar la transmisión de operaciones. Si algún proceso no se ejecuta no es posible continuar.

De igual forma, una lista de procesos de fin de día consistiría en: deshabilitar la transmisión, deshabilitar el sistema, respaldar la base de datos en otro dispositivo, respaldar la información en tablas históricas y eliminar las operaciones procesadas y canceladas.

Para controlar estos procesos utilizamos la entidad CONTROL, que está compuesta de un identificador del proceso, una breve descripción del mismo, un estado que indica si ya se realizó o no y la fecha y hora del servidor en la cuál se efectuó el proceso. En la Figura 4-18 encontramos el modelo creado para controlar y configurar el sistema.

La razón de nombrar a la columna "estado" con este nombre en lugar de llamarla "realizado", obedece a las formas normales. Dentro de la normalización encontramos que una columna solamente puede representar un concepto y no variar según el renglón. En todos los renglones de esta entidad, en la columna "estado", se almacena el valor de realizado o no realizado. Los únicos dos renglones que cambian el significado de la columna son los que pertenecen al acceso al sistema y a la transmisión de operaciones, cuyos casos esta columna indica si está habilitado o no.

⁷ Algunos Bancos Centrales prefieren procesar las operaciones en este punto del día, otros prefieren procesarlos hasta que se presente el momento.

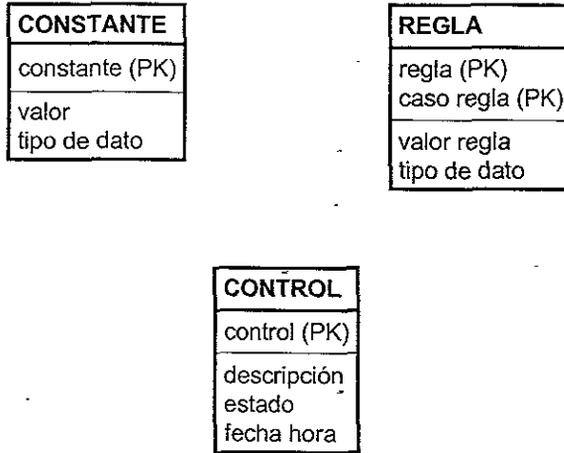


Figura 4-18. Entidades requeridas para la configuración y control de procesos.

El modelo entidad – relación completo está ilustrado en los diagramas 4-1, 4-2, 4-3, 4-4, 4-5, 4-6, 4-7 y 4-8. Como el diagrama está fraccionado en la encontramos el plano general del diagrama.

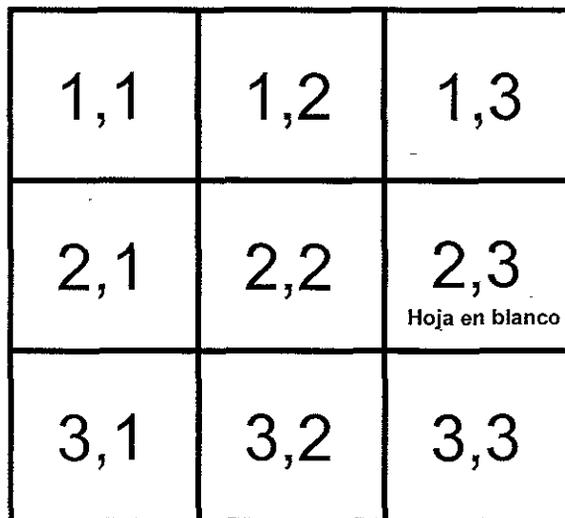
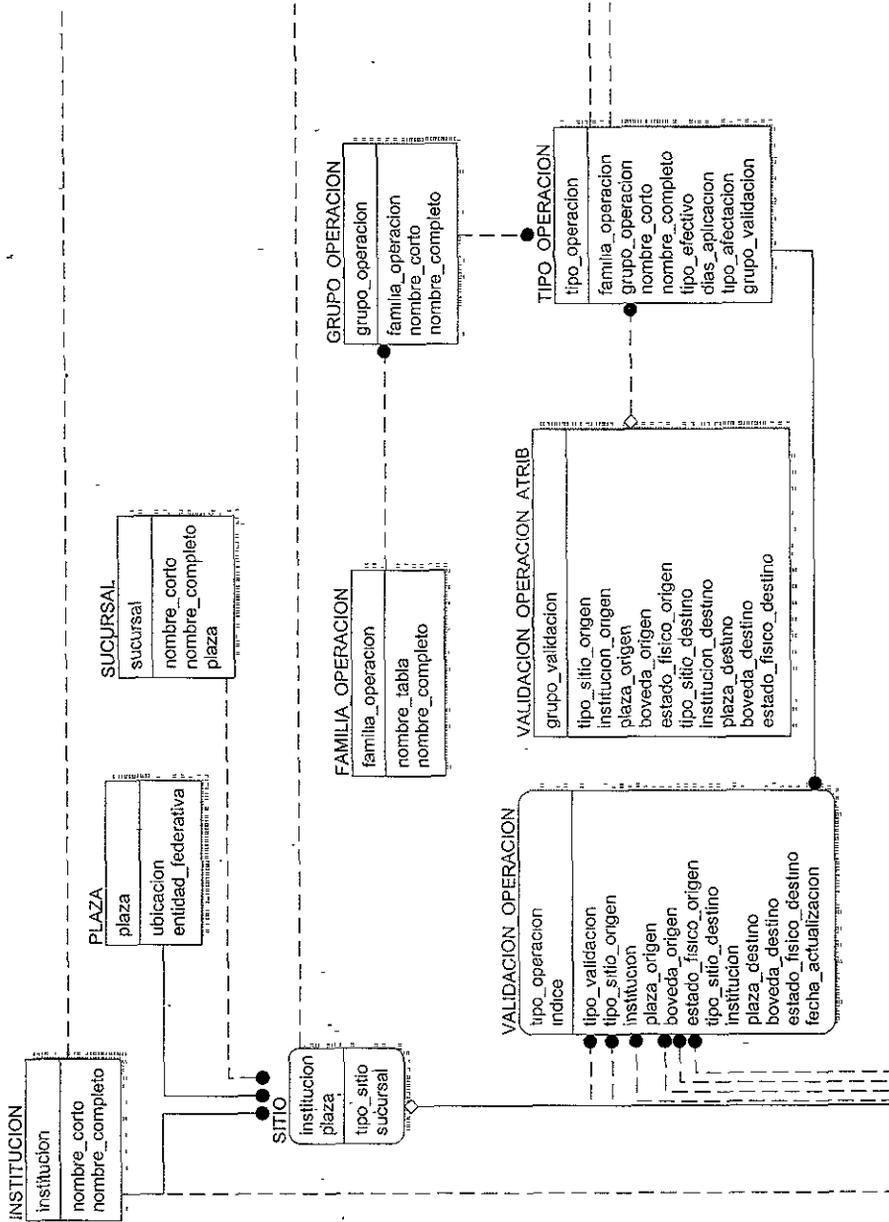
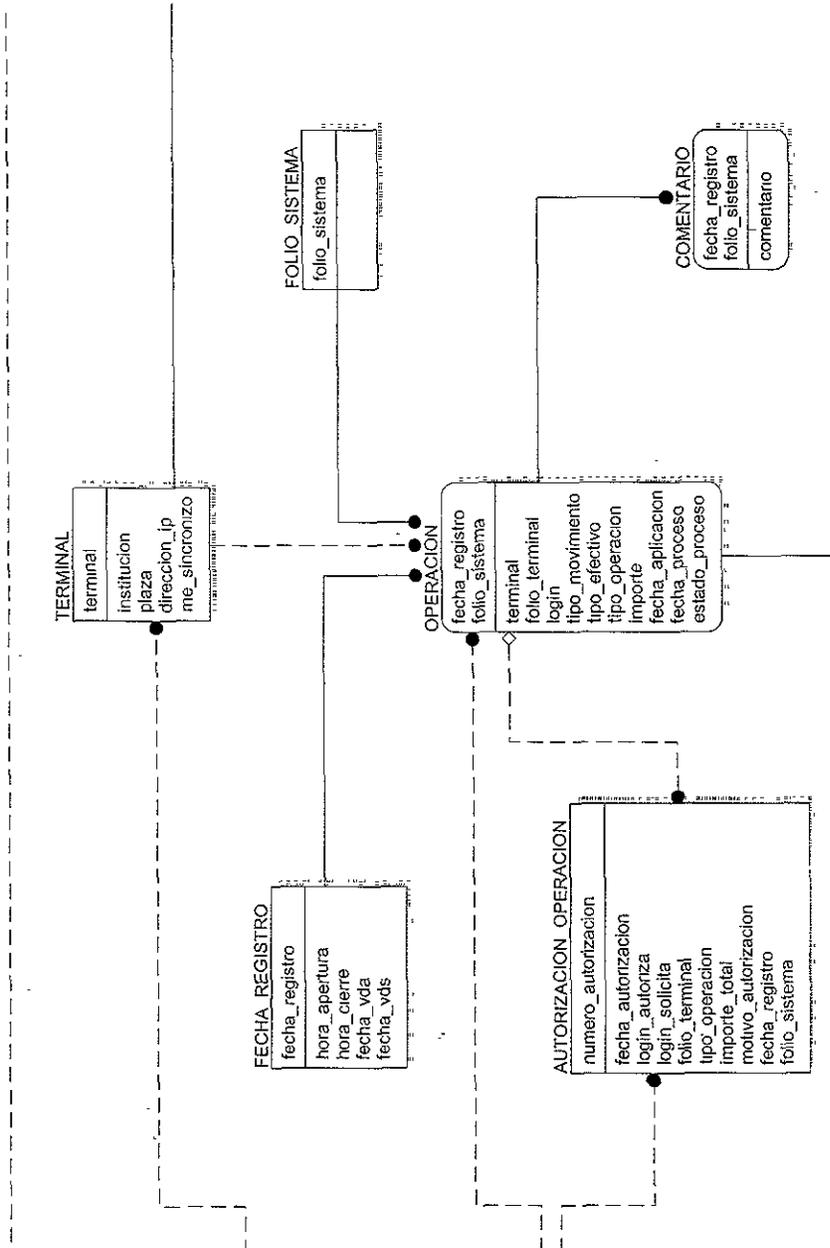
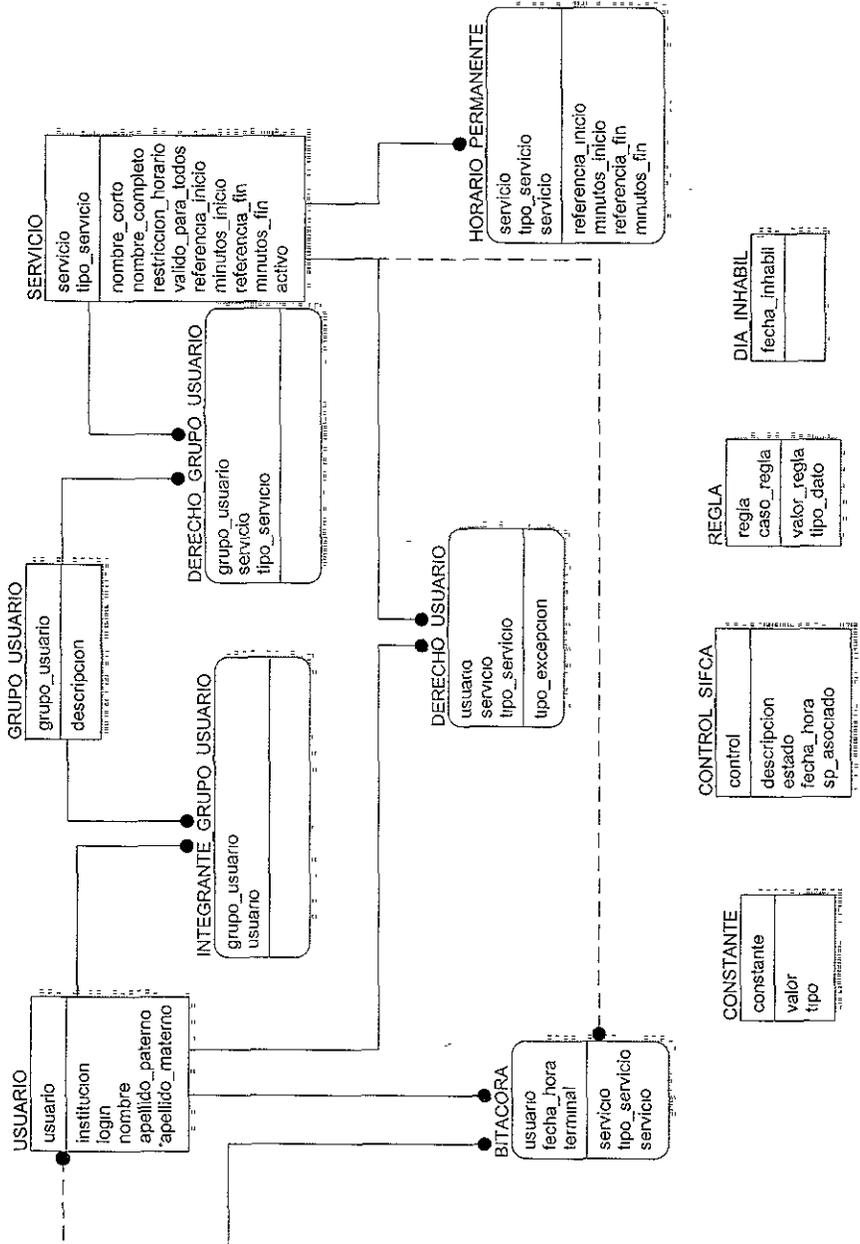
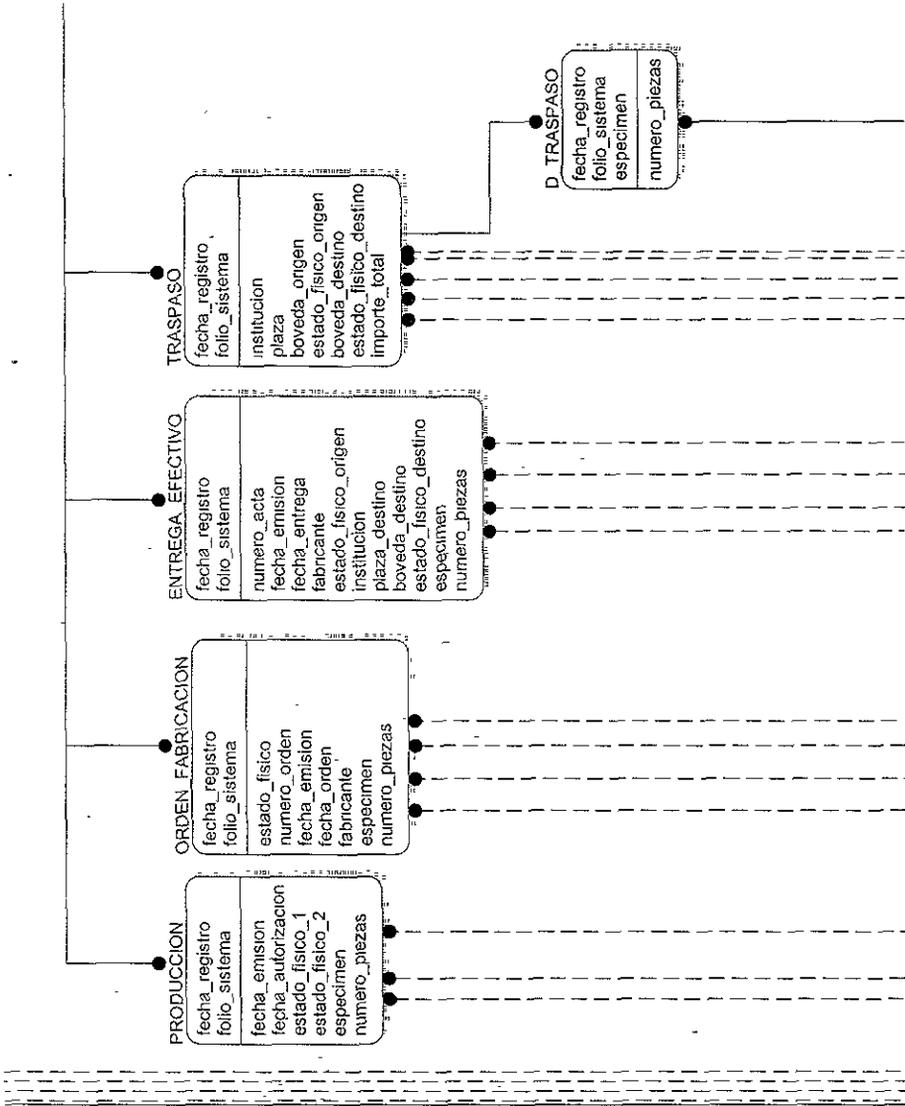


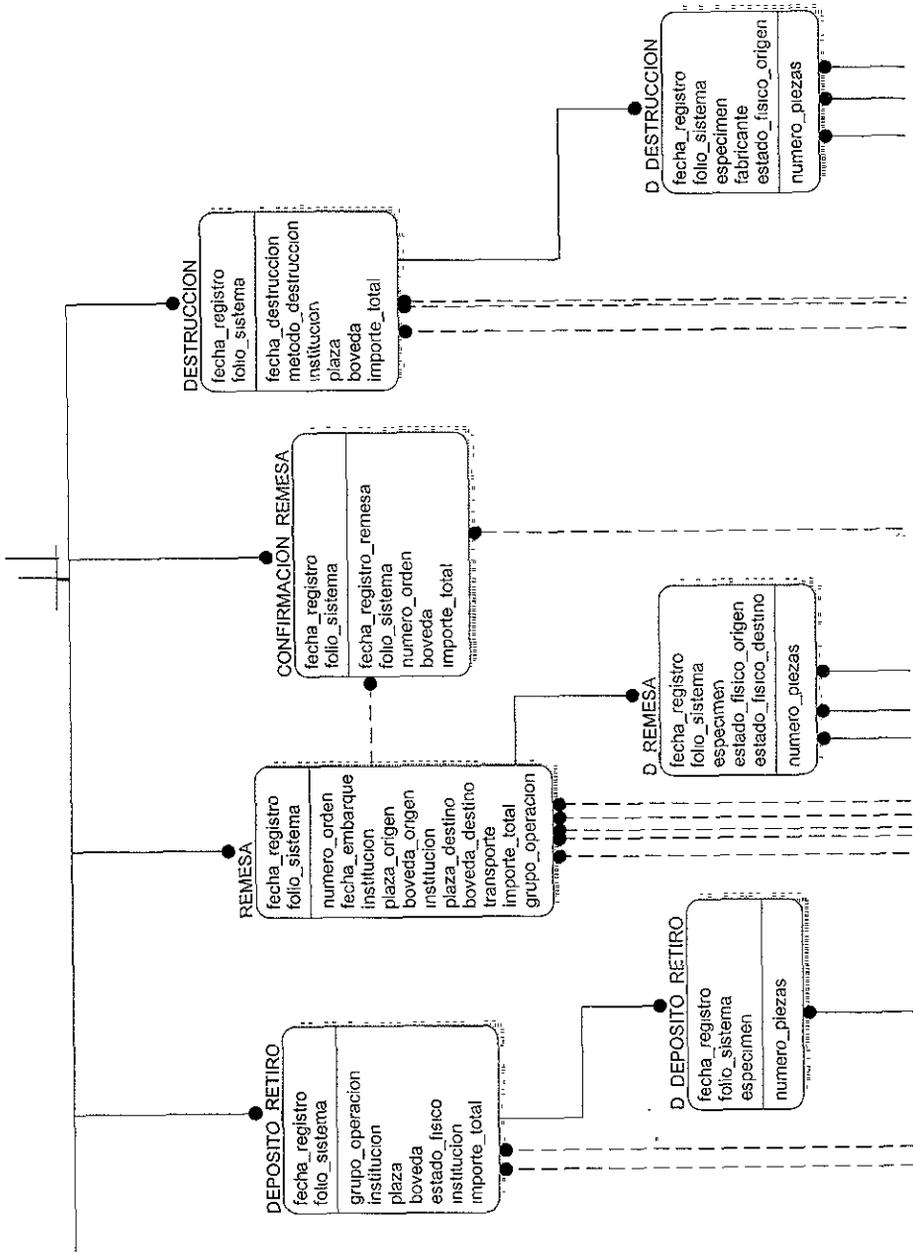
Figura 4-19. Plano general para interpretar el modelo lógico E-R.

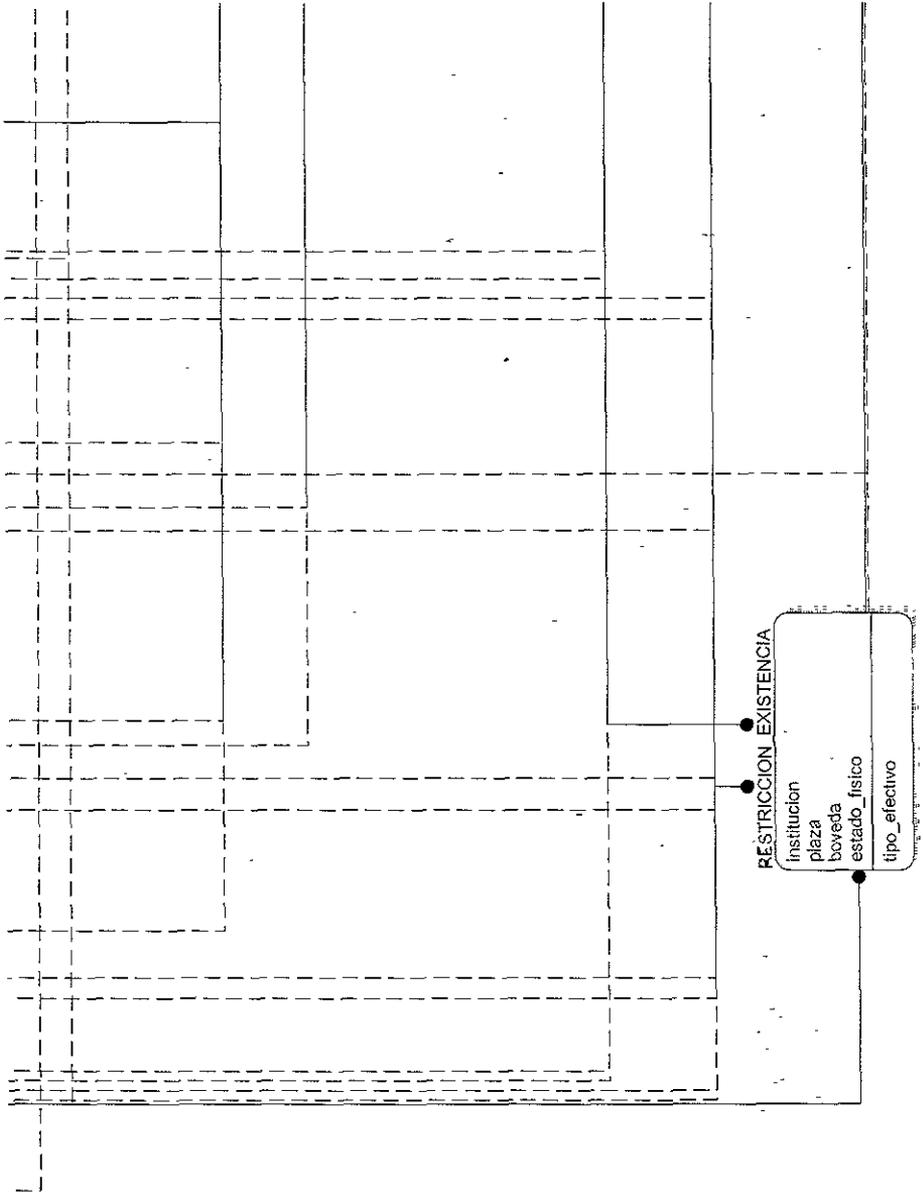


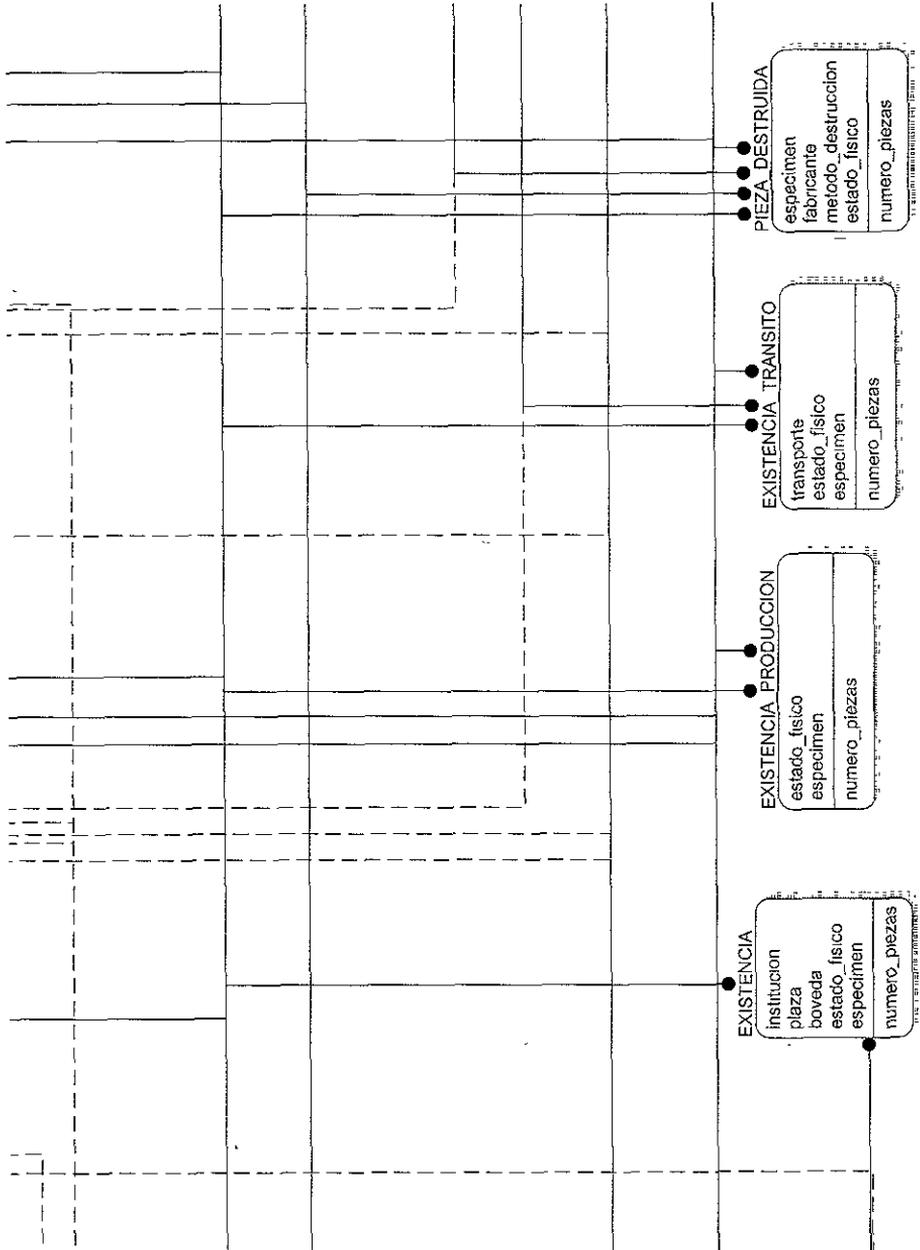


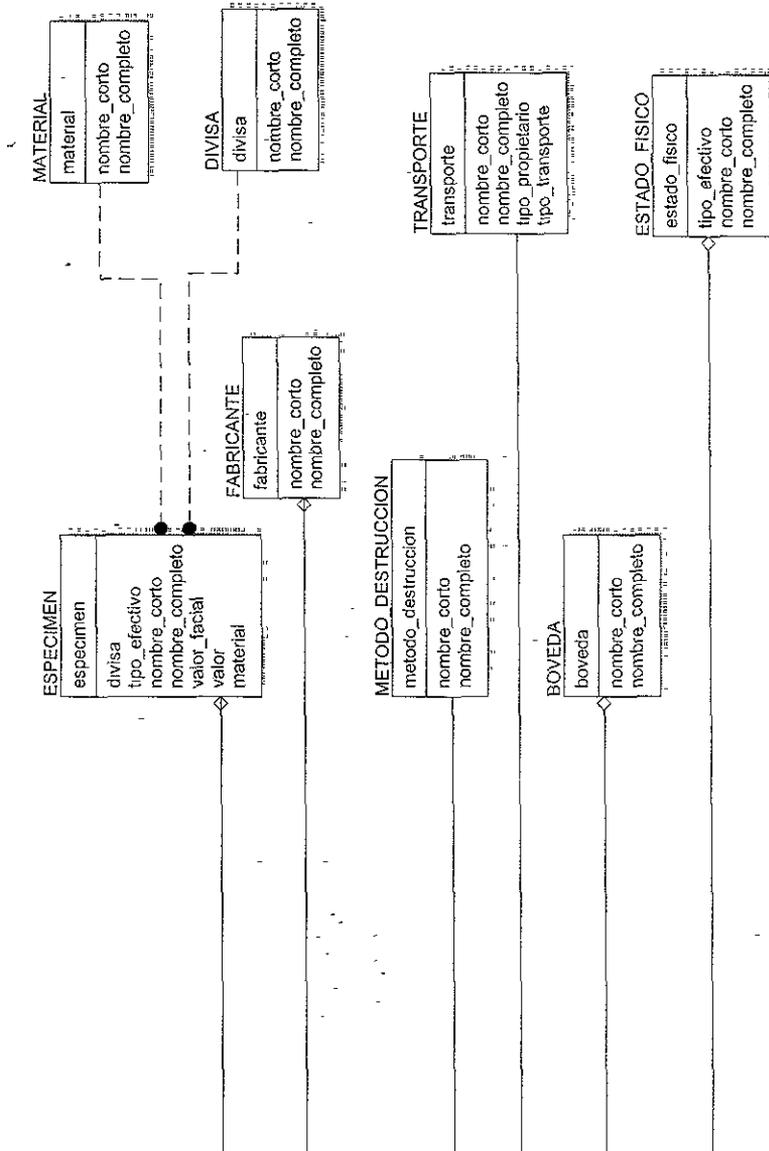












4.3 Normalización

Una vez terminado el modelo entidad – relación, es necesario verificar que cumple con las formas normales ya mencionadas, de lo contrario es necesario normalizar. En este punto del diseño se crean entidades asociativas cuando existen relaciones “muchos a muchos”, también es necesario verificar que la información que se almacena en cada atributo de cada entidad sea atómico, es decir, que ya no puede descomponerse en varios atributos.

Dado que en el mismo diseño se consideraron las formas normales, encontramos que no hay relaciones “muchos a muchos”. La mayoría de las entidades están normalizadas, exceptuando aquellas que, en su momento, se indicaron las razones por las cuales no conviene normalizar. En las entidades sobrantes encontramos que la entidad USUARIO no está debidamente normalizada, porque el atributo nombre puede ser descompuesto en nombre, apellido paterno y apellido materno.

Hay que tomar en cuenta que entre más normalizado se encuentre un diseño menor será la porosidad y la redundancia de datos que no forman parte de una llave primaria. Uno de los preceptos del modelo relacional es evitar columnas cuyo contenido pueda ser obtenido del contenido del renglón y de sus relaciones, un ejemplo es el importe del movimiento, dado que se tiene el número de piezas de cada espécimen y, por otro lado se tiene el valor real de cada pieza, el importe se puede calcular al momento de realizar las consultas.

Sin embargo, un modelo normalizado, en algunas ocasiones, puede invertir demasiado tiempo al resolver consultas o al insertar información. Una práctica común en el diseño de bases de datos es desnormalizar, esto es, una vez que se tiene un diseño depurado se ajusta para mejorar el rendimiento, esto representa una regresión a las formas normales más elementales.

Algunas de las prácticas más comunes son:

- Reagrupar entidades
- Dividir entidades
- Crear atributos calculados
- Agregar atributos en forma redundante

Al reagrupar entidades se busca disminuir el número de entidades involucradas para resolver una consulta crítica. Al reagrupar posiblemente se obtenga una entidad con una cierta porosidad. Un claro ejemplo de esta práctica es la entidad VALIDACIÓN OPERACIÓN de nuestro diseño, salvo que fue diseñada así desde el inicio, como se recordará se podrían haber creado varias entidades para

almacenar las reglas del negocio y en su lugar se creó una sola entidad con un cierto grado de porosidad.

Dividir las entidades consiste en una entidad con demasiados atributos puede ser dividida en dos entidades con una relación uno a uno, esto, en el modelo físico, disminuye el contenido de los renglones en páginas de datos⁸ permitiendo almacenar más renglones por página, lo que desemboca en un menor número de accesos para resolver consultas. En este caso la información utilizada con mayor frecuencia es separada del resto para formar otra entidad. Un ejemplo puede ser un catálogo de proveedores, supongamos que dicho catálogo tiene como atributos una clave de proveedor, su nombre y sus datos fiscales como su R.F.C., dirección, colonia, entidad federativa y código postal, si la información más utilizada es la clave del proveedor y su nombre. Al desnormalizar la entidad PROVEEDOR sólo tendría estos atributos y la información fiscal se tendría en otra entidad.

Para resolver consultas críticas, en algunas ocasiones se crean columnas con valores calculados, por ejemplo, si se tiene una entidad que contiene los movimientos contables que afectaron una cuenta y por otro lado se tiene el saldo inicial de la misma, se podría conocer el saldo en cualquier momento sumando los importes de los movimientos, sin embargo, si es una consulta muy frecuente realizada por un gran número de usuarios en forma simultánea, es muy probable que el tiempo de respuesta no satisfaga a los usuarios, en estos casos se puede crear un atributo calculado que muestre el saldo actual de la cuenta, reduciendo el número de accesos a páginas de datos.

La redundancia de información consiste en agregar atributos de los cuales se puede obtener información siguiendo las relaciones, esta práctica es útil para reducir el número de tablas involucradas para resolver una consulta, por ejemplo, retomemos el caso del proveedor, supongamos que el diseño pertenece a una empresa que genera una gran cantidad de facturas, así que tenemos una entidad más llamada FACTURA y los directivos de la empresa desean una relación de cuánto han facturado a cada R.F.C., en este caso podría incluirse el R.F.C. en la entidad FACTURA, a pesar de poderse obtener utilizando la entidad que contiene los datos fiscales.

Para desnormalizar se debe analizar el objetivo del sistema, las consultas críticas y aquellas que se realizan con mayor frecuencia y siempre se debe estar consciente del impacto que tiene cada modificación al diseño. El hecho de desnormalizar no implica que todo el diseño debe modificarse, sólo deben ajustarse aquellas entidades para las cuáles se tenga una justificación sólida y que realmente mejoren el rendimiento de la aplicación.

En nuestro diseño encontramos varias entidades que deben desnormalizarse.

⁸ Una página de datos es un espacio de tamaño fijo donde son almacenada la información.

El primer caso es el atributo grupo de operación. En la entidad DEPÓSITO RETIRO se almacenan los depósitos y los retiros indistintamente, ya que el tipo de operación indica de qué operación está contenida en el renglón. Una consulta crítica consiste en obtener el número de depósitos y el número de retiros realizados por cada banco. Si no agregamos este atributo tendremos que involucrar las entidades DEPÓSITO RETIRO, OPERACIÓN, TIPO OPERACIÓN y GRUPO OPERACIÓN para proporcionar esta información, sin embargo, si lo agregamos sólo requerimos de las dos primeras entidades de la lista.

En la entidad REMESA encontramos el mismo caso, una consulta importante es identificar cuántos envíos y cuántas concentraciones hay vigentes, definamos el envío cuándo la remesa sale de las instalaciones de un Banco Central y concentraciones cuándo sale de un Corresponsal.

Como se puede observar no es necesario agregar el atributo "grupo operación" en todas las entidades, ya que, salvo las dos entidades mencionadas, no hay una consulta crítica que justifique la redundancia.

El segundo caso es el importe del movimiento. Existen varias consultas críticas donde el importe del movimiento es muy importante, por ejemplo, el importe total operado en cada sitio por tipo de operación. Otro ejemplo es para búsquedas de operaciones cuándo sólo se conoce el importe. Si consideramos que el número de movimientos en un día puede ser considerable, resulta desgastante calcular el importe de cada operación, más aún si consideramos que tendríamos que consultar todas las tablas de operaciones. Así, una de las entidades que requiere este atributo calculado es OPERACIÓN, que es donde está la información común de todos los movimientos. Este atributo también nos sirve para reducir el margen de error de captura, ya que se verifica que el importe total del movimiento coincida con la suma de los importes de cada espécimen.

En las entidades TRASPASO, DEPÓSITO RETIRO, REMESA y DESTRUCCIÓN, este atributo es útil para resolver las consultas de los movimientos sin tener que considerar los detalles de cada operación.

En el caso de CONFIRMACIÓN REMESA, este atributo sirve como medio de validación para reducir el riesgo de un error en la captura. En las demás entidades de operaciones este atributo no es necesario.

Las demás tablas no requieren ningún otro atributo para mejorar el rendimiento de la aplicación. Así obtenemos el modelo definitivo, incluido en el apéndice B.

Una vez definido el modelo entidad – relación para el sistema de flujos de caja, es necesario crear el modelo físico. En el modelo físico las entidades se convierten en tablas y los atributos en columnas. A cada columna se le asocia el tipo de dato que le corresponda y, en su caso, se le asocian los *defaults*, *check constraint* y reglas. También se definen los índices adecuados para obtener el

mejor rendimiento posible, considerando las consultas más frecuentes y las consultas críticas.

En nuestro caso no utilizaremos ninguno de estos elementos. Este diseño no contempla un valor por *default* para ninguna de sus columnas. En el caso de los *check constraint*, a pesar de ser muy útiles para mantener la integridad referencial y más rápidos que realizar la validación por medio de *triggers*, sin embargo, los mensajes de error son muy generales, lo que puede resultar confuso para el usuario. Con la intención de crear un producto "amigable" con el usuario la integridad referencial está contenida en los *triggers*, de tal forma que se proporcione un mensaje de error más claro, dónde se indique en que columna se intentó escribir un valor sin referencia.

El caso de las reglas es el mismo, en una regla se puede definir el dominio de la columna, pero el mensaje de error no ilustra cuál regla no se cumplió, en su lugar se definió la entidad REGLA en la cuál están contenidos los dominios para cada regla y, al igual que en el caso anterior, en los *triggers* se lleva a cabo esta validación.

En cuanto a los índices, tomamos las llaves primarias de cada una de las entidades para convertirlas en los índices *clustered*. En algunos casos es necesario crear índices *non – clustered* para mejorar el rendimiento de algunas consultas. En las tablas OPERACIÓN, REMESA y USUARIO encontramos la necesidad de crear otros índices.

La tabla OPERACIÓN tiene como índice *clustered* a la fecha de registro y el folio de sistema, adicionalmente creamos un índice *non – clustered* con las columnas fecha de registro, terminal y folio de la terminal y creamos otro índice *non – clustered* que acepta valores duplicados en la columna fecha de aplicación. El primer índice *non – clustered* se utiliza para facilitar la ubicación de operaciones enviadas desde una terminal y el segundo para identificar las operaciones cuya fecha valor es futuro.

En la tabla REMESA se creó un índice *non – clustered* en la columna número de orden, con la intención de facilitar la localización de una remesa en particular.

Por último, en la tabla USUARIO se creó un índice *non – clustered* en la columna *login*, la razón es que internamente se maneja un número de usuario y es necesario relacionarlo con la clave de acceso al RDBMS, dado que no es "sano" relacionar tablas del sistema con tablas de usuario ya que pudiera bloquear al RDBMS, se puede obtener la clave y así localizar el número de usuario. No se utilizó el *login* para identificar al usuario internamente, principalmente, por el espacio que ocuparía en la BITÁCORA. El número de usuario se puede definir como entero mientras que el *login* es una cadena de caracteres variable.

4.4 Diccionario de datos

Una vez definidos los elementos a utilizar definamos el diccionario de datos. Aquí mostramos, para cada tabla, todas las columnas, el tipo de dato con el cual fueron definidas, si aceptan valores nulos, si forman parte de la PK y si están definidas como FK. Para una mayor claridad dividimos el diccionario en las siguientes secciones:

- Catálogos
- Existencias
- Permisos de ejecución de los servicios
- Controles de operación
- Operaciones
- Validaciones
- Configuración y control

En la Tabla 4-1 se muestra el diccionario de datos de las tablas consideradas catálogos. Se puede observar que están definidas en varias tablas las columnas nombre corto y nombre completo. El nombre corto es una abreviación o una descripción breve que permite al usuario identificar su significado, por ejemplo, el nombre corto de una divisa es USD (*United State Dollar*, Dólar de los Estados Unidos) mientras que su nombre completo es "Dólar de los Estados Unidos de Norteamérica".

Tabla	Columna	Tipo de Dato ⁹	Acepta Nulos	Forma la PK	Es FK
SUCURSAL	sucursal	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
	plaza	string	N	N	N
INSTITUCIÓN	institución	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N

continúa ...

⁹ El tipo de dato *string* es una cadena de caracteres. El tipo de dato *money* es numérico.

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	Es FK
PLAZA	plaza	string	N	S	N
	ubicación	string	N	N	N
	entidad federativa	string	N	N	N
SITIO	institución	entero	N	S	S
	plaza	string	N	S	S
	tipo sitio	entero	N	N	N
	sucursal	entero	N	N	S
BÓVEDA	bóveda	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
ESTADO FÍSICO	estado físico	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
	tipo efectivo	entero	N	N	N
DIVISA	divisa	string	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
MATERIAL	material	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
ESPÉCIMEN	espécimen	entero	N	S	N
	divisa	string	N	N	S
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
	tipo efectivo	entero	N	N	N
	valor	numérico	N	N	N
	valor facial	numérico	N	N	N
	material	entero	S	N	S

continúa ..

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	Es FK
FABRICANTE	fabricante	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
DÍA INHÁBIL	fecha inhábil	fecha – hora	N	S	N
FAMILIA OPERACIÓN	familia operación	entero	N	S	N
	nombre tabla	string	N	N	N
	nombre completo	string	N	N	N
GRUPO OPERACIÓN	grupo operación	entero	N	S	N
	familia operación	entero	N	N	S
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
TIPO OPERACIÓN	tipo operación	entero	N	S	N
	familia operación	entero	N	N	N
	grupo operación	entero	N	N	S
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
	días aplicación	string	N	N	N
	tipo afectación	string	N	N	N
	tipo efectivo	entero	N	N	N
MÉTODO DESTRUCCIÓN	método destrucción	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
TRANSPORTE	transporte	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
	tipo propietario	entero	N	N	N
	tipo transporte	entero	N	N	N

continúa

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	Es FK
SERVICIO	servicio	entero	N	S	N
	tipo servicio	entero	N	S	N
	nombre corto	string	N	N	N
	nombre completo	string	N	N	N
	valido para todos	carácter	N	N	N
	restricción horario	carácter	N	N	N
	minutos inicio	entero	N	N	N
	referencia inicio	carácter	N	N	N
	minutos fin	entero	N	N	N
	referencia fin	carácter	N	N	N
activo	carácter	N	N	N	
HORARIO PERMANENTE	servicio	entero	N	S	N
	tipo servicio	entero	N	S	S
	minutos fin	entero	N	N	N
	minutos inicio	entero	N	N	N
	referencia fin	carácter	N	N	N
	referencia inicio	carácter	N	N	N
TERMINAL	terminal	entero	N	S	N
	institución	entero	N	N	S
	plaza	string	N	N	S
	dirección ip	string	N	N	N
	me sincronizo	carácter	N	N	N
GRUPO USUARIO	grupo usuario	entero	N	S	N
	descripción	string	N	N	N
USUARIO	usuario	entero	N	S	N
	institución	entero	N	N	S
	login	string	N	N	N
	nombre	string	N	N	N
	apellido paterno	string	N	N	N
	apellido materno	string	N	N	N

Tabla 4-1. Diccionario de datos de las tablas utilizadas como catálogos.

Ahora definamos el diccionario de datos para las tablas donde se controlan las existencias. En la Tabla 4-2 encontramos las definiciones para los distintos tipos de existencias que se manejan.

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	Es FK
EXISTENCIA	institución	entero	N	S	S
	plaza	<i>string</i>	N	S	S
	bóveda	entero	N	S	S
	estado físico	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	<i>money</i>	N	N	N
EXISTENCIA PRODUCCIÓN	estado físico	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	<i>money</i>	N	N	N
EXISTENCIA TRÁNSITO	transporte	entero	N	S	S
	estado físico	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	<i>money</i>	N	N	N
PIEZA DESTRUIDA	fabricante	entero	N	S	S
	estado físico	entero	N	S	S
	método destrucción	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	<i>money</i>	N	N	N

Tabla 4-2. Diccionario de datos para las tablas utilizadas en el control de existencias.

En la Tabla 4-3 encontramos el diccionario de datos para las tablas donde se controlan los permisos sobre los servicios, tanto para los grupos como para los usuarios. De igual forma, En la Tabla 4-4 encontramos el diccionario de datos para el conjunto de tablas relacionadas con el control de las operaciones: la fecha de registro, el folio del sistema y la bitácora.

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	Es FK
INTEGRANTE GRUPO USUARIO	grupo usuario	entero	N	S	S
	usuario	entero	N	S	S
DERECHO GRUPO USUARIO	grupo usuario	entero	N	S	S
	servicio	entero	N	S	S
	tipo servicio	entero	N	S	S
DERECHO USUARIO	usuario	entero	N	S	S
	servicio	entero	N	S	S
	tipo servicio	entero	N	S	S
	tipo excepción	carácter	N	N	N

Tabla 4-3. Diccionario de datos de las tablas utilizadas para controlar permisos.

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
FECHA REGISTRO	fecha registro	fecha - hora	N	S	N
	fecha vda	fecha - hora	N	N	N
	fecha vds	fecha - hora	N	N	N
	hora apertura	fecha - hora	N	N	N
	hora cierre	fecha - hora	N	N	N
FOLIO SISTEMA	folio sistema	entero	N	S	N
BITÁCORA	fecha hora	fecha - hora	N	S	N
	terminal	entero	N	S	S
	usuario	entero	N	S	S
	servicio	entero	N	N	N
	tipo servicio	entero	N	N	S

Tabla 4-4. Diccionario de datos para las tablas de control de operaciones

En la Tabla 4-5 encontramos el diccionario de datos de todas las tablas dónde se almacenan las operaciones del día.

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
OPERACIÓN	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	terminal	entero	N	N	S
	folio terminal	entero	N	N	N
	login	<i>string</i>	N	N	N
	tipo efectivo	entero	N	N	N
	tipo movimiento	carácter	N	N	N
	tipo operación	entero	N	N	S
	importe	<i>money</i>	N	N	N
	fecha aplicación	fecha - hora	N	N	N
	fecha proceso	fecha - hora	S	N	N
	estado proceso	carácter	N	N	N
	AUTORIZACIÓN OPERACIÓN	número autorización	<i>string</i>	N	S
fecha autorización		fecha - hora	N	N	N
fecha registro		fecha - hora	S	N	S
folio sistema		entero	S	N	S
folio terminal		entero	N	N	N
importe total		<i>money</i>	N	N	N
login autoriza		<i>string</i>	N	N	N
login solicita		<i>string</i>	N	N	N
motivo autorización		<i>string</i>	N	N	N
tipo operación		entero	N	N	S
COMENTARIO	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	comentario	<i>string</i>	N	N	N

continúa ...

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
AUTORIZACIÓN	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	estado físico 1	entero	N	N	S
	estado físico 2	entero	N	N	S
	espécimen	entero	N	N	S
	fecha autorización	fecha - hora	N	N	N
	fecha emisión	fecha - hora	N	N	N
	número piezas	money	N	N	N
ORDEN FABRICACIÓN	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	fabricante	entero	N	N	S
	estado físico	entero	N	N	S
	espécimen	entero	N	N	S
	fecha emisión	fecha - hora	N	N	N
	fecha orden	fecha - hora	N	N	N
	número orden	entero	N	N	N
número piezas	entero	N	N	N	
ENTREGA EFECTIVO	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	fabricante	entero	N	N	S
	estado físico origen	entero	N	N	S
	institución	entero	N	N	S
	plaza destino	string	N	N	S
	bóveda destino	entero	N	N	S
	estado físico destino	entero	N	N	S
	espécimen	entero	N	N	S
	número acta	string	N	N	N
	fecha emisión	fecha - hora	N	N	N
	fecha entrega	fecha - hora	N	N	N
número piezas	money	N	N	N	

continúa ...

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
TRASPASO	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	institución	entero	N	N	S
	plaza	string	N	N	S
	bóveda origen	entero	N	N	S
	estado fisico origen	entero	N	N	S
	bóveda destino	entero	N	N	S
	estado fisico destino	entero	N	N	S
importe total	money	N	N	N	
D TRASPASO	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	money	N	N	N
REMESA	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	institución origen	entero	N	N	S
	plaza origen	string	N	N	S
	bóveda origen	entero	N	N	S
	institución destino	entero	N	N	S
	plaza destino	string	N	N	S
	bóveda destino	entero	S	N	S
	grupo operación	entero	N	N	N
	fecha embarque	fecha - hora	N	N	N
	transporte	entero	N	N	S
	número orden	string	N	N	N
importe total	money	N	N	N	

continúa ...

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
D REMESA	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	estado fisico origen	entero	N	S	S
	estado fisico destino	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	money	N	N	N
CONFIRMACIÓN REMESA	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	número orden	string	N	N	N
	bóveda	entero	N	N	S
	fecha registro remesa	fecha - hora	N	N	S
	folio sistema remesa	entero	N	N	S
	importe total	money	N	N	N
DEPÓSITO RETIRO	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	institución	entero	N	N	S
	plaza	string	N	N	S
	bóveda	entero	N	N	S
	estado físico	entero	N	N	S
	banco usuario	entero	N	N	S
	grupo operación	entero	N	N	N
	importe total	money	N	N	N
D DEPÓSITO RETIRO	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	money	N	N	N

continúa ...

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
DESTRUCCIÓN	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	bóveda	entero	N	N	S
	institución	entero	N	N	S
	plaza	<i>string</i>	N	N	S
	método destrucción	entero	N	N	S
	fecha destrucción	fecha - hora	N	N	N
	importe total	<i>money</i>	N	N	N
D DESTRUCCIÓN	fecha registro	fecha - hora	N	S	S
	folio sistema	entero	N	S	S
	fabricante	entero	N	S	S
	estado físico origen	entero	N	S	S
	espécimen	entero	N	S	S
	número piezas	<i>money</i>	N	N	N

Tabla 4-5. Diccionario de datos de las tablas dónde se almacenan las operaciones.

En la Tabla 4-6 encontramos el diccionario de datos de las tablas utilizadas en las validaciones de las reglas del negocio.

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
RESTRICCIÓN EXISTENCIA	institución	entero	N	S	S
	plaza	<i>string</i>	N	S	S
	bóveda	entero	N	S	S
	estado físico	entero	N	S	S
	tipo efectivo	entero	N	N	N

continúa ...

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
VALIDACIÓN OPERACIÓN	tipo operación	entero	N	S	S
	índice	entero	N	S	N
	tipo validación	carácter	N	N	N
	tipo sitio origen	entero	S	N	N
	institución origen	entero	S	N	S
	plaza origen	string	S	N	S
	bóveda origen	entero	S	N	S
	estado físico origen	entero	S	N	S
	tipo sitio destino	entero	S	N	N
	institución destino	entero	S	N	S
	plaza destino	string	S	N	S
	bóveda destino	entero	S	N	S
	estado físico destino	entero	S	N	S
	fecha actualización	fecha – hora	N	N	N
VALIDACIÓN OPERACIÓN ATRIB	grupo validación	entero	N	S	N
	tipo sitio origen	carácter	S	N	N
	institución origen	carácter	S	N	N
	plaza origen	carácter	S	N	N
	bóveda origen	carácter	S	N	N
	estado físico origen	carácter	S	N	N
	tipo sitio destino	carácter	S	N	N
	institución destino	carácter	S	N	N
	plaza destino	carácter	S	N	N
	bóveda destino	carácter	S	N	N
	estado físico destino	carácter	S	N	N

Tabla 4-6: Diccionario de datos para tablas de validación de las reglas del negocio.

Por último, en la Tabla 4-7 encontramos el diccionario de datos de las tablas utilizadas para el control y la configuración del sistema.

Tabla	Columna	Tipo de Dato	Acepta Nulos	Forma la PK	ES FK
CONSTANTE	constante	<i>string</i>	N	S	N
	valor	<i>string</i>	N	N	N
	tipo	<i>string</i>	N	N	N
CONTROL	control	entero	N	S	N
	descripción	<i>string</i>	N	N	N
	estado	carácter	N	N	N
	fecha hora	fecha – hora	S	N	N
	sp asociado	<i>string</i>	S	N	N
REGLA	regla	<i>string</i>	N	S	N
	caso regla	<i>string</i>	N	S	N
	valor regla	<i>string</i>	N	N	N
	tipo dato	<i>string</i>	N	N	N

Tabla 4-7. Diccionario de datos para las tablas de configuración y control.

4.5 Relación con otros sistemas

Los sistemas de flujos de caja están íntimamente relacionados con los sistemas contables de los Bancos Centrales. Por tanto, aquellas operaciones que se tengan que reflejar en la contabilidad es necesario enviarlas. En este sentido podemos imaginar dos escenarios para alimentar el sistema contable, el primero consiste en generar los movimientos contables dentro del sistema de flujos de caja y enviárselos en forma de una póliza, el segundo es enviar las operaciones. Esta decisión compete directamente a los Bancos Centrales, sin embargo, creemos que dentro del sistema de flujos de caja no deben generarse los movimientos, ya que el objetivo del sistema es controlar existencias, el generar los asientos contables involucraría "contaminar" el sistema con funciones no propias de él. Por otra parte, al igual que el detalle de las operaciones es importante para los sistemas de flujos de caja, también lo es para los sistemas contables. Un sistema contable puede contener un sin fin de movimientos pero es más completo si tiene las operaciones que los generaron.

Es probable que no requieran toda la información que utiliza el sistema de flujos de caja, pero dada la necesidad de reflejar las operaciones en la contabilidad, la información de las operaciones cubren este requerimiento.

Para enviar en forma adecuada las operaciones al sistema contable es probable que se requieran catálogos que permitan encontrar una clave equivalente en el sistema contable, por ejemplo, en los sistemas de flujos de caja existen especímenes que identifican cada tipo de pieza, incluso por los materiales que la componen, este catálogo satisface las necesidades del área involucrada, sin embargo, uno o varios especímenes pueden agruparse en un espécimen distinto en el sistema contable. Para enviar las operaciones correctamente a la contabilidad podemos crear una tabla que contenga las equivalencias entre ambos catálogos y enviar la operación utilizando las claves equivalentes.

Una forma de reducir el número de tablas de equivalencia consiste en utilizar catálogos institucionales, donde se almacenen las claves de aquellos catálogos de interés general, por ejemplo, el catálogo de instituciones, de plazas, de divisas, de sucursales, etc. Si todos los sistemas utilizan estas tablas institucionales, todos los sistemas estarán "hablando el mismo idioma". Además, el utilizar información común permite distribuir las responsabilidades entre las diferentes áreas de la institución y mantiene actualizados todos los sistemas.

Si cada sistema utiliza su propio catálogo, cada administrador tiene que realizar funciones de actualización, lo que genera una duplicidad de funciones. Por el contrario, si el área encargada de actualizar la información que le concierne, sólo un área en la institución realiza esta función y todos los sistemas tienen información veraz, por ejemplo, si por la naturaleza de un área, tienen a su cargo la actualización del catálogo de instituciones, al suceder un cambio, ya sea una creación de alguna institución o una desaparición, esta área realiza el cambio en un repositorio central y, por tanto, la información institucional está actualizada.

A pesar de estar centralizada la información, las responsabilidades están distribuidas, ya que no necesariamente una sola área tendría que mantener todos los catálogos, al contrario, el área que posea la información se encargará de mantenerla en cuanto se presenten los cambios.

Con un esquema de esta naturaleza los sistemas podrían utilizar las tablas del repositorio central para resolver sus consultas. Asumamos que este repositorio es una base de datos institucional.

Los sistemas podrían utilizar directamente las tablas institucionales, sin embargo, representa un riesgo para sistemas críticos depender directamente de otra base de datos, porque si en algún momento esta base de datos no está disponible, el sistema no podría continuar su operación normal. Para evitar esta situación recomendamos que cada sistema realice una copia de la información institucional que necesite dentro de su propia base de datos, esto obviamente, genera duplicidad de información pero ofrece independencia operativa.

Así, los administradores de cada sistema realizarían la actualización de sus catálogos institucionales con la periodicidad que requiera, evitando duplicar la

función de mantener los catálogos si asumimos que la información institucional es definitiva.

En nuestro diseño, la información que identificamos como institucional, porque no la consideramos exclusiva de los sistemas de flujos de caja, es la relativa a:

- Instituciones.
- Plazas.
- Sucursales.
- Días inhábiles.
- Divisas.
- Una parte del catálogo de errores.
- Terminales.
- Usuarios.

Los catálogos de Instituciones, Plazas, Sucursales, Días inhábiles y Divisas, son utilizados por varias áreas de un Banco Central en su operación. También se pueden definir errores que se presentan en una gran cantidad de sistemas y tipificarlos en un catálogo único, por ejemplo, "institución inexistente", el cual lo pueden utilizar todos los sistemas que utilizan instituciones, o el error "fuera de horario" o "terminal inexistente", etc. Los errores generales y los específicos del sistema forman el catálogo de errores del sistema.

En cuanto terminales y usuarios el esquema es más complejo, podemos asumir que una terminal o un usuario puede ser reconocido dentro de uno o varios sistemas, pero difícilmente será usuario de todos los sistemas. Si pensamos que tanto una terminal como un usuario existen en forma única y un modelo relacional es una representación del mundo real, no es conveniente asignar una clave distinta a un objeto en diferentes sistemas, en su lugar podemos asignar una clave única a cada elemento y relacionarla con los sistemas que utiliza. De esta forma la actualización de terminales y usuarios no sería directamente de las tablas institucionales donde se almacene, teniendo que auxiliarnos en unas tablas que indiquen que elementos hay que copiar a la base de datos de cada sistema.

Una vez diseñado el modelo entidad – relación y planteadas las consideraciones relativas al manejo de información institucional procederemos a la codificación del sistema.

5 Codificación

En este capítulo se muestran los lineamientos de codificación seguidos y las partes más relevantes del código generado.

5.1 Lineamientos de codificación

Como este trabajo sólo abarca la parte del servidor, es necesario definir la nomenclatura utilizada y los lineamientos bajo los cuales se realizó la codificación de los servicios, *triggers* y demás objetos relativos a la base de datos.

Nomenclatura

Para distinguir los distintos objetos de la base de datos utilizamos la siguiente nomenclatura:

- El nombre de las tablas está en mayúsculas.
- El nombre de los atributos está en minúsculas.
- El nombre de los *triggers* está formado por el nombre de la tabla anteponiéndole una "t" seguida de una letra que indica la acción¹ y un guión bajo, por ejemplo el *trigger* de inserción de la tabla EXISTENCIA se llama ti_EXISTENCIA.
- El nombre de los *stored procedures* comienzan con spsif seguido de un guión bajo y del nombre del procedimiento. Si es un procedimiento de uso interno se agrega una "i" antes del primer guión bajo, por ejemplo, un procedimiento que puede invocar el usuario sería spsif_consulta_sitios y uno de uso interno sería spsifi_valida.

¹ Se utiliza "i" para la inserción, "u" para la modificación y "d" para la eliminación.

- El nombre de las transacciones indica su función global, por ejemplo “*escribe_operación*”.
- Dentro de cada transacción, antes de realizar una afectación a una tabla se coloca un *save point* cuyo nombre está formado por el prefijo “tr_” y el nombre de la tabla, por ejemplo, tr_EXISTENCIA.
- El nombre de las variables que almacenan el valor de una columna, tienen el mismo nombre de la columna.

Lineamientos

- La única forma de que el usuario puede interactuar con el sistema es a través de *stored procedures*, que como se mencionó anteriormente nos permiten optimizar el código a fin de obtener mejores tiempos de respuesta, también permiten tener más control sobre los renglones a devolver, así como realizar las validaciones de ejecución pertinentes. En cuanto a seguridad, los *stored procedures* mantienen encapsulado el código por lo que las tablas utilizadas no tienen que ser del conocimiento del usuario.
- Todos los *stored procedures* devuelve como código de retorno (*return status*) un cero cuando haya terminado correctamente, de lo contrario devuelve el código de error correspondiente. Si el procedimiento devuelve valores en las variables de salida, en caso de error, estas variables están iniciadas con valores nulos.
- En los procedimientos de envío de operaciones se devuelve la fecha de registro y el folio asignado por el sistema en variables de salida.
- Solamente en el servidor, dentro de los *stored procedures*, se abren y cierran las transacciones, reduciendo así el tiempo de contención en las tablas. El cliente jamás podrá abrir una transacción desde su aplicación.
- Cuando en varios procedimientos se afectan un conjunto similar de tablas, se procura efectuarlas siguiendo el mismo orden a fin de evitar “*dead locks*”.
- Las validaciones de integridad referencial y el procesamiento de las operaciones son realizadas en los *triggers*, a fin de evitar errores si el *dbo* (*database owner*, dueño de la base de datos) inserta una operación sin utilizar un *stored procedure*.
- Se procura, dentro de lo posible, reutilizar *stored procedures* de uso interno dentro de los *triggers* y de otros *stored procedures*.

- A fin de facilitar la asignación de permisos de ejecución de los *stored procedures* a los usuarios, todos los objetos (tablas, *triggers*, *stored procedures*, etc.) son creados por el *dbo*.
- Se cuenta con grupos de seguridad en la base de datos, de tal forma que ningún usuario está fuera de algún grupo. Los privilegios de ejecución están asignados a los grupos.
- Toda la información institucional se obtiene, vía *stored procedures*, de la base de datos donde reside.

Una vez definidos los lineamientos y la nomenclatura utilizada veamos como está estructurado el plan de codificación.

5.2 Estructura de la codificación

Existen partes funcionales que son utilizadas en una gran cantidad de objetos dentro de la base de datos, por ejemplo la validación de horario que se realiza en todos los procedimientos a los cuales tiene acceso el usuario. Estas partes funcionales las llamaremos en lo sucesivo procedimientos comunes y pueden ser invocadas desde cualquier *stored procedure* o *trigger*. Por claridad, dividiremos estas partes comunes en procedimientos de validación y procedimientos de extracción de valores constantes o reglas.

Los procedimientos de validación verifican el horario del servicio, los privilegios del usuario, el permiso de interactuar con el servidor, etc. Los procedimientos de extracción de constantes y reglas, proporcionan los valores solicitados en el tipo de dato adecuado, por ejemplo, la obtención del tipo de sitio "sucursal" o el tipo de efectivo "billete". Estos valores son complementos para realizar las validaciones específicas y/o el procesamiento de las operaciones. Como se planteó en el capítulo anterior, se pudieron generar los objetos "reglas" y asociarlos a las columnas, sin embargo, se optó por mantener un esquema alterno que permita devolver un mensaje de error más amigable para el usuario.

Otro tipo de procedimientos son los de interacción, a través de ellos los usuarios envían operaciones al servidor, consultan la información, ya sea de las operaciones o de las existencias y obtienen sus reportes. Todos los procedimientos requieren como parámetros de entrada el número de terminal, la clave de la institución y la clave de la plaza, estos últimos para distinguir el sitio de consulta, adicionalmente, cuando un mismo *stored procedure* puede resolver varios servicios, se envía la clave del servicio y el tipo de servicio, mismos que se utilizan para la validación de permisos y el registro en la bitácora.

Cuando una parte de un *stored procedure* puede ser reutilizada en otros procedimientos, se genera un *stored procedure* auxiliar el cual contiene el código reutilizable.

Como el procesamiento de las operaciones se realiza en los *triggers*, dentro de este código se insertan los renglones faltantes a las tablas de existencias, siempre y cuando cumplan con las validaciones establecidas en RESTRICCIÓN EXISTENCIA.

Ahora veamos las partes más relevantes del código generado.

5.3 Codificación

Comencemos con los procedimientos de uso común. Los primeros procedimientos importantes son los encargados de validar privilegios y horarios de los servicios. Existen dos procedimientos que realizan estas funciones, el primero valida los servicios de consultas y reportes y el segundo tiene una ampliación para validar que la transmisión esté habilitada y que el sitio del movimiento esté abierto. Para reutilizar el código, el segundo procedimiento realiza sus validaciones adicionales y ejecuta el primer procedimiento de validación.

A continuación se muestra el código `spsifi_valida`, que es el procedimiento encargado de realizar las validaciones de horarios y permisos sobre el servicio.

```
create proc spsifi_valida
    @terminal      smallint,
    @institucion  int,
    @plaza        char(5),
    @servicio     smallint,
    @tipo_servicio tinyint
as
declare @status    int, @general    int, @fecha_hora  datetime,
        @usuario   int, @control   int, @estado     char(1),
        @habilitado int, @serv_cap  int

declare @constante varchar(30), @valor_cte  varchar(30),
        @val_regla  varchar(10)

/*
** -----
** Se obtiene la fecha y hora del Servidor y las constantes
** requeridas
** -----
*/

select @fecha_hora =getdate()
```

CAPÍTULO 5. Codificación

```
select @constante ='SISTEMA_HABILITADO'

exec @status =spsifi_da_valor_constante @constante,
    @valor_cte output
    if (@status !=0) return @status
select @habilitado =convert(int, @valor_cte)

/*
** -----
** Se valida que el sistema esté habilitado
** -----
*/

exec @status =spsifi_da_edo_CONTROL @habilitado, @estado output
    if (@status !=0) return @status

if (@estado !='S')
begin
/*
** -----
** Se obtienen los TIPOS DE SERVICIO que si pueden
** realizarse cuando el sistema está deshabilitado
** -----
*/

    exec @status =spsifi_regla_tipo_servicio 'CONTROL',
        @control output
        if (@status !=0) return @status

    exec @status =spsifi_regla_tipo_servicio 'GENERAL',
        @general output
        if (@status !=0) return @status

    if (@tipo_servicio not in (@control, @general))
        return 41139
end

/*
** -----
** Se hacen las validaciones pertinentes
** -----
*/

exec @status =spsifi_valida_derecho_usuario @servicio,
    @tipo_servicio,
    @usuario output
    if (@status !=0) return @status

exec @status =spsifi_valida_terminal @terminal, @institucion,
    @plaza
    if (@status !=0) return @status
```

CAPÍTULO 5. Codificación

```
exec @status =spsifi_valida_horarios @servicio, @tipo_servicio,
                                     @fecha_hora
    if (@status !=0) return @status

/*
** -----
** Si todo sale bien Bitacoriza
** -----
*/

begin tran val_escribe_bitacora
    save tran trBITACORA
    insert into BITACORA values (@usuario, getdate(), @terminal,
                                @servicio, @tipo_servicio)

    select @status =@@error
        if (@status !=0)
            begin
                rollback val_escribe_bitacora
                return @status
            end
commit tran val_escribe_bitacora

return 0
```

Ahora veamos el código del procedimiento encargado de validar la inserción de operaciones

```
create proc spsifi_valida_envio
    < parámetros de entrada >
as
    < declaración de variables >

/*
** -----
** Valida que la Transmisión esté habilitada
** -----
*/

select @constante = 'TRANSMISION_HABILITADA'
exec @status =spsifi_da_valor_constante @constante,
                                     @valor_cte output
    if (@status !=0) return @status
select @transmision =convert(int, @valor_cte)
```

CAPÍTULO 5 Codificación

```
exec @status =spsifi_da_edo_CONTROL @transmision, @estado output
    if (@status !=0) return @status

if (@estado !='S') return 41141

exec @status =spsifi_regla_estado_sitio 'CERRADO',
    @cerrado output
    if (@status !=0) return @status-

exec @status =spsifi_sitio_cerrado @institucion, @plaza,
    @estado_sitio output
    if (@status !=0) return @status

if (@estado_sitio =@cerrado) return 41212

/*
** -----
** Inicializa Variables
** -----
*/

select @grupo_operacion =grupo_operacion,
    @tipo_afectacion =tipo_afectacion,
    @grupo_validacion =grupo_validacion,
    @restriccion_validacion =restriccion_validacion
from TIPO_OPERACION
where tipo_operacion =@tipo_operacion

if (@grupo_operacion is NULL) return 41066

/*
** -----
** Se obtiene el tipo Servicio y se validan derechos de usuario
** -----
*/
exec @status =spsifi_regla_tipo_servicio 'CAPTURA',
    @t_servicio output
    if (@status !=0) return @status

exec @status =spsifi_valida @terminal, @institucion, @plaza,
    @tipo_operacion, @t_servicio
    if (@status !=0) return @status

/*
** -----
** Valida que el folio terminal no exista con la misma fecha en
** la tabla OPERACION. Si el movimiento por insertar es alta.
** -----
*/
```

CAPÍTULO 5. Codificación

```
if (@tipo_movto = 'A')
begin
  if exists (select * from OPERACION
             where fecha_registro = @fecha_registro
                and folio_terminal = @folio_term)
    return 41190

  if exists (select * from OPERACION
             where folio_terminal   =@folio_term
                and fecha_aplicacion =@fecha_aplicacion)
    return 41235
end

return 0
```

Veamos a continuación como se validan los horarios de los servicios.

```
create proc spsifi_valida_horarios
           @servicio      int,
           @tipo_servicio tinyint,
           @fecha_hora    datetime
as
declare @status int

declare @hora_apertura  datetime, @referencia_inicio char(1),
        @hora_cierre    datetime, @minutos_inicio      int,
        @hora_inicio    datetime, @referencia_fin      char(1),
        @hora_fin       datetime, @minutos_fin         int,
        @valida_horario char(1), @fecha_vmd           datetime

/*
** -----
** Obtiene la hora de apertura y cierre de FECHA_REGISTRO
** -----
*/

exec @status=spsifi_da_fecha_vmd_hape_hcie @fecha_vmd      output,
                                           @hora_apertura  output,
                                           @hora_cierre    output

if (@status !=0) return @status

/*
** -----
** Se obtiene las referencias que delimitan el horario
** -----
*/
```

CAPÍTULO 5. Codificación

```
select @valida_horario    =restriccion_horario,
       @referencia_inicio =referencia_inicio,
       @minutos_inicio    =minutos_inicio,
       @referencia_fin     =referencia_fin,
       @minutos_fin       =minutos_fin
from SERVICIO
where servicio            =@servicio
   and tipo_servicio     =@tipo_servicio

if (@referencia_inicio is NULL) return 30024

/*
** -----
** Si (@valida_horario = 'N') no es necesario validar horarios
** así que termina, de lo contrario verifica que esté dentro del
** horario permitido. Utiliza la fecha y hora de la máquina.
** -----
*/

if (@valida_horario = 'N') return 0

if (@referencia_inicio = 'A')
    select @hora_inicio =dateadd (mi, @minutos_inicio,
                                   @hora_apertura)
else select @hora_inicio =dateadd (mi, @minutos_inicio,
                                   @hora_cierre)

if (@referencia_fin = 'A')
    select @hora_fin =dateadd (mi, @minutos_fin,
                               @hora_apertura)
else select @hora_fin =dateadd (mi, @minutos_fin,
                               @hora_cierre)

if (@fecha_hora not between @hora_inicio and @hora_fin)
    return 32002

return 0
```

Veamos como se verifica que el usuario tenga derecho sobre el servicio solicitado.

```
create proc spsifi_valida_derecho_usuario
    @servicio        int,
    @tipo_servicio   tinyint,
    @usuario         int        output,
    @institucion     int        output
as
```

```

/*
** -----
** Valida que el usuario indicado tenga derecho
** a ejecutar el servicio referenciado.
** -----
*/

declare @status int,
        @valido_para_todos char(1)

select @valido_para_todos =valido_para_todos
      from SERVICIO
     where servicio      =@servicio
        and tipo_servicio =@tipo_servicio

if (@valido_para_todos is NULL) return 40004

select @usuario      =usuario,
       @institucion =institucion
      from USUARIO
     where login =suser_name()

/*
** -----
** Si el usuario no está registrado marca error.
** -----
*/

if (@usuario is NULL)
    return 30136

/*
** -----
** Si el servicio es válido para todos termina
** -----
*/

if (@valido_para_todos = 'S') return 0

if not exists (select d.*
              from INTEGRANTE_GRUPO_USUARIO i,
                   DERECHO_GRUPO_USUARIO d
             where i.usuario      =@usuario
                and d.servicio    =@servicio
                and d.tipo_servicio =@tipo_servicio
                and d.grupo_usuario =i.grupo_usuario)

```

```

begin
/*
** -----
** Los grupos a los que pertenece el usuario no tienen
** autorización para ejecutar el @servicio, por lo que se
** valida si @usuario tiene derecho a ello en forma individual
** -----
*/
if not exists (select * from DERECHO_USUARIO
               where usuario      =@usuario
                  and servicio    =@servicio
                  and tipo_servicio =@tipo_servicio)
    return 40006 /* El usuario no esta autorizado
                ** para ejecutar este servicio */
end

return 0

```

Una variante en las validaciones es cuando el *stored procedure* está asociado en forma única a un servicio, en este caso no se envía el número de servicio porque puede ser obtenido por el servidor. Veamos el código de este procedimiento

```

create proc spsifi_val_stored_procedure
    < parámetros de entrada >
as
< declaración de variables >

select @servicio =servicio
    from SP_SERVICIO
    where stored_procedure =@stored_procedure
        and tipo_servicio =@tipo_servicio

if (@servicio is NULL)    return 40004

exec @status =spsifi_valida_bitacora @terminal, @institucion,
                                     @plaza,      @servicio,
                                     @tipo_servicio

    if (@status !=0)    return @status

return 0

```

Ahora veamos como son utilizados estos procedimientos dentro de los procedimientos que puede ejecutar el usuario. A continuación mostramos un procedimiento que resuelve la consulta de existencias, dado que el usuario puede interesarle algunos elementos se le facilitó el establecimiento de acotamientos opcionales, el usuario puede enviar valores en algunos parámetros que reduzcan el número de renglones que devuelve el *stored procedure*.

CAPÍTULO 5. Codificación

```
create proc spsif_cons_EXISTENCIA
    <parámetros de entrada >
as
/*
** -----
** Para siempre devolver el encabezado de las columnas, en caso
** de error se invoca a spsifi_cons_existencias_vacia y después
** se termina con el código de error en @status
** -----
*/
< declaración de variables >

exec @status =spsifi_regla_tipo_servicio 'REPORTE', @tserv output
    if (@status !=0)
        begin
            <manejo de errores >
            return @status
        end

exec @status =spsifi_val_stored_procedure
    @terminal, @institucion, @plaza,
    @nombre_sp, @tserv

    if (@status !=0)
        begin
            <manejo de errores_>
            return @status
        end

select @status,      s.tipo_sitio,      ex.institucion,      ex.plaza,
    ex.boveda,      ex.estado_fisico,      esp.tipo_efectivo,
    esp.divisa,      ex.especimen,      ex.numero_piezas,
    Importe =ex.numero_piezas * convert(money, esp.valor)
from EXISTENCIA ex, ESPECIMEN esp, SITIO s
where esp.especimen =ex.especimen
    and s.institucion =ex.institucion
    and s.plaza =ex.plaza
    and ((esp.tipo_efectivo =@tipo_efectivo) or
        (@tipo_efectivo is null))
    and ((esp.divisa =@divisa ) or (@divisa is null))
    and ((s.tipo_sitio =@tipo_sitio ) or
        (@tipo_sitio is null))
    and ((ex.institucion =@institucion ) or
        (@institucion is null))
    and ((ex.plaza =@plaza ) or (@plaza is null))
    and ((ex.boveda =@boveda ) or (@boveda is null))
    and ((ex.estado_fisico =@edo_fis ) or (@edo_fis is null))
    and ((ex.especimen =@especimen ) or (@especimen is null))

return 0
```

CAPÍTULO 5. Codificación

Una operación puede tener varios detalles, por ejemplo, un traspaso, en un movimiento de este tipo pueden incluirse diversas cantidades de diversos especímenes, dado que no es posible conocer en este momento cuántos detalles contiene una operación, es necesario definir un protocolo que permita insertar varios detalles sin definir demasiados parámetros. El protocolo consiste en enviar los detalles concatenados, utilizando como separador un tabulador. El número de elementos que forma la cadena depende del número de columnas que tenga la tabla donde se almacena esta información. Por otro lado, dado que la lógica de procesamiento está en los *triggers*, para cancelar una operación es factible insertar un nuevo renglón para que el procesamiento sea siempre de la misma forma, de esta forma, además, se cuenta con el historial de una operación, lo mismo sucede al ser modificada, siempre se tendrá el rastro de todas las modificaciones que se realicen a una operación. A continuación se muestra un ejemplo de cómo se codificó un procedimiento de inserción.

```
create proc spsif_envio_de_traspaso
    < parámetros de entrada >
as
/*
** -----
** Actúa de acuerdo al @tipo_movimineto de la siguiente manera:
**
** Alta          : sólo ejecuta inserción del nuevo registro.
** Cancelación  : sólo ejecuta cancelación del registro
**               indicado en los parámetros output del sp
** Modificación:
**   - Ejecuta cancelación
**   - Inserta el nuevo registro de alta para la modificación
** -----
*/
<declaración de variables>

/*
** -----
** Se crean las tablas temporales empleadas en todos los
** procedimientos antes mencionados, ya que estas no pueden ser
** creadas una vez que se ha iniciado una transacción.
** -----
*/

create table #traspaso      (registro      varchar(15),
                           renglon       numeric(7,0)  identity)
create table #detalle_trasp (especimen    smallint,
                           num_piezas   money,
                           renglon     numeric(7,0)  identity)
create table #detalle     (especimen    smallint,
                           num_piezas   money)
```

CAPÍTULO 5. Codificación

```
/*
** -----
** INICIA LA TRANSACCION **
** -----
*/

begin tran envio_traspaso

if (@tipo_movto in ('C','M')) /* verifica si se cancela */
begin

save tran cancela_traspaso

exec @status =spsifi_cancela_traspaso
           @fecha_recepcion output, @folio_sistema output,
           @terminal output, @fol_term_alta,
           @fol_term_orig output, @comentario,
           @tipo_movto, @folio_sistema_orig

if (@status !=0)
begin
<manejo de errores>
return @status
end

end

if (@tipo_movto in ('A','M')) /* Si hay alta que insertar */
begin
select @fecha_recepcion =NULL,
       @folio_sistema =NULL

save tran inserta_traspaso

exec @status =spsifi_inserta_traspaso <parámetros>
if (@status !=0)
begin
< manejo de errores >
return @status
end

end

commit tran envio_traspaso

drop table #traspaso
drop table #detalle_trasp
drop table #detalle

return 0
```

Ahora veamos como es que se lleva a cabo la escritura en las tablas.

```
create proc spsifi_inserta_op_traspaso
    <parámetros de entrada>
as
< declaración de variables >

select @fecha_recepcion =fecha_vmd
    from FECHA_REGISTRO

/*
** -----
** Validación de datos de Operación
** -----
*/

exec @status =spsifi_valida_envio < parámetros >
    if (@status !=0)    return @status

if ((@detalle1 = ' ') or (@detalle1 is NULL))
    select @detalle1 =NULL

if ((@detalle2 = ' ') or (@detalle2 is NULL))
    select @detalle2 =NULL

if ((@detalle1 is NULL) and (@detalle2 is NULL))
    return 41011

while (@detalle1 is not NULL)
    begin
        exec @status =spsifi_desconcatena_x_tab    @detalle1 output,
                                                    @elemento output

        insert into #traspaso
        select rtrim(ltrim(@elemento)) where @elemento !=' '
    end

while (@detalle2 is not NULL)
    begin
        exec @status =spsifi_desconcatena_x_tab    @detalle2 output,
                                                    @elemento output

        insert into #traspaso
        select rtrim(ltrim(@elemento)) where @elemento !=' '
    end
```

CAPÍTULO 5. Codificación

```
select @num_elementos =max (renglon) from #traspaso

if (@num_elementos %2 !=0)
begin
  < manejo de errores >
  return 41012
end

select @i =1
while (@i < @num_elementos)
begin
  /*
  ** Valida que los campos numéricos no contengan caracteres
  ** diferentes a números
  */
  if exists (select * from #traspaso
            where registro like '%[^0-9]%' and renglon =@i)
  begin
    < manejo de errores >
    return 32224
  end

  select @especimen =convert(smallint, registro) from #traspaso
  where renglon =@i

  select @num_piezas =convert(money,      registro) from #traspaso
  where renglon =@i+1

  insert into #detalle_trasp values (@especimen, @num_piezas)

  exec @status =spsifi_prep_renglon_EXISTENCIA
          @institucion, @plaza, @boveda_orig,
          @edo_fis_orig, @especimen
  if (@status !=0)
  begin
    < manejo de errores >
    return @status
  end

  exec @status =spsifi_prep_renglon_EXISTENCIA
          @institucion, @plaza, @boveda_dest,
          @edo_fis_dest, @especimen
  if (@status !=0)
  begin
    < manejo de errores >
    return @status
  end

  select @i =@i +2
end
```

CAPÍTULO 5. Codificación

```
select @suma_importes =sum(d.num_piezas * e.valor)
  from ESPECIMEN e, #detalle_trasp d
 where e.especimen =d.especimen

if (@suma_importes !=@importe)
  begin
    < manejo de errores >
    return 41019
  end

/*
** -----
** Comienzan las actualizaciones
** -----
*/

save tran trFOLIO_SISTEMA

update FOLIO_SISTEMA
  set folio_sistema =folio_sistema +1

select @status =@@error
  if (@status !=0)
    begin
      < manejo de errores >
      return @status
    end
select @folio_sistema =folio_sistema
  from FOLIO_SISTEMA

if (@folio_sistema is NULL)
  begin
    < manejo de errores >
    return 32226
  end

/*
** Se inserta la información general de la operación
*/

save tran trOPERACION

insert into OPERACION values (...)

select @status =@@error
  if (@status !=0)
    begin
      < manejo de errores >
      return @status
    end
```

CAPÍTULO 5. Codificación

```
/*
** Insertamos Encabezado
*/
save tran trTRASPASO
insert into TRASPASO values (...)
select @status =@@error
       if (@status !=0)
           begin
               < manejo de errores >
               return @status
           end

/*
** Insertamos los detalles
*/

save tran trD_TRASPASO

insert into D_TRASPASO
select @fecha_recepcion, @folio_sistema, especimen, num_piezas
       from #detalle_trasp

select @status =@@error
       if (@status !=0)
           begin
               < manejo de errores >
               return @status
           end

delete #traspaso
delete #detalle_trasp

return 0
```

Los procedimientos para las demás operaciones son similares al que acabamos de mostrar.

Veamos ahora como se procesan las operaciones en los *triggers*.

```
create trigger ti_TRASPASO on TRASPASO
for insert
as
begin
< declaración de variables >

/*
** Se verifica que solamente se haya escrito un renglón
*/

select @renglones =@@rowcount
```

CAPÍTULO 5. Codificación

```
if (@renglones !=1)
begin
/* Se hace rollback solamente al save point */
rollback trTRASPASO
raiserror 31211 "Solo se permite insertar un renglón a la vez"
return
end

/*
** Se toman los datos insertados
*/

select @fecha_registro =fecha_registro,
@folio_sistema =folio_sistema,
@institucion =institucion,
@plaza =plaza,
@bodega_origen =bodega_origen,
@edo_fis_origen =estado_fisico_origen,
@bodega_destino =bodega_destino,
@edo_fis_destino =estado_fisico_destino,
@importe_total =importe_total
from inserted

/*
** Se toma información de la tabla OPERACION de la operación
** correspondiente y se valida que no esté procesada
*/

exec @status =spsifi_da_p_procesar_OPERACION <parámetros>
if (@status !=0)
begin
< manejo de errores >
return
end

/*
** Verifica que el tipo de operación capturado sea aceptable para
** la tabla correspondiente según la familia
*/

exec @status =spsifi_val_familia @tipo_operacion, @nombre_tabla
if (@status !=0)
begin
< manejo de errores >
raiserror @status "La operación no es de esta Familia"
return
end
```

CAPÍTULO 5 Codificación

```
/*
** Se verifica que existan las combinaciones sitio - bóveda
** - estado_fisico origen y destino y además que acepte el
** tipo de efectivo de la operación.
*/

exec @status =spsifi_valido_en_RESTRICCION @institucion, @plaza,
                                           @bodega_origen, @edo_fis_origen,
                                           @tipo_efectivo

    if (@status !=0)
        begin
            < manejo de errores >
            return
        end

exec @status =spsifi_valido_en_RESTRICCION @institucion, @plaza,
                                           @bodega_destino, @edo_fis_destino,
                                           @tipo_efectivo

    if (@status !=0)
        begin
            < manejo de errores >
            return
        end

/*
** Se obtiene el tipo de sitio
*/

exec @status =spsifi_da_tipo_sitio @institucion, @plaza,
                                   @tipo_sitio output

    if (@status !=0)
        begin
            < manejo de errores >
            return
        end

/*
** Se valida que el movimiento sea válido según el tipo de
** operación, es decir, se valida que con el tipo de operación
** se permita traspasar del origen al destino.
*/

exec @status =spsifi_movimiento_valido @tipo_operacion,
                                       @tipo_sitio, @institucion,
                                       @plaza, @bodega_origen,
                                       @edo_fis_origen, @tipo_sitio,
                                       @institucion, @plaza,
                                       @bodega_destino, @edo_fis_destino
```

CAPÍTULO 5. Codificación

```
    if (@status !=0)
    begin
        < manejo de errores >
        return
    end
return
end
```

Ahora veamos el trigger de la tabla de detalle.

```
create trigger ti_D_TRASPASO on D_TRASPASO
for insert
as
begin
    < declaracion de variables >

    if (@@rowcount =0) return

    /*
    ** Se valida que todos los detalles sean de una sola operacion
    */

    if (((select count (distinct fecha_registro) from inserted) !=1)
    or
        ((select count (distinct folio_sistema) from inserted) !=1))
    begin
        rollback trD_TRASPASO
        raiserror 41059 "Solo se permite detalles de una transaccion a
        la vez"

        return
    end

    /*
    ** Obtiene la llave de la transacción (fecha_registro y
    **                                     folio_sistema)
    */

    select distinct @fecha_registro =fecha_registro,
                   @folio_sistema =folio_sistema
    from inserted

    if (@fecha_registro is NULL) or (@folio_sistema is NULL))
    begin
        < manejo de errores >
        return
    end
```

CAPÍTULO 5. Codificación

```
/*
** Se valida que no existan especímenes duplicados en el detalle
** de la operación
*/

if exists (select especimen from inserted
           group by especimen having count(*) >1)
begin
  < manejo de errores >
return
end

/*
** Obtiene el encabezado de la operación (TRASPASO) y la
** información general como el tipo de movimiento (Alta /
** Cancelación) y el importe total (OPERACION) y se valida
** que no este procesada previamente.
*/

select @institucion      =institucion,
       @plaza            =plaza,
       @bodega_origen   =bodega_origen,
       @edo_fis_origen  =estado_fisico_origen,
       @bodega_destino  =bodega_destino,
       @edo_fis_destino =estado_fisico_destino,
       @importe_total   =importe_total
from TRASPASO
where fecha_registro    =@fecha_registro
   and folio_sistema   =@folio_sistema

if (@institucion is NULL)
begin
  < manejo de errores >
return
end

select @tipo_movimiento  =tipo_movimiento,
       @tipo_efectivo    =tipo_efectivo,
       @tipo_operacion   =tipo_operacion,
       @divisa           =divisa,
       @importe_operacion =importe,
       @fecha_aplicacion =fecha_aplicacion,
       @estado_proceso   =estado_proceso
from OPERACION
where fecha_registro    =@fecha_registro
   and folio_sistema   =@folio_sistema
```

CAPÍTULO 5. Codificación

```
if (@tipo_operacion is NULL)
begin
  < manejo de errores >
  return
end

if (@estado_proceso != 'I')
begin
  < manejo de errores >
  return
end

/*
** Obtiene el tipo de afectación correspondiente al tipo de
** operación. Si el tipo de movimiento es Cancelación, se debe
** efectuar la afectación contraria
*/

exec @status = spsifi_da_tipo_afectacion
                                @tipo_operacion,
                                @tipo_movimiento,
                                @afecta_elem_1    output,
                                @afecta_elem_2    output,
                                @afecta_elem_3    output

    if (@status != 0)
    begin
      < manejo de errores >
      return
    end

/*
** Valida que todos los especímenes existan en ESPECIMEN y sus
** tipos de efectivo sean coherente con el registrado en la
** operación.
*/

if exists (select i.especimen
           from inserted i, ESPECIMEN E
           where E.especimen = *i.especimen
              and E.tipo_efectivo in (@tipo_efectivo, 3)
           group by i.especimen
           having count (E.especimen) = 0)
begin
  < manejo de errores >
  return
end
```

CAPÍTULO 5 Codificación

```
/*
** Valida que la suma de los importes por especimen sea igual al
** importe total
*/

select @suma_detalle = sum(i.numero_piezas * E.valor)
      from inserted i, ESPECIMEN E
      where E.especimen = i.especimen

if (@suma_detalle !=@importe_operacion)
  begin
    < manejo de errores >
    return
  end

/*
** Se valida que todos los especímenes sean de la misma divisa
** y que su divisa sea la registrada en la operación.
*/

if exists (select E.divisa
          from inserted i, ESPECIMEN E
          where E.especimen = i.especimen
          and E.divisa !=@divisa)
  begin
    < manejo de errores >
    return
  end

/*
** Toma la fecha de proceso del sistema
*/
select @fecha_vmd =fecha_vmd
      from FECHA_REGISTRO

if (@fecha_vmd is NULL)
  begin
    < manejo de errores >
    return
  end

/*
** Verifica que la fecha de aplicación de la transacción sea la
** misma asignada en el servidor como fecha de proceso para ese
** día (fecha_vmd).
*/

if (@fecha_aplicacion !=@fecha_vmd)
  return
```

CAPÍTULO 5. Codificación

```
/*
** -----
** Agrega los registros de existencias que faltan
** iniciándolos con número de piezas igual a 0.
** -----
*/

save tran trEXISTENCIA

insert into EXISTENCIA
select t.institucion, t.plaza, t.boveda_origen,
t.estado_fisico_origen, i.especimen, 0
  from inserted i, EXISTENCIA e, TRASPASO t
 where t.fecha_registro      =i.fecha_registro
    and t.folio_sistema      =i.folio_sistema
    and t.institucion        *=e.institucion
    and t.plaza              *=e.plaza
    and t.boveda_origen      *=e.boveda
    and t.estado_fisico_origen *=e.estado_fisico
    and i.especimen          *=e.especimen
 group by t.institucion, t.plaza, t.boveda_origen,
t.estado_fisico_origen, i.especimen
having count(e.institucion) = 0

select @status = @@error
  if (@status !=0)
  begin
    < manejo de errores >
  return
  end

save tran trEXISTENCIA

insert into EXISTENCIA
select t.institucion, t.plaza, t.boveda_destino,
t.estado_fisico_destino, i.especimen, 0
  from inserted i, EXISTENCIA e, TRASPASO t
 where t.fecha_registro      =i.fecha_registro
    and t.folio_sistema      =i.folio_sistema
    and t.institucion        *=e.institucion
    and t.plaza              *=e.plaza
    and t.boveda_destino     *=e.boveda
    and t.estado_fisico_destino *=e.estado_fisico
    and i.especimen          *=e.especimen
 group by t.institucion, t.plaza, t.boveda_destino,
t.estado_fisico_destino, i.especimen
having count(e.institucion) = 0
```

CAPÍTULO 5. Codificación

```
select @status = @@error
  if (@status !=0)
    begin
      < manejo de errores >
      return
    end

/*
** Actualiza las existencias **
*/

save tran trEXISTENCIA

update EXISTENCIA
  set EXISTENCIA.numero_piezas = E.numero_piezas +
      (i.numero_piezas *
       @afecta_elem_1)

  from inserted i, EXISTENCIA E
 where E.institucion = @institucion
    and E.plaza = @plaza
    and E.boveda = @boveda_origen
    and E.estado_fisico = @edo_fis_origen
    and E.especimen = i.especimen
select @status = @@error
  if (@status !=0)
    begin
      < manejo de errores >
      return
    end

save tran trEXISTENCIA

update EXISTENCIA
  set EXISTENCIA.numero_piezas = E.numero_piezas +
      (i.numero_piezas *
       @afecta_elem_2)

  from inserted i, EXISTENCIA E
 where E.institucion = @institucion
    and E.plaza = @plaza
    and E.boveda = @boveda_destino
    and E.estado_fisico = @edo_fis_destino
    and E.especimen = i.especimen
select @status = @@error
  if (@status !=0)
    begin
      < manejo de errores >
      return
    end
```

CAPÍTULO 5. Codificación

```
/*
** Asigna fecha de proceso y cambia el estado de la operación a
** (P)rocesada
*/
select @nuevo_estado = 'P'

save tran mod_estado_OPERACION

exec @status =spsifi_mod_estado_OPERACION < parametros >
    if (@status !=0)
        begin
            < manejo de errores >
            return
        end

return
end
```

Por último, veamos algunos ejemplos de cómo se mantiene la integridad referencial en caso de eliminación de renglones, en primer lugar veamos el *trigger* de existencias para *delete* y posteriormente el *trigger* de *delete* para un sitio.

```
create trigger td_EXISTENCIA on EXISTENCIA
for delete
as
begin
if exists (select * from deleted where numero_piezas !=0)
    begin
        rollback trEXISTENCIA
        raiserror 40033 "EXISTENCIA. No se puede borrar renglones con
            saldo diferente de cero"
    end

return
end

return
end
```

```
create trigger td_SITIO on SITIO
for delete
as
begin
< declaracion de variables>
/*
** Validaciones de no existencia en tablas de operación
*/
```

CAPÍTULO 5. Codificación

```
if exists (select * from OPERACION t, deleted d
           where t.institucion_movimiento =d.institucion
                and t.plaza_movimiento =d.plaza)
    select @bandera = 'E'

/*
** Como en las remesas el sitio destino no es el sitio del
** movimiento es necesario verificar que no esté referenciado
*/

if (@bandera is null) and
    exists (select * from REMESA t, deleted d
           where t.institucion_destino =d.institucion
                and t.plaza_destino =d.plaza)
    select @bandera = 'E'

if (@bandera is not null)
    begin
        rollback trSITIO
        raiserror 30054 "SITIO. Existe información vigente en tablas
                        de operación"

        return
    end

save tran trTERMINAL
delete TERMINAL
    from TERMINAL t, deleted d
    where t.institucion =d.institucion
          and t.plaza =d.plaza

select @status =@@error
    if (@status !=0)
        begin
            < manejo de errores >
            return
        end

save tran trRESTRICCION_EXISTENCIA
delete RESTRICCION_EXISTENCIA
    from RESTRICCION_EXISTENCIA t, deleted d
    where t.institucion =d.institucion
          and t.plaza =d.plaza

select @status =@@error
    if (@status !=0)
        begin
            < manejo de errores >
```

CAPÍTULO 5. Codificación

```
        return
    end

save tran trVALIDACION_OPERACION
delete VALIDACION_OPERACION
    from VALIDACION_OPERACION t, deleted d
    where t.institucion_origen =d.institucion
        and t.plaza_origen      =d.plaza

select @status =@@error
    if (@status !=0)
        begin
            < manejo de errores >
            return
        end

save tran trVALIDACION_OPERACION
delete VALIDACION_OPERACION
    from VALIDACION_OPERACION t, deleted d
    where t.institucion_destino =d.institucion
        and t.plaza_destino     =d.plaza

select @status =@@error
    if (@status !=0)
        begin
            < manejo de errores >
            return
        end
    -end

return
end
```

El código de los objetos mostrados en este capítulo ilustra la forma de codificar los distintos objetos con que cuenta la base de datos, aun cuando se generaron más 700 procedimientos y alrededor de 120 *triggers*, el código de los objetos mostrados en este capítulo son representativos de la lógica empleada.

Ahora veamos algunas de las pruebas realizadas, así como el plan de liberación y las consideraciones para el mantenimiento del sistema.

6 Pruebas, liberación y mantenimiento

En este capítulo se muestran algunas de las pruebas realizadas al sistema para validar su correcto funcionamiento, también se muestran los planes de liberación, capacitación y mantenimiento.

6.1 Pruebas

Antes de liberar cualquier sistema es necesario realizar una serie de pruebas que garanticen que los resultados obtenidos son los esperados y que se están cumpliendo los requerimientos que los usuarios plantearon. En este punto revisaremos cuáles fueron las pruebas realizadas, incluyendo las pruebas de año 2000, que nos permiten prever posibles problemas cuando llegue dicho año.

Pruebas de unidad

Todos y cada uno de los *stored procedures* y *triggers* fueron probados en forma individual verificando que funcionaran correctamente, por ejemplo, se probó el procedimiento que verifica el horario en diferentes momentos para garantizar que valida en forma adecuada, retornando un código de error cuando un servicio está fuera de horario.

Pruebas funcionales

Como los *stored procedures* y los *triggers* utilizan otros *stored procedures* y *triggers*, en estas pruebas se verificó que en su conjunto realizarán todas las acciones esperadas de forma adecuada. Por ejemplo, la validación que se realiza en cada procedimiento verifica el horario y permisos de ejecución, estas validaciones son realizadas por procedimientos distintos, sin embargo, al ejecutar el procedimiento de validación debe realizarlas y devolver el código de error adecuado.

Pruebas de implantación

Una vez que la base de datos y todos los elementos que la componen fueron creados se procedió a probar nuevamente los procedimientos para verificar que en el ambiente de producción los resultados eran consistentes con las pruebas realizadas con anterioridad. Además se probó que el esquema de seguridad de la base de datos estuviera bien definido, es decir, que los usuarios no tuvieran acceso directo a las tablas y estuviera bien definido el conjunto de procedimientos que puede invocar.

Pruebas de año 2000

Una parte importante de las pruebas fue simular operaciones con fechas posteriores al 2000 para verificar el adecuado manejo de fechas y así garantizar la continuidad de operación en los años venideros. Estas pruebas incluyeron modificar la fecha del servidor para que todo el ambiente simulara estar en el año 2000. Las fechas utilizadas para estas pruebas fueron: 31 de diciembre de 1999, 1° y 3 de enero del 2000, 28 y 29 de febrero del 2000, ya que el año 2000 es bisiesto.

6.2 Liberación y Capacitación

Una vez que el usuario ha aprobado los resultados de las pruebas se procedió a la liberación y capacitación del mismo.

6.2.1 Liberación

Al liberar se crearon todas las tablas, *triggers* y *stored procedures* que conforman el sistema, probándolos nuevamente, con la intención de garantizar que su comportamiento es el mismo que presentaron en su etapa de codificación.

Se crearon todas las cuentas de acceso de los usuarios del sistema, notificando a cada usuario su cuenta y contraseña, solicitando a cada uno de ellos que cambiará su contraseña a la brevedad posible. Con la intención de ofrecer seguridad se hicieron varias revisiones a la tabla del sistema que contiene las cuentas, verificando la fecha en la que cada usuario cambio su contraseña hasta que todos los usuarios realizaron el cambio.

Se definieron los grupos de usuarios y se les asociaron los privilegios correspondientes a cada grupo, posteriormente se agregaron las cuentas a los grupos respectivos, como se ilustra en la Figura 6-1.

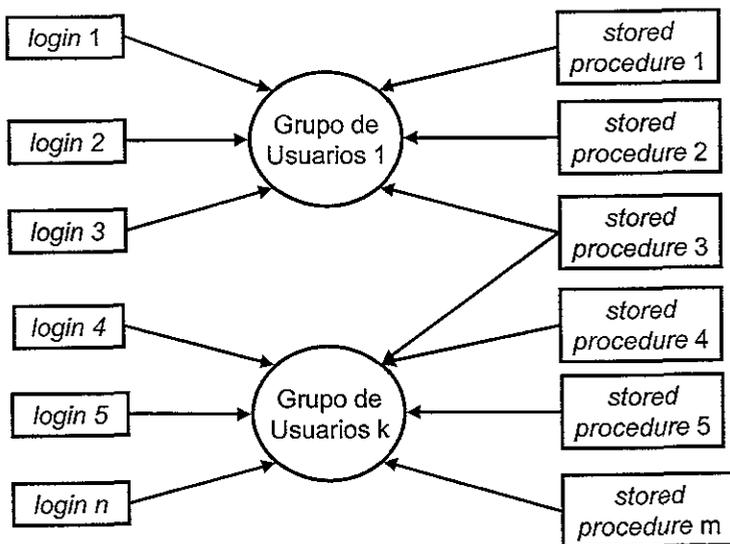


Figura 6-1 Esquema de seguridad basado en el RDBMS

Se cargaron los catálogos institucionales y locales para poder realizar la carga de datos para las tablas de existencias. Cuando las existencias ya estaban en las tablas se confrontaron contra los registros del usuario como se muestra en la Figura 6-2.

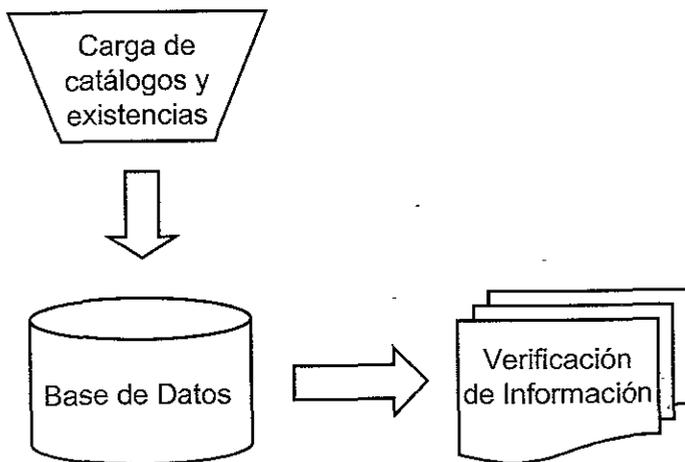


Figura 6-2. Carga de datos.

Después de verificar y aprobar el contenido de las tablas de existencias, se procedió a instalar el esquema de contingencias. Este esquema consiste en contar con al menos un servidor de respaldo el cuál es actualizado en forma remota, como se muestra en la Figura 6-3.

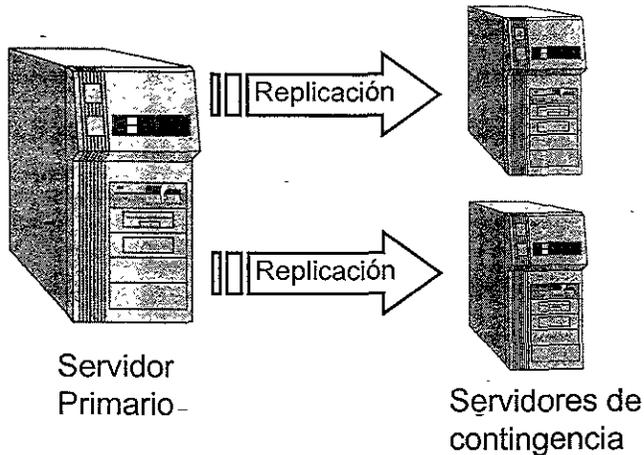


Figura 6-3. Esquema de replicación general.

Para actualizar la información en el servidor de contingencia se utilizó replicación. La replicación puede ser de dos formas, replicar ejecuciones o replicar movimientos.

Replicar ejecuciones significa repetir la ejecución de las instrucciones en el servidor de respaldo, por ejemplo, al ejecutar una inserción de una operación vía un *stored procedure*, en el servidor de contingencia se ejecutaría nuevamente el procedimiento con los mismos valores en los parámetros de entrada, sin embargo, este esquema puede provocar inconsistencias en el servidor de contingencia, por ejemplo, si un procedimiento fue invocado unos instantes antes del fin de su horario válido, y dado que hay una latencia que varía según los canales de comunicación entre los servidores, puede darse el caso que una operación que fue aceptada en el servidor primario sea rechazada en el servidor de contingencia.

La otra forma de replicar, consiste en realizar todos los movimientos en las tablas del servidor de contingencia, para esto, la herramienta encargada de replicar utiliza una tabla conocida como *log* (registro) de transacciones (ver Figura 6-4). En esta tabla se almacenan todos los movimientos a las tablas, también es utilizada por el manejador para deshacer una transacción cuando ejecuta un *rollback*; como se replican solamente las transacciones aceptadas,

mientras una transacción no esté concluida no se replicará. Este esquema permite contar con todas las operaciones independientemente del horario garantizando que toda la información estará en el servidor de contingencia, por esta razón éste fue el tipo de replicación utilizada.

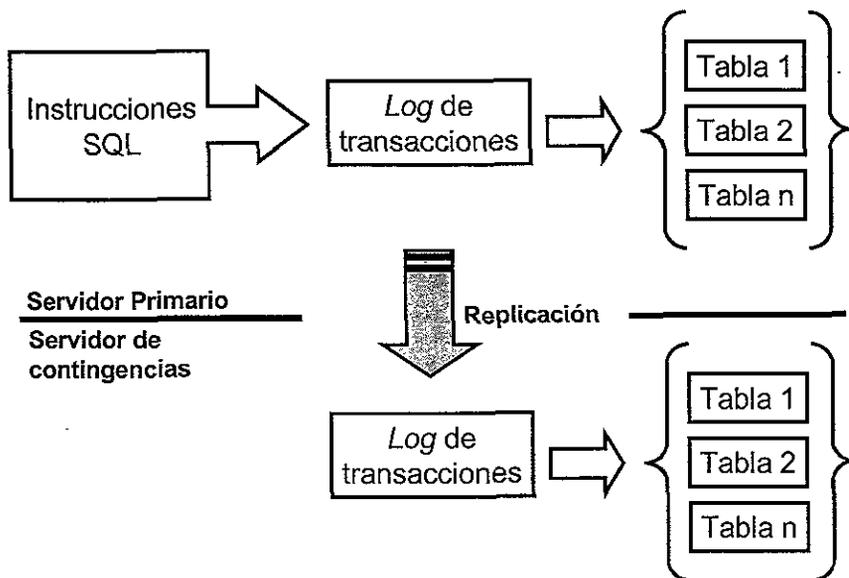


Figura 6-4. Replicación de movimientos.

6.2.2 Capacitación

A pesar de que el sistema presentado no incluye la interfaz con el usuario, es importante que el usuario conozca, en términos generales, el funcionamiento del servidor. Si el usuario entiende cómo se realizan las afectaciones a las existencias, en caso de un error de captura, cuenta con más elementos para detectar y corregir el error.

La capacitación mínima incluye:

- Manejo de existencias
- Operaciones
- Esquema de validaciones
- Esquema de permisos

- Esquema de seguridad
- Replicación

El manejo de existencias se refiere a la explicación de la estructura de las tablas donde se almacenan los diferentes tipos de existencia.

En cuanto el procesamiento de las operaciones incluye la forma de enviar las operaciones al servidor y que tablas de existencias afecta cada tipo de operación, pretendiendo que el usuario conozca a fondo que representa cada afectación.

En el esquema de validaciones se explica cómo se valida el cumplimiento de las reglas del negocio y cómo se definen.

Al tocar el tema del esquema de permisos se explica cómo está conformado el esquema, la utilización de los servicios, la posible creación de grupos de usuarios para facilitar la administración y como asociar los permisos tanto a los grupos de usuarios como a cada usuario.

En el esquema de seguridad se menciona el funcionamiento de los grupos definidos en el manejador y la importancia de la contraseña. También se menciona el funcionamiento y establecimiento de los horarios, así como el manejo del control general del sistema, desde el cual se controlan los distintos procesos que hay que realizar.

El punto de la replicación de datos, consiste en explicar qué es la replicación y como funciona en términos generales, indicando al usuario cuál es (o son) los servidores de contingencia y qué hay que hacer en caso de tener que cambiar de servidor.

Dado que la capacitación a los usuarios de la interfaz depende del equipo de trabajo que la codifique, para los fines del servidor basta que los usuarios conozcan a grandes rasgos cómo funciona y qué sucede en el servidor cuando realizan alguna petición.

6.3 Mantenimiento

Todos los sistemas, en algún momento de su vida, requieren ajustes para soportar los cambios en la forma de procesar la información. Esta fase la podemos clasificar en cuatro grupos, el mantenimiento ocasionado por un cambio en la forma de proceso, el mantenimiento generado por nuevas operaciones, el mantenimiento preventivo y el mantenimiento correctivo.

El mantenimiento por cambio en la forma de proceso surge cuando cambian las políticas de la empresa y es necesario ajustar el sistema para que cumpla con

las nuevas reglas. El mantenimiento por nuevas operaciones surge cuando se detecta un nuevo requerimiento que el usuario desea incluir en el sistema.

Normalmente cada modificación al código del sistema requiere utilizar la metodología del ciclo de vida, sin embargo, dado el esquema diseñado para manejar las existencias y las operaciones la inclusión de nuevos requerimientos es sencillo, si se trata de un nuevo tipo de existencia bastará con definir la tabla correspondiente y modificar los *triggers* de las tablas donde están las operaciones que afectan a este nuevo tipo de existencias. En el caso de nuevas operaciones, bastará agregar la o las tablas necesarias debajo de la tabla operación, suponiendo que no era factible incorporarla a las tablas existentes.

A pesar de contar con una gran cantidad de procedimientos de uso interno que pueden ser reutilizados, en el caso de consultas y reportes es posible que se tengan que generar nuevos procedimientos que permitan la extracción de la información requerida.

Con la intención de mejorar el rendimiento de las consultas solicitadas con mayor frecuencia, se puede realizar mantenimiento preventivo, este tipo de mantenimiento nos permite depurar el código de los procedimientos previendo problemas que pueden surgir cuando un servicio tiene gran demanda. Para identificar cuáles procedimientos son los más solicitados podemos revisar periódicamente la bitácora.

Ningún sistema está exento de contener algún tipo de error en la codificación, algunos errores pueden ser tan pequeños como el redondeo, cuando el usuario detecta alguna anomalía en los resultados es necesario corregir el problema, este tipo de arreglos es conocido como mantenimiento correctivo. Por la forma de diseño de este sistema, el mantenimiento correctivo no debe presentar mayores problemas, dado que está "granulado", es sencillo identificar qué procedimiento está fallando y solucionarlo en un solo punto, el problema se puede presentar en varios servicios, sin embargo es muy probable que el número de procedimientos a corregir sea considerablemente menor, por ejemplo, si se detectará un error en la forma de validar el horario, todos los servicios presentarían problemas en la ejecución, sin embargo, sería necesario modificar el código de sólo un procedimiento para solucionar el problema.

Una vez que el sistema fue probado e implantado, concluyamos este trabajo.

7 Resultados y conclusiones

En este capítulo se muestran los resultados obtenidos del sistema así como las conclusiones de este trabajo.

7.1 Resultados

Hoy en día se requiere que los administradores de los sistemas tengan mayor dominio sobre las aplicaciones, dado que las reglas del negocio son cambiantes. Estas personas deben tener control sobre la información y el acceso a ella, así como el conocimiento de la forma en la que funciona el sistema, lo que les permite enfrentar situaciones anormales en la operación del mismo. Por esta razón el sistema no debe considerarse como una "caja negra" que es alimentada y devuelve los resultados, es conveniente que los encargados de controlar la operación conozcan cómo funciona el sistema en su interior para realizar más eficientemente sus funciones.

El sistema desarrollado permite al administrador tener un amplio dominio sobre el establecimiento de las reglas del negocio; puede establecer las validaciones pertinentes, establecer los permisos de acceso a la información por cada usuario o grupos de usuarios y, en algunos casos, agregar nuevos tipos de operación sin la intervención del personal de sistemas.

No pretendemos que los administradores sean personas expertas en el desarrollo de sistemas, pero es importante que cuenten con los elementos necesarios para enfrentar situaciones imprevistas y tomar las decisiones pertinentes.

En la etapa inicial puede resultar agobiante el establecimiento de las reglas del negocio en el sistema, sin embargo, una vez superado este punto el mantenimiento de éstas es más simple.

Una de las metas de desarrollo era formar un sistema que ofreciera flexibilidad a los cambios y mayor control de la aplicación al administrador, dando así mayor responsabilidad a quién debe tenerla, que es quién realmente conoce la

operación. También se buscó que el diseño fuera flexible para la incorporación de nuevos requerimientos y facilitar el mantenimiento del mismo.

En general, el sistema desarrollado permite realizar las modificaciones necesarias de una forma sencilla. Por la estructura de las tablas de operaciones y manejo de existencias, agregar una nueva familia de operaciones o un tipo de existencias no contemplado es sencillo. En el caso de las operaciones, basta con agregar la nueva tabla de familia y, si así lo requiere, su detalle para que la operación comience a ser integrada al sistema.

El manejar un esquema de tres capas ofrece la ventaja de controlar la información que se devuelve a diferentes niveles, siendo el administrador el que decide quién puede ver la información. Además, permite optimizar el código mejorando el tiempo de respuesta de cada servicio, se tiene control absoluto de las transacciones y la forma en que se realizan, reduciendo el tiempo de contención en las páginas de datos y disminuyendo la probabilidad de *dead locks*. Dado que la estructura de las tablas es independiente al servicio, existen servicios dedicados a la inserción de operación que complementan la información, lo que permite crear llaves primarias artificiales que faciliten el manejo interno de las operaciones. Por otro lado, este esquema permite realizar alteraciones a la estructura de las tablas y los servicios en forma centralizada, evitando en varios casos, tener que modificar los clientes. También facilita este esquema la interacción con otros sistemas, principalmente si éstos están desarrollados en la misma plataforma.

Sin embargo, el costo de utilizar este esquema se refleja en la gran cantidad de código que es necesario generar para satisfacer las necesidades de los usuarios, si bien un mismo servicio puede mostrar la información acotada por criterios establecidos por el usuario, es frecuente encontrar consultas ocasionales que en su momento son de gran relevancia pero que con el tiempo su uso disminuye.

Al utilizar catálogos institucionales, se reduce la responsabilidad del administrador de mantener actualizada información de la cual él no es el encargado en forma directa, evitando así realizar actualizaciones extemporáneas y permitiéndole dedicarse a su trabajo sin utilizar recursos para tareas que no le competen.

7.2 Conclusiones

El sistema obtenido, a pesar de no contar con el cliente, contiene la lógica de procesamiento y las reglas del negocio, que son dos elementos vitales para el correcto funcionamiento del sistema. A pesar de la gran cantidad de código, el esquema de interacción utilizado permite desarrollar los clientes en más de una plataforma comportándose de igual manera.

El objetivo de este sistema es controlar las existencias y las operaciones que dan origen a los cambios en las existencias; el producto final no solamente satisface este objetivo, también contiene información de gran utilidad para otro tipo de usuarios, principalmente a aquellos que realizan los estudios de necesidad de efectivo en las distintas plazas.

El sistema soporta cualquier cantidad de operaciones, por la estructura jerárquica utilizada en el manejo de operaciones, el alcance del sistema puede crecer tanto como requerimientos existan.

En cuanto mantenimiento, podemos decir que es sencillo, tanto para el administrador como para el personal de sistemas. El mantenimiento de las reglas del negocio, permisos, información catalogada, horario de operación, etc., que es responsabilidad del administrador, está contenida en tablas, por lo que el mantenimiento es sólo en los datos. Para el mantenimiento que debe realizar personal de sistemas tanto preventivo, correctivo como nuevos requerimientos, puede realizarse con relativa facilidad, dado que una gran cantidad de procedimientos son utilizados únicamente en forma interna, en algunas ocasiones un comportamiento anormal en varios servicios puede corregirse modificando el código de un solo procedimiento.

Este sistema fue desarrollado en el Manejador de Bases de Datos Relacional que mejor se adaptaba al contexto de la solución, sin embargo, el modelo puede aplicar en cualquier otra plataforma.

El modelo generado en este trabajo puede ser aplicado no únicamente al control de flujos de caja, puede adoptarse como solución en cualquier problema de control de existencias, dado que el tratamiento dado a los billetes o monedas fue como mercancía y no como dinero.

Un punto de mejora en este sistema es el acceso controlado a las tablas para obtener consultas muy específicas que difícilmente se volverán a necesitar. Con el esquema actual, se necesita que una persona de sistemas desarrolle un servicio que satisfaga las necesidades de información, lo que involucra un cierto tiempo de codificación. Si para este tipo de necesidades se le otorgan permisos a los administradores del sistema y ellos son capaces de entender el modelo relacional, se evitaría generar código "inútil" sin descuidar las necesidades de información.

Finalmente, algo muy importante de resaltar es la influencia que nuestra máxima casa de estudios ha dejado en nuestra vida profesional, ya que con responsabilidad y dedicación es factible resolver cualquier problema que se nos presente, ya sea laboral o personal. Las enseñanzas y experiencias que recibimos a través del tiempo por parte de los profesores y compañeros que compartieron sus conocimientos con nosotros, se pueden ver materializadas en proyectos como el que se presentó en este trabajo.

A

Comandos SQL

En este apéndice se presenta la sintaxis de los comandos de *Transact - SQL* más comunes. Por claridad se han agrupado en varios temas con la finalidad de facilitar su comprensión.

Comandos básicos

Los comandos básicos para interactuar con la base de datos se muestran a continuación.

SELECT

Se utiliza para extraer información de las tablas.

```
select [all | distinct] select_list
[into [[database.]owner.]table_name]
[from [[database.]owner.]{view_name|table_name
  [(index index_name [ prefetch size ][lru|mru])]}
  [holdlock | noholdlock] [shared]
[, [[database.]owner.]{view_name|table_name
  [(index index_name [ prefetch size ][lru|mru])]}
  [holdlock | noholdlock] [shared]]... ]
[where search_conditions]
[group by [all] aggregate_free_expression
  [, aggregate_free_expression]... ]
[having search_conditions]
[order by {[[[database.]owner.]{table_name.|view_name.}]
  column_name | select_list_number | expression}
  [asc | desc]
  [, {[[[database.]owner.]{table_name|view_name.}]
  column_name | select_list_number | expression}
  [asc | desc]]...]}
[compute row_aggregate(column_name)
  [, row_aggregate(column_name)]...
  [by column_name [, column_name]...]]
[for {read only | update [of column_name_list]}]
[at isolation {read uncommitted | read committed |
  serializable}] [for browse]
```

INSERT

Agrega renglones a una tabla o vista.

```
insert [into]
  [(database.[owner.]){table_name|view_name}
  [(column_list)]
  {values (expression [, expression]...)
  |select_statement }
```

UPDATE

Cambia el contenido de los renglones existentes.

```
update [[database.]owner.]{table_name | view_name}
  set [[(database.)owner.]{table_name.|view_name.}]
  column_name1 = {expression1|NULL|(select_statement)}
  [, column_name2 = {expression2|NULL|(select_statement)}]...
[from [(database.)owner.]{view_name|table_name
  [(index index_name [ prefetch size ] [lru|mru])]}]
  [, [(database.)owner.]{view_name|table_name
  [(index index_name [ prefetch size ] [lru|mru])]}]
  ...]
[where search_conditions]
```

DELETE

Elimina renglones de una tabla.

```
.delete [from] [[database.]owner.]{view_name|table_name}
  [where search_conditions]

delete [[database.]owner.]{table_name | view_name}
  [from [(database.)owner.]{view_name|table_name
  [(index index_name [ prefetch size ] [lru|mru])]}]
  [, [(database.)owner.]{view_name|table_name
  [(index index_name [ prefetch size ] [lru|mru])]}]
  ...]
  [where search_conditions]
```

Elementos de una Base de datos

Con los comandos que se muestran a continuación es factible crear y destruir una base de datos. También se muestran los comandos que nos permiten crear tablas, índices, reglas, *stored procedures*, *triggers*, etc.

CREATE DATABASE Crea una base de datos. Este comando se ejecuta desde la base de datos maestra y se requiere que sea ejecutado por algún usuario que tenga privilegios de dba (*database administrator*, administrador de bases de datos).

```
create database database_name
  [on {default | database_device} [= size]
    [, database_device [= size]]...]
  [log on database_device [= size]
    [, database_device [= size]]...]
  [with override] [for load]
```

DROP DATABASE Elimina una o más bases de datos del servidor. Este comando puede ser ejecutado por un dba o por el dbo (*database owner*, dueño de la base de datos), por esta razón es importante contar con un estricto control de cuáles usuarios tienen esa atribución.

```
drop database database_name [, database_name]...
```

Los siguientes comandos pueden ser ejecutados por un dbo o algún usuario con privilegios.

CREATE TABLE Crea una nueva tabla y de manera opcional *constraints* de integridad.

```

create table [database.]owner.]table_name
  (column_name datatype
    [default {constant_expression | user | null}]
    {[{identity | null | not null}]
    | [{constraint constraint_name}
      {{unique | primary key}
      [clustered | nonclustered]
      [with {fillfactor | max_rows_per_page}= x]
      - [on segment_name]
      | references [[database.]owner.]ref_table
        [(ref_column)]
      | check (search_condition)}}]...
  | [constraint constraint_name]
    {{unique | primary key}
    [clustered | nonclustered]
    (column_name
    [{, column_name}...])
    {fillfactor | max_rows_per_page}= x]
    [on segment_name]
  | foreign key (column_name
    [{, column_name}...])
    references [[database.]owner.]ref_table
      [(ref_column [{, ref_column}...])]
  | check (search_condition)
    [{, {next_column | next_constraint}}]...
  [with max_rows_per_page = x]
  [on segment_name]

```

DROP TABLE

Elimina una tabla y todos los índices, *triggers*, y permisos asociados a ella.

```

drop table [[database.]owner.]table_name
[, [[database.]owner.]table_name ]...

```

ALTER TABLE

Agrega columnas; agrega, cambia o elimina *constraints*; particiona o elimina las particiones existentes en una tabla. Las nuevas columnas deben aceptar valores nulos o bien tener asociado un *default*.

```

alter table [database.[owner].]table_name
  {add column_name datatype
    [default {constant_expression | user | null}]
    [{identity | null}]
    | [[constraint constraint_name]
      {unique | primary key}
      [clustered | nonclustered]
      [with {fillfactor | max_rows_per_page} = x]
      [on segment_name]
      | references [[database.]owner.]ref_table
        [(ref_column)]
      | check (search_condition)}}}
  {[, next_column]}
| add [{constraint constraint_name}
  {unique | primary key}
  [clustered | nonclustered]
  (column_name [{, column_name}...])
  [with {fillfactor | max_rows_per_page} = x]
  [on segment_name]
  | foreign key (column_name [{, column_name}...])
    references [[database.]owner.]ref_table
      [(ref_column [{, ref_column}...])]
  | check (search_condition)}
| drop constraint constraint_name
| replace column_name
  default {constant_expression | user | null}
| partition number_of_partitions
| unpartition}

```

CREATE INDEX

Crea un índice sobre una o más columnas en una tabla.

```

create [unique] [clustered | nonclustered] index index_name
  on [[database.]owner.]table_name (column_name
    [, column_name]...)
  [with {{fillfactor | max_rows_per_page} = x,
    ignore_dup_key, sorted_data,
    [ignore_dup_row | allow_dup_row]}]
  [on segment_name]

```

DROP INDEX

Elimina índices.

```

drop index table_name.index_name [, table_name.index_name]...

```

CREATE DEFAULT Especifica el valor a insertar en una columna (*default*) si es que no se proporciona un valor de manera explícita al momento de realizar una inserción.

```
create default [owner.]default_name
as constant_expression
```

DROP DEFAULT Elimina *defaults*.

```
drop default [owner.]default_name
[, [owner.]default_name]...
```

CREATE RULE Crea una regla (*rule*). Establece el dominio de los valores aceptables para una columna en particular.

```
create rule [owner.]rule_name
as condition_expression
```

DROP RULE Elimina una regla definida por el usuario.

```
drop rule [owner.]rule_name [, [owner.]rule_name]...
```

CREATE VIEW Crea una vista que es una forma alternativa de visualizar los datos en una o más tablas.

```
create view [owner.]view_name
[(column_name [, column_name]...)]
as select [distinct] select_statement
[with check option]
```

DROP VIEW Elimina una o varias vistas.

```
drop view [owner.]view_name [, [owner.]view_name]...
```

CREATE PROCEDURE Crea un procedimiento almacenado que puede recibir uno o más parámetros.

```
create proc[edure] [owner.]procedure_name[;number]
  [(@parameter_name
    datatype [(length) | (precision [, scale])]
    [= default][output]
  [, @parameter_name
    datatype [(length) | (precision [, scale])]
    [= default][output]]...[)]
  [with recompile]
as SQL_statements
```

DROP PROCEDURE Elimina procedimientos almacenados.

```
drop proc[edure] [owner.]procedure_name
  [, [owner.]procedure_name] ...
```

CREATE TRIGGER Crea un *trigger*.

```
create trigger [owner.]trigger_name
  on [owner.]table_name
  for {insert , update , delete}
as SQL_statements
```

O bien, usando la cláusula "*if update*":

```
create trigger [owner.]trigger_name
  on [owner.]table_name
  for {insert , update}
as
  [if update (column_name) [{and | or} update (column_name)]...]
  SQL_statements

  [if update (column_name) [{and | or} update (column_name)]...
  SQL_statements]...
```

DROP TRIGGER Elimina un *trigger*.

```
drop trigger [owner.]trigger_name [, [owner.]trigger_name]...
```

Manejo de transacciones

Los siguientes comandos se utilizan para abrir una transacción, aceptarla o rechazarla y manejar puntos de recuperación.

BEGIN TRANSACTION Marca el punto de inicio de una transacción definida por el usuario.

```
begin tran[saction] [transaction_name]
```

COMMIT TRANSACTION Marca el punto final de una transacción definida por el usuario.

```
commit [tran[saction] | work] [transaction_name]
```

SAVE TRANSACTION Marca un punto intermedio dentro de una transacción con la finalidad de no deshacer todas las alteraciones realizadas antes de dicho punto.

```
save tran[saction] [savepoint_name]
```

ROLLBACK Deshace todas las afectaciones realizadas desde el inicio de la transacción o bien desde el *save point*.

```
rollback {tran[saction] | work}  
         [transaction_name | savepoint_name]
```

Ciclos de control

El RDBMS tiene algunas de estructuras de control, los principales comandos se muestran a continuación

IF - ELSE

Permite evaluar condiciones en las sentencias SQL.

```
if logical_expression
  statement
[else
  [if logical_expression]
  statement]
```

WHILE

Ejecuta una serie de sentencias mientras la condición evaluada sea verdadera.

```
while logical_expression
  statement
```

BEGIN - END

Delimita una serie de comandos SQL, mismos que serán ejecutados dentro de una sentencia de control, por ejemplo *if ... else*.

```
begin
statement block
end
```

Permisos

Los siguientes comandos son utilizados para otorgar o retirar permisos de *select insert, update* o *delete* en las tablas o vistas y de ejecutar *stored procedures*.

GRANT

Asigna permisos a usuarios o grupos de usuarios para poder tener acceso a los objetos de la base de datos.

```
grant {all [privileges] | permission_list}
  on {table_name [(column_list)]
     |view_name[(column_list)]
     |stored_procedure_name}
  to {public | name_list | role_name} with grant option]
```

REVOKE

Retira permisos a usuarios o grupos de usuarios de acceso de los objetos de la base de datos.

```
revoke [grant option for]
  {all [privileges] | permission_list}
  on {table_name [(column_list)]
     |view_name[(column_list)]
     |stored_procedure_name}
  to {public | name_list | role_name}
```

B

Glosario de términos

Institución	Organismo o empresa que tiene relación con el Banco Central, por ejemplo, las instituciones bancarias.
Plaza	Es un poblado donde existe al menos una sucursal de alguna institución bancaria.
Bóveda	Lugar físico destinado al almacenamiento de billete o moneda.
Ventanilla	Lugar físico donde se realizan distintos procesos con los billetes o monedas como recuento, ordenamiento, clasificación. Las ventanillas solamente tienen piezas en su poder durante el día operativo, al cierre del día deben estar vacías.
Ventanilla de atención	Tipo de ventanilla donde se reciben o entregan billetes o monedas.
Estado físico	Es el estado de desgaste que tiene cada billete o moneda.
Estado lógico o de proceso	Es el proceso en el cuál se encuentra un billete o moneda, principalmente cuando está en autorizado o en proceso de fabricación.
Pieza	Un billete o moneda.
Espécimen	Pieza de cierta denominación, valor real y características de fabricación.
Banco Usuario	Sucursal de una institución bancaria que está facultada para acudir al Banco Central a realizar depósitos y retiros de efectivo a nombre de la institución a la que pertenece.

Bloqueo	Sucede cuando un proceso mantiene en forma exclusiva una o varias páginas de datos en la base de datos. Mientras una página o tabla está bloqueada ninguna otra sesión puede accederla. Si una sesión requiere una página bloqueada se esperará hasta que la sesión que la tiene la libere.
Dead lock	Es un estado en la base de datos que ocurre cuando dos o más sesiones tienen bloqueadas páginas de datos o de índices y requieren acceder alguna de las páginas que tiene bloqueadas la otra sesión.
Transacción	Conjunto de instrucciones agrupadas en un bloque, tiene la característica que no se puede afectar parcialmente, es decir, se aplica en su totalidad o no aplica ninguna instrucción.
Transacción	En el contexto de los flujos de información, una transacción es una operación o movimiento.
Commit	Es la aceptación de una transacción. Se utiliza para indicar que termina la transacción y que todas las afectaciones que la conforman son aceptadas.
Save point	Punto de referencia que facilita deshacer una transacción, normalmente se utiliza para el manejo de errores.
Rollback	Es el rechazo de una transacción. Se utiliza para deshacer todas las afectaciones de la transacción realizadas hasta este punto. Si se declara un <i>save point</i> se pueden deshacer solamente las instrucciones contenidas después del <i>save point</i> .
Join	Es el producto cartesiano de dos o más tablas. Mediante la utilización de llaves primarias y foráneas, y de algunos otros criterios de selección se obtiene la intersección del contenido de las tablas.
Outer Join	Es una particularidad de los <i>joins</i> , que consiste en mostrar el contenido de una tabla y aquellos renglones que tengan una correspondencia en la otra tabla, mostrando un valor nulo en aquellos renglones de la primera tabla que no tienen correspondencia en la segunda.
SQL	<i>Structured Queries Language</i> (Lenguaje de Consultas Estructuradas). Es el lenguaje mediante el cual se realizan las peticiones de información, se realizan inserciones, modificaciones y eliminación de renglones.

GLOSARIO

Recuento	Proceso mediante el cuál se verifica el número de piezas reportadas corresponda a la realidad, además se verifica que todas las piezas sean de la misma denominación.
Corresponsal	Sucursal bancaria que realiza algunas de las funciones del Banco Central.
Sitio	Lugar donde se almacena efectivo y se realizan operaciones de caja. Cada sitio es identificado por la clave de la institución a la que pertenece y la clave de la plaza donde se ubica.
Billete Mutilado	Billete que por ruptura está incompleto.
Billete Parchado	Billete roto unido con algún adhesivo.
Traspaso	Operación que consiste en trasladar efectivo de un lugar a otro dentro del mismo sitio.
Remesa	Operación que consiste en trasladar efectivo de un sitio a otro. Esta operación se compone de dos fases, el envío y la recepción.
Envío	Fase inicial de una remesa que consiste en retirar efectivo del sitio origen y embarcarlo en el transporte previsto. Dentro del sistema el envío involucra disminuir las existencias del sitio origen y aumentar las piezas en tránsito.
Recepción	Fase final de una remesa que consiste en desembarcar efectivo del transporte y agregarlas al sitio destino. Dentro del sistema la recepción involucra disminuir las piezas en tránsito y aumentar las existencias del sitio destino.
Depósito	Operación que realizan los bancos comerciales que consiste en entregar dinero en efectivo en el Banco Central o en sus Corresponsales con abono a su cuenta electrónica.
Retiro	Operación que realizan los bancos comerciales que consiste en retirar dinero en efectivo del Banco Central o de sus Corresponsales con cargo a su cuenta electrónica.
Autorización	Operación mediante la cuál, la autoridad competente, permite al Banco Central fabricar una determinada cantidad de efectivo.
Fabricación	Proceso en el cuál el dinero toma su forma física.
Destrucción	Proceso mediante el cuál, las piezas deterioradas, son destruidas.

Bibliografía

Constitución Política de los Estados Unidos Mexicanos.

BARROS DE CASTRO, Antonio, LESSA, Carlos Francisco, *"Introducción a la economía, un enfoque estructuralista"*, Editorial Siglo veintiuno, 49a edición.

DATE, C.J., *"Introducción a los Sistemas de Bases de Datos"*, Segunda edición, Ed. Addison-Wesley Iberoamericana, S.A., México, 1990.

GEREZ, Víctor, MIER, Mauricio, *"Desarrollo y Administración de Programas de Computación (Software)"*, Segunda edición, Ed. Compañía Editorial Continental, S:A: de C:V:, México, México, 1985.

IAN, Claudio, *"Apuntes de Ingeniería de programación"*.

KENDELL & KENDELL, *"Análisis y Diseño de Sistemas"*, Tercera edición, Ed. Prentice Hall Hispanoamericana, S.A., México, 1997.

LONEY, Kevin, *"ORACLE, Manual del Administrador versión 6 y 7.x"*, Primera edición, Ed. McGraw-Hill Publishing Company/Interamericana de España, S.A., Madrid, España, 1995.

LÓPEZ FUENSALIDA, Antonio, *"Metodología de Desarrollo en el camino hacia el CASE"*, Primera edición, Ed. RA-MA, Madrid, España, 1990.

MARTIN, Tim, *"DB2/SQL Manual para programadores"*, Primera edición, Ed. McGraw-Hill Publishing Company/Interamericana de España, S.A., Madrid, España, 1991.

PRESSMAN, Roger S., *"Ingeniería del Software un enfoque práctico"*, Tercera edición, Ed. Addison-Wesley Iberoamericana, S.A., México, 1990.

TRAN, Viet G., *"Introduction to DB2 Programming"*, Segunda edición, Ed. McGraw-Hill Publishing Company, Estados Unidos, 1989.

Bibliografía

HALFHILL, Tom, "UNIX vs Windows NT", BYTE, May 1996 Volume 21, number 5.

NANCE, Barry, "Enterprise Data Managers", BYTE, November 1995 Volume 20 number 11.

NANCE, Barry, "El mejor Sistema Operativo para servicios Web: ¿Unix o NT?", BYTE México, México, Marzo 1998.

NANCE, Barry, "Relations with your data", BYTE , June 1998 Volume 23 number 6.

SINHA, Alok, "Client-Server Computing", Communications of the ACM, July 1992 Volume 37 Number 7.

Banco de México

<http://www.banxico.org.mx>

Banco de Argentina

<http://www.bcra.gov.ar>

Banco de España

<http://www.bde.es/>

BCE (Banco Central Europeo)

<http://www.bde.es/ume/cuestio/cuestio7.htm>

Federal Reserve Board (Washington), "Purposes and Functions"

<http://www.bog.frb.fed.us/pf/pdf/frspf2.pdf>

Dirección General de Servicios de Cómputo Académico

<http://www.asc.unam.mx/>

Universidad de Guadalajara

<http://www.udg.mx/Jat/Jat95/integrarsis.html>

"Implementation of an open systems application environment RDBMS comparative analysis"

<http://www.zeda.com/rdbmsbd.htm>

The Standish Group International, Inc., "Solaris vs NT"

<http://www.standishgroup.com/syst.html>

"A Technical Comparison of Oracle and Sybase"

<http://www.zeda.com/rdbmsbd.htm>

The University of Texas , "Overview of the Relational Model"

<http://www.microlib.cc.utexas.edu/cc/dbms/utinfo/reimod/>

Informix

<http://www.informix.com>

Sybase

<http://www.sybase.com>

Oracle

<http://www.oracle.com/>

SUN

<http://www.sun.com/>

Microsoft

<http://www.microsoft.com/>