FACULTAD DE CIENCIAS

DEPARTAMENTO DE MATEMÁTICAS

EXPERIENCIA PROFESIONAL

Implementación de un nuevo modelo de consulta de información y reestructura de arquitectura de datos

Trabajo por experiencia profesional presentado por Erick Celso Zavaleta González para obtener el grado de Licenciado en Ciencias de la Computación

---

Supervisados por: Dr. Canek Peláez Valdés

Dedicado a...

Índice general

Índice de figuras

Índice de tablas

Agradecimientos

¡Muchas gracias a todos!

Capítulo 1

Introducción

El proyecto que se presenta en este trabajo fue la implementación y reestructura de la infraestructura de base de datos, así como los cambios en procesos y arquitectura de datos de una institución financiera que por razones de acuerdos de confidencialidad no puedo mencionar su nombre. Dentro de las actividades que realicé se encuentra un análisis del software a utilizar, así como la definición y desarrollo de programas para convertir los datos de diferentes bases de datos y plataformas hacia un repositorio de datos único que ayudó a eficientar los tiempos de respuesta y las consultas realizadas.

El marco teórico del proyecto se basa en la arquitectura de datos, que en tecnología de la información se compone de los modelos, políticas, reglas y/o estándares que determinan qué datos se recolectan y cómo se guardan, ordenan, integran y usan en los sistemas de datos de una institución.

Hablar de datos es entrar en un mundo complejo donde existen diferentes perspectivas de datos, dependiendo del uso que se les quiera dar o de las personas que lo utilizan. Cada grupo de personas tiene su propia perspectiva sobre el manejo de datos: manejo de grandes volúmenes, acceso al detalle de los datos de manera instantánea, manejo de la integridad, datos de acceso exclusivo, etcétera. La arquitectura de datos nos ayuda a que estos diferentes tipos de datos y diferentes necesidades puedan coexistir de una forma conjunta de acuerdo a las necesidades de cada área o empresa.

Actualmente no existe ningún secreto para el manejo de datos y su respectiva arquitectura; en ambos casos, es importante entender los datos en términos de su infraestructura: es decir, se requiere conocer la infraestructura que rodea a los datos para llevar a cabo un uso adecuado de los mismos.

Así mismo, hacemos notar que dentro de cualquier empresa u organi-

zación se pueden encontrar diferentes tipos de datos: estructurados y no estructurados. Los primeros son datos predecibles y que normalmente son manejados en una base de datos (SMBD): registros, atributos, llaves, índices, etcétera. Por su parte, los datos no estructurados no son predecibles y, como su nombre lo indica, no tienen una estructura bien definida, usualmente son de difícil acceso y generalmente se requiere una búsqueda más profunda para hacer consultas; por ejemplo, una cadena de caracteres en un texto libre.

Todos estos conceptos serán detallados y aplicados en los siguientes capítulos de este trabajo.

Capítulo 2

Análisis de requerimientos

En este capítulo se detallan los requerimientos técnicos y funcionales del proyecto, así como el proceso de selección de software requerido para atender a las necesidades de la institución financiera.

2.1. Proceso de selección de software

Parte importante para llegar a la solución propuesta fue la selección del software en el cual se desarrolló el nuevo proceso ETL[1]. La selección de software requirió de varios pasos previos a la toma de la decisión. El proceso de selección de software de acuerdo a la metodología de Accenture fue la siguiente:

1. Establecer los requerimientos de negocio.
2. Establecer los requerimientos de evaluación de las herramientas de ETL.
3. Generar una lista larga de proveedores.
4. Generar una lista corta (generalmente 3 o 4 proveedores) de candidatos.
5. Realizar un RFP (*Request For Approval*) para evaluar a los candidatos de manera individual y detallada.
6. Realizar la recomendación final del producto.

Este proceso se detalla en la figura 2.1:

[1]Extracción, Transformación y Carga (*load* en Inglés).

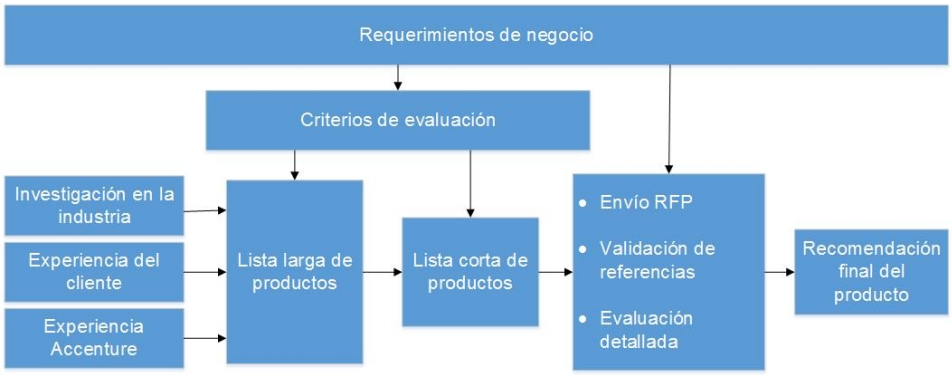

Figura 2.1: Proceso de selección de software

En las siguientes secciones se definirán cada uno de los pasos del proceso.

2.1.1. Requerimientos de funcionalidad del software

Con base en las reuniones realizadas con el área de desarrollo de sistemas de la institución financiera se identificaron una serie de requerimientos que debía de cumplir la herramienta seleccionada. Estos requerimientos se detallan a continuación.

- Se requería una conexión a correo electrónico mediante el protocolo SMTP para envío de informes de termino de procesos o de errores en los mismos.
- Tener una bitácora (*log*) de procesos que tuviera la información de cada uno de los procesos de extracción/carga (inicio, fin, estado de término, errores –en caso de haberlos–).
- La herramienta debía de tener la capacidad de contar con un esquema de limpieza de datos que permitiera identificar y corregir los errores en nombres, direcciones, RFCs, números de teléfono, y algunos otros datos relevantes para la institución financiera.
- Permitir invocar a la herramienta ETL desde otras aplicaciones tales como PeopleSoft, aplicaciones desarrolladas en Java y .NET.
- Permitir control y detalle de los errores presentados en los procesos o en cada uno de los componentes del flujo de datos. El control se llevaría

a cabo mediante el envío, por parte del ETL, de errores a tablas de bitácora o archivos de texto en donde se pudieran identificar de una manera sencilla, los errores presentados.

- Contar con un esquema de datos redundante, como puede ser un grupo (*cluster*), bases de datos replicadas, balanceo de cargas con tolerancia a fallos, etcétera.

- Contar con un esquema de control de versiones que soporte el trabajo de múltiples usuarios al mismo tiempo.

- La herramienta debía tener la capacidad para entregar diferentes archivos a los sistemas destino (archivos de texto, tablas en base de datos, XML, PDF, Excel).

- Contar con un esquema de seguridad basado en roles y jerarquías de usuario que permitieran a los usuarios acceder a datos específicos de la base de datos de acuerdo a su rol.

- Contar con pantallas de configuración que permitieran a los administradores de la herramienta tener un mayor control sobre los procesos y los usuarios. Así mismo, que fuera una herramienta fácil de usar visualmente.

- Permitir a la nueva herramienta la convivencia con las tecnologías con las que contaba la institución financiera (Windows 2003, SQL Server 2005, Oracle, DB2, AS400, Windows Vista y *Mainframes*).

- Contar con una herramienta de fácil manejo de metadatos.

- Contar con un ambiente de diseño y desarrollo 100 % visual que permitiera a los desarrolladores implementar las soluciones de extracción y carga de una manera sencilla.

- Permitir la operación y administración de la herramienta de una forma remota.

- Generar componentes reutilizables entre aplicaciones y entre procesos.

- La herramienta debía permitir a los usuarios la creación de funciones personalizadas que cumplieran con los estándares de la empresa y que no fueran parte de las configuraciones predefinidas de la herramienta.

- Capacidad para calendarizar procesos; es decir, la herramienta debería ser capaz de ejecutar los procesos de forma automática a la hora y día especificados.
- Soportar modelos de minería de datos que ayudarían a la institución financiera a realizar un análisis de sus datos y poder definir estrategias de mercado para los diferentes productos que ofrecían.
- Capacidades de soporte para DataWarehouse y Business Intelligence, este requerimiento era para cumplir con el plan estratégico de la institución.
- Capacidad de la herramienta para publicar procesos como *Web Services* a fin de poder acceder a ellos de manera remota o ejecutarlos a través de un portal con los permisos correspondientes.
- Capacidades de detección y corrección de errores en los procesos, mediante un proceso de depuración (*debugging*) que sería ejecutado paso a paso.
- Como un requerimiento básico, la herramienta debería de almacenar la información extraída en una sola fuente de datos, para la solución propuesta fue una base de datos de paso llamada base de datos *stage*.
- Explotación de la información de una base centralizada para realizar reportes.
- Consulta de datos históricos, en especial de transacciones, para poder ser explotados desde otro ambiente o aplicación existente en la institución financiera.
- La herramienta debería de ser capaz de obtener solamente la información que tuvo cambios entre los diferentes periodos de extracción, ayudando al rendimiento del proceso y extrayendo una cantidad menor de datos.

2.1.2. Requerimientos técnicos del software

Por su parte los requerimientos técnicos solicitados por la institución para seleccionar la herramienta fueron los siguientes:

- Mejorar el rendimiento de la extracción de datos de 90 minutos a 45 minutos (tiempo tentativo) con una cantidad de 6 millones de registros.

- Mejorar el tiempo de transformación de datos utilizando componentes para limpieza y calidad de los datos.
- Mejorar el rendimiento de la herramienta cuando el volumen de datos exceda los 100 millones de registros.
- Que la arquitectura de la herramienta fuera compatible con la arquitectura de la institución y la arquitectura futura que se planteó en el presente proyecto.
- La herramienta debía ser capaz de poder integrarse son los sistemas de la institución (PeopleSoft, Core bancario, sistema de reportes, sistema de recursos humanos, pagos a terceros, etcétera.)
- La herramienta debía ser capaz de ejecutarse en diferentes plataformas (Windows, Linux, Unix, *Mainframes*).
- Conectividad con los sistemas fuente (ODBC, OLAP, LDAP) y con diferentes sistemas manejadores de bases de datos como Oracle, DB2, SQL Server, Informix, Sybase, Teradata.
- Capacidad para crear funciones y métodos desde cualquier lenguaje de programación estructurado y que pudiera ser ejecutado desde la herramienta ETL.

Aunque el gobierno de datos no era parte del alcance del proyecto, sí era parte de la estrategia de crecimiento de la institución, y por ello la herramienta debería de soportar esta funcionalidad.

2.2. Criterios de evaluación

Como mencionamos al inicio del documento, parte importante de la selección de sofware fue la definición de los criterios de evaluación de la herramienta ETL a utilizar como parte del proyecto. Dichos criterios nos ayudaron a realizar una evaluación objetiva de las diferentes herramientas y tener un panorama general de los productos que cumplían con las características y necesidades de la institución financiera. Tomando como base esas necesidades y la experiencia de Accenture se definieron los criterios de selección de software.

Los criterios se agruparon en conceptos genéricos que describen la funcionalidad de cada área. Estos criterios se listan a continuación:

1. *Capacidades del servicio.*
2. *Opciones de integración.*
3. *Ambiente de la herramienta.*
4. *Soporte y capacitación.*
5. *Técnicas adicionales de integración de datos.*
6. *Manejo de la información.*
7. *Estrategias del producto.*
8. *Estrategias corporativas.*
9. *Costos.*
10. *Convenios con otros proveedores.*
11. *Finanzas de la compañía.*

2.2.1. Lista de proveedores

Una vez que se definieron los criterios de evaluación y se realizó el análisis de requerimientos con el equipo de la institución financiera, el siguiente paso fue dar una lista larga de proveedores candidatos para ser evaluados y que en ese momento eran las mejores opciones dentro del mercado. La lista contenía a 6 proveedores de software: Oracle (Oracle Warehouse Builder), Informática (Power Center), IBM (Infosphere Information Server), Microsoft (Integration Services SSIS), SAP (SAP-Business Objects), SAS Institute (SAS).

2.2.2. Evaluación de proveedores

Con la lista larga de proveedores definida, la siguiente tarea que se llevó a cabo fue la evaluación de cada uno de los proveedores con el fin de establecer un grupo reducido de candidatos que representaría la lista final de selección de la herramienta ETL a utilizar dentro del proyecto. De acuerdo a las necesidades de la Institución financiera se definieron los siguientes porcentajes para los criterios de evaluación

La prioridad de cada uno de los criterios de evaluación se realizó con base en las necesidades de la institución y en la funcionalidad de cada uno de ellos.

Criterio	Porcentaje de evaluación	Prioridad
Capacidades del servicio	14 %	1
Manejo de la información	14 %	2
Opciones de integración	11 %	3
Ambiente de la herramienta	11 %	4
Costos	11 %	5
Soporte y capacitación	9 %	6
Estrategias del producto	6 %	7
Convenios con otros proveedores	6 %	8
Técnicas adicionales de integración de datos	6 %	9
Estrategias corporativas	6 %	10
Finanzas de la compañía proveedora	6 %	11

Tabla 2.1: Porcentajes y prioridades de evaluación.

Con toda la información que se recolectó, realicé un análisis de cada proveedor y su herramientas presentadas, así como sus factores diferenciales y las características principales soportadas. Esta evaluación se presenta en la siguiente tabla con los resultados finales:

2.3. Lista corta de proveedores

Con base en los criterios de selección y al análisis de las capacidades de cada una de las herramientas, se obtuvo la lista corta de los proveedores. Los proveedores seleccionados fueron los siguientes:

- IBM
- Informática
- Oracle
- Microsoft

Si bien SAP-Business objects tenía una calificación mayor, el negocio decidió no incluirlo en la lista final debido a que no se contaba con capacitación suficiente de parte del proveedor. Así mismo, Microsoft se incluyó a petición explicita de la institución.

		Microsoft SSIS	Oracle OWB	Informática Power center	IBM IIS	SAPBusiness Objects	SAS
	Porcentaje						
Capacidades del servicio	**14.29 %**	**3**	**3.5**	**4.5**	**4**	**3.75**	**4**
Escalabilidad y rendimiento		4	4	5	5	4	4
Alta disponibilidad		4	3	4	4	4	3
Seguridad		3	4	5	3	4	5
Plataformas de ejecución soportadas		1	3	4	4	3	4
Opciones de integración	**11.43 %**	**3**	**3**	**4**	**4**	**3.4**	**3.6**
Conectividad con sistemas fuente		4	3	5	5	5	5
Conectividad para la carga		4	3	5	5	5	5
Servicios Web		4	4	5	5	3	3
Conexión a correo electrónico		1	1	1	1	0	1
Reusabilidad		2	4	4	4	4	4
Ambientes de la herramienta	**11.43 %**	**3.2**	**4**	**4.4**	**4.2**	**4.4**	**3.6**
Visualización del ambiente de diseño y desarrollo		4	4	4	4	4	4
Manejo de errores		4	4	5	4	5	4
Ambiente de colaboración		2	4	4	4	4	3
Manejo y modelado de datos y metadatos		2	4	4	4	4	4
Administración		4	4	5	5	5	3
Soporte y capacitación	**8.57 %**	**4.25**	**4.75**	**4.25**	**4.5**	**3.75**	**4.75**
Soporte		4	5	5	5	5	4
Capacitación		4	5	3	5	3	5
Documentación		4	4	5	5	3	5
Soporte a diferentes lenguajes		5	5	4	3	4	5
Técnicas adicionales de integración de datos	**5.71 %**	**4**	**2.5**	**4**	**3.5**	**3.5**	**1**
EII		3	3	4	4	3	0
Cambio en la captura de datos		5	2	4	3	4	2
Manejo de la información	**14.29 %**	**3.6**	**3.8**	**4.4**	**4**	**3.4**	**3.2**
Reglas de transformación		3	3	4	4	3	3
Perfiles de datos		5	5	4	5	4	4
Calidad de datos		4	3	4	5	4	4
Visualización de datos		2	4	5	2	4	4
Contenido no estructurado		4	4	5	4	2	1
Estrategias de producto	**5.71 %**	**4**	**4**	**4**	**5**	**5**	**3**
Estrategias corporativas	**5.71 %**	**3**	**3**	**4**	**3.5**	**3**	**3**
Contribución del producto		3	3	5	3	3	3
Porcentaje de ganancias		3	3	3	4	3	3
Costos	**11.43 %**	**4.2**	**3.8**	**2.4**	**2.4**	**3.2**	**2.4**
Promedio del precio de venta		5	5	1	1	2	2
Estructura de precios		3	3	1	1	3	1
Modularidad de precios		5	5	5	5	5	5
Pruebas de concepto		3	3	3	3	3	3
Esquema de licenciamiento		5	3	2	2	3	1
Convenios con otros proveedores	**5.71 %**	**2.66**	**3.33**	**3.33**	**4.33**	**2.66**	**1.66**
Licenciamiento de terceros		1	2	4	5	2	1
Venta del producto por terceros		5	3	3	3	3	2
Integradores de sistemas		2	5	3	5	3	2
Finanzas de la compañía	**5.71 %**	**4**	**3**	**4.66**	**4.33**	**4.33**	**5**
Ganancias		3	2	5	5	3	5
Crecimiento de ganancias		4	2	4	3	5	5
Estado del proveedor		5	5	5	5	5	5
Total	100.00 %	168.91	171.18	190.45	186.76	172.65	158.21
Calificación total		**3.54**	**3.52**	**4.0**	**3.98**	**3.67**	**3.20**

Tabla 2.2: Evaluación de proveedores.

La recomendación proporcionada por Accenture fue Informática debido a que al ser una empresa independiente enfocada a la integración de datos, no estaba casado con ninguna base de datos específica, como sí lo están el resto de los participantes, y esto permitiría una mejor integración a las necesidades de la institución. Basados en esta recomendación la institución financiera se decidió por esta solución para implementar sus nuevos procesos ETL.

2.4. Análisis de fortalezas y debilidades de la institución financiera

Durante el proyecto se identificaron fortalezas así como mejoras y oportunidades de crecimiento dentro de la organización. Las fortalezas identificadas en la compañía son:

- Uso de una sola herramienta de ETL (Microsoft SQL 2005).
- Plataforma empresarial estandarizada (Microsoft).
- Las personas involucradas en el proyecto contaban con el conocimiento del negocio, aplicaciones y datos, necesarios para el desarrollo del mismo.
- Se tiene una claridad en las necesidades del negocio e inclusive existen iniciativas de integración, limpieza e implementación de un proceso de calidad de datos.

Por otra parte se identificaron debilidades dentro de la organización que no permitían un mejor manejo de la información así como explotarla de manera correcta. Dichas debilidades se mencionan a continuación:

- Se requería un mayor entendimiento del proceso de extracción de información.
- Mejorar la calidad de datos que se enviaban a los sistemas destino.
- No se contaba con un proceso de limpieza de datos en el sistema origen.
- No se tenía un proceso de identificación de errores y su correspondiente solución.
- No se tenía un esquema de gobierno de datos ni manejo de datos maestros.

Una vez identificadas las fortalezas y debilidades dentro de la compañía a nivel funcional, también se realizó un análisis de las capacidades tecnológicas con las que se contaba. Principalmente se enfocó el proyecto a las capacidades del flujo de datos; es decir, aquellas herramientas tecnológicas que no permitían un flujo de datos eficiente entre el sistema central y los sistemas destino. La identificación de ventajas y desventajas dentro del flujo de datos nos permitió enfocarnos en aquellos componentes que tenían que ser atendidos de inmediato.

Las desventajas identificadas en el flujo son:

1. Se contaba con múltiples procesos de extracción para obtener la misma información, ocasionando que el flujo de información entre sistemas fuera más lento.
2. Se tenía una sobrecarga de los procesos en el sistema origen de datos.
3. Se tenía redundancia de información; es decir, se repetía la información extraída.
4. No se tenía un proceso estándar de calidad de los datos.
5. No se tenía documentado el mapeo de datos entre el sistema origen y el sistema destino
6. No se tenía definido el flujo de datos maestros comunes a todos los sistemas.

Del análisis realizado se identificó que el estado final del proyecto debería de ser:

1. Extracción de datos del sistema central una sola vez y con un único proceso ETL.
2. Creación de un mapeo de datos entre sistema origen y destino.
3. En fases posteriores tener un esquema de manejo de datos maestros; así como limpieza y calidad en los datos extraídos y almacenados.
4. Contar con un sistema de gobierno de datos.
5. Establecer las bases para la creación de un modelo de inteligencia de negocios alineados al plan estratégico de la compañía.
6. Creación de un repositorio central de datos que almacene toda la información extraída y que nos permita tener una arquitectura más robusta y centralizada para el manejo de información.

2.5. Definición de los requerimientos

Existían varios requerimientos solicitados por los diversos colaboradores de la compañía con los cuales Accenture había conversado. Con base en las conversaciones se definieron los siguientes requerimientos tanto funcionales como técnicos.

2.5.1. Requerimientos funcionales de negocio

1. Analizar cada uno de los datos que están siendo extraídos de los sistemas origen por cada uno de los procesos ETL implementados en la compañía.

2. Implementar procesos de limpieza de datos durante la extracción del sistema

   - Eliminación de caracteres especiales en los nombres de los usuarios.
   - Substitución de las abreviaturas existentes en los nombres; por ejemplo: "Ma."; el cuál es interpretado como María.
   - Realizar la depuración de los RFCs de los usuarios ya sea agregando o eliminando caracteres para que éste sea correcto.

3. Realizar los siguientes procesos de transformación durante la extracción de datos del sistema origen:

   - Transformación del formato de fecha Juliana a formato DD/MM/AAAA.
   - Transformación del tipo de dato (numérico, decimal, fecha, carácter, booleano).

4. Identificar cada uno de las datos que son extraídos del sistema origen por medio de otras fuentes de información con la finalidad de identificar si existe una duplicidad en los procesos de extracción.

5. Creación de un repositorio central donde sea almacenada toda la información extraída que es requerida por los diferentes sistemas de la compañía para que desde su repositorio central se distribuya la información requerida por cada sistema.

6. Desarrollo de criterios de unificación de la información que es extraída, esto con la finalidad de optimizar los tiempos de extracción de datos que realiza cada desarrollo ETL.

7. Se requiere que los 20 archivos planos generados actualmente sean originados por la herramienta ETL que se implementará.

8. Para la generación del archivo *Currency_Exchange.txt* se requiere realizar la investigación del tipo de cambio de forma automática o con una interfaz gráfica en lugar de realizar la consulta y actualización de ese valor de forma manual de ese valor después de acceder al sitio del SAT.

9. Por necesidades del negocio y debido a que dentro del plan estratégico de la compañía está previsto el inicio de operaciones en otros países, será requerido el desarrollo de catálogos de información que tengan datos y códigos sobre el tipo de moneda, ciudades, estados, países, y códigos postales.

10. El sistema antilavado debe realizar la carga de información contenida en los archivos planos que fueron originados con la extracción, pagos a terceros y PeopleSoft de las 08:00 horas del día para el inicio de las actividades

11. Construcción de una tabla de datos cuyos datos serían extraídos del sistema central como insumo para el sistema de lavado de dinero; dicha tabla debería de permitir la operación del sistema de riesgos dentro de la compañía. Dicha tabla puede presentar hasta un día de retraso de información; esto es debido a que la información está disponible en T+1 días.

12. Para uno de los sistemas se requirió poblar 12 tablas con datos extraídos del sistema central. Dicho proceso tomó aproximadamente 2 horas.

13. Diseño y capacitación por parte del proveedor de la herramienta ETL seleccionada. Dicha capacitación incluyó los módulos para la operación y administración de la herramienta.

14. Diseño y desarrollo de un modelo que permitiera la administración de metadatos de la información que estaba siendo extraída del sistema central.

2.5.2. Requerimientos de limpieza de datos

Como parte de la definición de requerimientos funcionales que debía cumplir la herramienta, existe una sección que tenía que ver con la calidad y limpieza de datos; estos requerimientos solicitados por parte del área de tecnología comprendían que el proceso ETL realizara la limpieza de datos de los campos descritos a continuación:

- **RFC**. Se requería que el RFC se encontrara estandarizado y cumplieran con los requerimientos establecidos por el buró de crédito
- **Nombres de socios**. Se requería que los nombres de socios no tuvieran caracteres especiales como puntos, comas, retornos de carro, paréntesis, corchetes, porcentaje, comillas y caracteres de 16 bits.
- **Fechas de morosidad**. Se solicitó que las fechas para el cálculo de la morosidad se encontraran en un formato de fecha correcto yyyymmdd y que no se permitieran valores nulos o negativos para este tipo de dato.
- **Direcciones de socios**. Fue requerido que las direcciones de los socios tuvieran una nomenclatura estándar para nombres de calles y colonias.
- **Ciudades/Municipios**. Los códigos y nombres de ciudades y municipios debían de contar con un estándar y estos deberían de estar corregidos de a cuerdo a la información proporcionada por SEPOMEX.
- **Teléfonos**. Los teléfonos de los clientes debían contar con la clave lada y todos tenían que ser de 10 dígitos para números fijos y de 13 dígitos para números celulares. Los números telefónicos no deberían de tener guiones o caracteres especiales.

Como parte de la estandarización de direcciones era requerido que todos los códigos postales se encontraran conforme a la lista proporcionada por SEPOMEX y todos deberían de ser de 5 dígitos.

En el siguiente capítulo se detallará el diseño funcional y técnico del proyecto.

Capítulo 3

Diseño funcional y técnico

En este capítulo se detalla el diseño funcional de la implementación, así como la arquitectura de alto nivel y las diferentes capas que intervienen en el desarrollo del proyecto. Detallamos el diseño de los programas ETL y los componentes de la capa de datos.

3.1. Arquitectura de aplicaciones de alto nivel

La arquitectura definida para las aplicaciones destino y fuente que se consideraron dentro del alcance del proyecto se muestran en la figura 3.1; dentro de esta arquitectura de aplicaciones se consideró el proceso ETL que se construyó.

La arquitectura definida contenía al núcleo bancario como sistema fuente de información y enviaba la información a la herramienta ETL seleccionada para realizar los procesos de extracción, transformación, clasificación y carga hacia los sistemas destino y hacia cada uno de los tipos de datos que necesitaban los equipos y bases de datos destino.

3.2. Arquitectura de aplicaciones detallada

La arquitectura diseñada se encontraba dividida en cuatro capas que representan el flujo de información entre cada una de ellas. La capa de interfaz de usuario donde se realizaron todas las tareas de monitoreo, manejo de la herramienta y visualización de datos en el sistema destino. La capa de presentación únicamente contenía el servidor de correo que envía la información a la capa de interfaz de usuario. Por su parte la capa de integración contenía los procesos de la herramienta ETL; así como la base de datos *stage*

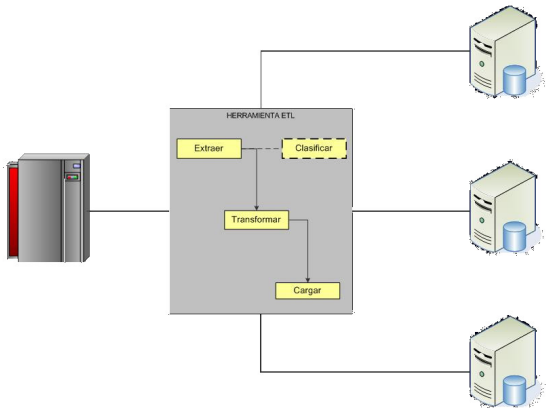

Figura 3.1: Arquitectura de aplicaciones.

y la base de datos de catálogos; esta capa también incluía a las aplicaciones que enviaban información complementaria a los sistemas destino. Finalmente contábamos con la capa de datos que contenía a los sistemas destino, el sistema fuente y al servidor de control de usuarios a través del directorio activo. La funcionalidad de cada capa así como el flujo de datos se describen a continuación.

3.2.1. Capa de datos

La capa de datos contenía diferentes funcionalidades dependiendo del servidor de datos del que estemos hablando. El núcleo bancario tenía la funcionalidad de enviar datos a la herramienta ETL a través del protocolo TCP/IP. Por sus parte el directorio activo al manejar información de los usuarios de la herramienta tenía la funcionalidad de autenticar a cada usuario que accedía a la herramienta otorgándole los permisos necesarios para ejecutar las tareas dentro de la herramienta ETL; la comunicación entre la herramienta ETL y el directorio activo se realizó mediante el protocolo LDAP.

Por su parte los sistemas destino deberían de tener la capacidad de recibir los datos de parte de la herramienta ETL; así como de parte de los ETLs actuales de la institución financiera y que complementaban la infor-

mación del núcleo bancario; los sistemas destino reciben los datos mediante el protocolo TCP/IP independientemente de la herramienta ETL que los envíe.

3.2.2. Capa de integración

La capa de integración tiene la funcionalidad de integrar los datos entre las diferentes aplicaciones; la herramienta ETL contiene todos los servicios para la extracción de datos del núcleo bancario, transformarlos y cargarlos en los diferentes sistemas destino de la capa de datos. La herramienta ETL tiene la funcionalidad de enviar los datos a la base de datos *stage*[1] así como a la base de datos de catálogos. El proceso debía enviar vía FTP los archivos de texto a una dirección específica dentro de un servidor para que el ETL del sistema antilavado tome los archivos y complemente la información que será enviada vía TCP/IP a la aplicación de usuario final. A su vez, el proceso ETL también envía información a uno de los ETLs del sistema de reportes para que sea complementada por parte del sistema de conciliación bancaria y sea enviada a la base de datos final de reportes.

3.2.3. Capa de presentación

La capa de presentación de la institución financiera contenía solamente un servidor de correo; la principal funcionalidad del servidor de correo era proporcionar a los usuarios la información de los procesos; es decir, recibía información del estado de los procesos ETL y a su vez enviaría la información de este estado al usuario final.

3.2.4. Capa de interfaz de usuario

La funcionalidad de la capa de interfaz de usuario era administrar la información que se recibe del servidor de correo así como de las diferentes aplicaciones con las que contaba la institución financiera. Esta capa era capaz de soportar un ambiente gráfico que permitía a los usuarios administrar, construir y diseñar sus propios elementos.

[1]Las bases de datos llamadas *stage* son bases de datos temporales o de paso antes de llevar la información a su destino final.

3.3. Tecnología conceptual

A continuación definiremos el diseño conceptual de la arquitectura usada para el proyecto, este diseño conceptual está dividido en cinco grandes áreas: ambiente de desarrollo, herramienta ETL, ambiente de operación, aplicaciones de la institución financiera y la capa de infraestructura; de cada una de ellas hablaremos durante este capítulo.

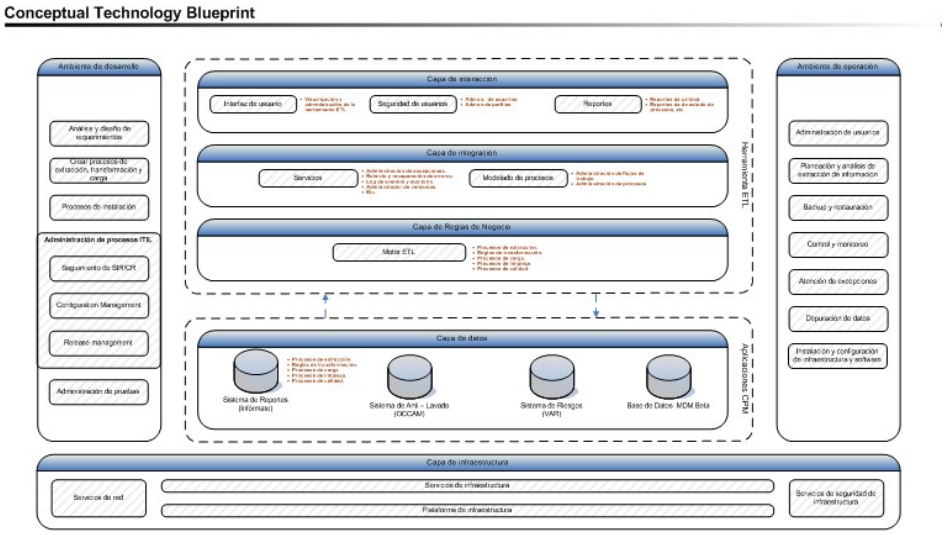

Figura 3.2: Tecnología conceptual.

3.3.1. Ambiente de desarrollo

Dentro del ambiente de desarrollo se encuentran definidas todas las actividades que se requieren tener para la construcción de los procesos ETL, así como la administración de los procesos y las pruebas. El ambiente de desarrollo tiene los siguientes componentes:

- *Análisis y diseño de requerimientos*: Dentro de la etapa de análisis y diseño de requerimientos encontramos la documentación de las fuentes y destinos de datos que serán procesadas mediante el ETL; el análisis de requerimientos abarca los requerimientos funcionales y técnicos que requiere la institución financiera para poder generar los datos en

los sistemas destino. Por su parte el diseño de los requerimientos incluyó el diseño funcional y técnico de todos los procesos a implementar dentro del ETL así como la definición de las reglas de programación y estándares que se tenían que seguir para un mejor desarrollo del proceso.

- *Creación de procesos de extracción, transformación y carga*: Una vez realizado el análisis y diseño de los procesos, el siguiente paso es construir los procesos ETL dentro de la herramienta seleccionada; para realizar este desarrollo se deberá de basar en el diseño funcional, en las reglas de transformación y en las reglas de negocio definidas dentro del propio diseño. El proceso de extracción para la primera iteración fue solamente del núcleo bancario ICBS, mientras que las transformaciones y la carga se realizó de acuerdo al sistema al que se cargaron los datos.

- *Procesos de instalación*: Los procesos de instalación se refieren a los procesos a seguir para instalar el ETL dentro de los diferentes ambientes (desarrollo, pruebas y producción); este proceso también incluyó la instalación de la herramienta de ETL en la que se desarrollaron los procesos. Parte importante del proceso dentro del ambiente de desarrollo fue la administración de los procesos de ITIL, como parte de este proceso se tenía el seguimiento de reportes de incidencias (SIR) y controles de cambio (CR), la administración de configuraciones y la administración de versiones.

- *Reporte de incidencias y controles de cambio*: Se llevó a cabo un control de incidencias y control de cambios para todos los elementos desarrollados dentro del ETL.

- *Administración de configuraciones*. La administración de configuraciones fue parte importante dentro del ambiente de desarrollo por lo que fue necesario tener un registro de todas las configuraciones que se realizaron dentro de la herramienta.

- *Administración de versiones*. Toda herramienta ETL debe tener una administración de versiones para llevar el control de todos los cambios hechos en el desarrollo.

- *Administración de pruebas*: Como todo proyecto de implementación, fue necesario llevar una administración de las pruebas; las pruebas que

se llevaron a cabo dentro de este proceso fueron: pruebas unitarias, pruebas de integración, pruebas de desempeño y pruebas de usuarios.

3.4. Diseño de los programas ETL

Como parte de la herramienta ETL se integraron tres áreas que tenían relación con los requerimientos de negocio: la capa de interacción, la capa de integración y la capa que contenía las reglas de negocio.

- Capa de interacción. Esta capa se refería a la interacción que tenían los usuarios con la herramienta ETL; así como la explotación de todos los elementos de la misma. Las principales funciones que se ejecutaron dentro de esta capa fueron la interfaz de usuario, la administración de protocolos de seguridad de usuarios y los reportes.

- Capa de integración. En esta capa hicimos referencia a todos los servicios y modelado de procesos necesarios para el buen funcionamiento de nuestra herramienta ETL. Los principales procesos que se tomaron en cuenta fueron los siguientes:

  - **Servicios**. Servicios necesarios para el manejo de errores, excepciones, versiones, así como servicios adicionales necesarios para los procesos de extracción, transformación y carga. Tales servicios se dividieron en: administración de excepciones, reinicio y recuperación de errores, registros de eventos y auditoría y administración de versiones.
  - **Modelado de procesos**. Fueron principalmente los procesos necesarios para la creación de los programas ETL.

- Reglas de negocio. Dentro de este proceso definimos las reglas para la transformación y carga correcta de los datos en la base de datos, así como la funcionalidad correcta que requería la entidad financiera. La principal herramienta que se tuvo en esta capa, fue el motor ETL y sus principales tareas fueron las siguientes:

  - **Procesos de extracción.** Definición de todos los procesos utilizados para realizar la extracción de datos
  - **Reglas de transformación.** Definición de todas las transformaciones utilizadas desde la extracción de datos hasta la carga de los mismos en la nueva estructura de base de datos.

- **Proceso de carga de datos.** Descripción de los procesos de carga de datos a cada uno de los sistemas destino.
- **Proceso de limpieza de datos.** Descripción de los procesos de limpieza de datos como: direcciones, teléfonos, nombres, RFCs y fechas.
- **Procesos de calidad.** Descripción de los procesos a seguir de acuerdo a los estándares de datos para tener una mejor calidad en la información.

3.5. Capa de datos

La capa de datos estaba conformada por los diferentes sistemas desde donde se extrajo la información así como los sistemas donde se almacenarían los datos. Como sistema para extracción de datos sólo se consideró un sistema, mientras que la carga se realizó hacia tres diferentes sistemas de la institución financiera. A continuación describo el tipo de información que se procesaba en cada uno de los sistemas destino:

- Se tenía el sistema de detección de fraudes o posible mal manejo de una cuenta; esta detección se realizaba mediante el monitoreo de las cuentas que se enviaban a cada uno de los archivos generados por el procesador principal de transacciones bancarias.
- El sistema de manejo de riesgos contenía la información de las cuentas con pagos vencidos o que presentaban algún grado de morosidad. Esta información también se obtenía del procesador principal de transacciones bancarias y sería procesada por la herramienta ETL que se implementó.
- Finalmente, existía el sistema para el manejo de los reportes operativos que requería la organización. La principal tarea de este sistema era generar reportes regulatorios previamente definidos, así como extracción de información bajo demanda.

En el siguiente capítulo presentaremos el diseño detallado de la solución implementada y cada uno de los procesos de extracción, transformación y carga.

Capítulo 4

Diseño detallado

En este capítulo se presenta el diseño detallado de la solución implementada en el proyecto. Se detalla el proceso de extracción, transformación y carga de información.

4.1. Dependencias y frecuencia de las extracciones de datos

Como primer paso dentro del proceso de extracción se identificaron las dependencias de los sistemas origen con otros procesos, tareas o reglas de negocio dentro de la organización, así como la frecuencia esperada o deseada de la extracción de datos. La extracción de datos se realizó con una periodicidad diaria; esto debido a que al tratarse del sistema principal de la institución financiera la información se actualizaba con esa periodicidad. La herramienta ETL nos permitió identificar aquellas tablas y campos que sufrían modificaciones, de tal forma que se extrajo sólo la información que sufría cambios a lo largo del día. La extracción de datos se realizó en un horario no operativo, a fin de no afectar la operación de la institución; además este proceso dependía del termino del proceso de cierre diario de operaciones. Solamente una vez que el proceso hubiera terminado comenzaba la extracción por parte de la herramienta ETL; este proceso contaba con una ventana de procesamiento de 5 horas. La extracción fue almacenada en la base de datos temporal dentro de la herramienta ETL.

4.1.1. Extracción inicial

En la extracción inicial se realizaron las siguientes transformaciones:

- Transformación de fecha juliana a fecha gregoriana.
- Cambio en los tipos de datos al momento de almacenarlos en la tabla temporal; principalmente transformación del tipo de dato *char* al tipo de dato *varchar* y los datos numéricos a decimal.

Al momento de que se terminaba la extracción de datos, se cerraba la conexión a la base de datos del procesador principal de transacciones bancarias a fin de liberar la carga de trabajo del mismo para continuar con su operación diaria.

4.1.2. Extracciones subsecuentes

Después de realizada la extracción inicial desde el sistema fuente y con los datos almacenados en el repositorio temporal, las siguientes extracciones se realizaron mediante extracciones incrementales con base en la adición, actualización o eliminación de datos de las tablas seleccionadas. Se consideró realizar la extracción total de los registros sólo en caso de que las modificaciones o actualizaciones en la base de datos fueran muchas; las condiciones para realizar la extracción completa fueron las siguientes:

1. Existían cambios en más del 50 % de los registros.
2. Los registros extraídos no se encontraban actualizados o contenían errores.
3. Los registros extraídos no cuentan con la calidad necesaria.

4.1.3. Relación entre las entidades de datos

En la figura 4.1 se puede visualizar la relación entre las diferentes entidades y los datos que comparten cada uno de ellos.

En la misma podemos observar que los diferentes sistemas manejan el mismo tipo de información; socios, cuentas, créditos y saldos.

4.2. Diseño para la transformación de datos

La transformación de los datos tenía varios procesos que se deberían seguir, entre los cuales se encontraba la limpieza de datos, los estándares y procedimientos utilizados, así como las reglas de transformación definidas para cada uno de los sistemas destino. La transformación de los datos se

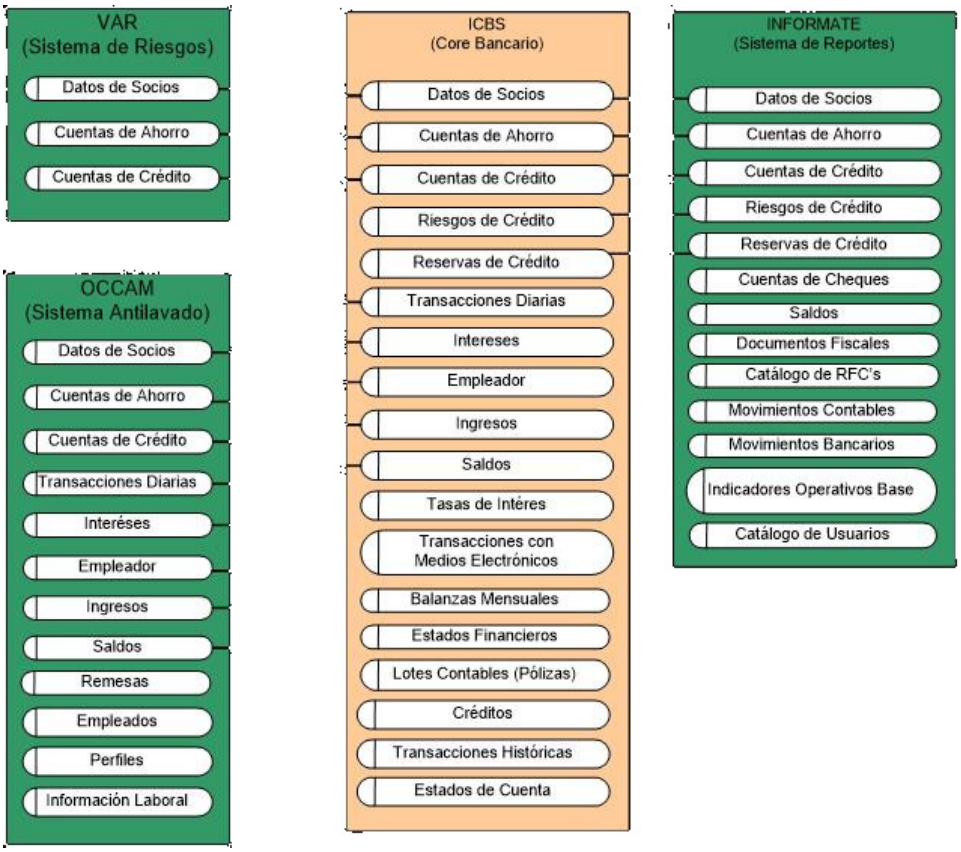

Figura 4.1: Datos relacionados en las entidades.

realizó dentro de la misma base de datos de paso y se almacenó en otra instancia de esta misma base de datos. La figura 4.2 muestra cómo se realizaron las transformaciones requeridas.

A continuación se detallan las reglas utilizadas para realizar la transformación de datos.

4.2.1. Limpieza de la fuente de datos

La limpieza de la fuente de datos **NO** estaba dentro del alcance del proyecto; sin embargo se realizaron las transformaciones de la fuente de datos para realizar la limpieza de datos hacia los sistemas destino. Como parte de un proceso de calidad de la información y limpieza de datos se

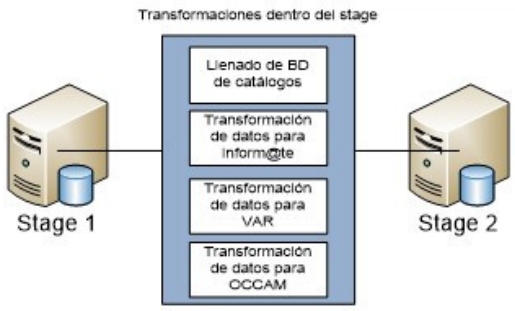

Figura 4.2: Transformaciones en base de datos temporal.

definieron los siguientes criterios:

1. Los nombres de los socios debían contener solamente caracteres alfabéticos; no se permitieron caracteres especiales como los siguientes: “\”, “.”, “#”, “$”, “%”, “&”.

2. Los números telefónicos debían contener solamente caracteres numéricos y debían ser de 10 posisciones para teléfonos fijos y 13 posiciones para teléfonos celulares.

   - Si *Tipo_Telefono* = Casa y Longitud (Telefono) =10 Entonces Telefono_Casa SiNo Error
   - Si *Celular* = Casa y Longitud (Telefono) = 13 Entonces Telefono_Celular SiNo Error

3. El RFC de los socios debería ser de 10 o 13 posiciones para las personas físicas y 12 posiciones para las personas morales. En caso de no contar con un RFC valido, el registro era rechazado y enviado a una tabla de auditoría para su corrección dentro del sistema fuente de parte del área de tecnología de la compañía.

4. Todos los códigos postales debían ser de cinco dígitos, en caso de existir algún registro con mayor o menor número, estos debían de ser enviados a una tabla de auditoría para su corrección dentro del sistema fuente. Se consideró una homologación de datos para los códigos de ciudades y estados. Todos los códigos de estados se homologaron a tres dígitos que corresponden a las tres primeras letras de cada estado; se realizó

una excepción para el caso de Chiapas (CHS) que podría confundirse con Chihuahua (CHI).

5. Las fechas se homologaron al formato yyyymmdd y no debían permitir valores nulos o fechas inválidas.

4.2.2. Estándares y procedimientos utilizados para la interrelación de los datos

Para realizar una estandarización de los datos se creó una base de datos temporal que contenía la información de todos los campos con los mismos datos pero que su sistema origen era diferente. La creación de esta tabla temporal fue crear la base para generar una tabla de referencias cruzadas que representara un gobierno de datos o MDM (*Master Data Management*) por sus siglas en inglés.

4.2.3. Reglas utilizadas para la transformación de datos

Como regla general para todos los procesos de extracción se realizaron las siguientes transformaciones.

- Transformación de fechas julianas a fechas gregorianas en formato yyyy/mm/dd.
- Transformación de los tipos de dato origen a los tipos de la base de datos *stage* generada en SQL Server.

Los procesos de transformación para cada destino fueron diferentes debido a las características de cada uno de los archivos de salida. Dichas transformaciones se realizaron tomando como principal fuente de datos la base de datos temporal.

Sistema de reportes

- Todos los campos de fecha que tengan un sufijo _d deberían de ser convertidas a formato de fecha (yyyymmdd) para todas las tablas de este sistema. Se pobló la fecha de vencimiento _D, tomando como base el campo fecha de vencimiento de la tabla de facturación. La regla de transformación que se definió fue la siguiente:

```
FEC_VENCIMIENTO_5_D = CAST(
  RIGHT(
```

```
LTRIM(
  RTRIM(
    FEC_VENCIMIENTO_5)
  ), 2
) +
LEFT(
  RIGHT(
    LTRIM(
      RTRIM(
        FEC_VENCIMIENTO_5
      )
    ), 4
  ), 2
) +
REPLICATE(
  '0', 2 - (
    LEFT(
      LTRIM(
        RTRIM(
          FEC_VENCIMIENTO_5
        )
      ),
      ABS(
        4 - len(
          LTRIM(
            RTRIM(
              FEC_VENCIMIENTO_5
            )
          )
        )
      )
    )
  )
) +
LEFT(
  LTRIM(
    RTRIM(
      FEC_VENCIMIENTO_5
    )
  ),
  ABS(
    4 - len(
      LTRIM(
        RTRIM(
          FEC_VENCIMIENTO_5
        )
      )
    )
  )
```

```
  ) AS DATETIME
) WHERE FEC_VENCIMIENTO_5 > 0
```

- Para el campo de fecha de último saldo se tomó la fecha de vencimiento más antigua que se tenía.
- Para obtener el número de créditos de un socio se realizó la siguiente regla.

```
NO_CREDITOS =  NO_CREDITOS
  WHERE NO_SOCIO IN NO_SOCIO de la tabla CIF_CREDITOS.
Si NO_CREDITOS = NULL Entonces NO_CREDITOS = 0
```

- La descripción de la información correspondiente a las garantías debía de estar vacía en la tabla destino.
- Para almacenar la fecha de último monto con saldo se tomó la fecha de vencimiento más antigua que se tenía para cada registro dentro de la tabla de saldos LN_SALDOS.
- Se realizó una validación sobre el tipo de cuenta para confirmar su valor igual a 1 (Personas físicas) o 6; en estos casos se colocó un valor 20 dentro del campo Tipo_producto de la tabla TA_SALDOS.
- El número de socio para poblar la tabla TA_SALDOS se obtuvo de la tabla de cuentas por socio, pero solamente de aquellas cuentas cuyo tipo de producto era 4 u 8 y que tuvieran un saldo en la tabla TA_SALDOS.
- Se calculó el número de créditos de cada socio y este valor se grabó en la tabla CIF_INFGEN; en caso de que el número de créditos fuera nulo se colocó un valor por omisión cero.
- Se insertó el número de socio dentro de la tabla TA_CTAXSOCIO para aquellos números de cuenta menores a 20000000000 y cuya clave de relación fuera 'SOW' u 'OWN'. El número de socio eran los 10 dígitos de la derecha de cada número de cuenta.
- Se definió que para el campo descripción perteneciente a la tabla LN_GARANTIAS debería poblarse con un valor NULL a pesar de tener mapeado un campo del cuál se extraía la información.

- El campo Periodo de la tabla LN_SALDOS se asignó con un valor fijo = 'M'.
- El campo frecuencia de la tabla LN_SALDOS se asignó con un valor fijo = 30.

La regla de transformación que se utilizó durante la extracción.

Reglas para el sistema para prevención de lavado de dinero

Las reglas definidas para este sistema son las siguientes:

- Se definieron algunos valores estáticos para los tres primeros campos de cada uno de los archivos generados; estos valores son los siguientes:
  1. Valor fijo 'CMPS' para el primer campo.
  2. Valor fijo 'T' para el segundo campo.
  3. Valor fijo 'F' para el tercer campo.
- Para la generación del archivo de cuentas se tomaron sólo aquellas cuentas que tuvieron movimientos durante el día.
- Solamente se tomaron en cuenta los siguientes tipos de cuenta: Parte social, cuenta mexicana, servicuenta y cuentamiga.
- Si el tipo CIF es una persona moral (CUTYP = 5), entonces el campo CompanyLegalId se pobló con dicho valor, en caso contrario se asignó un valor constante vacío "".

4.3. Arquitectura aplicativa

La arquitectura aplicativa requerida para un proceso de extracción, transformación y carga de datos está comprendida por una herramienta ETL y algunos componentes de software adicionales para lograr la integración de la solución.

El modelos de aplicaciones por lotes (*batch*) que se muestra a continuación ilustrará los componentes de la arquitectura que fueron requeridos para la implementación del proyecto. Este modelo fue desarrollado con base en las mejores prácticas para el uso de aplicaciones por lotes de tipo ETL.

4.3.1. Capas de la arquitectura

El esquema de las aplicaciones tipo por lotes incluye los servicios necesarios para apoyar los procesos que se ejecutan durante largos periodos de tiempo y que ejecutan tareas de manera repetitiva. En este esquema de aplicación, un programa generalmente lee un número grande de registros de una base de datos o de un archivo, procesa los datos y luego escribe datos transformados a una base de datos o a un archivo.

Del marco de trabajo (*framework*) de aplicaciones por lotes encontramos las siguientes capas principales:

- Ejecución.
- Proceso por lotes.
- Aplicaciones por lotes.
- Negocios.
- Datos.

El marco de trabajo comprende componentes que brindan servicios a las capas principales de la cuales son:

- Servidores de aplicación.
- Flujos de trabajo de usuarios.
- Captura de datos.
- Servicios por lotes.
- Servicios de reportes.
- Servicios de gestión.
- Servicios de seguridad.
- Servicios comunes.

En las siguientes secciones se describirá la responsabilidad de cada una de las capas.

4.3.2. Componentes de la capa de presentación

Dentro de la capa de presentación se tienen varios componentes entre ellos el panel de consulta y la interfaz de usuario que nos ayuda a la revisión y monitoreo del proceso por lotes; así como los resultados que éste arroja.

4.3.3. Servicios de la capa de presentación

Los servicios que brindan los componentes de esta capa de a acuerdo al marco de trabajo utilizado son los siguientes:

1. Monitoreo de programa.
2. Reportes.
3. Servicios comunes.

Monitoreo de programas

Este servicio proporciona las herramientas para monitorear cuáles programas y aplicaciones están siendo ejecutados, así como la información a nivel sistema del uso de recursos. El monitoreo de programas puede actuar como entrada principal en las decisiones de balanceo de cargas.

Reportes

Reportes permiten al usuario dar formato a los resultados de una consulta dentro de la producción de reportes de calidad.

4.3.4. Componentes de la capa de ejecución

Los componentes encontrados en esta capa de acuerdo al marco utilizado son los siguientes:

- Calendarizador de tareas (*scheduler*).
- *Script* de ejecución de programas.
- Proceso por lotes.

Los componentes de esta capa son los encargados de comenzar con el proceso de ejecución y se describen a continuación:

Calendarizador

El servicio de calendarización proporciona la estructura de control para programas por lotes; esto es, administra el flujo de su procesamiento. Un proceso de calendarización debe de proporcionar una interfaz de usuario para entrar y dar mantenimiento a la información y a la calendarización de los procesos por lotes.

Scripts de ejecución de programas

Este servicio es responsable de establecer y restaurar el ambiente en el cuál el proceso por lotes es ejecutado; también debe ser capaz de ser ejecutado por el calendarizador así como recibir y/o enviar parámetros del calendarizador al proceso por lotes, tales como el nombre del proceso a ser ejecutado. El *script* debe proporcionar los procedimientos y guía necesarios para ejecutar el proceso por lotes.

Proceso por lotes

Un proceso por lotes controla la ejecución de la aplicación. Un proceso por lotes es típicamente un script de ejecución que control el orden del proceso, nombres de archivos de entradas/salidas y dependencias entre procesos por lotes.

4.3.5. Servicios de la capa de trabajos programados (*Jobs*)

Los servicios que soportan a los componentes de esta capa de acuerdo al framework utilizado son los siguientes:

- Reinicio / Recuperación.
- Trabajos particionados.
- Servicios comunes

Reinicio y recuperación

El servicio de reinicio y recuperación permite a los proceso programados apagarse y reiniciarse en puntos apropiados después de una interrupción anormal o de la intervención del usuario.

Trabajos particionados

Usando una partición, permite que múltiples versiones de aplicaciones programadas se ejecuten concurrentemente. El propósito de esto es reducir el tiempo transcurrido que se requiere para procesar la información.

4.4. Arquitectura de negocio

Esta capa se divide en 4 grandes módulos que contienen toda la lógica de negocio los cuales son:

1. Extracción
2. Limpieza.
3. Transformación.
4. Carga

4.4.1. Módulo de extracción

Extracción es el proceso que consiste de los módulos que extraen los datos de una fuente(s) de datos determinada. El módulo de extracción está formado de los siguientes componentes.

1. Monitor de tareas programadas.
2. Monitor de *Triggers*
3. Adaptadores de fuente de datos.
4. Filtrado y búsqueda de datos.
5. Estandarización de tipos de tato

4.5. Arquitectura de datos

Como se explicó en los capítulos anteriores, la institución financiera contaba con diferentes procesos ETL que extraían información del sistema principal. Estos sistemas trabajaban de manera independiente por lo que cada sistema destino tenía su base de datos temporal antes de enviarla a su destino. La arquitectura se encuentra detallada en la figura 4.3.

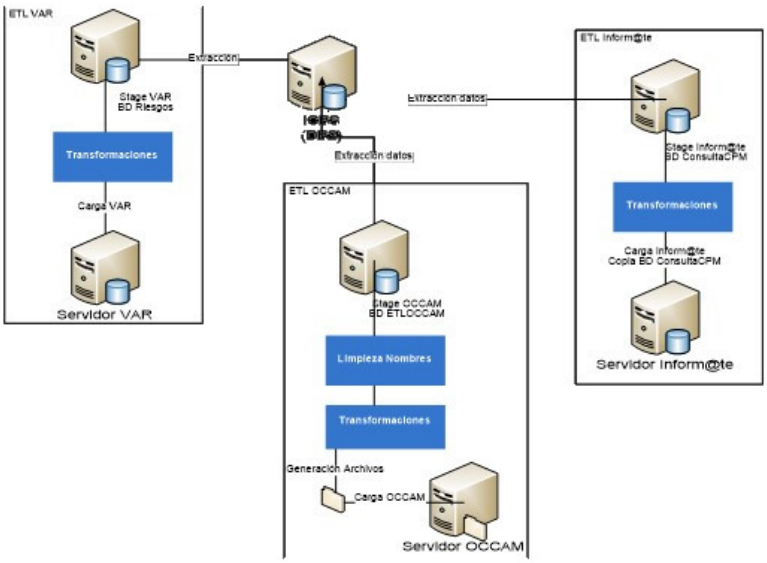

Figura 4.3: Arquitectura de datos actual.

Como se observa en la figura 4.3, existían muchas bases de datos intermedias entre la fuente de datos y los sistemas destino; la arquitectura definida dentro de este proyecto, nos permite tener una sola base de datos temporal, así como una sola capa de integración de datos. El siguiente diagrama muestra el diseño de la arquitectura desarrollada:

4.6. Arquitectura de infraestructura

Como parte de la solución propuesta se describirá la infraestructura con la cuál la institución financiera debe de tener en los diferentes ambientes para la implementación de la solución ETL. A lo largo del capítulo se especifican los componentes de hardware y software necesario.

4.6.1. Ambiente de producción

Se describe a continuación las especificaciones de los servidores físicos requeridos.

El diagrama de red de la infraestructura mencionada se muestra a continuación

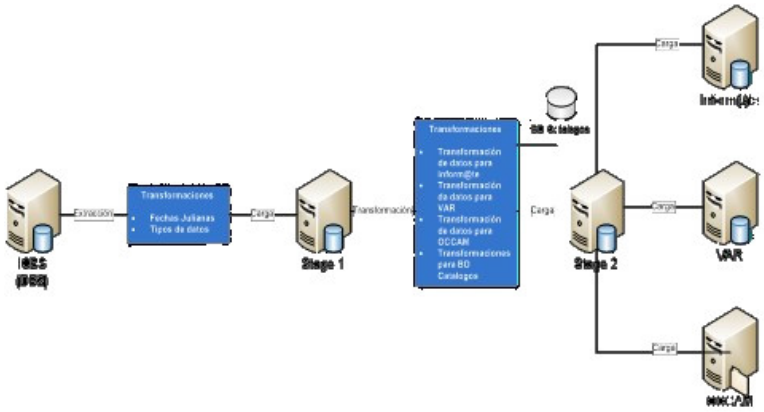

Figura 4.4: Arquitectura de datos final.

Modelo	Servidor	Sistema Operativo	
iSeries	ICBS PRO	OS400	
Dell tipo rack	Informate PRO DB	Windows 2003 estándard	Con
Dell tipo rack	Informate PRO DB	Windows 2003 estándard	
HP Proliant BL 460c G1 Tipo Blade	OCCAMPRO	Windows 2003 estándar	Visual es

Tabla 4.1: Infraestructura requerida.

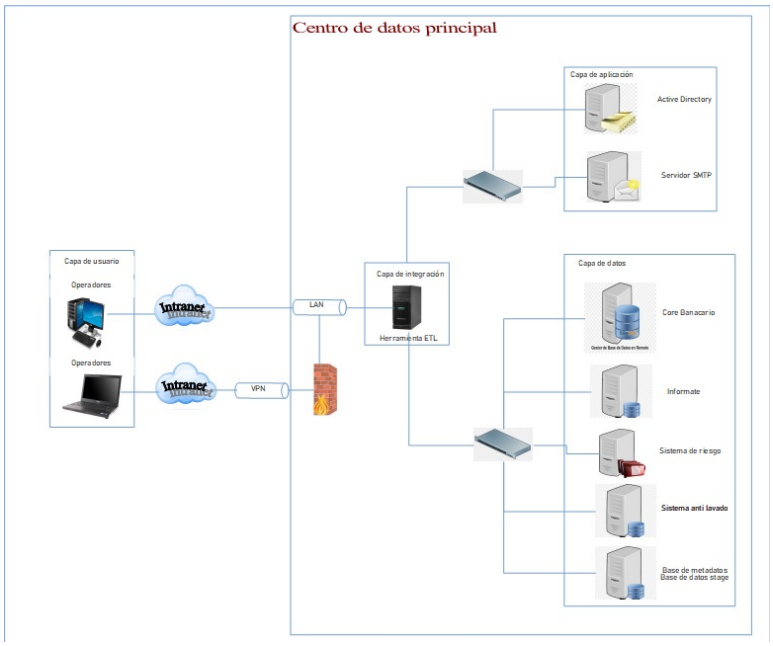

Figura 4.5: Infraestructura de aplicación.

Capítulo 5

Modelo de datos

En este capitulo se mostrará el diseño de datos lógico y físico generado para el proyecto que estamos presentando, correspondiente a la reestructuración del sistema actual.

El modelo de datos lógico diseñado, muestra la reestructuración de la base de datos de información que se realizó en la institución financiera. Como lo mencionamos en el presente trabajo, la institución contaba con muchas fuentes de datos que hacian muy complicado el manejo de información así como el gobierno de datos.

El diseño del modelo se basó en identificar las entidades principales como socios, cuentas, transacciones y créditos para centralizar toda la información de los diferentes sistemas además de tener una fuente de datos para la generación de los reportes de las diferentes áreas.

En la siguiente imagen se muestra el diseño lógico generado.

5.1. Modelo de datos lógico

5.2. Modelo de datos físico

Una vez generado el modelo de datos lógico, se procedió a diseñar el modelo de datos físico que se implementó a nivel base de datos.

Este modelo se presenta en la siguiente imagen:

5.3. Modelo de datos físico

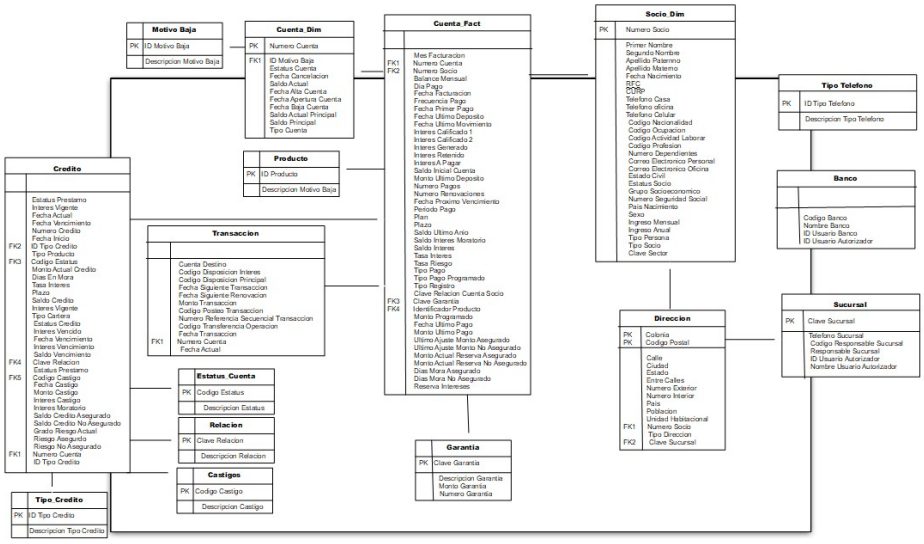

Figura 5.1: Modelo lógico de la base de datos.

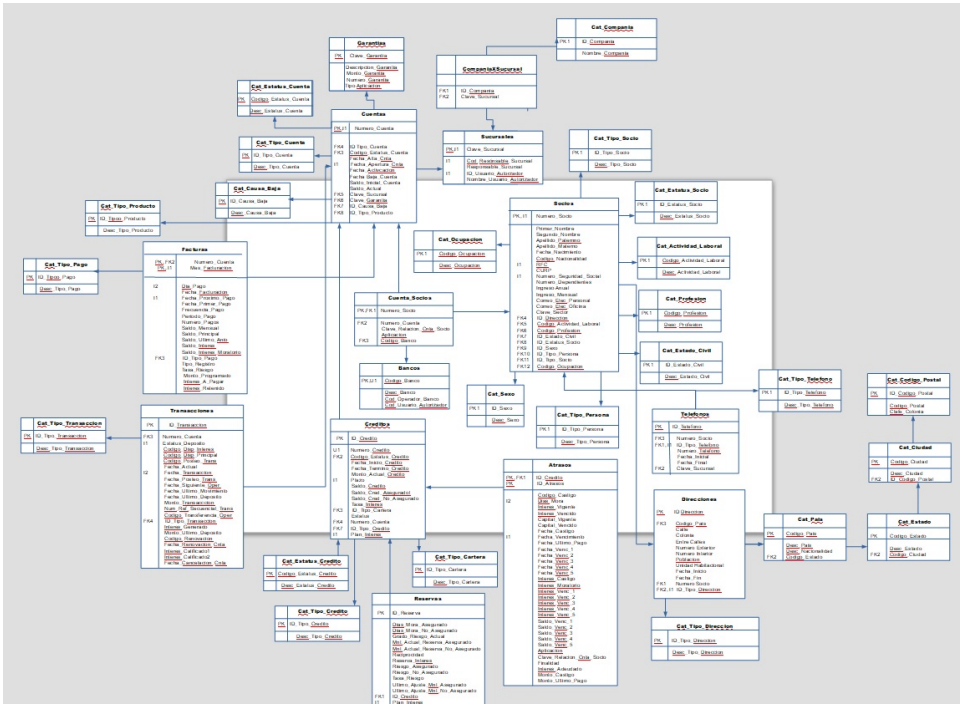

Figura 5.2: Modelo físico de la base de datos.

Capítulo 6

Estándares de programación

La convención de nombres tiene la ventaja de mejorar la lectura y comprensión de la aplicación, tanto para el desarrollo como para el mantenimiento; así mismo, tiene soporte para ciertos aspectos de la operación debido a que permite una mejor interpretación de la información contenida en cada elemento de la base de datos o procesos de ETL. Varios aspectos son posibles o más fáciles si se tienen definidos los estándares y convenciones de nombres:

- Actualizaciones (cambio de versiones) a la base de datos o software.
- Cambio al modelo de datos físico o lógico.
- Mover una aplicación de un servicio a otro.
- Mejor administración de la base de datos (uso de utilerías y *scripts*).

Los aspectos que serán cubiertos en este capitulo serán:

- Lineamientos de la base de datos.
- Uso de nombres y convenciones.
- Reglas para crear nombres de los objetos de las bases de datos.
- Recomendaciones para crear nombres de proyectos.

6.1. Lineamientos de base de datos

Algunos de los lineamientos que se siguieron al momento de realizar la construcción de la base de datos se enlistan a continuación:

1. En cada construcción de consultas, se verificó siempre los planes de ejecución; es decir, reducir al máximo el uso de memoria y capacidad de las máquinas. Además evitar que las consultas respondieran con escaneo de índices (`index scan`) o escaneo de tablas (`table scan`).
2. Evitar el uso del enunciado `WHERE ... IN` y utilizar `WHERE EXIST`.
3. Evitar el uso del `OR` y sustituirlo por otro tipo de consulta o por el enunciado `CASE`.
4. Evitar hacer uso de los operadores como `NOT` o <>.
5. En la ejecución de consultas, después del enunciado `WHERE`, hacer siempre referencia a las columnas con índices.
6. En enunciados de inserción de registros (`INSERT`) siempre colocar el nombre de las columnas. Lo anterior ayuda a evitar problemas de carga/inserción derivado de cambios en la estructura de la tabla.
7. Evitar el uso de comodines; es decir, escribir siempre el nombre de las columnas requeridas para reducir el tráfico en la red.
8. Mantener los enunciados SQL lo más concisos posibles.
9. Para el proceso de carga masiva de datos, se recomendó borrar primero los índices antes de iniciar la carga, reconstruyéndolos al final del proceso.
10. Siempre comentar el código generado.
11. No usar el prefijo "`_sp`" para nombrar procedimientos alamacenados (`stored procedures`).
12. Uso de limitadores en la obtención de registros de cualquier tabla.
13. Evitar el uso de tablas temporales en el procesamiento de datos. En algunos casos se utilizaron tablas temporales en lugar de consultas complejas.
14. Evitar el uso de cursores. En su lugar se utilizaron enunciados `WHILE`.
15. Después de la cláusula `ORDER BY`, poner el nombre de las columnas en lugar de valores numéricos que hagan referencia al orden de las columnas.
16. Cumplir con los estándares definidos para la base de datos.

6.2. Reglas para crear nombres de objetos en la base de datos

Las siguientes reglas fueron definidas para realizar objetos dentro de la base de datos.

- **Caracteres.** Se debía cumplir que:
  - Los nombres de los elementos debían consistir solamente de letras(A-Z), números (0-9) y guión bajo (_).
  - Los nombres no debían tener espacios.
  - Los nombres no debían tener comas o comillas
- **Longitud.** Si bien la longitud de los nombres es propia de cada herramienta de software, en el caso de Oracle no puede ser mayor a 30 bytes con las siguientes excepciones: nombre de base de datos hasta 8 bytes y nombre de ligas de base de datos hasta 128 bytes.
- **Palabras reservadas.** No estaba permitido el uso de palabras reservadas.
- **Separadores.** Uso del carácter '_' para nombres de objetos de base de dato que estuvieran compuestos de varias palabras. No usar '_' en el nombre de la base de datos.
- **Identificadores numéricos (##).** En las convenciones de números, el símbolo `##` es usado muchas veces para indicar que varios objetos de la base de datos deben de diferenciarse unos de otros.

6.3. Recomendaciones para crear nombres de proyectos

Una de las metas de utilizar convenciones de nombres es hacer que los nombres utilizados sean lo más simple posible, ya que el nombre debe ser fácil de recordar y debe seguir ciertas reglas que sean igualmente fáciles de explicar.

Las siguientes reglas no fueron obligatorias para el proyecto, pero se tomaron algunas de sus recomendaciones.

- **Nombres de proyectos.** Para la definición de nombres se deben cumplir dos metas principales: el nombre del objeto debe ser tan corto como sea posible y debe ser autodescriptivo. Una regla que hay que seguir es que cuando el nombre no pueda ser corto se debe elegir un nombre autodescriptivo pensando en que probablemente no sea la misma persona la que realice la implementación y el mantenimiento.

- **Verbos.** Si los nombres de los objetos contienen un verbo, siempre se tiene que iniciar la palabra con él. Por ejemplo es mejor decir `Buscar_Nombre_Cliente` que `Nombre_Cliente_Buscar`.

- **Palabras simples.** Utilizar palabras simples para describir un objeto. Usar `Buscar_Complemento` o sólo `Complemento` en lugar de `Buscar_Y_Complementar`.

- **Expresiones cotidianas.** Las expresiones cotidianas no deben ser usadas ya que pueden causar confusión al momento de leer o implementar.

- **Uso de formas singulares y plurales.** Las formas singulares son más comunes que las plurales en el uso diario; son mas cortas y simples de entender, por lo que es recomendable usarlas en lugar de las plurales. Como una regla de consistencia, cuando se define el nombre de la base de datos se puede utilizar la forma singular o plural, pero no ambas.

- **Uso de abreviaciones.** Si es necesario tener que usar abreviaciones, las siguientes recomendaciones nos pueden ayudar a definir una buena abreviación (tomaremos como ejemplo la palabra `ERROR`).

    - Obtener todas las consonantes o caracteres especiales de la palabra (`ERROR` sería `RRR`).
    - Si la palabra original comienza con vocal, mantener la vocal en su lugar (`ERROR` sería `ERRR`).
    - Si la regla antes descrita resulta en una abreviación que contiene más de dos consonantes idénticas, solo dos consonantes deben permanecer (`ERROR` sería `ERR`).
    - Si el resultado de la abreviación es difícil de entender, usar una abreviación que sea más comprensible. Por ejemplo usar `PROD` para describir producto en lugar de `PRDCT`.

6.4. Nombres estándar para la base de datos

Parte importante dentro de la programación son los nombres usados en la base de datos; a continuación se definen algunas recomendaciones que se deben seguir para tener la información en nuestro sistema de una forma entendible y fácil de leer.

6.4.1. Estándares para la creación de un modelo lógico

Atributos

1. Todos los nombres lógicos deben ser expresados en términos de negocio con las siguientes excepciones:
   - Acrónimos de negocio similares pueden ser usados de manera limitada.
   - El nombre `ID` debe ser usado siempre para identificadores.
2. Todos los nombres deben de iniciar con letra mayúscula.
3. Todos los nombres son singulares.
4. Las entidades deben de ser únicas dentro del modelo.
5. No se deben usar abreviaciones al menos que sea requerido por consideraciones de espacio y se deberán seguir las recomendaciones de la sección anterior.
6. Los nombres deben de consistir e iniciar con caracteres alfabéticos. Los números no son recomendables pero pueden ser usados después del inicio en caso de ser requeridos. Caracteres especiales no deben de ser usados.
7. Múltiples términos en el nombre deben estar separados por espacio, las diagonales o guiones no pueden ser usados.

Entidades

1. Un nombre de entidad debe ser un nombre o una frase en forma singular.
2. Todos los nombres de entidades son nombres de negocio y pueden terminar en una palabra que los clasifique.

3. Entidades asociadas deben ser creadas usando un nombre con un significado de negocio en lugar de la concatenación de 2 entidades.

4. La entidad asociada debe ser sufijo con una abreviación para la referencia cruzada.

Relaciones

Todas las relaciones tienen significado de una frase verbal y deben estar en minúsculas.

Definiciones

Las definiciones deben ser una frase detallada de los atributos que típicamente contiene referencias al sujeto componente del atributo. La definición debe consistir de varios enunciados que describan concisamente los datos. Ejemplos de datos deben proporcionarse siempre que sea posible.

Propiedades definidas por el usuario

Para todos los objetos, los comentarios deben ser proporcionados en las propiedades definidas por el usuario. Las funciones definidas por el usuario (`UDF`) son muy similares a los procedimientos almacenados con la diferencia de que estos pueden ser usados en los enunciados `SELECT`; si no fuera así, tanto los procedimientos almacenados como las funciones definidas por el usuario serían iguales. Las funciones definidas por el usuario deberán de tener el sufijo `_udf`.

6.4.2. Estándares para la creación de un modelo físico

Nombre de tablas

La regla principal para crear los nombres de tablas físicas es que deben estar basadas en los nombres de las tablas descritas en los estándares del modelo lógico visto en la sección anterior, con excepción de las siguientes reglas.

- El guión bajo reemplazará a los espacios que dividen los términos en el nombre.

- Uso de términos estándar apropiados. Los nombres deben ser amigables al usuario y consistentes con el mismo término utilizado por el modelo lógico.

- Los nombres no deben de contener palabras reservadas propias de la base de datos como: `type`, `name`, `column`, `row`, etc.
- No se deben usar acentos, diéresis ni la letra `ñ`.

Nombres de elementos de las tablas

Los fundamentos para crear los nombres de los elementos físicos de la base de datos deben de estar basados en los nombres de los atributos de negocio definidos en los estándares para el modelo lógico con las siguientes excepciones:

- El carácter '_' reemplazará a los espacios que separan los términos en el nombre
- Uso de términos estándar apropiados. Los nombres deben de ser amigables al usuario y consistentes con el mismo término utilizado en la definición del modelo lógico.
- Las llaves técnicas o sustitutas deben tener el sufijo `_TK`.
- Los nombres de las restricciones (*constrains*) deben ser de la siguiente forma:
  - `Nombretabla` es el nombre de la tabla para el cual la restricción aplica.
  - `tipo` identifica el tipo de restricción y solamente puede ser una de las siguientes opciones:
    - `pk`, llave primaria.
    - `fk`, llave foránea.
    - `c`, validar restricción.
  - `desc` es información adicional correspondiente al nombre de las columnas usadas en la restricción o a su uso.

Restricciones por omisión y de comprobación

Se debe de usar el nombre de la columna a la cuál la restricción está unida y colocar un sufijo `def` para restricciones por omisión (*default*) y `chk` para restricciones de comprobación (*check*) respectivamente.

Nombres de procedimientos almacenados

Los procedimientos almacenados (*stored procedures*) realizan tareas por el usuario por lo que es bueno seguir una serie de reglas para la definición de los nombres. Las reglas que se deberán seguir son las siguientes:

- Usar la plantilla `sp_x_yyyyyy` para nombrar a los procedimientos almacenados, donde:
  - `sp` denota que se trata de un procedimiento almacenado.
  - `x` denota el tipo de procedimiento almacenado a ejecutar:
    - `d` procedimiento de borrado.
    - `i` procedimiento de inserción.
    - `s` procedimiento de selección.
    - `u` procedimiento de actualización.
    - `x` procedimiento que combina inserción, selección y actualización.
  - `yyyyyy` descripción del procedimiento almacenado.
  - Usar un verbo para describir la acción del procedimiento almacenado.
  - Opcionalmente se puede usar un prefijo que agrupe los procedimientos almacenados de una misma tabla.

Variables

Para variables que almacenan el contenido de las columnas, se puede utilizar la misma regla que utilizamos para nombrar columnas. A continuación se describen las reglas generales:

- No definir nombres largos y complicados para las tablas u otros objetos de las bases de datos.
- No usar espacios en los nombres de los objetos de la base de datos ya que los espacios hacen confuso el acceso a las herramientas y aplicaciones de la base de datos. En el caso obligatorio de tener que usar espacios, asegurarse de colocar el nombre dentro de corchetes.
- No usar palabras reservadas para nombrar objetos de base de datos, porque esto puede resultar en situaciones impredecibles.

Capítulo 7

Conclusiones

Las bases en desarrollo de software, bases de datos, estructuras de datos, lenguajes de programación y, por encima de todo, análisis y solución de problemas que obtuve dentro de la carrera de Ciencias de la Computación, me ayudaron a que el proyecto que presento en este trabajo se realizara de manera exitosa.

La implementación de este tipo de proyectos de reestructuración de información está siendo un tema recurrente en muchas instituciones ya que no se tiene, en muchos casos, una cultura de centralización de información y sobre todo de gobierno de datos. Las instituciones deben de poner especial atención a la consistencia de la información de manera que todas las areas tengan acceso a los mismos datos, evitando así que cada área opere de manera indepeniente y tenga sus propios procesos o que presenten diferentes reportes por tener acceso a diferentes fuentes de información.

Como se mencionó anteriormente, los conocimientos adquiridos durante la carrera de ciencias de la computación me permitieron llevar a cabo con éxito la implementación de este proyecto. La metodología aprendida en la ingeniería de software y las clases de bases de datos permitieron realizar un análisis detallado de la problematica y de esta manera proponer un modelo nuevo para la institución financiera.

Agradezco todas y cada una de las clases porque gracias a ellas tuve el éxito que se requería para el proyecto.