



Instituto Politécnico Nacional

**Escuela Superior de Física y
Matemáticas**

**Diseño y construcción del Sistema de
Control de Indicadores para Teléfonos de
México**

T E S I S

MEMORIA DE EXPERIENCIA PROFESIONAL

Que para obtener el título de licenciado en Física y
Matemáticas

P R E S E N T A

Francisco Javier De la Vega Magaña



México, D. F., Julio de 2006

A mis padres, esposa e hijos que siempre han creído en mí.

A mi familia, amigos y personas que han sido parte de mi vida.

La vida misma es un regalo que Dios me dio. Le agradezco por todo lo que me ha dado en mi existencia.

Doy gracias a mis padres, por los cuidados y consejos que me dieron en mi formación como persona, y especialmente, por su cariño y confianza en mí.

A mi esposa que ha contribuido enormemente para terminar mis metas, por su apoyo y amor. A mis hijos, porque su sonrisa me da aliento para superarme.

Agradezco a mis amigos que me han impulsado constantemente con sus consejos para concluir esta meta.

A mi profesor y asesor Germán Gonzáles Santos, porque en él encontré una imagen del profesionalista honesto y dedicado, porque su enseñanza fue más allá que una simple clase, y porque me hizo comprender, con sus actos, que el esfuerzo rinde frutos que no se destruyen fácilmente.

Agradezco al personal de la Subdirección de Sistemas de Información Corporativa de Telmex, en especial a Héctor Palafox y Arturo Celada, por su apoyo para elaborar este trabajo.

Contenido

1. INTRODUCCIÓN	9
2. PROBLEMÁTICA EN EL PROCESAMIENTO DE INDICADORES EN LA SUBDIRECCIÓN DE SISTEMAS DE INFORMACIÓN CORPORATIVA .	13
2.1 Antecedentes	13
2.1.1 La importancia de la toma de decisiones en Telmex.	13
2.1.2 El proceso de la toma de decisiones	15
2.1.3 La información en la toma de decisiones	16
2.1.4 Operación de la SSIC.....	18
2.2 Análisis situacional de la SSIC.....	22
3. DISEÑO DEL SISTEMA.....	29
3.1 Modelos del diseño	29
3.1.1 Diseño de datos.	30
3.1.2 Diseño de procesos.....	41
3.1.3 Diseño de infraestructura.	53
4. DESARROLLO DEL SISTEMA.....	57
4.1 Administración del sistema	57
4.1.1 Módulo de registro de indicadores	58
4.1.2 Módulo de construcción de fórmulas	59
4.1.3 Módulo de administración de cargas.....	61
4.2 Algoritmos importantes	64
4.2.1 Proceso de agregación.....	66
4.2.2 Proceso de acumulación.....	69
4.2.3 Proceso de cálculo de indicadores	71
4.3 Implantación del sistema	79

5. RESULTADOS.....	83
5.1 Resultados obtenidos	83
5.2 Discusión sobre los resultados obtenidos.....	84
6. CONCLUSIONES Y DESARROLLOS FUTUROS.....	87
7. BIBLIOGRAFÍA.....	89
APÉNDICE A DIAGRAMA DE ACTIVIDAD	91
APÉNDICE B DISEÑO DE BASE DE DATOS RELACIONALES	95
B.1 Entidad.....	96
B.2 Relación.....	96
B.3 Atributo.....	97
B.4 Identificador	97
APÉNDICE C CÓDIGO DE LOS PROCESOS.....	99
C.1 Principal de agregación y acumulación.....	99
C.2 Proceso de agregación.....	103
C.3 Proceso de acumulación.....	111
C.4 Principal de cálculos.	119
C.5 Recursividad de cálculos.....	129
C.6 Proceso de cálculos.	132
C.7 Obtención de datos.....	141

Capítulo 1

Introducción

La privatización de Teléfonos de México (Telmex), en diciembre de 1990, representó para la nueva dirección de la empresa establecer un plan de trabajo a mediano y largo plazo, con objetivos específicos de crecimiento y modernización de los servicios de las telecomunicaciones, así como ofrecer nuevos productos y servicios de la mayor calidad.

A la fecha, el crecimiento de la empresa ha sido alcanzado en los distintos servicios y productos que ofrece. Algunos de los servicios que han tenido un crecimiento importante son:

- Líneas telefónicas: 320%.
- Teléfonos públicos: 780%.
- Se crea el servicio de Internet y tiene casi 2 millones de usuarios.
- Poblaciones con servicio: 200%.
- Presencia de Telmex en otros países con los servicios de telefonía local: Argentina, Brasil, Colombia, Chile, Perú y Estados Unidos de América.

Decidir a que mercados incursionar, que áreas fortalecer, que nuevos nichos de mercado establecer o que nuevos negocios desarrollar, es parte fundamental para alcanzar los objetivos de la empresa. Tomar las decisiones correctas para lograr los objetivos corporativos es una responsabilidad enorme para el desarrollo y crecimiento de la empresa. La omisión de alguna suposición, restricción o regulación puede ser fatal para la empresa misma.

La información es un elemento importante en la toma de decisiones de la empresa. La recopilación debe ser confiable y oportuna. El tipo de información requerida, el momento de obtenerla y donde encontrarla son decisiones de los directores, junto con los gerentes de las áreas dedicadas a la obtención y concentración de los datos.

Los indicadores son uno de los elementos de información utilizados en Telmex para entender los asuntos de su negocio y empleados en la toma de decisiones; ya que son los que sintetizan situaciones importantes y, en la mayoría de los casos, interesa conocer su comportamiento temporal.

La concentración de la información estratégica, su procesamiento y su publicación son parte de las funciones de la Subdirección de Sistemas de Información Corporativa de Telmex (SSIC).

El presente trabajo, muestra el diseño y construcción de uno de los principales sistemas de información de esta subdirección: el Sistema de Control de Indicadores (SCI). Desarrollado como herramienta de trabajo para procesar los datos solicitados de las diferentes fuentes para obtener los valores de los indicadores, el SCI facilita el trabajo del personal operativo de la subdirección y minimiza la operación manual y semiautomática que realizaba anteriormente.

El documento contiene un panorama del diseño y construcción del SCI, pasando por la descripción de la solución propuesta y los resultados obtenidos de la construcción de dicho sistema de información.

El capítulo 2 inicia con un resumen de los eventos importantes de Telmex en los últimos 15 años. Se enfatiza la importancia que ha tenido la toma de decisiones en el crecimiento de la empresa, a través de los años, y el papel que ha jugado la SSIC en el procesamiento de la información que utilizan para la toma de decisiones. En este capítulo, también se describen los antecedentes de los problemas existentes en la obtención de los indicadores y se presenta, en forma resumida, el análisis de la situación de ese momento, para

determinar como el sistema de información ayudaría en la solución de los problemas detectados.

El capítulo 3 presenta los objetivos del SCI. Además, algunos de los modelos utilizados para el diseño del sistema: modelo de datos, modelo de procesos y modelo de la arquitectura de cómputo. Dentro del modelo de datos, presentaremos la estructura de datos utilizada para los procesos de agregaciones, acumulaciones y cálculos.

El capítulo 4 está dedicado a la funcionalidad de los módulos principales y la importancia de cada uno de ellos. Se describen a detalle las etapas del procesamiento de indicadores, desde que llega la información a la subdirección, pasando por la carga del archivo a la base de datos, la validación de datos, los cálculos de indicadores y transferencia de datos entre servidores. Del mismo modo, hablaremos de los procesos internos del sistema, tales como los disparadores automáticos de procesos y del envío automático de correo electrónico, que sirven para el control de flujo de procesos.

Este capítulo incluye la descripción del módulo de construcción de fórmulas para la obtención de indicadores, y el detalle del proceso de generación de cálculos de los mismos. También, describe los procesos de agregaciones y acumulaciones. Se mencionan las ventajas que ofrecen estos procesos dentro de la generación de valores de indicadores. A su vez, presenta el pseudo código de estos procesos.

El capítulo 5 está dedicado a los resultados obtenidos por el sistema. Se discuten estos resultados y se comentan los beneficios que tiene el SCI sobre la operación anterior.

Finalmente, se mencionan algunos trabajos futuros que son recomendables realizar para mejorar ciertos módulos y procesos del sistema, y para ampliar la funcionalidad del Sistema de Control de Indicadores.

Capítulo 2

Problemática en el procesamiento de indicadores en la Subdirección de Sistemas de Información Corporativa

2.1 Antecedentes

2.1.1 La importancia de la toma de decisiones en Telmex.

La privatización de Teléfonos de México (Telmex), en diciembre de 1990, representó para la empresa establecer un plan de trabajo a mediano y largo plazo, con objetivos específicos de crecimiento y modernización que buscaba extender los beneficios de las telecomunicaciones a más mexicanos; así como ofrecer productos y servicios de mayor calidad con tecnología de vanguardia, que se establecieron en el Título de Concesión bajo el que fue privatizada Telmex.

Los retos establecidos fueron superados ampliamente en el número de líneas en servicio, telefonía pública, así como en el número de comunidades con cobertura. Adicionalmente, Telmex realizó importantes avances en el desarrollo y construcción de la red de fibra óptica así como en la digitalización de la planta. El cable submarino Columbus II empezó operaciones conectando a 57 empresas de telefonía en 41 países.

En 1997, el mercado mexicano de Larga Distancia se abrió a la competencia. Telmex cumplió ampliamente y sin contratiempos con todos los requerimientos impuestos por las autoridades nacionales con el fin de realizar una transición a la competencia en la industria de las telecomunicaciones. La plataforma tecnológica de Telmex se preparó para interconectar a los nuevos operadores de larga distancia en el país. Desde el 1º de enero de ese año, inició la competencia en las 60 principales ciudades del país donde el 75% de los clientes decidieron quedarse con los servicios que ofrece Telmex.

Posteriormente, en 1999, Telmex asumió un compromiso en un plan a dos años denominado Telmex20/20 enfocado en continuar el proceso de modernización y crecimiento de la infraestructura telefónica. El Plan cumplió con sus objetivos 6 meses antes de lo previsto. Específicamente, se alcanzó el 100% de digitalización de la planta de telefonía local, así como contar con 20 millones de servicios entre líneas fijas, inalámbricas, cuentas de Internet y enlaces dedicados a la transmisión de datos.

En el periodo de 1990 a 2003, Telmex invirtió alrededor de 27 mil millones de dólares. El año 2003 fue complicado no sólo por las condiciones económicas que se manifestaron con un crecimiento moderado de la economía, sino también por un entorno regulatorio de incertidumbre que afectó las decisiones de inversión de Telmex. Sin embargo, la empresa obtuvo una ventaja de este año, consolidando los logros alcanzados en los años previos y buscando obtener una mayor eficiencia en el control de gastos de los procesos básicos.

A partir de 2003 comienzan las negociaciones para adquirir algunas empresas de telefonía de Sudamérica. Actualmente se han adquirido las empresas de Telefonía de Argentina, Brasil, Colombia, Chile, Perú y Estados Unidos.

En noviembre de 2005, Telmex lanza el servicio de video teléfono; con esto, México es el primer país en el continente Americano en ofrecer este servicio. Los directivos apuestan a que este servicio crecerá a mediano plazo. Por lo pronto, pretenden extenderlo en las zonas rurales, y a mediano plazo lo harán en todo el continente.

La tabla 2.1 muestra el crecimiento que ha tenido la empresa después de su privatización. Las cifras muestran los servicios más representativos de la empresa.

CONCEPTO	1990	2004
Líneas telefónicas	5,352,824	17,172,278 (*)
Teléfonos públicos	92,073	725,483 (**)
Cuentas de Internet		1,741,296 (*)
Poblaciones con servicio	10,621	20,848 (*)
Tráfico local (millones de llamadas)	8,950	26,782 (*)
Tráfico de larga distancia nacional (millones de llamadas)	4,375	16,700 (*)
Digitalización de la red	30.9%	100% (**)
Número de concesionarios interconectados	10	20(*)
NOTA: (*) cifras a septiembre de 2004 (**) cifras a diciembre de 2003		

Tabla 2.1. Crecimiento de los servicios ofrecidos por Telmex, [1].

2.1.2 El proceso de la toma de decisiones

El crecimiento de Telmex a lo largo de 15 años es resultado de las estrategias de crecimiento que se han adoptado. Decidir a que mercados incursionar, que áreas fortalecer, que nuevos nichos de mercado establecer o que nuevos negocios desarrollar es parte del trabajo que deben asumir los directores de la empresa. **La toma de decisiones** es un asunto primordial para el desarrollo y crecimiento de la empresa. Tomar las decisiones correctas para lograr los objetivos corporativos es una responsabilidad enorme. Omitir alguna suposición, restricción o regulación puede ser fatal para la empresa misma.

Para García J. [2], la toma de decisiones corporativa consiste en encontrar una acción adecuada para resolver una situación problemática presente o futura, en la que, además, hay una serie de sucesos inciertos. Para analizar la situación, hay que determinar los elementos que son relevantes y obviar los que no lo son, analizar las relaciones entre ellos y la forma

que tenemos de influir en ellos. Este paso puede dar lugar a problemas, cuando se tienen en cuenta aspectos irrelevantes y se ignoran elementos fundamentales del problema. Una vez determinada cual es la problemática, se analiza con profundidad, para tomar decisiones. Es necesario elaborar modelos de acciones alternativas, extrapolarlas para imaginar el resultado final y evaluar estos, teniendo en cuenta la incertidumbre de cada suceso que lo compone y el valor subjetivamente se le asigna ya sea consciente o automáticamente. Así se obtiene una idea de las consecuencias que tendría cada una de las acciones alternativas que se han definido y que puede servir para elegir la conducta más idónea como el curso de acción que va a solucionar la problemática. Algunas de estas decisiones podrían involucrar la ganancia o pérdida de grandes sumas de dinero o el cumplimiento o incumplimiento de la misión y las metas de la empresa.

En este mundo cada vez más complejo, la dificultad de las tareas del personal que toma las decisiones aumenta día a día. La persona debe responder con rapidez a los acontecimientos que parecen ocurrir a un ritmo cada vez más acelerado. Además, ya no es posible que una sola persona conozca los detalles de todas las áreas de la empresa ni que tome todas las decisiones sobre su operación. Incluso dentro del relativamente limitado control que tiene un directivo, los factores que afectan sus decisiones son con frecuencia tan numerosos y sus efectos tan penetrantes que las decisiones forzadas ya no resultan aceptables. En consecuencia, para tomar decisiones efectivas, se exige que justifiquen sus decisiones basándose en la información oportunamente analizada y resumida. Fernández H. [4], muestra un análisis más detallado del proceso de la toma de decisiones.

2.1.3 La información en la toma de decisiones

La información es un elemento fundamental en la toma de decisiones de la empresa y para que ésta sea útil deben tenerse en cuenta los aspectos siguientes:

- **Constancia:** Debe provenir de fuentes que proporcionen la información sin interrupciones.
- **Confiable:** Los datos recibidos deben corresponder con la información real.

- **Oportunidad:** La información debe recibirse en el momento en que se le requiere. Es de poco interés si se proporciona posteriormente. En la actualidad, los tiempos con los que se tiene que contar con la información se hacen cada vez más cortos.
- **Fuente actual e histórica:** La información presente e histórica nos presenta el comportamiento sobre un asunto u operación a través del tiempo.
- **Relevancia:** Debemos contar con la información necesaria y suficiente, para analizar un asunto u operación. También, es importante evitar la información redundante.
- **Fuente externa e interna:** La información interna y externa a la empresa, nos ofrece un contexto más amplio de las variables que pueden afectar al comportamiento de la empresa.

Gran parte de la información estratégica usada en Telmex para la toma de decisiones son indicadores. Entenderemos por un **indicador** toda cantidad estadística, obtenida mediante la agregación de datos primarios o mediante operaciones matemáticas que sintetizan situaciones o acciones importantes relacionadas con la operación de la empresa. Interesa, en la mayoría de los casos, conocer la evolución temporal de los indicadores. Para su elaboración, es necesario contar con la información adecuada y de todos los niveles jerárquicos de la empresa. Latorre R. [5] presenta una definición más completa de lo que es un indicador y cuales son sus características.

Entre las ventajas que se tienen con el uso de indicadores, en la toma de decisiones, se encuentran:

- Reduce la complejidad del asunto que se analiza sin interferencias subjetivas.
- Identifica tendencias del tema que se está tratando.
- Presenta líneas de acción.
- Permite la comparación contra los objetivos prefijados.

Es recomendable que un sistema de indicadores tenga un número reducido de ellos; mientras menos indicadores tiene un sistema es mejor, ya que es más manejable y permite realizar un seguimiento periódico a bajo costo.

2.1.4 Operación de la SSIC.

La concentración de la información estratégica, su procesamiento y publicación es una de las funciones de la SSIC de Telmex. El procesamiento consiste en recibir la información de las diferentes fuentes o entidades; filtrar la información para detectar y corregir errores y realizar los procesos necesarios para obtener los indicadores.

Por las funciones que realiza dentro de la empresa y por las responsabilidades que tiene, la SSIC es considerada estratégica para la toma de decisiones.

El área se formó en 1995. Desde entonces, se encarga de distribuir los reportes de indicadores a los directores corporativos y operativos. Inicialmente, la entrega se realizaba en carpetas engargoladas, haciendo un promedio de doce carpetas por periodo. El procedimiento usado para procesar la información presentaba varios problemas, originando que, en varias ocasiones, fuera necesario realizar jornadas laborales, inclusive, de hasta 36 horas continuas, para obtener el producto final.

Hasta el 2000, el procedimiento en la SSIC para la obtención de los indicadores era el siguiente:

La SSIC recibía la información para procesar de 66 entidades de Telmex, llamadas “áreas”, que se encuentran distribuidas por todo el país. La información se encontraba en archivos de formatos distintos, donde las áreas almacenaban los resultados de su operación, como por ejemplo: resultados financieros, inventarios, ventas, quejas recibidas, quejas atendidas, líneas empresariales, líneas dedicadas, servicios múltiples como llamadas de larga distancia y servicio medible.

La tarea de la SSIC era reunir toda la información con los mismos criterios corporativos y convertirla en un mismo formato para procesarla.

El procesamiento consistía en:

- Consolidar, por entidad y periodo, los valores de los indicadores que recibían de las áreas.
- Generar valores acumulados, a partir de la información histórica almacenada.
- Generar resultados de otras 13 entidades que se obtienen a partir de las 66 áreas.
- Generar nuevos resultados que se obtienen a partir de operaciones entre indicadores.

El proceso es sencillo. Sin embargo, la cantidad de información recibida y generada, hacía que el proceso fuera laborioso. Cada archivo recibido contenía información requerida para obtener de uno a diez indicadores, es decir, un promedio de 50 archivos por tema. Por lo tanto, tenían que administrar una cantidad grande de archivos, por periodo. Además, si consideramos que el número de indicadores era de 250 por tema, entonces, los procesos que realizaban era sobre 20 mil registros por periodo, (Ver los datos procesados en los años de 1995 a 1999 que muestra la tabla 2.2).

El producto final era un informe mensual, llamado **Informe de Gestión**, que consistía de un reporte ejecutivo, que contenía unas 10 hojas. Para generar este informe la SSIC requería, aproximadamente, 12 personas que se dedicaban al procesamiento de indicadores; se auxiliaban de herramientas para la explotación de la información, por ejemplo dBase, Clipper y Fox Pro. Estos “manejadores de bases de datos” eran utilizados, principalmente, para unificar el formato de la información que les llegaba. Los sistemas operativos oficiales en las áreas administrativas de Telmex eran MSDOS con el ambiente de trabajo Windows 3.11.

El manejo de múltiples herramientas ocasionaba trabajo extra al momento de integrar la información en un solo archivo. Sin embargo, debido a que no había problemas graves en las entregas de los informes, los gerentes no consideraron necesaria la unificación de las herramientas.

Cada operador de la SSIC era responsable de obtener los resultados de un subconjunto del total de los indicadores. Sus resultados eran entregados a un coordinador que se encargaba de concentrar todos los indicadores en un solo archivo. Después de concentrados, se generaban otros indicadores, a partir de fórmulas donde intervienen indicadores de

diferentes responsables. La última parte del proceso general era la generación del Informe de Gestión. Aunque era un reporte sencillo, la urgencia de la entrega ocasionaba que el reporte resultara laborioso, principalmente la verificación detallada de los resultados finales.

De 1995 a 1999, trabajaron con las herramientas mencionadas. En 1998 crearon un primer programa en Fox Pro para generar indicadores calculados; es decir, obtener el valor de un indicador a partir de una operación aritmética donde intervienen dos o más indicadores. Este programa de cómputo les ahorró trabajo, sobre todo para obtener los indicadores financieros, que son de gran importancia para la Dirección General. Sin embargo, las operaciones se encontraban en código, lo que resultaba poco ágil, al momento de considerar una nueva fórmula o cambios en alguna operación existente; ya que se requería hacer el cambio en código y compilar para crear una nueva versión del programa.

Paralelamente, se fueron dando cambios en la organización de Telmex, que fueron aprovechados por la Subdirección. Uno de estos cambios fue la unificación de los sistemas transaccionales (sistemas donde se registra en línea la información operativa, por ejemplo: quejas solucionadas, nuevas líneas telefónicas, etc.). Con esto, ahora, las áreas registraban su operación en estos sistemas. La consulta de la información, de todas las entidades, era posible realizarla desde las computadoras de los gerentes o personal asignado para extraer los datos. Por lo tanto, ahora solamente se tenía que manipular un archivo en lugar de los 66 que anteriormente se recibían por tema. Parte de la información que entregaban las áreas a la SSIC, era registrada en los sistemas transaccionales. El resto de los datos seguían recibiendo en archivos por separado.

Fue hasta el 2000 que las nuevas necesidades de los Directores de Telmex, requirieron considerar nuevas variables para el análisis en la toma de decisiones; lo que originó la creación de nuevos indicadores, llegando a ser aproximadamente 650. El aumento en la cantidad de datos a procesar originó que la SSIC comenzara a tener problemas críticos en su operación; por lo tanto, el Subdirector del área, junto con sus gerentes, decidieron construir un sistema de información, robusto e integral, que permitiera la automatización completa del

proceso de indicadores. Fue así como a partir del año de 2000, inició el proyecto para construir el Sistema de Control de Indicadores y el Sistema de Información.

La tabla 2.2 representa la cantidad de información que la SSIC ha procesado a través de los años. Las cifras que aparecen del 2001 al 2005, es información que se ha procesado con el SCI.

AÑO	VOLUMEN DE INFORMACION			
	MENSUAL	SEMANAL	DIARIA	TOTAL
1995	230,000	0	0	230,000
1996	230,000	0	0	230,000
1997	230,000	0	0	230,000
1998	230,000	0	0	230,000
1999	230,000	0	0	230,000
2000	1,137,600	0	0	1,137,600
2001	1,542,980	0	0	1,542,980
2002	2,902,315	919,031	0	3,821,346
2003	5,208,663	3,268,536	0	8,477,199
2004	5,826,102	6,509,772	1,105,160	13,441,034
2005*	6,622,126	7,088,593	11,749,350	25,460,069

Tabla 2.2. Volumen de información procesada en la SSIC.

El aumento de la información a través de los años es debido a que han considerado nuevas entidades (oficinas de trabajo y centros de trabajo), que son entidades hijas de las áreas. Además, la cantidad de indicadores ha aumentado de 650 a más de 1400, y la periodicidad mensual pasa a ser mensual, semanal y diaria.

En paralelo al desarrollo del SCI, en el 2002 se liberó otro sistema, llamado Sistema de Información (SI), el cual complementa el procesamiento de los indicadores, sustituyendo la impresión del Informe de Gestión. Su implementación se dio gracias al apoyo recibido por la Dirección General de Telmex y la Dirección de Sistemas. Con este sistema, los reportes de indicadores se presentan en una página Web dentro de la Intranet de Telmex. Por medio de un esquema de seguridad y privilegios, establecidos por Telmex, permite la consulta de

información a los directores, subdirectores y gerentes de la empresa. De esta forma, la información está disponible a más gente, no importando en que parte de la República Mexicana se encuentre. Otra de las ventajas, es que la información es actualizada constantemente y no es necesario esperar toda la información para procesar y generar los resultados, como sucedía anteriormente.

Existieron otros esfuerzos encabezados por otras áreas que pensaron que el solo hecho de emplear herramientas de información de vanguardia solucionarían los problemas, pero fracasaron en el intento.

2.2 Análisis situacional de la SSIC

El análisis situacional es una técnica usada en varias áreas de conocimiento para detectar problemas en el funcionamiento de la operación de un proceso, en general.

Hanel J. [3] nos presenta una definición general de situación y análisis situacional. Una **situación** es un conjunto de problemas o preocupaciones que incluyen 3 elementos:

- Problemas actuales cuyos efectos se sienten en este momento.
- Asuntos sobre los que hay que decidir ahora para atacar un problema o para aprovechar una oportunidad.
- Problemas potenciales con una probabilidad de ocurrencia en el futuro.

La técnica de Análisis Situacional incluye la evaluación y aclaración de situaciones, desglosar situaciones complejas en componentes manejables y el mantenimiento del control sobre los eventos. Hace referencia al análisis de datos pasados, presentes y futuros, que proporciona una base para seguir un proceso de planeación.

El proceso de la técnica está dividido en cuatro etapas básicas:

1. **Reconocer las situaciones.** Comenzar con un listado de los temas a resolver. Algunas de las preguntas que ayudan a determinar los asuntos a resolver son: ¿qué oportunidades se

presentan?, ¿qué amenazas existen?, ¿qué problemas recientes no se han resuelto?, ¿en qué recomendaciones estamos trabajando ahora?, ¿qué decisiones deben tomarse ya?, ¿qué proyectos, sistemas o planes importantes están a punto de implantarse? y en definitiva ¿qué situaciones nos preocupan?

2. **Separar las situaciones en componentes manejables.** Consiste en dividir situaciones complejas en situaciones más pequeñas definidas con mayor claridad. Algunas preguntas que ayudan para determinar si es necesario dividir una situación son: ¿estamos hablando de una cosa o de varias?, ¿qué está pasando realmente en esta situación?, ¿qué es lo que en realidad está preocupándonos acerca de esta situación?, ¿qué aspectos comprende este conjunto o lfo?, ¿de qué parte se trata?
3. **Priorizar.** Decidir en qué orden se va a trabajar con las situaciones desglosadas. Importante, urgente y/o factible.
4. **Plan de resolución.** Seleccionar el proceso apropiado para resolver cada situación.

El análisis situacional es empleado por la ingeniería del software para determinar cuales son los problemas que se presentan en un proceso de manejo de información y como deben ser atacados por medio de la construcción de sistemas informáticos.

El análisis situacional de la operación de la Subdirección de Sistemas de Información Corporativa detectó varios problemas en su funcionamiento. La figura 2.2 muestra un desglose de las actividades realizadas para la elaboración del reporte ejecutivo; este proceso parte de la recepción de la información y termina en la creación de la carpeta de los reportes de indicadores. Es importante señalar que la información proviene de origen sin manipulación y con catálogos únicos de agregación.

En el apéndice A se describe el significado de los elementos del diagrama de actividades.

A continuación se describen los procedimientos y problemas encontrados en el procesamiento de los indicadores:

1. **Verificar la estructura de la información:** Verificar que la estructura del archivo se encuentre tal como lo solicitó la subdirección. Los problemas que se presentan en este proceso son:
 - 1.1. Los operadores no abren siempre los archivos para verificar que tengan la estructura correcta.
 - 1.2. En ocasiones, los operadores verifican los primeros registros, pero los errores pueden presentarse en los registros posteriores.
 - 1.3. No verifican siempre el orden de los campos.
2. **Analizar datos:** Revisar que los valores contenidos en el archivo son válidos. Los problemas que se presentan en este proceso son:
 - 2.1. Los operadores no validan siempre los valores del archivo.
 - 2.2. Errores humanos al validar los valores del archivo:
 - 2.2.1. Existencia de una fecha única y válida.
 - 2.2.2. Existencia de las entidades e indicadores en sus catálogos.
 - 2.2.3. Formato correcto de la fecha.
 - 2.3. Tardar más tiempo del necesario en la validación de los datos.
 - 2.4. Repetir la tarea cuando se detectan errores del procedimiento anterior.
3. **Generar datos agregados y acumulados:** Un valor agregado de un indicador es el resultado de realizar la operación (suma o promedio) de los valores de varias entidades que pertenecen a otra de mayor nivel, por ejemplo: la suma del número de llamadas reportadas en diferentes áreas de trabajo, representa el total de número de llamadas de la división a la que pertenecen. Para generar este valor se cuenta con un diagrama de jerarquías, que en adelante llamaremos **árbol de agregación**, el cual indica la relación que existe entre las entidades.

El valor marginal de un indicador es el valor generado en el periodo (mes, semana o día); por ejemplo: el número de quejas recibidas en el periodo (mes). El valor acumulado de un indicador es el valor que se acumula a través de los periodos; en el mismo ejemplo: el número de quejas recibidas en lo que va del año (la suma de las quejas recibidas en todos los meses que van del año). Las operaciones que se aplican (suma, promedio o inventario) dependen del indicador que se esté procesando. En Telmex se utilizan 4 tipos

de valor por indicador: **M**arginal, **A**cumulado del marginal, **P**rograma y **A**cumulado del programa. Los problemas que se presentan en la generación de estos valores se encuentran los siguientes:

- 3.1. Repetir la tarea cuando se detectan errores de los procedimientos anteriores.
- 3.2. Problemas en la generación de valores agregados:
 - 3.2.1. Error en la asociación de las entidades.
 - 3.2.2. Error al decidir que operación aplicar en la agregación del indicador.
 - 3.2.3. Errores manuales al trabajar con una cantidad grande de datos.
 - 3.2.4. Utilizar demasiado tiempo para realizar las operaciones, debido a que se hace de manera semiautomática.
 - 3.2.5. Errores al generar el resultado cuando se trata de árboles de agregación complicados.
 - 3.2.6. Los resultados, en ocasiones, son incompletos, debido a que las áreas no entregan su información a tiempo para procesarla.
- 3.3. Problemas en la generación de valores acumulados:
 - 3.3.1. Error al decidir que operación aplicar en la acumulación de un indicador.
 - 3.3.2. Error en la generación del valor agregado, debido a que se basan en el acumulado del periodo anterior y éste puede presentar un error.
 - 3.3.3. Utilizar demasiado tiempo para realizar las operaciones, debido a que se hace de manera semiautomática.
 - 3.3.4. Los resultados, en ocasiones, son incompletos, debido a que las áreas no entregan su información a tiempo para procesarla.
4. **Verificar datos concentrados:** Después de que cada persona genera sus resultados, un coordinador se encarga de verificar los datos que se van a concentrar. Los problemas que se presentan en esta parte del proceso general son:
 - 4.1. Los operadores envían una versión anterior de los datos, por lo tanto, se tienen que repetir los procesos posteriores tantas veces como se presente este problema.
 - 4.2. Al no existir un repositorio de información común para compartir catálogos, el personal llega a generar nombres diferentes para un indicador o un mismo nombre

para diferentes indicadores, generando errores al concentrar los datos de todo el personal.

4.3. Repetir la tarea cuando se detectan errores de los procedimientos anteriores.

5. **Realizar cálculos de indicadores:** El coordinador junta los datos de todos los operadores y aplica ciertos cálculos que han sido programados en un modulo de cómputo. El problema que se presenta en este procedimientos es:

5.1. Los cálculos que se realizan actualmente por medio de un programa de cómputo está restringido a las fórmulas definidas. Para adicionar o modificar una fórmula hay que modificar el programa y posteriormente compilarlo.

6. **Verificar datos históricos:** El coordinador recibe, de cada uno de los operadores, los datos procesados y los datos históricos de los indicadores. El coordinador es responsable de verificar que la información histórica sea correcta. Los problemas que se presentan en esta parte del proceso son:

6.1. Los operadores no detectan cambios en datos históricos de sus indicadores.

6.2. El tiempo utilizado para verificar que la información histórica sea correcta puede ser grande.

6.3. Repetir la tarea cuando se detectan errores de los procedimientos anteriores.

7. **Elaborar reportes:** El coordinador, junto con el personal operativo, genera una serie de reportes que van desde tablas comparativas hasta hojas de cálculos con referencias cruzadas. Los problemas que se presentan en esta parte del proceso son:

7.1. La elaboración de los reportes consumen mucho tiempo.

7.2. Repetir la tarea cuando se detectan errores de los procedimientos anteriores.

8. **Armar la carpeta:** El coordinador, junto con los operadores, imprime y engargola las hojas para armar la carpeta. Los problemas que se presentan en esta parte del proceso son:

8.1. La tarea consume mucho tiempo.

8.2. Repetir la tarea cuando se detectan errores de los procedimientos anteriores.

La tabla 2.3 presenta un resumen de los problemas encontrados durante el procesamiento de indicadores.

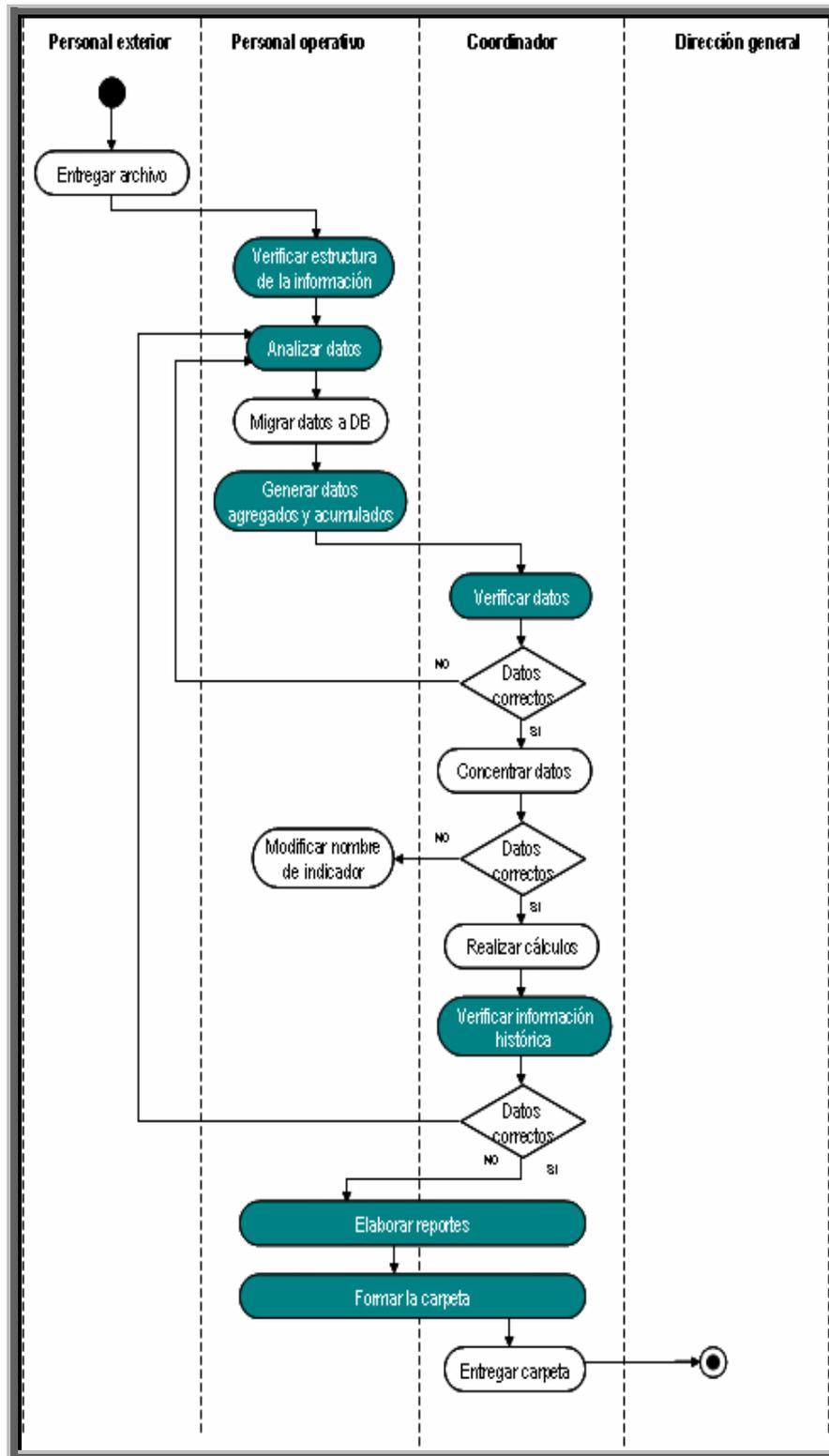


Figura 2.2. Diagrama de actividades de la situación actual.

Cuando se realiza el análisis situacional de la operación, se adquiere un conocimiento profundo del funcionamiento de la operación, proceso o negocio. Este conocimiento es la base de los ingenieros para desarrollar el nuevo diseño del sistema, sin olvidar todos los aspectos que tienen que ser corregidos. Con esto, el ingeniero de sistemas aporta, en muchas ocasiones, soluciones que el cliente o usuario, nunca imaginó.

No.	PROBLEMA
1	Falta de capacidad de algunos operadores, para revisar completamente y con detalle la información que les llega.
2	Algunos operadores no utilizan el catálogo común de entidades e indicadores.
3	Reprocesos manuales que consumen mucho tiempo.
4	Muchos de los procesos que son manuales o semiautomáticos son muy tardados.
5	Existe trabajo repetitivo debido a los errores que se presentan de las tareas anteriores.
6	Las operaciones de agregaciones, acumulaciones y cálculo de indicadores son muy laboriosos y con gran probabilidad de errores en la obtención de los resultados.
7	No tienen una lista bien definida de las operaciones que se tienen que realizar en las agregaciones y acumulaciones de cada indicador.
8	Algunos operadores usan el resultado del valor acumulado del periodo anterior para obtener el valor del periodo actual, sin considerar que dicho dato puede tener algún error, provocando que el resultado también esté mal.
9	En ocasiones existe atraso en la entrega de la información por parte de algunas entidades, provocando que los resultados sean incompletos para los periodos posteriores. En algunos casos esta información llega atrasada, por lo que se procesa para el periodo que se recibe, pero no se regeneran los valores de los periodos posteriores.
10	Complejidad para controlar la última versión de la información histórica.
11	El proceso de cálculos está restringido a los cálculos que se codifiquen dentro de un programa que les genera los resultados.
12	Resistencia al cambio por parte de los operadores.

Tabla 2.3. Resumen de problemas.

Capítulo 3

Diseño del sistema

3.1 Modelos del diseño

En el diseño de sistemas se aplican técnicas y principios con el propósito de definir dispositivos, procesos o sistemas, con suficiente detalle como para permitir su interpretación y construcción. Yourdon E. [6] muestra con detalle algunas de estas técnicas.

Los modelos de diseño elaborados para la construcción del SCI son:

1. **Diseño de datos:** Transforma el modelo del dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el sistema.
2. **Diseño de procesos:** Describe el funcionamiento que deben tener los programas y como deben realizarlo.
3. **Diseño de infraestructura:** Describe la infraestructura de cómputo sobre la que va a funcionar el dispositivo, proceso o sistema a construir.

El diseño debe contemplar todos los requisitos explícitos contenidos en el análisis y debe cumplir todos los requisitos implícitos que desea el cliente.

Debe ser una guía que pueda leer y entender el personal que construya el código y los que prueban y mantienen el sistema. El diseño debe proporcionar una idea completa de lo que es el sistema, enfocando los dominios de datos y comportamiento del sistema desde el punto de vista de la implementación.

3.1.1 Diseño de datos.

El diseño de datos es esencial en la construcción de cualquier sistema de información. El modelo utilizado en la construcción del SCI fue el Entidad Relación. El Apéndice B muestra las características de dicho modelo.

Un diagrama entidad relación va acompañado de un diccionario de datos, el cual describe el objetivo de la entidad y los datos contenidos en ella. También, las reglas que tengan que cumplir los datos, como por ejemplo un rango de datos, el valor que debe de tener por omisión, etc.

Los diagramas entidad relación de las figura 3.1 y 3.2 muestran la estructura de datos que se utilizan para generar los valores agregados y acumulados de los indicadores.

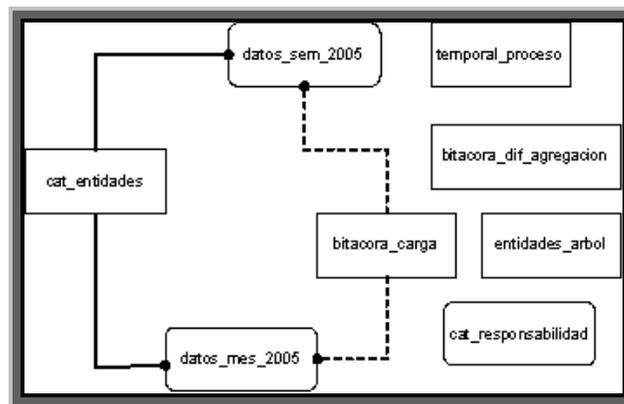


Figura 3.1. Modelo entidad relación del procedimiento de agregaciones.

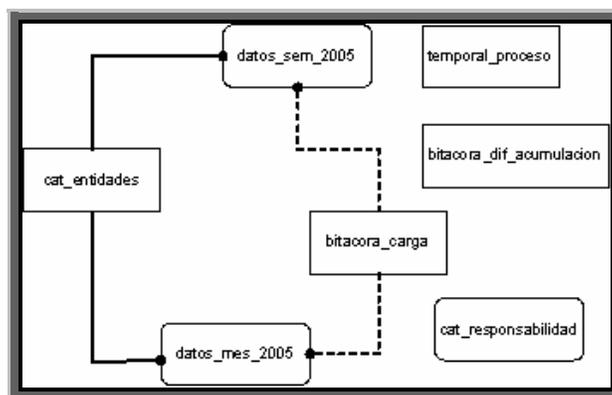


Figura 3.2. Modelo entidad relación del procedimiento de acumulaciones.

A continuación se presenta el diccionario de datos de los tres procesos. Las tablas se presentan en orden alfabético.

bcp_calculados: Tabla auxiliar para generar un archivo de los indicadores calculados que son de un producto diferente al original.

Nombre	Descripción	Tipo	Llave	Requerido
Entidad	Entidad de trabajo de Telmex.	Char(5)		SI
Indicador	Identificador del indicador calculado.	Char(10)		SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)		SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)		SI
Valor	Valor del indicador que viene en el archivo cargado.	Flota		SI
Fecha	Fecha del valor procesado.	Varchar(10)		SI

bitacora_carga: Lleva el control de las cargas realizadas y los estados en que se encuentran, de las periodicidades mensual y semanal.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga a procesar.	Int	PK	SI
Producto	Clave del producto que engloba a un conjunto de indicadores.	Char(7)		SI
Fecha_carga	Fecha y hora en que se está realizando la carga.	Datetime		SI
Archivo	Nombre del archivo cargado.	Varchar(100)		SI
Periodicidad	Periodicidad de la información (semanal, mensual).	Char(1)		SI
Status_proceso	Estado en el que se encuentra la carga.	Char(2)		SI
Usuario	Clave del responsable del indicador.	Char(10)		SI

bitacora_carga_dia: Lleva el control de las cargas realizadas y los estados en que se encuentran, de la periodicidad diaria.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga a procesar.	Int	PK	SI
Producto	Clave del producto que engloba a un conjunto de indicadores.	Char(7)		SI
Fecha_carga	Fecha y hora en que se está realizando la carga.	Datetime		SI
Archivo	Nombre del archivo cargado.	Varchar(100)		SI
Periodicidad	Periodicidad de la información (diaria).	Char(1)		SI
Status_proceso	Estado en el que se encuentra la carga.	Char(2)		SI
Usuario	Clave del responsable del indicador.	Char(10)		SI
Año	Año de la información a procesar	Smallint		SI

bitacora_dif_agregacion: Almacena las diferencias encontradas entre los datos agregados cargados y los calculados por el sistema.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga a procesar.	Int		SI
Entidad	Entidad de trabajo de Telmex.	Char(5)		SI
Indicador	Mnemónico del indicador.	Char(10)		SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)		SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)		SI
Periodo	Periodo del valor del indicador a procesar.	Smallint		SI
Fecha	Fecha del valor a procesar.	Char(10)		SI
Valor_carg	Valor del indicador que viene en el archivo cargado.	Float		SI
Valor_calc	Valor del indicador, generado por el sistema.	Float		SI
Responsable	Clave del responsable del indicador.	Char(1)		SI
Periodicidad	Periodicidad de la información (semanal, mensual o diaria).	Char(1)		SI

bitacora_proceso_calc: Almacena una bitácora de actividad de las sentencias que se ejecutan en el proceso de cálculos.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga procesada.	Int		SI
Periodicidad	Periodicidad de la información (semanal, mensual o diaria).	Char(1)		SI
Hora	Hora del proceso.	Datetime		SI
Descripción	Descripción del proceso realizado.	Varchar(150)		SI

calculos_modificados: Almacena los indicadores cargados que participan en fórmulas y cuyos registros ya se encuentran en la tabla de datos.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga a procesar.	Int		SI
Entidad	Entidad de trabajo de Telmex.	Char(5)	FK	SI
Indicador	Identificador del indicador.	Char(10)	FK	SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)		SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)		SI
Valor	Valor del indicador que viene en el archivo cargado.	Float		SI
Fecha	Fecha del valor a procesar.	Char(10)		SI
Año	Año del valor del indicador a procesar.	Smallint		SI
Periodo	Periodo del valor del indicador a procesar.	Smallint		SI
Nivel	Nivel de la entidad (centro de trabajo, área, división, etc.).	Char(1)	FK	SI
Valor_anterior	Valor del indicador antes de la carga del archivo.	Float		SI

calculos_insertados: Almacena los indicadores cargados que participan en fórmulas y que son nuevos en la tabla de datos.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga a procesar.	Int		SI
Entidad	Entidad de trabajo de Telmex.	Char(5)	FK	SI
Indicador	Identificador del indicador.	Char(10)	FK	SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)		SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)		SI
Valor	Valor del indicador que viene en el archivo cargado.	Float		SI
Fecha	Fecha del valor a procesar.	Char(10)		SI
Año	Año del valor del indicador a procesar.	Smallint		SI
Periodo	Periodo del valor del indicador a procesar.	Smallint		SI
Nivel	Nivel de la entidad (centro de trabajo, área, división, etc.).	Char(1)	FK	SI

cat_calculado_raiz_anidado: Contiene la relación de indicadores raíz con su calculado.

Nombre	Descripción	Tipo	Llave	Requerido
Periodicidad	Periodicidad de la información (semanal, mensual o diaria).	Char(1)	PK	SI
Indicador_calculado	Identificador del indicador calculado.	Char(10)	PK	SI
Tipo_calculado	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Indicador_raiz	Identificador del indicador raíz.	Char(10)	PK	SI
Tipo_raiz	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Periodo_relativo_raiz	Periodo relativo al periodo que se carga.	Smallint	PK	SI
Posicion_raiz	Posición del indicador dentro de la fórmula	Smallint	PK	SI

cat_entidades: catálogo de entidades de Telmex.

Nombre	Descripción	Tipo	Llave	Requerido
Entidad	Entidad de trabajo de Telmex a donde pertenece la información.	Char(5)	PK	SI
Descripcion	Descripción de la entidad de Telmex.	Char(50)		SI
Nivel	Nivel de la entidad (centro de trabajo, área, división, etc.).	Char(1)		SI
Fecha_alta	Fecha de alta del registro.	Datetime		SI
Fecha_baja	Fecha de baja lógica del registro.	Datetime		NO

cat_formulas: Contiene la relación del indicador calculado con su formula.

Nombre	Descripción	Tipo	Llave	Requerido
Indicador	Identificador del indicador calculado.	Char(10)	PK	SI
Periodicidad	Periodicidad de la información (semanal, mensual o diaria).	Char(1)	PK	SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Formula_corta	Fórmula que se tiene que aplicar para obtener al indicador calculado.	Varchar(8000)		SI

cat_niveles: Contiene la descripción de los diferentes niveles que se tienen dentro de las entidades de Telmex.

Nombre	Descripción	Tipo	Llave	Requerido
Nivel	Nivel de la entidad (centro de trabajo, área, división, etc.).	Char(1)	PK	SI
Descripcion	Nombre largo o descripción de la entidad	Char(40)		SI
Jerarquia	Jerarquía del nivel con respecto a los otros niveles.	Int		SI

cat_operadores: Almacena los operadores permitidos en la construcción de fórmulas.

Nombre	Descripción	Tipo	Llave	Requerido
Cve_operador	Clave del operador.	Smallint	PK	SI
Operador	Operador aplicado en la fórmula.	Varchar(20)		SI
Descripción	Descripción del funcionamiento del operador	Varchar(150)		SI
Activo	Indica si el operador está activo o inactivo.	Char(1)		SI

cat_periodos_relativos: Almacena la descripción de los diferentes periodos relativos que se pueden utilizar en una fórmula.

Nombre	Descripción	Tipo	Llave	Requerido
Periodicidad	Periodicidad de la información (semanal, mensual o diaria).	Char(1)		SI
Periodo_relativo	Valor relativo con respecto al periodo de carga.	Smallint		SI
Descripcion	Descripción del periodo relativo	Varchar(200)		SI

cat_responsabilidad: Contiene la relación entre indicadores y productos.

Nombre	Descripción	Tipo	Llave	Req.
Producto	Clave del producto que engloba a un conjunto de indicadores.	Char(7)	PK	SI
Indicador	Identificador del indicador.	Char(10)		SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)		SI
Periodicidad	Periodicidad de la información (semanal, mensual o diaria).	Char(1)		SI
Usuario	Clave del responsable del indicador.	Char(1)		SI
Ultima_actualizacion	Fecha y hora de la última modificación del registro.	Datetime		NO
Calc_agre	1: Se consideran los valores agregados generados por el sistema. 0: Se consideran los valores agregados cargados en el archivo.	Tinyint		NO
Agregacion	Tipo de operación a realizar: A: Suma, P: Promedio, N: No aplica.	Char(1)		NO
Calc_acum	1: Se consideran los valores acumulados generados por el sistema. 0: Se consideran los valores acumulados cargados en el archivo.	Tinyint		NO
Acumulacion	Operación: A:Suma,P:Promedio,S:Saldo,N:No apl.	Char(2)		NO

Cifras_control: Tabla para llevar un control de la información que se carga.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga procesada.	Int	PK, FK	SI
Año	Año del valor del indicador procesado.	Smallint	PK	SI
Periodicidad	Periodicidad de la información (semanal, mensual o diaria).	Char(1)	PK	SI
Periodo	Periodo del valor del indicador procesado.	Smallint	PK	SI
Indicador	Identificador del indicador.	Char(10)	PK, FK	SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Nivel	Nivel de la entidad (centro de trabajo, área, división, etc.).	Char(1)	PK, FK	SI
Modificados	Cantidad de registros modificados.	Int		SI
Insertados	Cantidad de registros insertados.	Int		SI
Nulos	Cantidad de registros con valor nulo.	Int		SI
Ceros	Cantidad de registros con valor 0.	Int		SI
Total	Cantidad total de registros procesados.	Int		SI
Reglones	Cantidad de renglones por nivel.	Int		SI

datos_mes_2005: Contiene los valores de los indicadores mensuales del año 2005.

Nombre	Descripción	Tipo	Llave	Requerido
Entidad	Entidad de trabajo de Telmex.	Char(5)	PK, FK	SI
Indicador	Identificador del indicador.	Char(10)	PK	SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)	PK	SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Periodo	Periodo del valor del indicador a procesar.	Smallint	PK	SI
Carga	No. de carga a procesar.	Int		SI
Valor	Valor del indicador que viene en el archivo cargado.	Float		SI
Fecha	Fecha del valor a procesar.	Datetime		SI

datos_mes_adic: Almacena los indicadores semanales de tipos especiales.

Nombre	Descripción	Tipo	Llave	Requerido
Entidad	Entidad de trabajo de Telmex.	Char(5)	PK, FK	SI
Indicador	Identificador del indicador.	Char(10)	PK, FK	SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)	PK	SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Fecha	Fecha del valor procesado.	datetime	PK	SI
Valor	Valor del indicador que viene en el archivo cargado.	Float		SI

datos_sem_2005: Contiene los valores de los indicadores semanales del año 2005.

Nombre	Descripción	Tipo	Llave	Requerido
Entidad	Entidad de trabajo de Telmex.	Char(5)	PK, FK	SI
Indicador	Identificador del indicador.	Char(10)	PK	SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)	PK	SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Periodo	Periodo del valor del indicador a procesar.	Smallint	PK	SI
Carga	No. de carga a procesar.	Int		SI
Valor	Valor del indicador que viene en el archivo cargado.	Float		SI
Fecha	Fecha del valor a procesar.	Char(4)		SI

datos_semanal_adic: Almacena los indicadores semanales de tipos especiales.

Nombre	Descripción	Tipo	Llave	Requerido
Entidad	Entidad de trabajo de Telmex.	Char(5)	PK, FK	SI
Indicador	Identificador del indicador.	Char(10)	PK, FK	SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)	PK	SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)	PK	SI
Fecha	Fecha del valor procesado.	Char(4)	PK	SI
Valor	Valor del indicador que viene en el archivo cargado.	Float		SI

entidades_arbol: Almacena las diferentes estructuras de árbol para el proceso de agregación.

Nombre	Descripción	Tipo	Llave	Requerido
Id_ent_arbol	Identificador del árbol de entidades.	Int		SI
Indentacion	Estructura de la rama dentro del árbol.	Varchar(50)		SI
Nivel_p	Nivel de la entidad padre.	Char(1)		NO
Nivel_h	Nivel de la entidad hijo.	Char(1)		NO
Padre	Entidad padre.	Char(5)		NO
Hijo	Entidad hijo.	Char(5)		NO

temporal_proceso: Tabla que almacena los resultados de cada una de las etapas del proceso general.

Nombre	Descripción	Tipo	Llave	Requerido
Carga	No. de carga a procesar.	Int		SI
Entidad	Entidad de trabajo de Telmex.	Char(5)		SI
Indicador	Identificador del indicador.	Char(10)		SI
Segmento	Tipo de segmentación de un negocio o servicio.	Char(2)		SI
Tipo	Tipo de dato (A, M, P, Q).	Char(1)		SI
Valor	Valor del indicador.	Float		SI
Fecha	Fecha del valor a procesar.	Char(10)		SI
Año	Año del valor del indicador a procesar.	Smallint		SI
Periodo	Periodo del valor del indicador a procesar.	Smallint		SI
Nivel	Nivel de la entidad (centro de trabajo, área, división, etc.).	Char(1)		SI

3.1.2 Diseño de procesos

El diseño de procesos describe a detalle los pasos o tareas que deben realizar para que cada proceso funcione correctamente y genere los resultados esperados. En esta sección presentaremos los diagramas de actividades de cada uno de los procesos o subprocessos que intervienen en el procesamiento de indicadores. El apéndice A describe el significado de los elementos que se presentan en los diagramas de actividades.

La figura 3.4 muestra el proceso general de carga de indicadores, acompañada con una descripción de cada subprocesso. Posteriormente, se presentan los procesos de agregaciones, acumulaciones y cálculos.

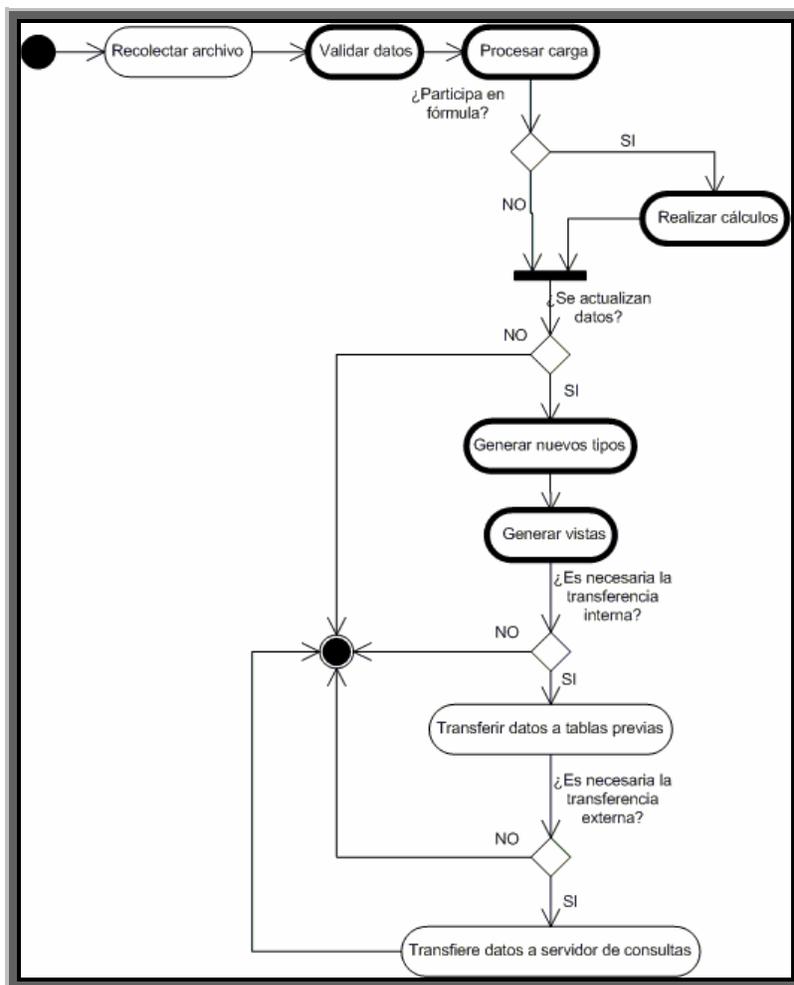


Figura 3.4. Proceso general de carga de indicadores.

Recolectar archivo: Dentro de una estructura de directorios en el servidor, previamente establecida, el proceso se encarga de verificar la existencia de algún archivo para procesar. Cuando detecta alguno, valida que el nombre sea correcto y que el contenido tenga la estructura requerida. En caso de no cumplir con estas condiciones el proceso rechazará el archivo, colocándolo en un subdirectorío de rechazo. En seguida, envía un correo electrónico al usuario responsable, informando que el archivo fue rechazado.

Validar datos: Después, que el archivo ha sido aceptado en el proceso anterior, este proceso valida el contenido del mismo. Verifica lo siguiente: que los campos numéricos, efectivamente, sean números; que el formato de la fecha sea correcto, que los indicadores y entidades que vienen en el archivo existan en los catálogos; que no existan registros duplicados; que los indicadores pertenezcan a un solo producto. En caso de fallar en alguna validación, el proceso se encarga de rechazar la información y enviar un correo electrónico al usuario responsable, informando la razón del rechazo.

Procesar carga: Realiza las operaciones de agregaciones y acumulaciones. Verifica que la información cumpla con ciertas características. Por último, almacena los valores de los indicadores en una tabla de procesamiento de datos.

Realizar cálculos: Este proceso lo presentaremos posteriormente con mayor detalle.

Generar nuevos tipos: Existen algunos indicadores a los que se les aplican ciertas operaciones matemáticas que se utilizan para análisis especiales, como por ejemplo: tendencia, cumplimiento, promedios y distintos tipos de ordenamiento, llamado rank. Después de realizada la operación, los valores se almacenan en una tabla establecida para estos datos con un tipo diferente a los M, A, P y Q.

Generar vistas: La información que se va a explotar por medio de consultas se almacena con cierta estructura de datos, de tal manera que las consultas puedan ser rápidas. El trabajo de este proceso es leer la información cargada y almacenarla con la estructura establecida para las consultas.

Transferir datos a tablas previas: La información procesada y almacenada en la estructura de datos, llamada vistas, es transferida al servidor de consulta. La información se envía, tal y como se encuentra en el servidor de procesamiento, pero a tablas previas, para que no interfiera la transferencia de la información con alguna consulta que se esté realizando en las tablas finales, debido a que la transferencia de datos resulta algo lenta.

Transferir datos a servidor de consulta: Después de ser transferida la información en las tablas previas, este proceso pasa la información a las tablas finales y envía correo electrónico al usuario responsable para informarle que el proceso terminó correctamente.

A continuación presentaremos a detalle los procesos de construcción de fórmulas, cálculos, agregaciones y acumulaciones.

3.1.2.1 Proceso de construcción de fórmulas.

El objetivo de la construcción de fórmulas es generar valores de indicadores, que llamaremos calculados, a partir de fórmulas donde intervienen otros indicadores, que llamaremos raíz.

Para obtener los valores calculados, es necesario que exista la fórmula o expresión matemática que lo pueda generar. Las fórmulas se generan dinámicamente por medio de un módulo de construcción de fórmulas. Las operaciones u operadores que se pueden aplicar son las permitidas por el manejador de BD (suma, resta, multiplicación, agrupación, funciones del manejador de BD.). De ser necesario, se pueden construir funciones de usuario, para complementar el funcionamiento básico permitido y ampliar las opciones para los operadores. Otra de las características de la construcción de fórmulas es que se pueden incluir indicadores raíz de tipos y periodos distintos al calculado, en decir, que para el indicador que se va a calcular, de tipo M, por ejemplo, se puede tomar en cuenta el mismo indicador u otros indicadores, de tipo A, por ejemplo, permitiendo que se generen más opciones de acuerdo a las necesidades de los usuarios. La figura 3.5 muestra el procedimiento para la construcción de fórmulas para el cálculo de indicadores.

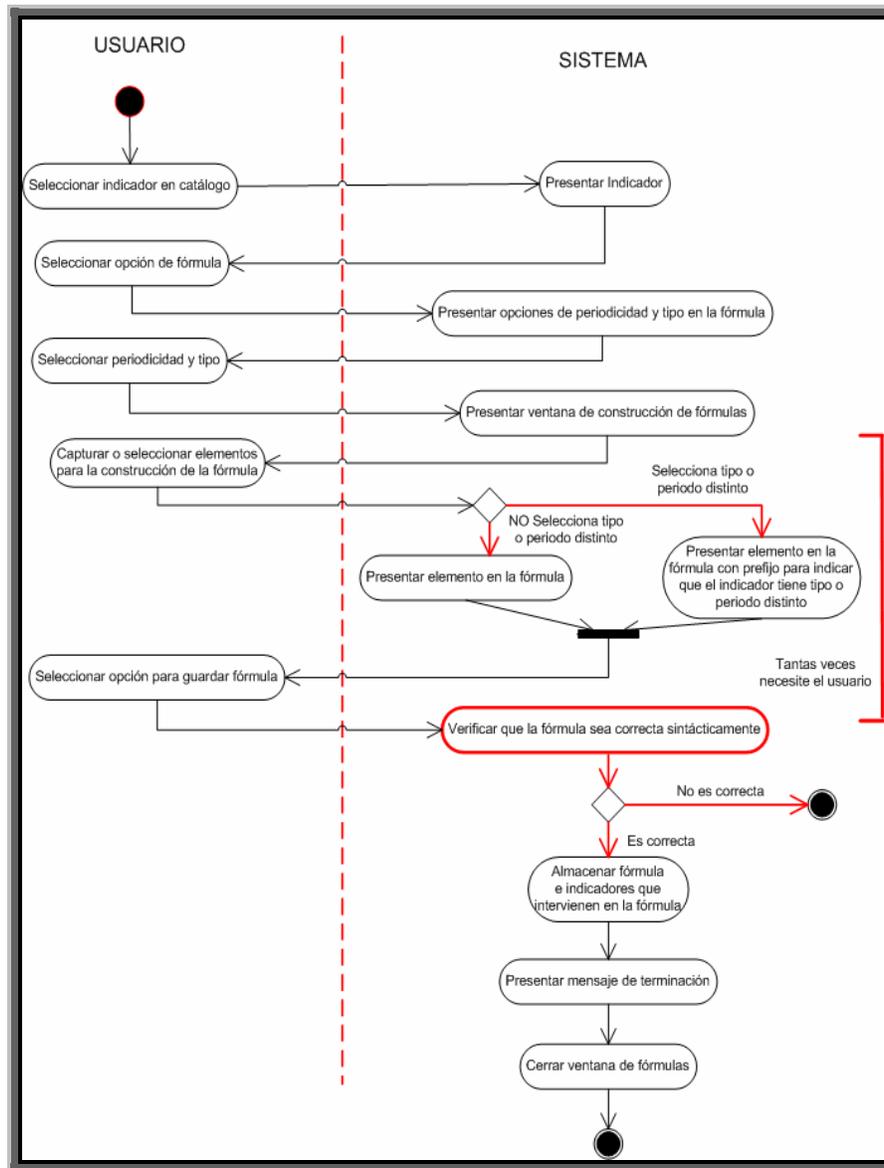


Figura 3.5. Construcción de fórmulas.

3.1.2.2 Proceso de cálculo de indicadores.

Después de haber registrado una fórmula, el sistema está preparado para generar los valores calculados del indicador. El proceso antecesor a los cálculos detecta cuales indicadores participan en una fórmula. Para informar al proceso de cálculos que debe trabajar, el proceso que le antecede debe establecer el estado "CC". Además, los indicadores que participan en la fórmula los almacena en una tabla de trabajo.

El proceso principal de cálculo de indicadores revisa constantemente la bitácora de cargas para verificar cuando hay una carga a la cual se le tiene que aplicar los cálculos, es decir que el estado de la carga sea “CC”.

La figura 3.6 muestra el procedimiento principal de cálculo de indicadores. La función principal de este proceso es colocar la información a procesar en una tabla de trabajo para que el siguiente proceso obtenga los datos de dicha tabla. Posteriormente, ya obtenidos los resultados de indicadores calculados, genera archivos de los indicadores calculados que son de un producto diferente a los indicadores del archivo cargado, donde el producto agrupa un conjunto de indicadores que especifica cierto tipo de información y además pertenece a un mismo usuario. Por último, almacena en las tablas de datos, los indicadores calculados del mismo producto.

El proceso de cálculos se debe realizar de manera recursiva, debido a que existen indicadores dentro de las fórmulas que participan dentro de otra fórmula. Por ejemplo, $c = a + b$ y $f = d + c$. En el ejemplo, la variable c se encuentra en dos fórmulas, en la primera es calculado y en la segunda es raíz, por lo tanto, primero se tiene que calcular c y después f . Algunos indicadores que han sido construídos en la SSIC se encuentran en varios niveles de recursividad de cálculos, por lo tanto, cada vez que el proceso calcula un conjunto de indicadores, debe verificar que los indicadores resultantes participan o no en otra formula. La salida de este proceso es el total de indicadores que se encuentran en los diferentes niveles de recursividad. La figura 3.7 muestra el procedimiento de recursividad de cálculos.

Un usuario es responsable de mantener actualizados los valores de los indicadores que le asignaron. Cuando se realiza una carga, puede presentarse el caso donde algunos indicadores del archivo disparen indefinidamente el procedimiento de recursividad de cálculos, aún de indicadores que no son de su responsabilidad. Aunque es necesario que se generen los resultados de todos los indicadores calculados, no es indispensable que se realice dentro del mismo periodo de tiempo que genera los indicadores de su responsabilidad. Por lo tanto, el proceso de recursividad está diseñado de tal manera para obtener los indicadores calculados del mismo producto de la carga original.

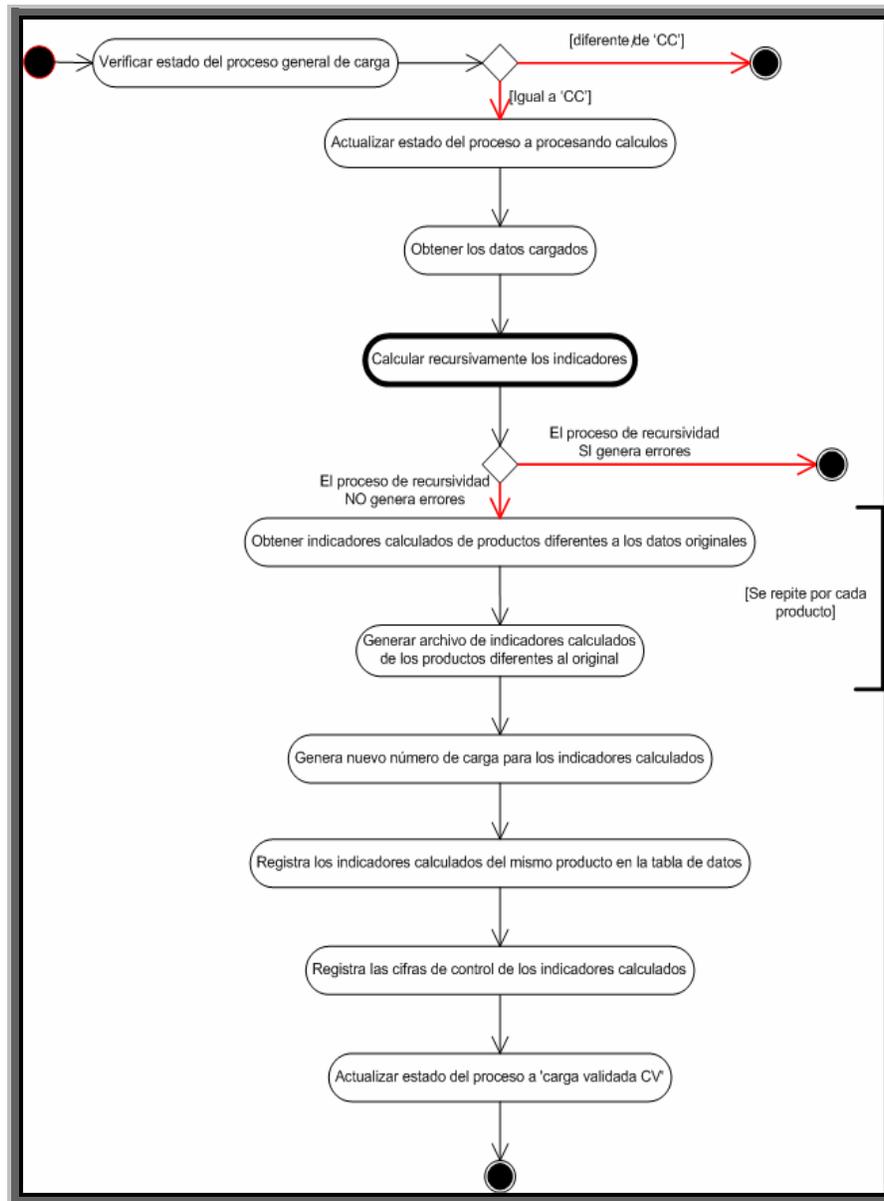


Figura 3.6. Procedimiento principal de cálculo de indicadores.

El procedimiento de cálculo de indicadores es la parte medular de la generación de los valores de los indicadores calculados. Después de obtener los valores de todos los indicadores que participan en una fórmula, se construye una tabla donde los indicadores que participan en la fórmula aparecerán, ahora, como columnas, en lugar de renglones. Por lo tanto, se construye la tabla mencionada con ciertos campos básicos que nos definirán nuestro conjunto identificador, conocido como llave. Posteriormente, se agrega columna por columna

correspondiente a cada indicador raíz que se utiliza en la fórmula. Al mismo tiempo, se construye una tabla por indicador, el cual contiene el total de valores de ese indicador. Después de generar todas las columnas de los indicadores se genera la columna donde se almacenará el resultado de la fórmula.

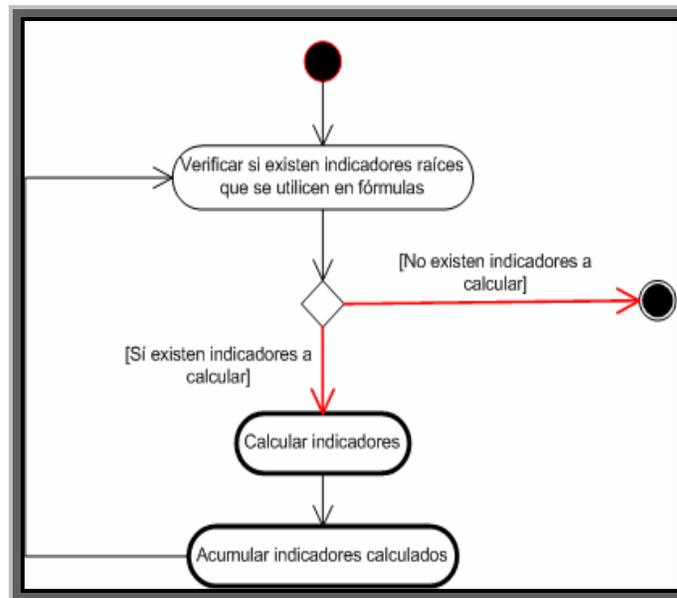


Figura 3.7. Procedimiento de recursividad de cálculo de indicadores.

Los valores generados en la columna de resultados son comparados contra los valores que puedan existir en las tablas de datos. En caso de existir diferencias entre ambos valores, el proceso debe enviar un correo electrónico al responsable de los indicadores, con las diferencias obtenidas. La figura 3.8 muestra el proceso de cálculo de indicadores.

Un subproceso, que tiene tanta importancia como la generación de los resultados de cálculos, es “obtención de datos que intervienen en las fórmulas”. Su función principal es obtener los valores de indicadores que participan en una fórmula y que no se encuentran en la información que se está cargando. Sin este proceso no sería posible generar todos los cálculos, debido a que las cargas no contienen, en la mayoría de los casos, todos los indicadores que participan en una fórmula. En caso de no contar con todos los indicadores necesarios para realizar las operaciones, el proceso se termina. El proceso realiza tareas complejas, debido a que considera todas las opciones de información que debe buscar.

Tomando como referencia los indicadores raíces de la carga, el procedimiento de obtención de indicadores que intervienen en las fórmulas, obtiene la lista de indicadores calculados. Posteriormente, elimina los indicadores calculados que a su vez puede ser raíz de otro calculado en un posterior cálculo. Verifica cuando la carga procesada es por medio de un recálculo. En este caso, elimina de la lista de calculados, todos aquellos que no aparezca en la lista de recálculos. Por último, elimina todos los registros de los indicadores cuya entidad no aparezca en su correspondiente árbol de agregación o de navegación. La figura 3.9 muestra las actividades del proceso.

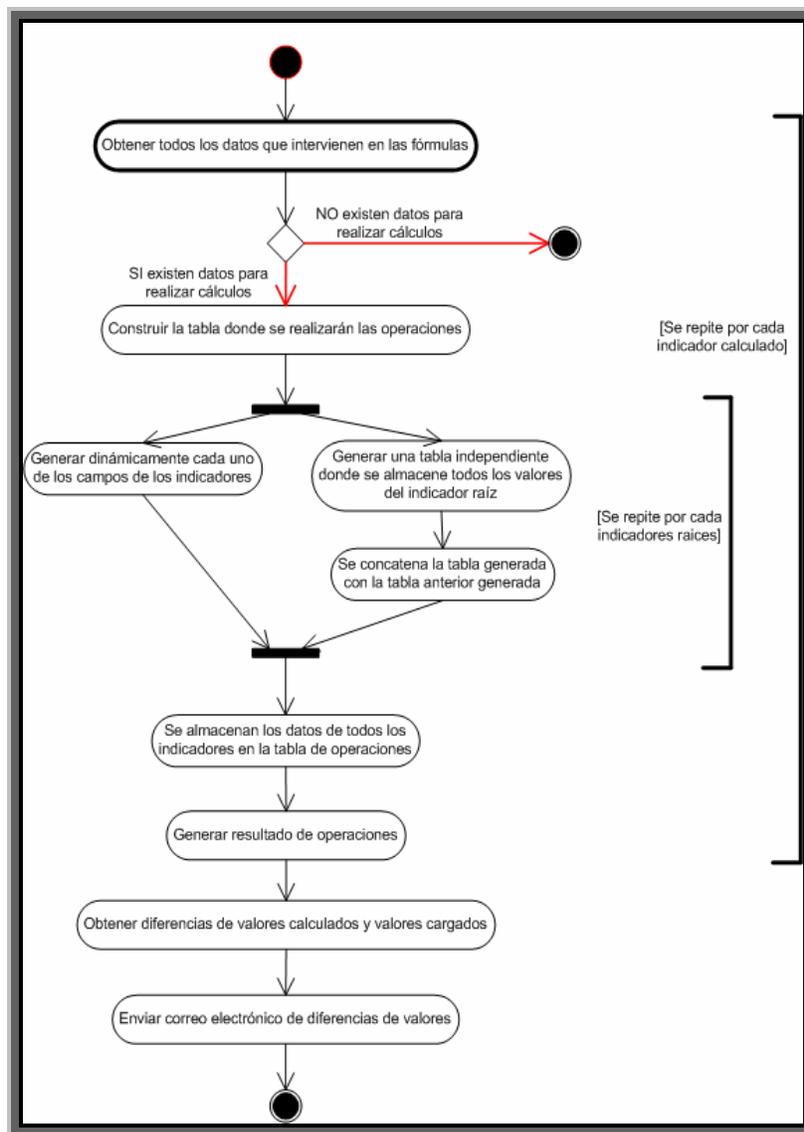


Figura 3.8. Procedimiento de cálculo de indicadores.

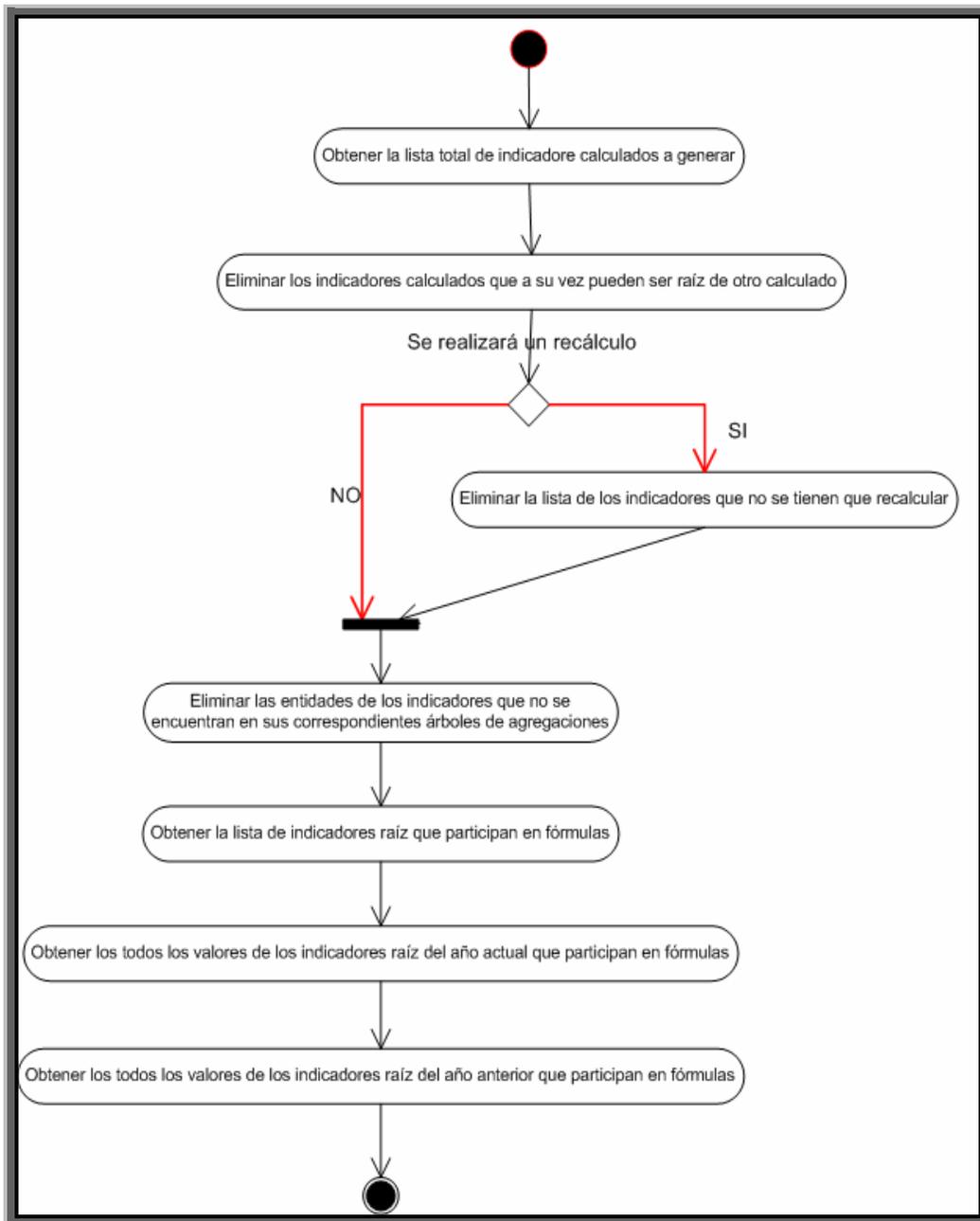


Figura 3.9. Procedimiento para obtener los datos que intervienen en las fórmulas.

3.1.2.3 Proceso de agregaciones.

Un valor agregado de un indicador es la agrupación (suma o promedio) de entidades hijas que pertenecen a una entidad padre. Un indicador A puede definir un conjunto de relaciones entidades padre-hijos distinto a un indicador B. Para modelar estas relaciones se creó una tabla donde fueron definidas. De esta manera, se puede generar varios conjuntos de relaciones, sin afectarle a las relaciones existentes. Al conjunto de relaciones de entidades padres con entidades hijas se le llama árbol de entidades y a las relaciones entre padres e hijos se les llama ramaje.

El proceso de agregación permite la agregación de entidades de diferentes niveles, es decir, que una entidad A, que tiene un nivel D, puede agregarse con una entidad B de nivel A. Esto es posible porque se define en el árbol de agregación. Por lo tanto, es de suma importancia que los árboles de agregación queden bien definidos para que se realicen las agregaciones tal como lo desean. La figura 3.10 muestra el proceso de agregación de indicadores.

3.1.2.4. Proceso de acumulación.

El valor acumulado de un indicador es la agrupación (suma, promedio o saldo) de los valores marginales a través del tiempo; es decir, de los periodos. Por ejemplo, el indicador A, cuya acumulación se definió como suma, tiene los valores marginales siguientes: 10 en enero, 14 en febrero y 12 en marzo. El valor acumulado de A en marzo es $10 + 14 + 12 = 36$.

El proceso genera los valores acumulados de los indicadores entre los periodos de un año; es decir, no es acumulativo entre años. Existen casos donde el indicador es creado a mitad de año, entonces, el número de periodos tiene que ser a partir de que se encuentra el valor del primer periodo almacenado en la base de datos. Lo mismo ocurre en caso de que se hayan registrado valores ceros los primeros periodos, el proceso toma en cuenta a partir de que existe un valor distinto de cero. La figura 3.14 muestra el proceso de acumulación de indicadores.

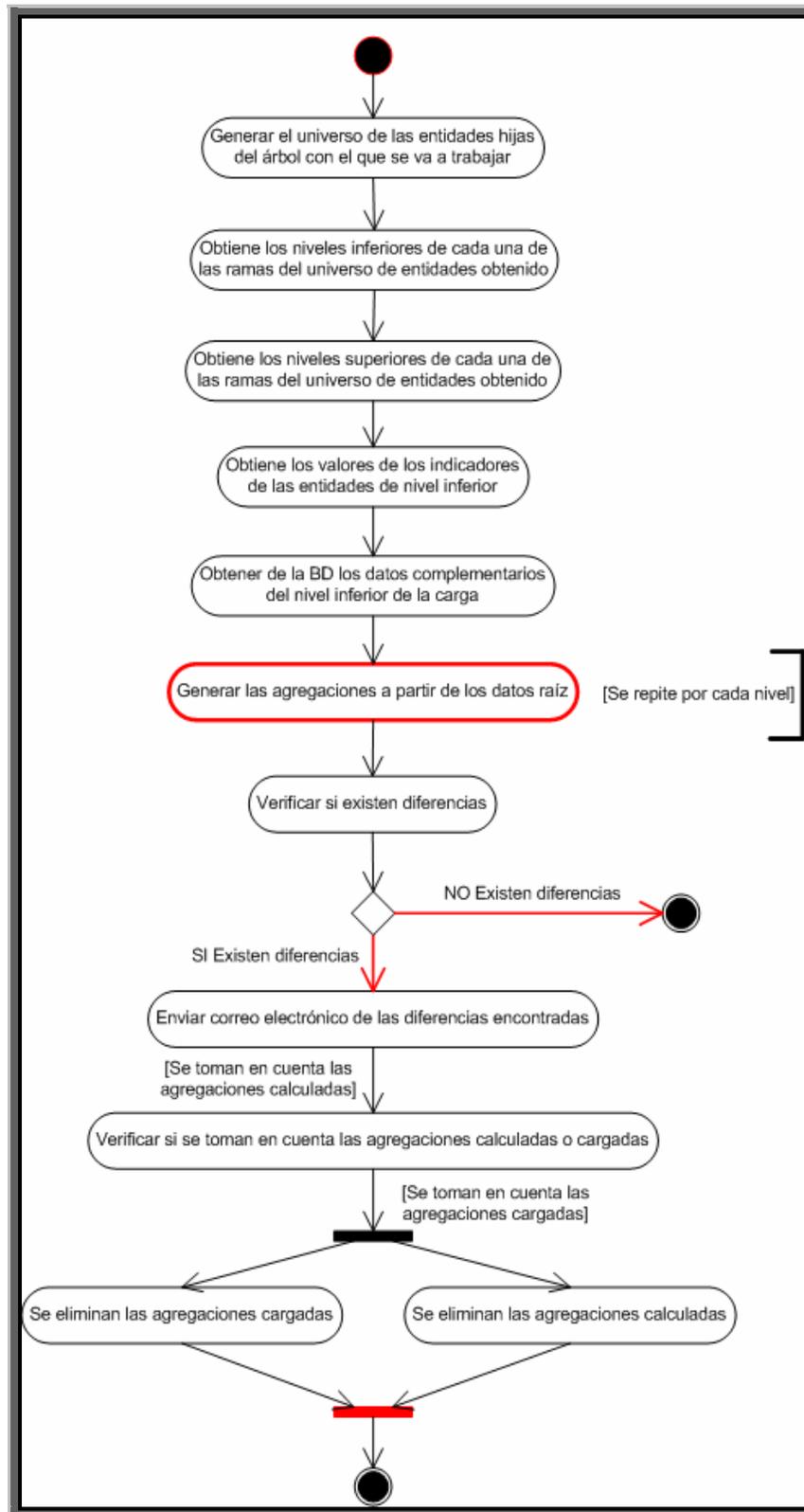


Figura 3.10. Procedimiento de agregaciones.

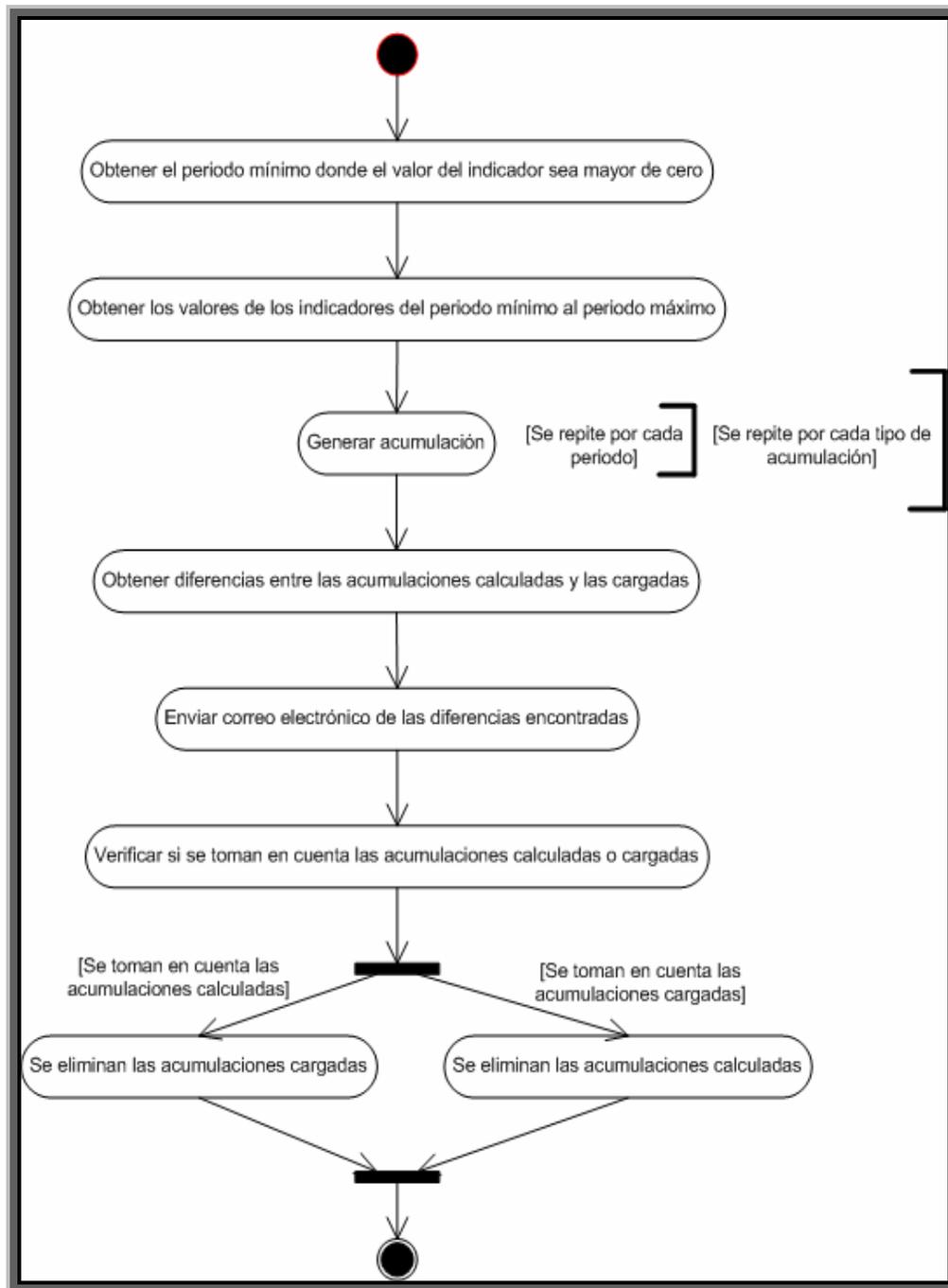


Figura 3.14. Procedimiento de acumulaciones.

3.1.3 Diseño de infraestructura.

El modelo de arquitectura de cómputo nos permite determinar cual es la arquitectura con la que funcionará mejor el sistema de información. En el caso del SCI, el trabajo de diseño se redujo a verificar que la infraestructura con que cuenta la SSIC era suficiente para que el sistema funcionara adecuadamente. La figura 3.15 nos muestra un diagrama general de la arquitectura de cómputo donde funciona, actualmente, el Sistema de Control de Indicadores y el Sistema de Información.

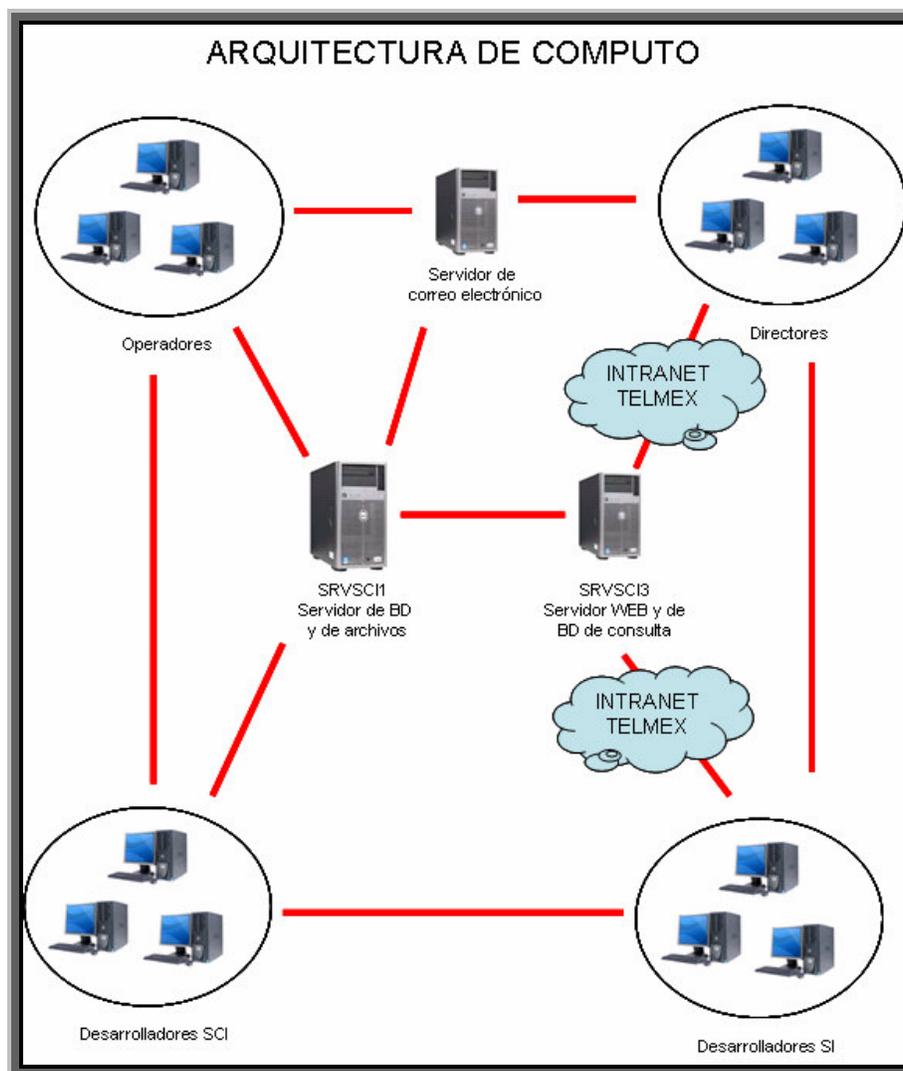


Figura 3.15. Arquitectura de cómputo.

A continuación describiremos las características del equipo de cómputo que se utiliza en el desarrollo y en la operación del sistema de cómputo.

- Servidor para el procesamiento de datos (SRVSCI1)
 - Servidor: HP ML570
 - Procesador: 4 procesadores Intel Xenon 3.33 GHz.
 - Memoria: 4 GB.
 - Discos duros: 1 de 40 GB, 2 de 70 GB y 4 de 150 GB.
 - Sistema operativo: Microsoft Windows 2003 Server.
 - Manejador de base de datos: Microsoft SQL Server 2000.

 - Servidor para la consulta de indicadores (SRVSCI3)
 - Servidor: DELL 6850
 - Procesador: 4 procesadores Intel Xenon 3.33 GHz.
 - Memoria: 4 GB.
 - Discos duros: 1 de 40 GB y 2 de 70 GB.
 - Sistema operativo: Microsoft Windows 2000 Server.
 - Manejador de base de datos: Microsoft SQL Server 2000.
 - Servidor Web: NetScape Enterprise Server.

 - Servidor de correo electrónico
 - Servidor: Desconocido.
 - Procesador: Desconocido.
 - Memoria: Desconocido.
 - Discos duros: Desconocido.
 - Sistema operativo: Desconocido.
 - Servicio de correo: Microsoft Exchange.

 - PC operadores
 - Procesador: Pentium II.
 - Memoria: 128 MB.
 - Disco duro: 10 GB.
 - Sistema operativo: Windows 98, Windows NT o Windows XP.
-

Servicio de correo: Contar con una cuenta de correo de Telmex y tener instalado el Outlook.

Otras características: estar conectado a la red de Telmex.

- PC Directores

Procesador: Pentium IV.

Memoria: 128 MB.

Disco duro: 20 GB.

Sistema operativo: Windows 2000 Server o Windows XP.

Servicio de correo: Contar con una cuenta de correo de Telmex y tener instalado el Outlook.

Otras características: estar conectado a la red de Telmex.

- PC Desarrolladores SCI

Procesador: Pentium III.

Memoria: 256 MB.

Disco duro: 40 GB.

Sistema operativo: Windows 2000 Server o Windows XP.

Servicio de correo: Contar con una cuenta de correo de Telmex y tener instalado el Outlook.

Herramientas de cómputo: Microsoft SQL Server 2000, Microsoft Visual Basic 6.0.

Otras características: estar conectado a la red de Telmex.

- PC Desarrolladores SI

Procesador: Pentium III.

Memoria: 512 MB.

Disco duro: 40 GB.

Sistema operativo: Windows 2000 Server o Windows XP.

Herramientas de cómputo: JBuilder 9, Microsoft SQL Server 2000.

Otras características: estar conectado a la red de Telmex.

Capítulo 4

Desarrollo del sistema

El SCI consta de un conjunto de módulos de administración y un conjunto de procesos que realizan el trabajo de procesamiento de indicadores.

Los módulos de administración permiten a los usuarios registrar y configurar las propiedades de los indicadores y los árboles de agregación de las entidades. Otras de sus funciones es el monitoreo del flujo de la carga de indicadores, desde el depósito del archivo hasta el almacenamiento de los indicadores en el servidor de consultas.

Los procesos realizan las tareas necesarias: carga, validación de datos y cálculos, y almacenamiento y transferencia de información; para que los indicadores puedan ser consultados por los directores. El procesamiento es totalmente automático, por lo tanto, los usuarios no intervienen, en lo absoluto, en la obtención de los resultados.

4.1 Administración del sistema

Para obtener los resultados que se esperan, es importante que los indicadores sean configurados adecuadamente. En este trabajo, únicamente, se presentan los módulos más importantes que ayudan a tener un entendimiento general del funcionamiento del sistema. Los módulos que se describen son:

- Registro de indicadores.
- Construcción de fórmulas.

- Administración de cargas.

4.1.1 Módulo de registro de indicadores

El primer paso en el procesamiento de un indicador es registrarlo en el sistema; definir sus características, tales como: nombre, unidad de medición, responsable, tipo de agregación, tipo de acumulación, fórmula, en caso de tenerla, etc. El módulo, permite la captura, registro y consulta de las propiedades del indicador. La figura 4.1 muestra la ventana de registro de indicadores.

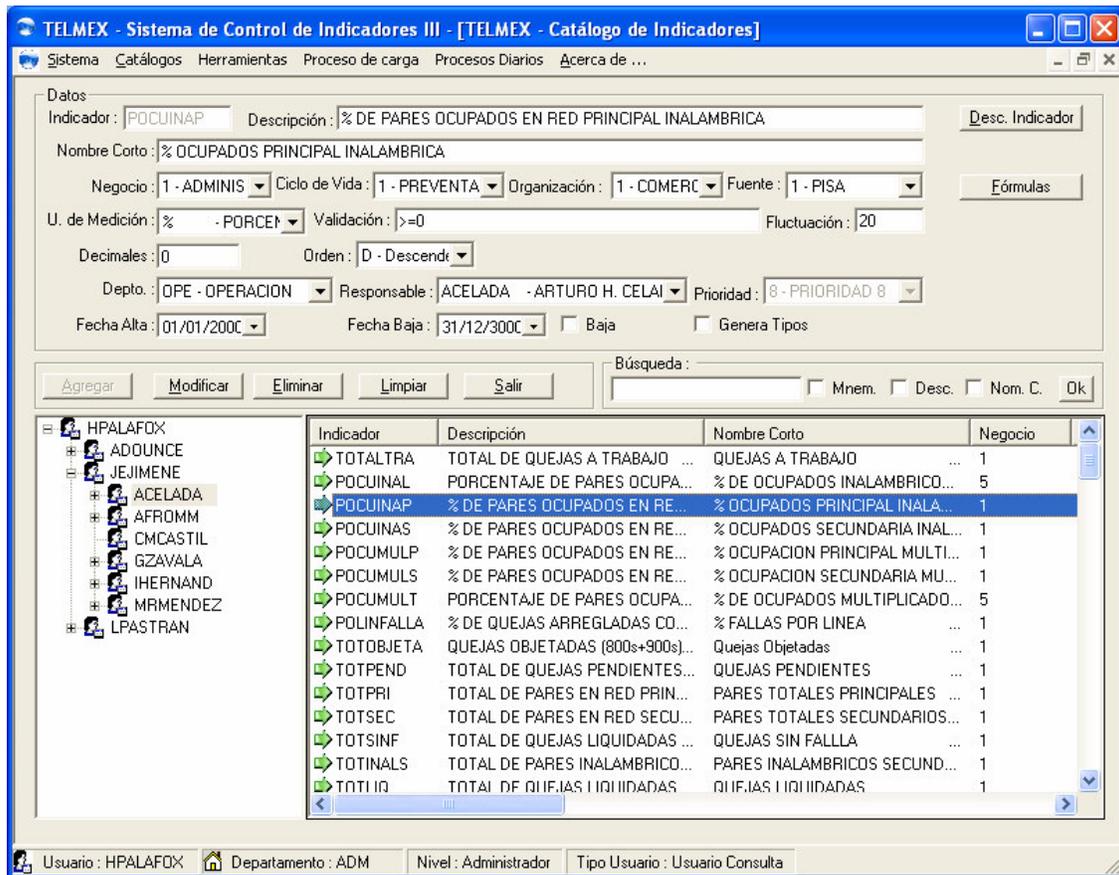


Figura 4.1. Ventana de registro de indicadores.

La parte superior de la ventana es la sección de edición de las propiedades del indicador. Los campos describen las propiedades del indicador; son de gran importancia, debido a que definirán el comportamiento de los procesos para este indicador. La parte

inferior izquierda de la ventana presenta un listado de usuarios con orden jerárquico. El lado derecho muestra una lista de indicadores que está asociado a los usuarios o responsables. Cuando es seleccionado un usuario, el sistema presenta los indicadores que tiene asociados. Cuando un indicador del listado es seleccionado, los datos se presentan en la sección de edición de datos. Si un usuario es jefe, entonces, puede ver y modificar todos los indicadores de la gente a su cargo. Sin embargo, las modificaciones quedan registradas en una bitácora de cambios, de tal manera que, se registra el usuario quien realizó el cambio y la fecha cuando se hizo.

4.1.2 Módulo de construcción de fórmulas

Una vez registrado el indicador, es posible que su valor no sea cargado, sino calculado a partir de otros. Para realizar esto, es necesario crear una fórmula, donde el resultado de su evaluación sea asignada al indicador.

En la parte superior derecha de la ventana de la figura 4.1, se aprecia el botón de “Fórmulas”, que sirve para activar el submódulo de construcción de fórmulas. Para entrar al submódulo es necesario seleccionar un indicador.

Las fórmulas son utilizadas en el proceso de cálculos durante la carga de datos. Posteriormente describiremos este funcionamiento del sistema.

La figura 4.2 muestra la ventana de construcción de fórmulas. En la parte superior de la ventana existen dos listas de datos, una muestra los indicadores disponibles en la BD; la segunda los operadores y palabras reservadas permitidas en la fórmula. La captura de la fórmula es directa. El usuario necesita saber cual es la nomenclatura que deben tener los indicadores. La notación de los indicadores es la siguiente:

- a) Los indicadores van entre corchetes

- b) El indicador de periodo o tipo distinto debe llevar un prefijo antes del nombre del indicador. El prefijo puede ser `_PN01` o `_TIPA`, donde 01 es el periodo anterior y A es el tipo del indicador, en este ejemplo, es tipo acumulado.

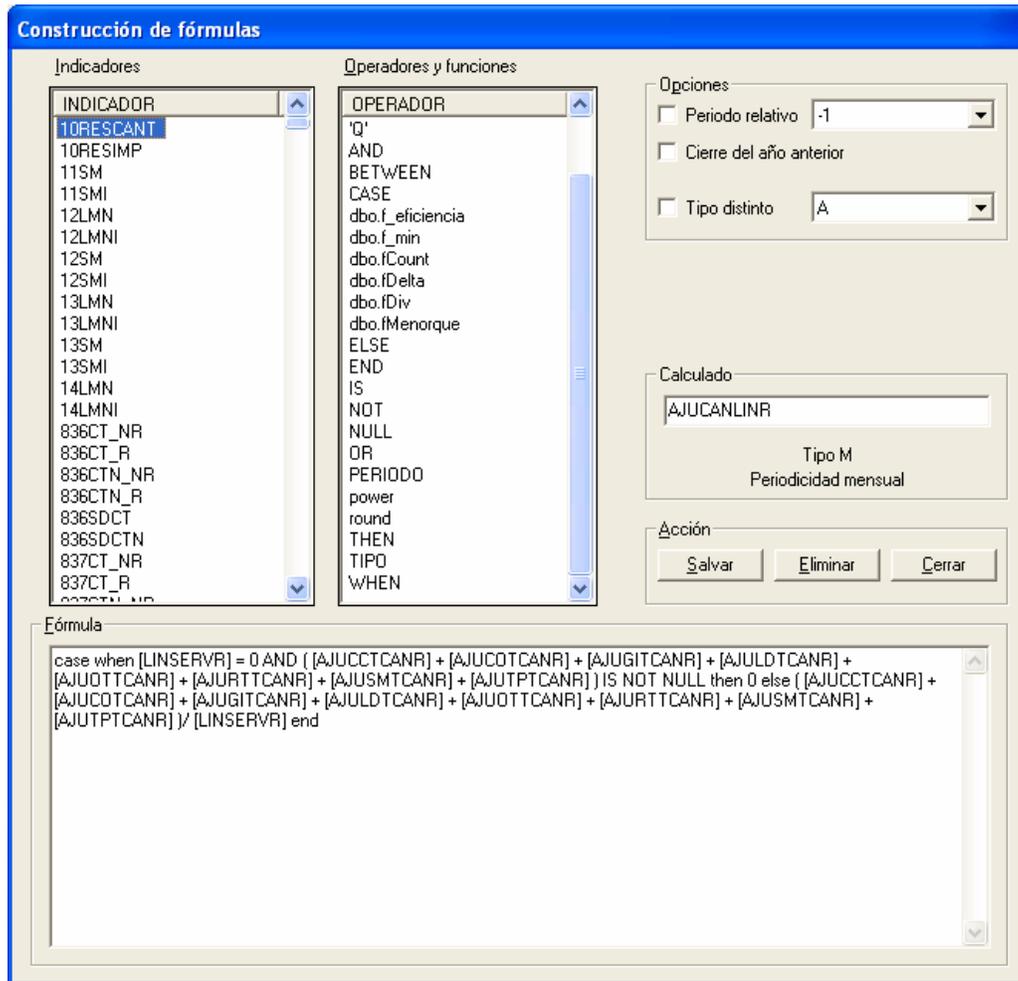


Figura 4.2. Ventana de construcción de fórmulas.

La validación de la sintaxis de la fórmula la realiza SQL Server. Cuando el usuario salva la fórmula, antes de almacenarla en la BD, se verifica que esté sintácticamente correcta. Esto se realiza construyendo una tabla con una columna por cada indicador y con una columna calculada, donde se coloca la fórmula. El manejador de BD, revisa la validez sintáctica de la fórmula. Si es correcta, permite construir la tabla, en caso negativo, no lo permite y envía un mensaje de error, que es manipulado por el sistema y se lo presenta al usuario.

4.1.3 Módulo de administración de cargas

Las etapas en el flujo del proceso general de cargas, depende de la definición del indicador. La figura 4.3 muestra las etapas por las que puede pasar una carga. Cada etapa define un determinado conjunto de procedimientos que se aplican a los indicadores. La gráfica muestra las claves M1, M2,..., M6, que representan, los disparadores, llamados “monitores”, en la SSIC. Son tareas programadas que se ejecutan automáticamente en periodos de tiempo preestablecido. Casi todos los monitores están definidos para ejecutarse cada minuto. La separación del flujo completo en monitores permite mayor control sobre el proceso total.

Una carga inicia al momento en que el primer monitor detecta un archivo para procesar. La figura 4.4 muestra la estructura de directorios de archivos que se diseñó en el servidor para que el personal operativo coloque los archivos para su carga. Los directorios de los usuarios tienen los privilegios de acceso desde el sistema operativo. Los privilegios se dan por usuario, por tal motivo, nadie puede ver y modificar lo de otro usuario.

El módulo de administración de cargas sirve para visualizar la etapa en que se encuentran las cargas de indicadores. Presenta el trabajo que está realizando el sistema. Todos los usuarios de la SSIC pueden entrar al módulo para observar el comportamiento de sus cargas. La figura 4.5 muestra la ventana del módulo de administración de cargas.

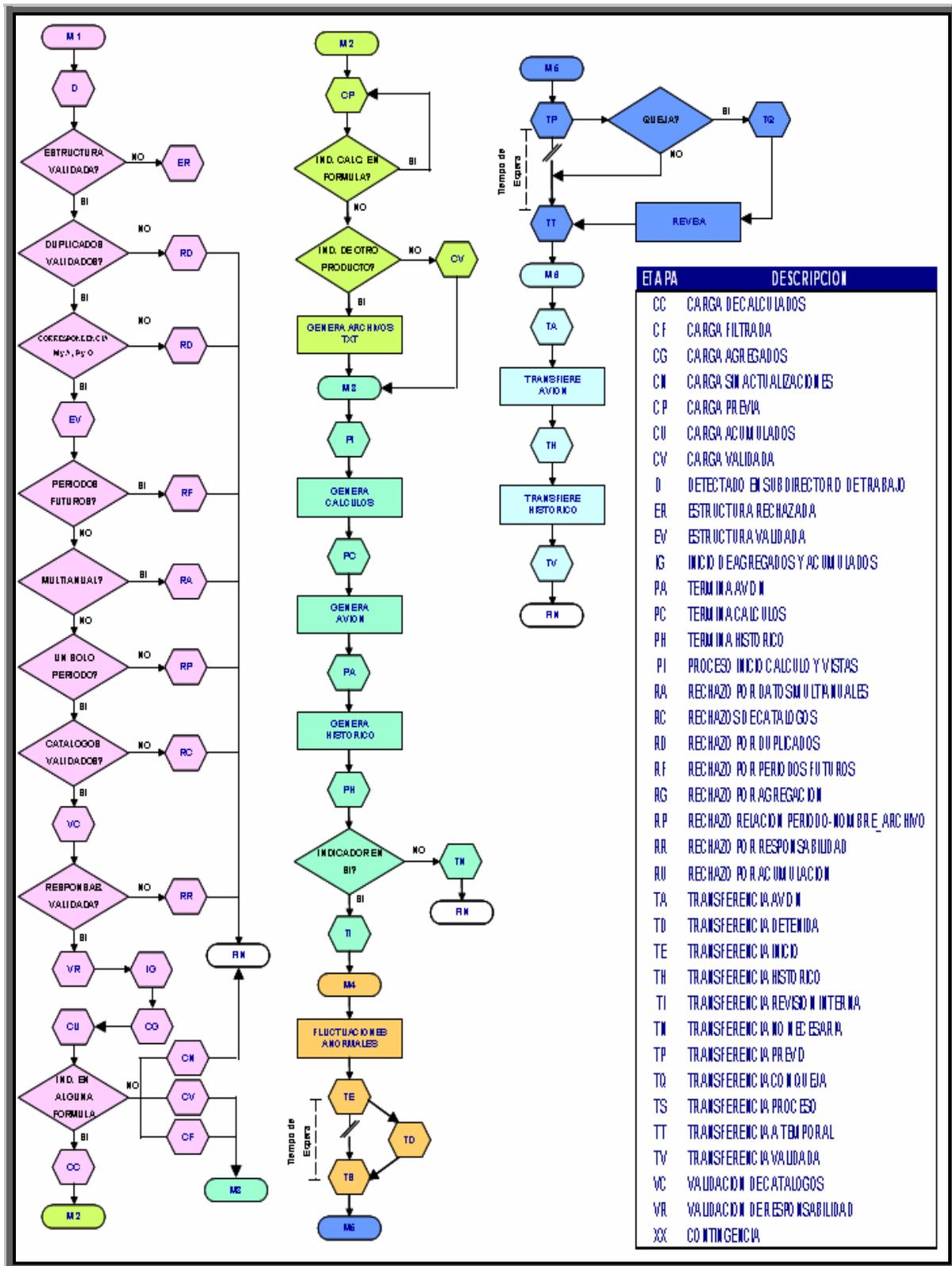


Figura 4.3. Diagrama de las etapas del proceso de carga de indicadores.

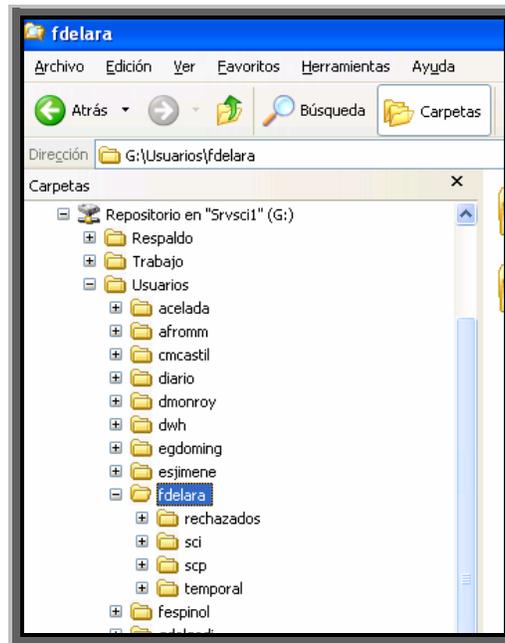


Figura 4.4. Estructura de directorios para colocar los archivos a procesar.

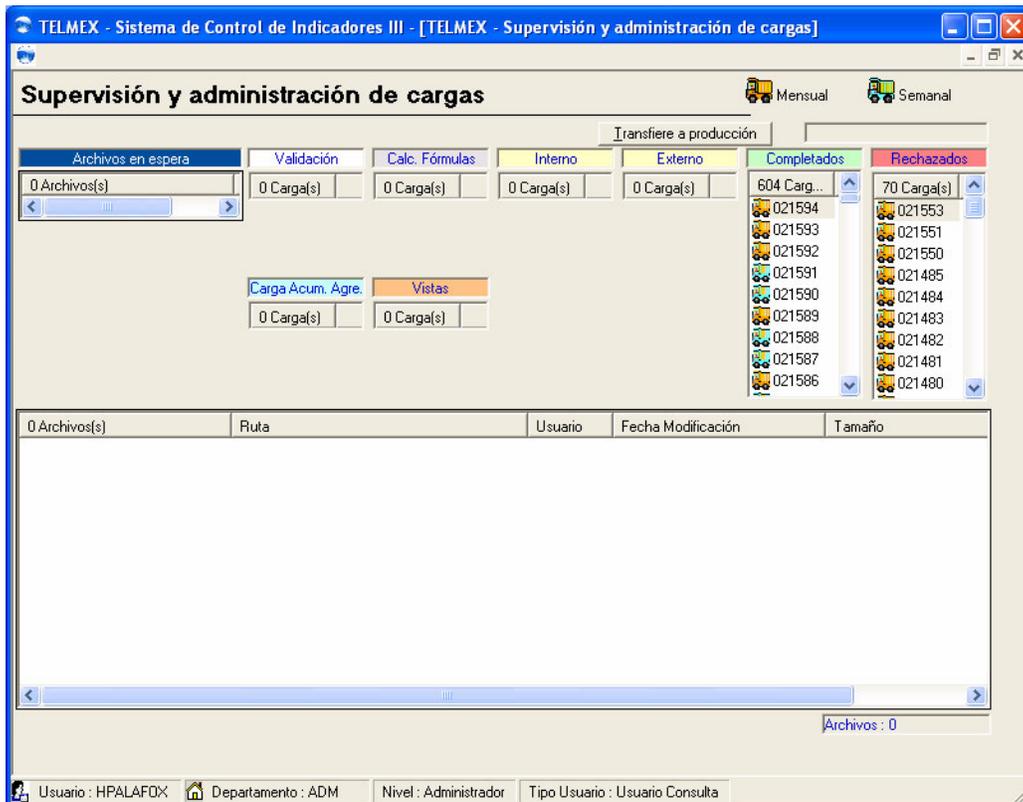


Figura 4.5. Ventana de supervisión y administración de carga de archivos.

4.2 Algoritmos importantes

El código de un programa es el elemento principal en la construcción de un sistema de información. Entender el funcionamiento del código, sin tener referencias, es complejo. Algunos programadores escriben sus algoritmos tan complejos que no llegan a ser entendibles para otros programadores. Una buena práctica al capturar el código es escribir comentarios de lo que hace el algoritmo.

Los algoritmos que se presentan en esta sección son: agregaciones, acumulaciones y cálculo de indicadores.

Para comprender el funcionamiento de los algoritmos que se describirán, comenzaremos por describir el formato de los archivos que se procesan en la carga. La estructura es la siguiente:

ENTIDAD	INDICADOR	SEGMENTO	TIPO	VALOR	FECHA
CD	INCLINSERV	T	M	150,321.00	30-03-2006
CH	INCLINSERV	T	M	132,425.00	30-03-2006

En el archivo, cada dato está separado por un tabulador. No contiene encabezados, únicamente los datos. Algunas condiciones que debe tener la información contenida en el archivo son:

- Los datos deben ser del mismo año.
- Los indicadores deben pertenecer a un mismo producto. Esta relación se establece en la tabla cat_responsabilidad.
- No pueden ser cargados valores de tipo A y Q sin tener su correspondiente valor de tipo M y Q, pero si puede darse el caso de cargar valores de tipo M y P sin cargar valores de tipo A y Q.
- La periodicidad (diaria, semanal, mensual) debe ser la misma.

Las tablas que se utilizan para realizar la mayor parte de los procesos, en la carga de indicadores, entre ellos: agregaciones, acumulaciones y cálculo de indicadores; tienen una estructura similar a la estructura del archivo. Es importante recordarlo, porque las sentencias SQL utilizadas están escritas considerando esta estructura.

Anterior a la automatización de las agregaciones, acumulaciones y cálculos, el personal de la SSIC realizaba las operaciones por separado. Los resultados obtenidos eran colocados en los archivos de carga, junto con los otros valores de indicadores. Durante la implantación de estos procesos fue necesario considerar que la información de agregados, acumulados y calculados podía estar en el archivo, por lo tanto, fue necesario programar una sección de comparación de datos cargados contra los obtenidos por el proceso.

El resultado obtenido de los tres procesos es independiente de cual se realice primero, pero para mantener un orden en el flujo de procesos, se decidió realizar los procesos en el siguiente orden: 1) agregaciones, 2) acumulaciones y 3) cálculo de indicadores. La figura 4.6 muestra el orden en que se ejecutan los 3 procesos que realizan operaciones matemáticas.

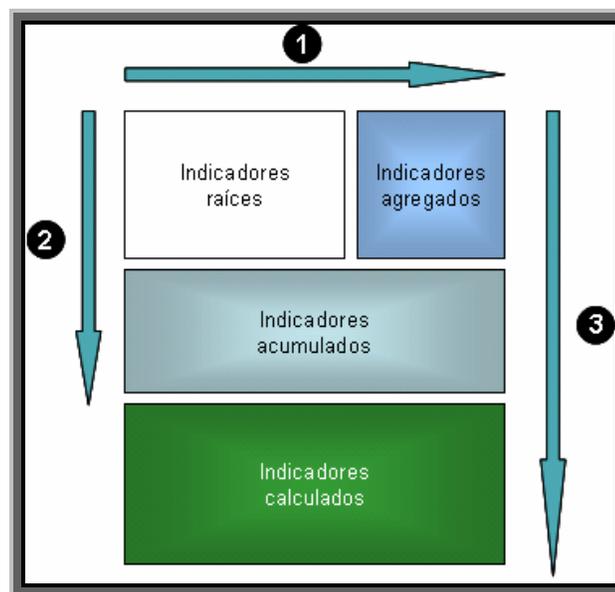


Figura 4.6. Orden en la generación de valores de indicadores.

A continuación se presentan los algoritmos generados para construir los procesos mencionados.

4.2.1 Proceso de agregación

Cuando se carga un archivo, no necesariamente contiene todas las entidades hijas para calcular las entidades padres, en estos casos el proceso tiene que buscar en la BD las entidades que no fueron cargadas.

Debido a que las agregaciones y acumulaciones se realizan una seguida de la otra fue necesario construir un procedimiento almacenado que ejecuta ambos procesos. El procedimiento almacenado, llamado `sp_Principal_Agrega_Acumula` se encarga de crear el ambiente de trabajo; almacena en una tabla la información que se procesará y recibe en la misma tabla el resultado de los valores agregados y acumulados. Después de haber obtenido los resultados, depura los datos que ya se encuentran en la BD, y en el caso que un usuario haya cargado resultados de indicadores agregados o acumulados, el procedimiento se encarga de compararlos con los generados por el propio proceso, y en caso de existir diferencias, el procedimiento envía un correo electrónico, indicando que los valores cargados son erróneos. El pseudo código 4.1 muestra las tareas que realiza el SP `sp_Principal_Agrega_Acumula`.

```
sp_Principal_Agrega_Acumula:
Entrada: No. Carga; Tipo de proceso.
Salida: 0 cuando termina sin problemas y <> 0 cuando se presenta un error.
INICIO
1) Crear tablas temporales auxiliares.
2) Almacenar los datos a procesar en una tabla temporal de trabajo.
3) Obtener los valores de inicio de la periodicidad, año y nombre de tabla de
almacenamiento.
4) Actualizar el estado del proceso a agregado.
5) Ejecutar el procedimiento de agregación para obtener los valores agregados.
6) Actualizar el estado del proceso acumulado.
7) Ejecutar el procedimiento de acumulación para obtener los valores acumulados.
8) De la información procesada se eliminan los que tiene mismo valor que la tabla de
almacenamiento.
9) Eliminar de la tabla de paso los registros de los indicadores agregados y acumulados que
han sido cargados y que calculó su valor el sistema.
10) Insertar a la tabla de paso los valores agregados y acumulados que fueron calculados
por el sistema.
FIN
```

Pseudo código 4.1. Procedimiento principal de agregación y acumulación.

Para generar un valor agregado es importante saber cual es la relación que existe entre entidades hijas y padres. En la figura 4.7 mostramos un ejemplo de árbol de agregación. Actualmente existen 21 árboles de agregación que son utilizados para los distintos indicadores. Por lo tanto, el procedimiento tiene que determinar cual es el que tiene que usar, dependiendo el conjunto de indicadores que tiene que procesar. El pseudo código 4.2 muestra las tareas que realiza para generar los cálculos de agregación.

Los problemas importantes resueltos en este proceso fueron:

- Realizar el recorrido del árbol a través de las ramas de los árboles.
- Generar un código, de tal manera, que una misma sentencia sirviera para todas variables a considerar: periodicidad, año, árbol a utilizar y operación matemática a realizar.

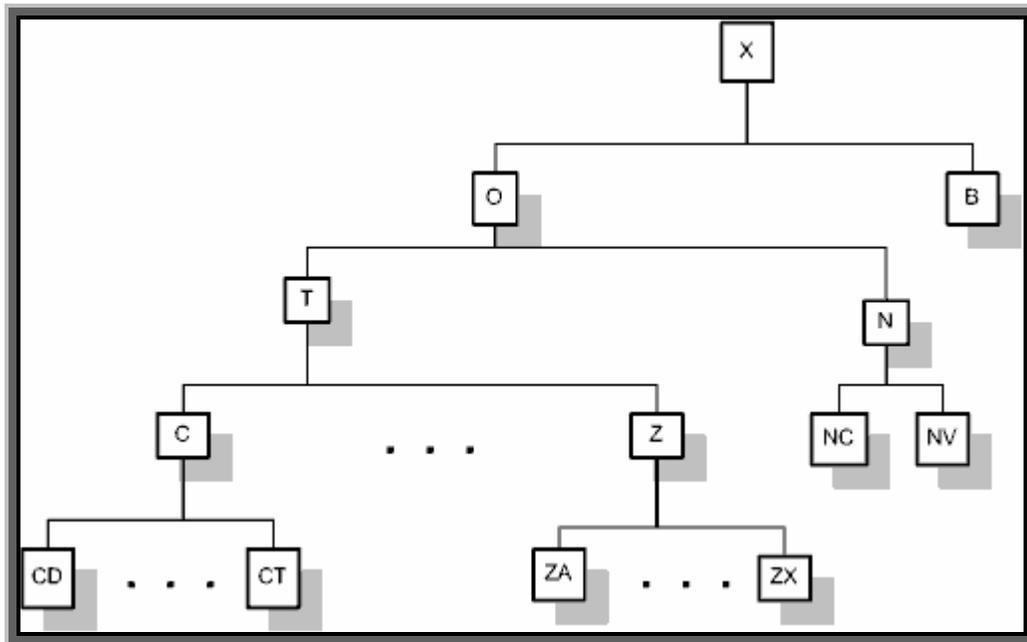


Figura 4.7. Diagrama de árbol de agregación.

```
sp_Agregacion:
Entrada: Periodicidad; año; Tipo de proceso.
Salida: 0 cuando termina sin problemas y <> 0 cuando se presenta un error.
        Resultado de agregaciones en la tabla de trabajo.
INICIO
1) Crear tablas temporales auxiliares.

2) Inicializar las variables usuario, mail, Número de Carga, producto, Número de árbol de
agregación, nombre de la tabla de datos y nombre del campo fecha.

3) Obtener el esquema del árbol de agregación para los indicadores cargados.

4) Obtener las entidades de los niveles inferiores del esquema de árbol de agregación.

5) Obtener las entidades padres de todos los niveles del esquema de árbol de agregación.

6) Obtener el universo de datos. Realizar un cruce de información entre el esquema de árbol
de agregación, contra los indicadores cargados.

7) Almacenar en una tabla auxiliar "#temporal_agrega" los indicadores cargados, cuyo nivel
de entidad coincide con el nivel inferior del esquema del árbol de agregación (4).

8) Obtener de la base de datos el resto de indicadores, cuya entidad se encuentra en el
esquema (3), pero no fue cargada.

9) Para cada nivel padre en el esquema del árbol de agregación se hace:

9.1) Realizar un cruce de información para obtener la agregación (suma o promedio) del
nivel que se está procesando y se almacena en la tabla auxiliar (7).

10) Comparar los valores cargados y los calculados para verificar si existe algún agregado
que fuera cargado, para determinar si existe diferencia. En caso de haber diferencias,
estas se almacenan en una tabla de trabajo.

11) Preguntar por el tipo de proceso.

11.1) Si el tipo de proceso = 0.

11.1.1) En caso de existir diferencias en valores agregados, el proceso manda un correo de
aviso.

11.1.2) Eliminar los indicadores calculados por el proceso y que tengan la configuración
para que prevalezcan los valores cargados.

11.1.3) Eliminar los indicadores cargados y que tengan la configuración para que prevalezca
el valor calculado por el proceso.

11.1.4) Almacenar los valores en la tabla de trabajo que contiene los indicadores cargados.

11.2) Si el tipo de proceso = 10.

11.2.1) En caso de existir diferencias en valores agregados, el proceso manda un correo de
aviso.

11.3) Si el tipo de proceso = 20.

11.3.1) Eliminar todos los indicadores calculados por el proceso y los sustituye por los
cargados.
FIN
```

Pseudo código 4.2. Proceso de agregación.

4.2.2 Proceso de acumulación

Para genera un valor acumulado es necesario tomar en cuenta los valores marginales de todos los periodos del año a la fecha. Debido a que la carga de información no es homogénea, se deben tener algunas consideraciones, para realizar las agregaciones:

- No todos los indicadores comienzan desde el periodo 1, esto se debe a que su creación pudo ser posterior al primer periodo. En algunos casos, los usuarios cargaron valores ceros en los primeros periodos, hasta antes del periodo en que se dio su creación y carga de valores, por tal motivo, se tienen que descartar esos registros.
- Una carga puede contener indicadores de varios periodos a la vez.
- Se realiza una carga de un periodo o periodos anteriores a la última fecha, lo que provoca que se tengan que generar las acumulaciones de los periodos posteriores a los que se están cargando.
- Verificar cuando un indicador que se carga tiene valores de más de un periodo.

Estas consideraciones provocan que sea más compleja la programación de los algoritmos. En la gráfica de la figura 4.8 se trata de representar las consideraciones mencionadas. El pseudo código 4.3 muestra las tareas que realiza el proceso para generar los valores agregados.

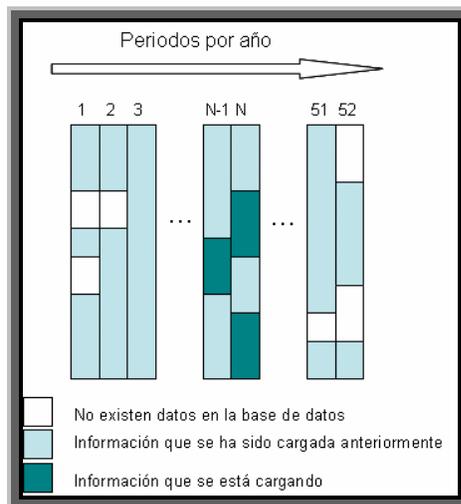


Figura 4.8. Representación gráfica de los datos acumulados en la BD.

```
sp_Acumulacion:
Entrada: Periodicidad; año; Tipo de proceso.
Salida: 0 cuando termina sin problemas y <> 0 cuando se presenta un error.
        Resultado de agregaciones en la tabla de trabajo.
INICIO
1) Crear tablas temporales auxiliares.

2) Inicializar las variables usuario, mail, Número de Carga, producto, nombre de la tabla
de datos, nombre del campo fecha y número máximo de periodo.

3) Obtener una tabla auxiliar pivote, donde la llave es entidad, indicador, segmento y
tipo, pero no se toma en cuenta el periodo.

4) Tomando como base la tabla pivote (3), se obtiene el periodo mínimo de cada indicador
que está almacenado en la base de datos.

5) Los valores de los indicadores cargados se pasan a una tabla de trabajo
"#temporal_acumula", cuya operación de acumulación esté configurada como (suma, promedio o
saldo).

6) Obtener de la base de datos los valores de los indicadores de periodos anteriores a los
de la carga y los deposita en la tabla de trabajo (5).

7) Almacenar en una tabla auxiliar "#temporal_acumula_max" los indicadores cuyo periodo es
el máximo.

8) Para cada tipo de operación (suma, promedio o saldo) realiza:

8.1) Del periodo mínimo al periodo máximo realiza.

8.1.1) Armar la sentencia a ejecutar.

8.1.2) Ejecutar la sentencia insert armada y almacena los datos en la tabla
"#indicadores_sumados".

8.1.3) Incrementar en 1 el periodo.

9) Se realiza una comparación entre los valores calculados y los cargados para verificar si
fueron cargado algunos valores agregados y determinar si existen diferencias.

10) Preguntar por el tipo de proceso.

10.1) Si el tipo de proceso = 0.

10.1.1) En caso de existir diferencias en valores acumulados, el proceso manda un correo de
aviso.

10.1.2) Eliminar los indicadores calculados por el proceso y que tengan la configuración
para que prevalezcan los valores cargados.

10.1.3) Eliminar los indicadores cargados y que tengan la configuración para que prevalezca
el valor calculado por el proceso.

10.1.4) Almacenar los valores en la tabla de trabajo que contiene los indicadores cargados.

10.2) Si el tipo de proceso = 10.

10.2.1) En caso de existir diferencias en valores acumulados, el proceso manda un correo de
aviso.

10.3) Si el tipo de proceso = 20.

10.3.1) Eliminar todos los indicadores calculados por el proceso y los sustituye por los
cargados.
FIN
```

Pseudo código 4.3. Proceso de acumulación.

4.2.3 Proceso de cálculo de indicadores

El proceso de cálculos se hizo complejo, debido a todas las consideraciones que estableció el cliente, que debía realizar. Entre las más importantes se encuentran las siguientes:

- Cuando los indicadores cargados participan en una fórmula, el sistema tiene que determinar si falta algún elemento dentro de la carga, en caso de faltar, el proceso tiene que buscar los valores de los indicadores que se encuentra en la BD; si existen, se realiza la operación, en otro caso, no.
- Debe permitir el anidamiento en las fórmulas, por ejemplo: $c = a + b$, $e = c + d$, donde c tiene un nivel de anidamiento como fórmula. El nivel de anidamiento puede ser variable. Existen casos que presentan hasta nueve niveles de anidamiento.
- En un cálculo pueden intervenir periodos distintos, por ejemplo: $b = a(p) - a(p-1)$, donde $a(p)$ es el valor del indicador del periodo que se está cargando y $a(p-1)$ es el valor del indicador del periodo anterior. En general, este valor se obtiene de la BD, puesto que ya se encuentra almacenado. Un periodo distinto puede ser -1, -3, -6 ó diciembre del año pasado. Este parámetro es variable y el usuario puede definir el periodo que necesite.
- Una carga puede contener varios periodos consecutivos, por lo tanto, se debe tener cuidado con la fórmulas, para no repetir resultados u obtener valores incorrectos.
- Una fórmula puede contener tipos distintos, por ejemplo: $c(M) = b(Q) / e(Q)$, donde $c(M)$ es el indicador calculado de tipo M, $b(Q)$ es el indicador b de tipo Q y $e(Q)$ es el indicador e de tipo Q.
- El proceso de cálculos debe ser capaz de soportar fórmulas donde intervenga una gran cantidad de indicadores. Hasta el momento, la fórmula más grande que tienen usa alrededor de cien indicadores.
- La información mensual, semanal y diaria se almacenan en tablas distintas y utilizan periodos distintos, por lo tanto el proceso debe funcionar para las distintas periodicidades.

- Una carga puede contener valores de indicadores calculados.

Debido a la complejidad del proceso, se crearon varios Procedimientos Almacenados (SP), de manera, que cada uno tuviera una funcionalidad bien definida, para ser utilizados en otros procesos, como por ejemplo la auditoria o recálculo, sin tener que modificar el código o tener que estar construyendo otro código para funcionalidades similares. La figura 4.9 muestra como se encuentran organizados los SP utilizados en el proceso de cálculos.

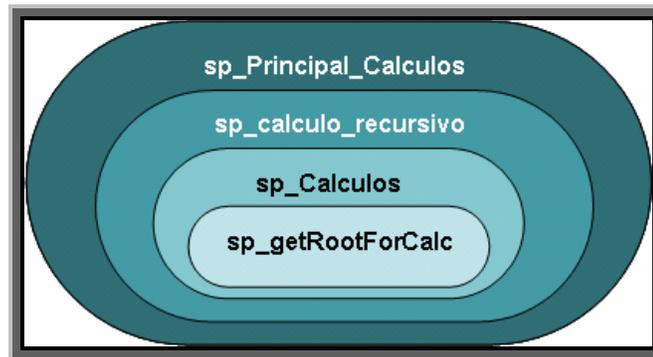


Figura 4.9. División del proceso en procedimientos definidos por funcionalidad.

4.2.3.1. **sp_Principal_Calculos.**

Obtiene los datos que se procesarán, los almacena en una tabla de trabajo que servirá para que posteriormente almacene los datos calculados. Cuando se cargan indicadores que son calculados, estos se comparan con los calculados por el proceso y envía un correo electrónico de los valores que resulten distintos entre los calculados y cargados.

Como mencionamos anteriormente, una carga debe contener indicadores del mismo producto, por lo tanto, los indicadores calculados que son de otro producto deben ser registrados en una carga distinta, para tal caso, se generan archivos de indicadores calculados, pero de productos diferentes, para que se procese como carga distinta.

El pseudo código 4.4 muestra las tareas que realiza el procedimiento almacenado, antes y después de que se generan los valores calculados.

```
sp_Principal_Calculos:
Entrada:
Salida: 0 cuando termina sin problemas y <> 0 cuando se presenta un error.
INICIO
1) Crear tablas temporales auxiliares.
2) Obtener el número de carga de bitacora_carga donde el estado = 'CC'.
3) Si el número de carga > 0.
3.1) Actualizar el estado de bitacora_carga a 'CR'.
3.2) Obtener los valores de las tablas calculos_insertados y calculos_modificados y los
deposita en una tabla de trabajo "#temporal_proceso".
3.3) Obtener el año de los datos cargados, el nombre de la tabla de datos, de acuerdo a la
periodicidad, el producto, usuario, correo electrónico y periodo mínimo a cargar.
3.4) Ejecutar el proceso de recursividad de cálculos.
3.5) Almacenar los resultados de los cálculos en la tabla "#temporal_proceso_total".
3.6) Como es posible que existan registros duplicados, por la recursividad, se filtran los
valores más actuales.
3.7) Se ejecuta un ciclo para los distintos productos, generados como el resultado de los
cálculos.
3.7.1) Obtener el nombre del nuevo archivo.
3.7.2) Crear el nuevo archivo y lo coloca en el directorio adecuado.
3.8) Los indicadores que fueron procesados en el ciclo anterior, son eliminados.
3.9) Comparar contra los datos almacenados en BD y se eliminan los valores calculados que
son iguales a los almacenados.
3.10) Se separan los registros de los indicadores calculados en tablas diferentes, de
acuerdo a sus características: modificados y nuevos.
3.11) Crear un nuevo número de carga.
3.12) Insertar y modificar los valores a la base de datos.
3.13) Insertar en una tabla de paso "temporal_proceso", para que utilice la información en
otros procesos.
3.14) Generar las cifras de control.
3.15) Actualizar el estado a "CV", para continuar el proceso.
3.16) Eliminar de las tablas de origen (3.2), la información procesada.
3.17) Eliminar de la tabla "indicadores_recalculados", en caso de que se tratara un
recálculo.
3.18) Si se procesaron indicadores de Latinoamérica, entonces se inserta en la tabla
"bitacora_productos_latam", para que otro programa genere un reporte en documento PDF de
los resultados obtenidos.
3.19) Si existen diferencias de indicadores cargados con los calculados.
3.19.1) Enviar correo electrónico al responsable para avisar que existen diferencias.
FIN
```

Pseudo código 4.4. Procedimiento almacenado principal.

4.2.3.2. sp_calculo_recursoivo.

Detecta los indicadores que tengan un nivel de anidamiento en fórmulas. El SP se ejecuta recursivamente tantas veces encuentre indicadores que participan en fórmulas para obtener el valor de otros indicadores calculados.

El código es sencillo y reducido, pero el poder que tiene su funcionalidad es grande, porque permite a los usuarios despreocuparse de actualizar los resultados de fórmulas que están relacionadas entre indicadores. El pseudo código 4.5 muestra las tareas de verificación que realiza el procedimiento almacenado para determinar si existe anidamiento en las fórmulas.

```
sp_calculo_recursoivo:
Entrada: Carga; periodicidad; producto; año; No. Recursividad
Salida: 0 cuando termina sin problemas y <> 0 cuando se presenta un error.
        Resultado de los cálculos almacenados en la tabla de trabajo.

INICIO
1) Si la tabla temporal "#temporal_proceso" contiene registros de indicadores que
participan en una fórmula, entonces.

2) Ejecutar el procedimiento almacenado de cálculos.

3) Eliminar de la tabla "#temporal_proceso" los indicadores raíz y deja los calculados.

4) Manda ejecutar el proceso de acumulaciones.

5) Si se generaron indicadores calculados, entonces.

5.1) Eliminar los indicadores calculados que se obtuvieron en una pasada anterior de la
recursividad y se consideran los nuevos, almacenándose en la tabla
"#temporal_proceso_total"

5.2) Eliminar los indicadores calculados que resulten ser de un producto distinto a la
carga original.

5.3) Eliminar los valores de los indicadores acumulados, para que en la siguiente pasada de
recursividad no sean tomados en cuenta para los cálculos.

5.4) Se manda a ejecutar a si mismo "sp_calculo_recursoivo"

FIN
```

Pseudo código 4.5. Procedimiento almacenado de recursividad.

4.2.3.3. sp_calculos.

Es la parte medular del proceso de cálculos. Se encarga de obtener los resultados de las fórmulas donde participan los indicadores cargados.

Para construir este proceso se utilizó una característica de las tablas de SQL Server. Una tabla en SQL Server puede tener columnas calculadas. Una columna calculada es una columna virtual no almacenada físicamente en la tabla. Se calcula a partir de una expresión mediante otras columnas de la misma tabla. Por ejemplo, una columna calculada puede tener la definición *costo AS precio * cantidad*. La expresión puede ser un nombre de columna no calculada, una constante, una función, una variable o cualquier combinación de estos elementos conectados mediante uno o más operadores.

La estructura que se utiliza para almacenar los datos es: entidad, indicador, segmento, tipo, periodo y valor. Por lo tanto, para hacer uso de la columna calculada se tuvo que utilizar una tabla auxiliar con la estructura: entidad, segmento, tipo, periodo, valor indicador1, valor indicador2, ..., valor indicadorN; donde los nombres de la columnas valor indicador1, ..., valor indicadorN son: indicador1, ..., indicadorN. Después que SQL Server genera los valores, se tiene que regresar a la estructura normal de trabajo y se elimina la tabla auxiliar, tal como se muestra en la figura 4.10. La generación de código se hizo compleja, debido a la parametrización del código para que funcionara para cualquier fórmula.

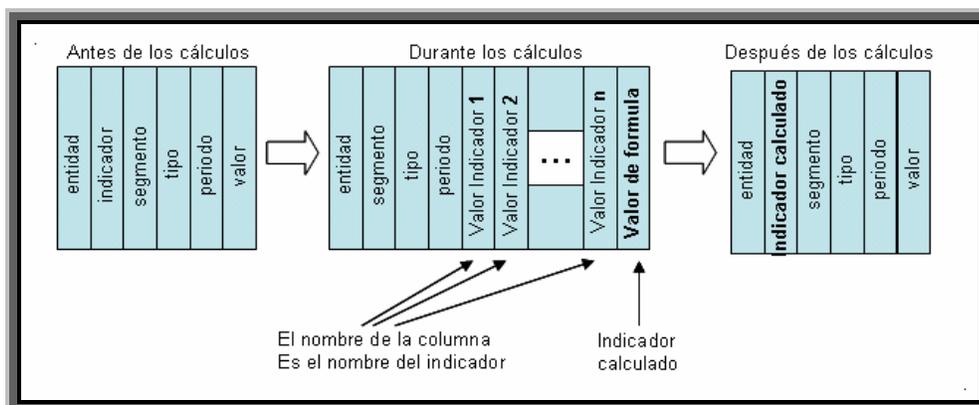


Figura 4.10. Manipulación de la información para crear tablas calculadas.

El pseudo código 4.6 presenta la forma como se generan los valores de los indicadores calculados.

```
sp_Calculos:
Entrada: Carga; periodicidad; año; Tipo de proceso.
Salida: 0 cuando termina sin problemas y <> 0 cuando se presenta un error.
        Resultado de los cálculos almacenados en la tabla de trabajo.

INICIO
1) Crear tablas temporales auxiliares.

2) Ejecutar el procedimiento almacenado "sp_getRootForCalc", para obtener los datos que
puedan faltar para realizar los cálculos.

3) Construir la tabla "#tabla_datos" donde se almacenarán los indicadores que participan en
fórmulas y que cumplen alguna de las condiciones: a) mismo tipo y periodo, b) periodo
distinto, c) tipo distinto y d) calculado.

3) Obtener las variables de usuario, correo electrónico y formato de fecha.

4) Para cada uno de los cálculos que se tienen que realizar.

4.1) Crear una tabla con los campos llave (año, entidad, tipo, periodo, segmento, fecha),
que posteriormente se le construirá la columna calculada.

4.2) Almacenar los datos a procesar en la tabla (4.1)

4.2) Almacenar en "#tabla_datos" los valores descritos en (3).

4.3) Para cada registro en #tabla_datos se hace.

4.3.1) Crear una tabla que contiene los valores llave (año, entidad, tipo, periodo,
segmento, fecha) y los valores del indicador del registro que se procesa.

4.3.2) Armar una sentencia (join), de la tabla (4.1) con cada una de las tablas (4.3.1), de
tal manera que cada indicador forme una columna y la fórmula sea una columna calculada.

4.4) Ejecutar la sentencia (4.3.2) para crear las columnas de los valores de los
indicadores y la columna calculada.

4.5) Los valores calculados generados en la construcción de la columna calculada son
pasados a la tabla auxiliar "#indicadores_calculados". De esta manera, tenemos los valores
de los indicadores calculados, con la estructura normal (vertical).

4.6)

5) Si el tipo de proceso <> 1

5.1) Eliminar de "#temporal_proceso" los indicadores cargados que son calculados y que han
sido generados por el proceso.

5.2) Insertar en "#temporal_proceso" los valores calculados, generados por el proceso.

FIN
```

Pseudo código 4.6. Procedimiento almacenado que genera los valores calculados.

4.2.3.4. sp_get_RootForCalc.

Obtiene los valores de los indicadores necesarios para realizar los cálculos y que no se encuentra en los valores cargados. El pseudo código 4.7 muestra como el procedimiento almacenado determina cuando obtener los valores necesarios que no han sido cargados y que participan en una fórmula.

El subproceso tuvo que solucionar varios problemas, entre los que se encuentran los siguientes:

- Determinar si un indicador es calculado o raíz, debido a que una carga puede contener indicadores calculados. Por ejemplo, se cargan los indicadores A, B y C, pero se encuentra registrada la fórmula $C = A + B$.
- Ordenar los niveles de cálculos, debido a que una carga puede contener valores de indicadores que participan en distintos niveles de anidamiento. Por ejemplo, se cargan los indicadores A, B, C, D, pero se encuentran registradas las fórmulas $C = A + B$, $F = C + D$. Es posible pensar que en el ejemplo anterior, se pueden realizar las dos operaciones en el mismo nivel, debido a que contamos con todos los datos raíz, pero puede resultar que el valor calculado C no es el mismo valor que el cargado, por tanto, en un segundo nivel de recursividad hay que calcular nuevamente el valor F, debido a que en el segundo nivel contamos con un valor actual de C.
- No todos los indicadores pueden tener las mismas entidades. Por ejemplo, los indicadores A y B tienen las entidades CA, CD, CH, CT; y el indicador $C = A + B$ tiene las entidades CA y CH, por lo tanto los valores de los indicador A y B no deben tomarse en cuenta para las entidades CD y CT.
- Los valores que se toman en cuenta deben ser marginales a menos que la fórmula indique lo contrario. Por procedimiento, es necesario realizar los cálculos sobre los valores marginales y posteriormente calcular el valor agregado sobre los resultados de los indicadores calculados.

```
sp_getRootForCalc:
Entrada: Carga, Periodicidad; Año
Salida: 0 cuando termina sin problemas y <> 0 cuando se presenta un error.
        Resultado de indicadores por calcular e indicadores que participan en los
cálculos.

INICIO
1) Crear tablas temporales auxiliares
2) Inicializar las variables año anterior, nombre de la tabla de datos.
3) Obtener todos los indicadores calculados, donde tiene al menos un elemento raíz.
4) Si es un recálculo, entonces
4.1) Eliminar todos los indicadores calculados que no están registrados como recálculo.
5) Verificar cuales son las entidades válidas de los indicadores calculados.
6) Eliminar todos los registros cuyas entidades no son válidas para los indicadores
calculados.
7) Ordenar la lista de indicadores calculados, debido a que los indicadores cargados pueden
intervenir en fórmulas que se encuentran en diferentes niveles de cálculo y unas pueden
intervenir en el resultado de otro.
8) Determinar cual es el nivel mínimo de cada indicador.
9) Eliminar los indicadores repetidos, debido a que un mismo indicador se presenta en
diferentes niveles de la recursividad.
10) Obtener los indicadores raíz que intervienen en todas las fórmulas que se van a
procesar.
11) Obtener por separado los indicadores raíz que cumplen que son de periodo fijo con
respecto a la carga.
12) Si existen indicadores raíz, entonces.
12.1) Obtener el periodo mínimo y máximo de toda la lista de valores.
12.2) Almacenar en la tabla "#temporal_calculo_aux", los valores de los indicadores raíz
que son del mismo año.
12.3) Si la periodicidad es mensual, entonces.
12.3.1) Obtener los valores de los indicadores que intervienen en fórmula y que tienen tipo
distinto a M, A, P y Q.
12.4) Si existe tabla de datos del año anterior al que se está cargando, entonces.
12.4.1) Obtener los valores de los indicadores raíz que su valor es de cualquier periodo
del año anterior.
12.4.2) Obtener los valores de los indicadores raíz que su valor es de un periodo fijo del
año anterior.
12.5) Todos los valores obtenidos son filtrados para obtener valores distintos y se
almacenan en "#temporal_calculo".
12.6) Almacenar los valores cargados en "#temporal_calculo", eliminando los que se hayan
obtenido de la base de datos.
12.7) Eliminar los indicadores donde su periodo sea mayor al periodo máximo.

FIN
```

Pseudo código 4.7. Procedimiento que obtiene los datos necesarios para la fórmula.

4.3 Implantación del sistema

Es el proceso de instalar equipos de cómputo, programas o componentes de información, y su configuración. Las tareas más importantes que se realizaron en la fase son:

1. Instalación y configuración de programas:
 - 1.1. Instalación y configuración de programas en los servidores:
 - 1.1.1. **Instalación del sistema operativo:** La instalación la realizó el personal de soporte técnico de Telmex. No necesitaba alguna configuración especial.
 - 1.1.2. **Configuración de red:** La instalación la realizó el personal de soporte técnico de Telmex. No necesitaba alguna configuración especial.
 - 1.1.3. **Instalación del manejador de la base de datos (SQL Server 2000):** Se hizo una instalación Standard, configurando, únicamente, el lenguaje para almacenar cadena de datos unicode con reconocimiento de caracteres latinoamericanos.
 - 1.2. **Construcción de la base de datos en ambos servidores:** La construcción de la base de datos se fue haciendo conforme se desarrollaba el sistema de cómputo. De igual forma, el llenado de las bases de datos, se hizo conforme se probaban cada uno de los módulos.
 - 1.3. **Creación de trabajos programados:** Los trabajos programados juegan un papel importante en el funcionamiento del sistema, ya que son los que disparan automáticamente la ejecución de una rutina o proceso. Uno de los trabajos programados se configuró en el sistema operativo, éste se activa cada minuto y sirve para la detección de archivos. En SQL Server se crearon el resto de los trabajos programados que se utilizaron. Estos dividen el proceso completo en varios pequeños (validación, carga, cálculos, generación de vistas y transferencia), que permiten un mejor control sobre el proceso completo. La construcción y configuración de estos trabajos programados es muy sencillo.

La figura 4.11 muestra el área de trabajo del administrador de la base de datos donde se puede crear los trabajos programados de SQL Server.



Figura 4.11. Panel de administración de trabajos programados.

La figura 4.12 muestra la ventana de generación y configuración de trabajos programados. Se tiene que definir que instrucciones se tienen que ejecutar, los lapsos de tiempo para activarse y que acciones realizar en caso de error.

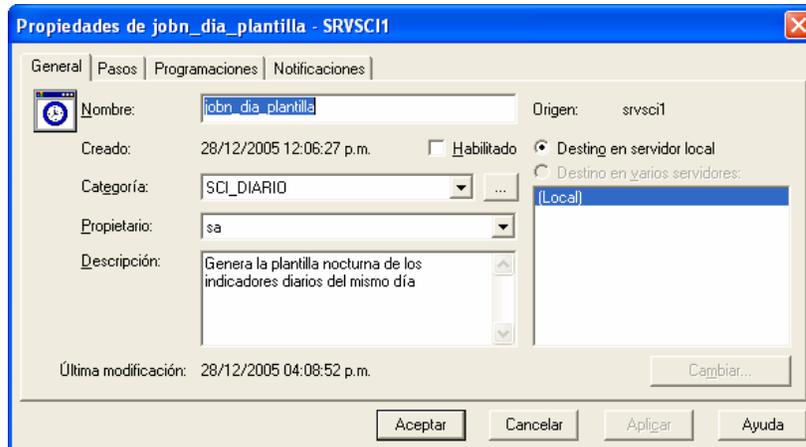


Figura 4.12. Ventana de configuración de trabajos programados.

- 1.4. **Configuración de correo electrónico:** Se aprovecharon los servicios de correo de Telmex y el servicio de correo de SQL Server para que el sistema notifique del proceso por medio de correo electrónico.

Con este servicio, el sistema envía notificaciones a los usuarios cuando la información está disponible para su revisión o cuando existe un problema en el proceso de una carga.

Para la configuración del servicio de correo electrónico, por medio de SQL Server, se tuvo que hacer lo siguiente:

- 1.4.1. Solicitar la alta de un usuario de correo a los administradores de los servidores de correos de Telmex. Para nuestro caso fue el usuario fue **sinfocorp**.
- 1.4.2. En el Outlook del servidor SRVSCI1 se configuró el perfil sinfocorp.
- 1.4.3. Ejecutar el administrador de SQL Server. Configurar el SQL Mail que se encuentra en los servicios de compatibilidad de SQL Server. La figura 4.13 muestra la ventana del administrador de SQL Server donde se configura el servicio de correo electrónico para SQL Server.

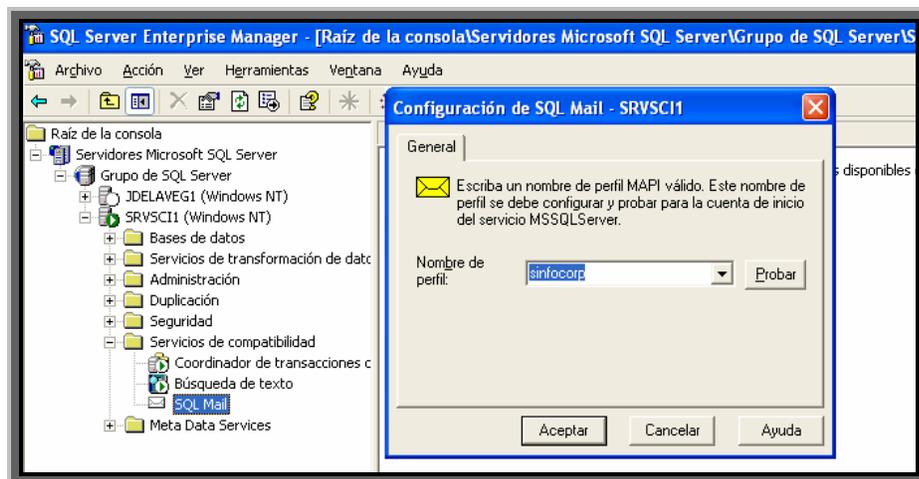


Figura 4.13. Panel de configuración de correo electrónico en SQL Server.

- 1.5. Configuración en las PC donde utilizan el SCI.
2. **Capacitación de los operadores sobre el sistema:** La capacitación del sistema se realizó a lo largo del proyecto. Cada vez que funcionaba un módulo, se les enseñaba cual era la funcionalidad del módulo y la manera de usarlo.

3. Pruebas de funcionamiento:

Operación del proceso y corrección de defectos: Después de que un módulo o proceso era terminado, se realizaban algunas pruebas de funcionamiento, con información real. La información resultante era analizada por el personal operativo de la subdirección y retroalimentaba a los programadores, en caso de existir algún defecto en los resultados. Al mismo tiempo se evaluaba el rendimiento de los procesos, para determinar si eran aceptados o si debían modificarse los algoritmos para acelerar los procesos.

Capítulo 5

Resultados

5.1 Resultados obtenidos

El objetivo principal de la construcción del SCI fue proporcionar a la Subdirección de Sistemas de Información Corporativa una herramienta de apoyo para el procesamiento de los indicadores de Telmex, donde la operación fuera sencilla; los tiempos de proceso, cortos; y la información, confiable.

La tabla 5.1 muestra una lista comparativa de la operación antes y después de la implantación del Sistema de Control de Indicadores. Los datos que se presentan son de la operación real.

Al mismo tiempo que se resolvían los problemas de la operación del procesamiento de indicadores, el desarrollo contribuyó con varios conceptos o componentes que fueron pensados y diseñados para el funcionamiento del SCI, que pueden ser utilizados en otros sistemas. Entre los conceptos y componentes desarrollados se encuentran:

- La configuración del correo electrónico dentro del manejador de BD para informar a múltiples usuarios el final del proceso.
- La separación y utilización de monitores, para tener mejor control del proceso total.
- La construcción de tablas con columnas calculadas para que sirvan de analizadores sintácticos de fórmulas.

- Construcción de sentencias SQL dinámicas que facilitan el crecimiento de la BD, sin tener que modificar las sentencias por cada cambio en la BD.

OPERACIÓN ORIGINAL	NUEVA OPERACIÓN
Entre 8 y 24 horas para procesar alrededor de 80 mil datos.	2 horas para procesar alrededor de 600 mil datos.
No se actualiza información de meses distintos al que se procesa.	Se actualiza información acumulada de meses posteriores y de cálculos, donde intervienen indicadores de periodos distintos.
Información disponible, únicamente, al director general y directores corporativos	Información disponible desde jefes de centrales u oficinas comerciales, con sus respectivos privilegios de consulta.
Una vez entregada la carpeta de reportes, no había actualización de los datos.	Los datos se actualizan en el momento que sea necesario
Hasta que todas las entidades entregaran la información, era posible generar los indicadores y la carpeta de reportes	Los reportes de indicadores se generan con la información que se va recibiendo y se actualiza al momento que van llegando nuevos datos.
Existían alrededor de 100 fórmulas fijas.	Existen más de 1,400 fórmulas que pueden ser modificadas o generar nuevas en cualquier momento por el personal operativo.
Se genera información mensual impresa y estática.	Se genera información mensual, semanal y diaria, dinámica y de forma automática.

Tabla 5.1. Lista comparativa del procesamiento de indicadores.

5.2 Discusión sobre los resultados obtenidos.

Algunos aspectos importantes que analizaremos de los resultados obtenidos son los siguientes:

- El tiempo de procesamiento se reduce de 30 horas a 1 hora, para 300 mil datos. Por lo que el ahorro en tiempo de trabajo es considerable.
- La actualización de la información, además de ser inmediata, se realiza para los resultados de los periodos futuros en donde interviene, lo que antes no se hacía.

- La distribución de los resultados es realizada a todo el personal que consulta la información en la Intranet de Telmex, minutos después de que se procesa.
- La generación de información semanal y diaria es de suma importancia, porque los directores pueden tener el comportamiento diario de la empresa.
- La construcción de un módulo de fórmulas y generación de cálculos de manera dinámica es uno de los procesos que tuvo una buena aceptación del Subdirector y sus Gerentes, ya que le evitó a su personal el cálculo extenuante y, en ocasiones, complejo, de indicadores.

El sistema ha aportado un gran beneficio a la operación de la Subdirección de Sistemas de Información Corporativa. Ahora, el tiempo dedicado a realizar tareas repetitivas, es utilizado en nuevas funciones que ha adquirido la Subdirección; esto, junto con la confiabilidad de la información, ha ocasionado que la Subdirección tenga mayor importancia dentro de la empresa.

Capítulo 6

Conclusiones y desarrollos futuros

El uso de sistemas para el procesamiento de información es más útil cada día; sobre todo cuando realizan tareas con grandes volúmenes de información, con un mínimo de errores.

El sistema de Control de Indicadores eliminó varios de los problemas que existían en la Subdirección. Además, mejoró, considerablemente, los tiempos de procesamiento. Esto ha generado que la información sea oportuna y confiable, aspectos relevantes para la toma de decisiones.

La operación de la Subdirección pasó a tener mayor importancia dentro de Telmex, porque ahora distribuye la información de indicadores a todos los directores y gerentes de la empresa que se encuentran en todo el país y Latinoamérica; y no sólo a los Directores corporativos, como lo hacía anteriormente.

La retroalimentación recibida por parte del personal operativo fue importante para tener éxito en la construcción del sistema de cómputo, porque ellos describieron todas las variables que pueden existir en el procesamiento de indicadores.

El desarrollo de componentes de software ayuda a que el código sea reutilizable. Algunos de los procesos, entre ellos las agregaciones, acumulaciones y cálculos, pueden ser reutilizados para generar procesos de auditoría de datos o reprocesos, donde sólo tienen que realizar el llamado de los procedimientos almacenados.

La velocidad para procesar los indicadores es aceptable para el usuario. Sin embargo, como se puede observar en la tabla 1.2, del capítulo 3, el volumen de información sigue creciendo con el paso del tiempo. Durante el desarrollo del sistema, se pudo observar que el tiempo de procesamiento de información aumenta en una proporción mayor que la del volumen de información a procesar; por ejemplo, una carga que tiene 5,000 valores tarda 2 minutos; pensaríamos que una carga de 50,000 valores tardaría 20 minutos; aproximadamente; sin embargo, tarda unos 35 minutos. Este comportamiento puede traer problemas con cargas mucho más grandes, porque el proceso comenzará a ser más lento. Por lo tanto, es conveniente modificar el proceso para que el aumento en el tiempo de procesamiento sea proporcional a la información que se carga.

Actualmente, no todos los indicadores que se procesan en el SCI se presentan en las páginas Web del SI. Sin embargo, algunos de estos indicadores son importantes para ciertos directores. Para ofrecerles esta información, la SSIC les procesa, semiautomáticamente por medio de Excel y Acrobat, los reportes que necesitan. Un nuevo desarrollo es automatizar completamente este proceso para que el sistema realice toda la operación, sin que el personal operativo tenga que meter las manos en el proceso.

Bibliografía

[1] Portal Telmex

http://www.telmex.com/explorer/esto/pt_seccion.jsp?p=esto_infcorp_intro_historia.html

http://www.telmex.com/explorer/esto/pt_seccion.jsp?p=esto_infcorp_19902002.html

[2] García J.; El proceso de la toma de decisiones y de resolución de problemas

<http://www.cop.es/colegiados/M-00451/tomadeciones.htm>

[3] Hanel J.; Análisis situacional; Universidad Autónoma Metropolitana; 1996.

[4] Fernández H.; Enfoque objetivo en la toma de decisiones

http://www.buscarportal.com/articulos/iso_9001_enfoque_objetivo_hacia_toma_decisiones.html

[5] Latorre R., Sanchidián R.; Indicadores

<http://www.usc.es/~calidade/ACSUG/www/PONENCIAS/ponencia6.ppt>

[6] Yourdon E.; Análisis estructurado moderno; Prentice-Hall; 1993.

[7] Muller P.; Modelado de objetos con UML; Gestión 2000; 1997.

[8] Marqués M.; El modelo entidad-relación; 2001.

<http://www3.uji.es/~mmarques/f47/apun/node83.html>

[9] Grupo Bybele; El modelo relacional; 2004.

http://kybele.escet.urjc.es/documentos/DBD/Tema8-MR_Dinamica.pdf

Apéndice A

Diagrama de actividad

Un diagrama de actividad se utiliza para seguir a través de un flujo de trabajo o procedimientos. Puede componerse de decisiones o procesos desarrollados en cada paso. El modelo revela los elementos esenciales del dominio del problema de una manera familiar para los usuarios. Modelar funciones complejas puede prevenir problemas más tarde cuando se intente escribir el código. El modelo hace más de lo que se imagina uno, ofrece suposiciones visibles y una fácil y correcta revisión. Muller P. [7] describe con detalle la utilidad de los diagramas de actividades. Además, presenta ejemplos que ayudan a entender como tienen que elaborarse.

Una actividad es un proceso o decisión. Un diagrama de actividades es una serie de actividades ligadas por disparadores y flechas que conectan cada actividad. La figura A.1 muestra la representación gráfica de estos elementos del modelo de actividades.



Figura A.1. Icono de actividad y disparador.

Una condición de guardia (corchetes) puede ser puestos en un disparador para forzar la navegación por el esquema de la actividad. La condición puesta dentro de los corchetes debe

ser verdadera antes de seguir el disparador asociado a la próxima actividad. La figura A.2 muestra la representación gráfica de la condición de guardia.

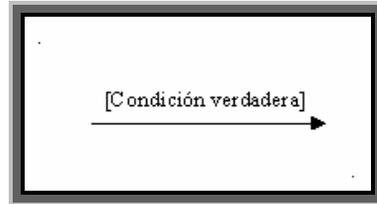


Figura A.2 Icono de condición de guardia.

El diamante es un icono de decisión, tal como lo muestra la figura A.3. Una actividad puede también ser una decisión. En este caso, una flecha sale del diamante; para cada posible valor de la condición establecida.

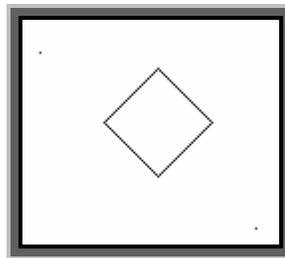


Figura A.3. Icono de diamante de decisión.

Una condición de guardia es puesta en cada flecha que sale del icono, controlando la dirección para seguir basado en el resultado de la decisión. También, cualquier método de la decisión puede tener cualquier número de valores de salida. Las decisiones no son limitadas a pruebas booleanas.

Típicamente, una decisión se utiliza para la lógica anidada de la decisión, una prueba dentro de una rama de otra decisión. La prueba para utilizar el diamante de la decisión en vez de una actividad cuando ningún procesamiento se incluye en el paso.

También se proveen iconos de comienzo y fin de un diagrama de actividad. Un punto sólido indica el comienzo de un flujo de actividades. Un ojo de toro indica el punto final del flujo de actividades. Puede haber más de un punto en el diagrama de actividad. La figura A.4 muestra los iconos del inicio y término del flujo de actividades.

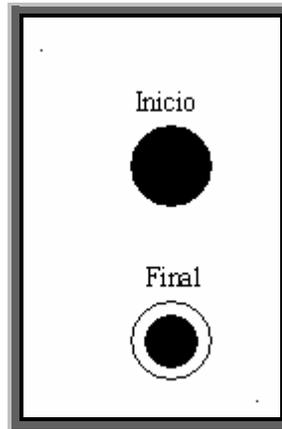


Figura A.4. Icono de comienzo y final del flujo de actividades.

La barra de sincronización se utiliza en ambos sentidos y sirve para unir varios flujos en uno solo, cuando tiene que existir una concurrencia. De igual modo una función se puede desprender en varios flujos. La figura A.5 muestra la representación gráfica de los dos tipos de sincronización.

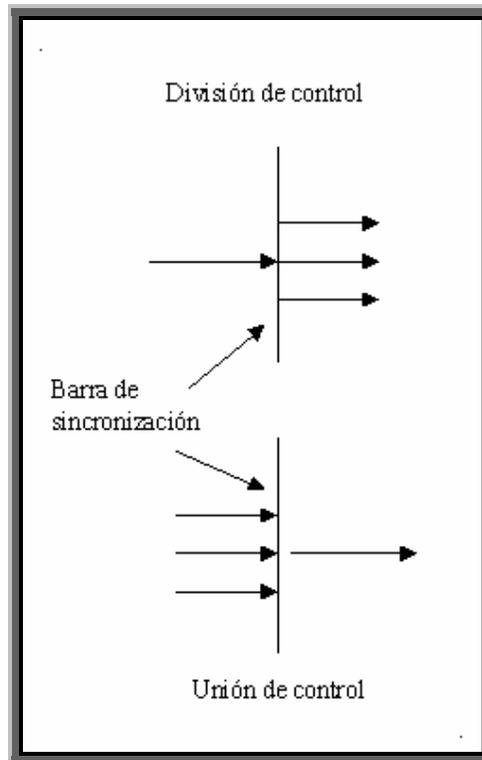


Figura A.5. Icono de sincronización de flujos.

Apéndice B

Diseño de base de datos relacionales

Una **base de datos** es un conjunto de datos, que pertenecen al mismo contexto, almacenados sistemáticamente para su uso posterior. En este sentido, una biblioteca puede considerarse una base de datos compuesta, en su mayoría, por documentos y textos impresos en papel e indexados para su consulta.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente consultar los datos de forma rápida y estructurada.

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos. Existen varios modelos de bases de datos. Los modelos de bases de datos no son cosas físicas, son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*; por lo general se refieren a algoritmos y conceptos matemáticos. El modelo utilizado en el SCI fue el Relacional.

Dentro del modelo relacional, existe el modelo de entidad-relación, que es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Marqués M. [8] y Grupo Bybele [9] hablan del modelo relacional de Peter Chen.

El modelo relacional fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

B.1 Entidad

Se trata de cualquier objeto u elemento (real o abstracto) en el cual se puede almacenar información en la base de datos. Hay dos tipos de entidades: fuertes y débiles. Una *entidad débil* es una entidad cuya existencia depende de la existencia de otra entidad. Una *entidad fuerte* es una entidad que no depende de otra para existir. La figura C.1 muestra la representación gráfica de una entidad.

B.2 Relación

Es una correspondencia o asociación entre dos o más entidades. Las entidades que están involucradas en una determinada relación se denominan *entidades participantes*. El número de participantes en una relación es lo que se denomina *grado* de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación *binaria*; si son tres las entidades participantes, la relación es *ternaria*; etc.

Una *relación recursiva* es una relación donde la misma entidad participa más de una vez en la relación, con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La *cardinalidad* con la que una entidad participa en una relación especifica el número mínimo y el máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es *obligatoria (total)* si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es *opcional (parcial)*. Las reglas que definen la cardinalidad de las relaciones son las *reglas de negocio*.

B.3 Atributo

Es una característica de interés o un hecho sobre una entidad o una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos.

Cada atributo tiene un conjunto de valores asociados denominado *dominio*. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un *atributo simple* es un atributo que tiene un sólo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un *atributo compuesto* es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto, cuando tienen afinidad en cuanto a su significado o su uso.

Por último, los atributos pueden ser derivados. Un *atributo derivado* es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

B.4 Identificador

Un identificador de una entidad es un atributo o conjunto de atributos que determina, de modo único, cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

En la actualidad, existen varios programas de cómputo que ayudan a construir los modelos Entidad-Relación de una manera más sencilla. Cada uno tiene una representación gráfica de los elementos del modelo Entidad-Relación. La figura B.1 muestra la representación gráfica que hace el programa ERWin de cada uno de los elementos del modelo Entidad-Relación.

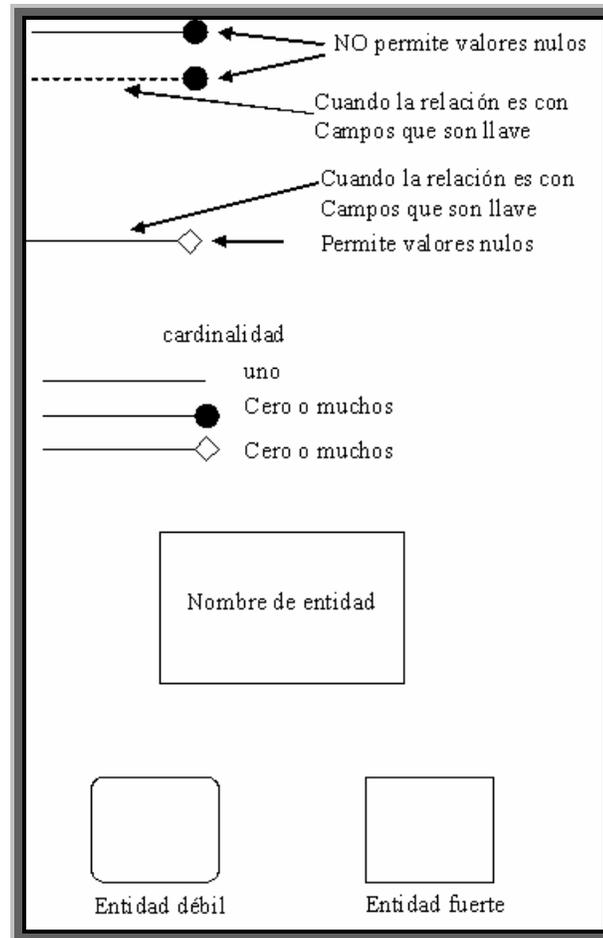


Figura B.1. Elementos del modelo entidad relación de ERwin.

Apéndice C

Código de los procesos

C.1 Principal de agregación y acumulación.

```
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Principal_Agrega_Acumula' AND type = 'P')
    DROP PROCEDURE sp_Principal_Agrega_Acumula
GO
--
*****
-- Nombre:          sp_Principal_Agrega_Acumula
-- Descripción:     A partir de un archivo de datos, concentra el llamado de los procedimientos almacenados
que
--                  realizan las operaciones de:
--                  * Agregados.
--                  * Acumulados.
--
-- Parámetros:
--                  @pnCarga:   Número de carga a procesar
--                  @pnProcesa: 1) Indica que se trata de una carga normal
--                              0) Indica que se trata de una carga directa
-- Return:          -1:Error, 0:Éxito
--
*****
CREATE PROCEDURE sp_Principal_Agrega_Acumula
    @pnCarga    int,
    @pnProcesa  int
AS
BEGIN
    ----- INICIA SECCION DE DECLARACIONES -----
    DECLARE @sSql          varchar(4000),
            @nError        int, @sError varchar(150),
            @sPeriodicidad char(1),
            @sAño          char(4),
            @nPeriodoMin   smallint,
            @nAgrega       smallint, @nAcumula smallint, @nCalcula smallint,
            @result        int,
            @sProducto     varchar(15), @sArchivo varchar(100),
            @sFecha        varchar(10),
            @sTabDatos     varchar(30),
            @nTipoProceso  int
    ----- TERMINA SECCION DE DECLARACIONES -----

    -- Establece el formato de fecha
    SET DATEFORMAT mdy

    --
    =====
    -- SECCION DE CONSTRUCCION DE TABLAS
    -----

    CREATE TABLE #cat_responsabilidad(
        indicador    varchar(10),
```

Apéndice C. Código de los procesos

```
        regleta          varchar(10)
    )
    CREATE TABLE #temporal_proceso(
        carga            int,
        entidad          varchar(10),
        indicador        varchar(10),
        segmento         varchar(2),
        tipo             char(1),
        valor            float,
        fecha            varchar(10),
        periodo         smallint,
        tipo_calc       tinyint -- 0 = cargado, 1 = auxiliar, 10 = agregado, 20 = acumulado, 30 =
calculado
    )
    CREATE INDEX ix_temp_proceso1 On #temporal_proceso (entidad, indicador, segmento, tipo, periodo, valor)
    CREATE INDEX ix_temp_proceso2 On #temporal_proceso (entidad, indicador, segmento, tipo, periodo, fecha)
    CREATE INDEX ix_temp_proceso3 On #temporal_proceso (entidad, indicador, segmento, tipo, periodo)
    CREATE INDEX ix_temp_proceso4 On #temporal_proceso (entidad, indicador, segmento, periodo)
    CREATE INDEX ix_temp_proceso5 On #temporal_proceso (indicador, tipo)
    CREATE INDEX ix_temp_proceso6 On #temporal_proceso (entidad)
    CREATE INDEX ix_temp_proceso7 On #temporal_proceso (tipo)

    -- Tabla que almacenará los indicadores que provienen de datos_sem, ya que estos no se tienen que
cargar nuevamente.
    CREATE TABLE #temporal_datos(
        entidad          varchar(10),
        indicador        varchar(10),
        segmento         varchar(2),
        tipo             char(1),
        valor            float,
        fecha            varchar(10),
        periodo         smallint
    )
    CREATE INDEX ix_temporal_datos1 On #temporal_datos (entidad, indicador, segmento, tipo, periodo, fecha)
    CREATE INDEX ix_temporal_datos2 On #temporal_datos (entidad, indicador, segmento, tipo, periodo)
    CREATE INDEX ix_temporal_datos3 On #temporal_datos (entidad, indicador, segmento, periodo)
    CREATE INDEX ix_temporal_datos4 On #temporal_datos (indicador, tipo)
    CREATE INDEX ix_temporal_datos5 On #temporal_datos (entidad)
    CREATE INDEX ix_temporal_datos6 On #temporal_datos (tipo)

    CREATE TABLE #tmp_insertados(
        carga            int,
        entidad          char(10),
        indicador        char(10),
        segmento         char(2),
        tipo             char(1),
        valor            float,
        fecha            char(10),
        año             smallint,
        periodo         smallint,
        nivel            char(1)
    )

    CREATE TABLE #tmp_modificados(
        carga            int,
        entidad          char(10),
        indicador        char(10),
        segmento         char(2),
        tipo             char(1),
        valor            float,
        fecha            char(10),
        año             smallint,
        periodo         smallint,
        nivel            char(1),
        valor_anterior  float
    )

    --
=====
    -- SECCION CARGA INICIALIZACION DE VALORES Y CARGA DE DATOS
    -----
    IF @pnProcesa = 1
        SET @nTipoProceso = 0
    ELSE IF @pnProcesa = 10
        SET @nTipoProceso = 10

insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'Inserta en
```

```

#temporal_proceso')
SELECT @nError = -1, @sError = 'Error al insertar datos en #temporal_proceso'
-- Carga los datos en la tabla de trabajo (tipo_calc = 0 significa que no son calculados)
INSERT #temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo, tipo_calc)
SELECT carga, entidad, indicador, segmento, tipo, valor, fecha, periodo, 0
FROM temporal_proceso
WHERE carga = @pnCarga
IF (@@error <> 0)
    GOTO err_calc

-- Inicialización de valores
SELECT @sPeriodicidad = periodicidad FROM bitacora_carga WHERE carga = @pnCarga
SELECT @sAño = convert(char(4), año), @nPeriodoMin = min(periodo)
FROM temporal_proceso WHERE carga = @pnCarga GROUP BY año
SELECT @sTabDatos = 'datos_' + CASE @sPeriodicidad WHEN 'M' THEN 'mes_' WHEN 'S' THEN 'sem_' END +
@sAño

SELECT @nError = -2, @sError = 'Error al insertar datos en #cat_responsabilidad'
-- Obtiene los datos necesarios de cat_responsabilidad para no hacer liga con todos los datos de la
tabla
INSERT #cat_responsabilidad
SELECT DISTINCT indicador, regleta FROM cat_responsabilidad WHERE tipo = 'M' AND periodicidad =
@sPeriodicidad
IF (@@error <> 0)
    GOTO err_calc

--
=====
-- SECCION DE AGREGACIONES
-----
-- Actualiza estado en la bitacora
UPDATE bitacora_carga SET status_proceso = 'CG' WHERE carga = @pnCarga
insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'Entra a
AGREGACIONES')
-- Ejecuta el proceso de AGREGACIONES
EXEC @nAgrega = sp_Agregacion_8 @sPeriodicidad, @sAño, @nTipoProceso
IF (@@error <> 0 OR @nAgrega < 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'Sale de
AGREGACIONES')

-- Actualiza estado en la bitacora
UPDATE bitacora_carga SET status_proceso = 'CU' WHERE carga = @pnCarga
SELECT @nError = -3, @sError = 'Error en la ejecución de sp_Acumulacion'
-- Ejecuta el proceso de ACUMULACIONES
insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'Entra al proceso de
acumulación')
EXEC @nAcumula = sp_Acumulacion_4 @sPeriodicidad, @sAño, @nTipoProceso
IF (@@error <> 0 OR @nAcumula < 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'Elimina los
indicadores agregados y acumulados que ya existen en datos_mmmm_yyyy')
SELECT @nError = -4, @sError = 'Error al eliminar datos de #temporal_proceso'
--
=====
-- Elimina los indicadores agregados y acumulados que ya existen en datos_mmmm_yyyy
SET @sSql = '
DELETE #temporal_proceso
FROM #temporal_proceso t WITH (NOLOCK), ' + @sTabDatos + ' d WITH (NOLOCK)
WHERE t.entidad = d.entidad
AND t.indicador = d.indicador
AND t.tipo = d.tipo
AND t.segmento = d.segmento
AND t.periodo = d.periodo
AND t.valor = d.valor '
EXEC(@sSql)
IF (@@error <> 0)
    GOTO err_calc

-----
-- Se eliminan de #insertados los indicadores agregados y acumulados que existen como cargados para
-- sustituirlos por los calculados.
insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'Elimina de
temporal_proceso los agregados y acumulados que han sido cargados')
DELETE temporal_proceso

```

```
FROM temporal_proceso tp WITH (NOLOCK), #temporal_proceso tg WITH (NOLOCK)
WHERE tp.entidad = tg.entidad
      AND tp.indicador = tg.indicador
      AND tp.segmento = tg.segmento
      AND tp.tipo = tg.tipo
      AND tp.periodo = tg.periodo
      AND tp.año = convert(smallint, @sAño)
      AND tp.carga = tg.carga
      AND tg.tipo_calc IN (10, 20)

-- Se insertan los indicadores agregados y acumulados en #insertados y #modificados
insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'inserta de
#temporal_proceso a temporal_proceso')
SELECT @nError = -7, @sError = 'Error al insertar datos en #insertados'
INSERT temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, año, periodo, nivel)
SELECT DISTINCT t.carga, t.entidad, t.indicador, t.segmento, t.tipo, t.valor, t.fecha,
convert(smallint, @sAño),
      t.periodo, e.nivel
FROM #temporal_proceso t WITH (NOLOCK), cat_entidades e WITH (NOLOCK)
WHERE t.entidad = e.entidad
      AND t.tipo_calc IN (10, 20)
IF (@@error <> 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), 'TERMINA
sp_Principal_Agrega_Acumula')

RETURN 0

err_cursor:
CLOSE prod_Cursor
DEALLOCATE prod_Cursor
insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), @sError)
RETURN @nError

err_calc:
insert bitacora_proceso_calc(carga, hora, descripcion) values (@pnCarga, getdate(), @sError)
RETURN @nError
END -- CREATE PROCEDURE sp_Principal_Agrega_Acumula_2
```

C.2 Proceso de agregación.

```

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Agregacion' AND type = 'P')
    DROP PROCEDURE sp_Agregacion
GO
--
*****
-- Nombre:          sp_Agregacion
-- Descripción:     Obtiene el valor de los indicadores de las entidades agregadas a partir de los datos
cargados de un
--                  archivo a una tabla temporal.
-- Parámetros:
-- Return:          -1:Error, 0:Éxito
--
*****
CREATE PROCEDURE sp_Agregacion
    @psPeriodicidad char(1),
    @psAño           char(4),
    @pnuComparar    smallint -- 0: Proceso normal de carga
                             -- 10: Para auditoría de carga directa. NO se hace sustitución de la
información cargada
                             --                y se envía correo en caso de diferencias.
                             -- 20: Para la auditoría y reagregaciones. No toma en cuenta los correos
AS
BEGIN
    ----- INICIA SECCION DE DECLARACIONES -----
    -----
    DECLARE    @sTabDatos      varchar(30),
               @sFecha        varchar(40),
               @sSql          varchar(8000),
               @nIdEntArbol   tinyint,
               @sNivelP       char(1),
               @sNivelH       char(1),
               @nIndentacion  smallint,
               @nIndentacion_2 smallint,
               @sNiveles      varchar(50),
               @sMensaje      varchar(500),
               @sCarga        varchar(10),
               @sUsuario      varchar(15),
               @sSubject      varchar(100),
               @sMail         varchar(50),
               @nCarga        int,
               @sProducto     varchar(15),
               @sEntidadP     varchar(10),
               @nuProcesados  int
    ----- TERMINA SECCION DE DECLARACIONES -----
    -----
    -- Establece el formato de la fecha
    SET DATEFORMAT mdy

    CREATE TABLE #cat_tipos(
        tipo_base char(1),
        tipo      char(1)
    )
    INSERT #cat_tipos VALUES('M', 'M')
    INSERT #cat_tipos VALUES('M', 'A')
    INSERT #cat_tipos VALUES('P', 'P')
    INSERT #cat_tipos VALUES('P', 'Q')

    SELECT @sUsuario = 'HPALAFOX', @sMail = 'hpalafox@telmex.com'
    SELECT TOP 1 @nCarga = carga FROM #temporal_proceso
    SET @sCarga = convert(varchar(10), @nCarga)

    SELECT TOP 1 @sUsuario = usuario, @sProducto = producto
    FROM #temporal_proceso t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #cat_tipos ct WITH
(NOLOCK)
    WHERE t.indicador = r.indicador
           AND t.tipo = ct.tipo
           AND r.tipo = ct.tipo_base
           AND r.periodicidad = @psPeriodicidad

```

Apéndice C. Código de los procesos

```
SELECT @nIdEntArbol = id_ent_arbol FROM productos.dbo.cat_productos WHERE id_producto = @sProducto
SELECT @sMail = e_mail FROM productos.dbo.cat_usuarios_producto WHERE id_responsable = @sUsuario

CREATE TABLE #temporal_agrega(
    id_ent_arbol tinyint,
    indentacion smallint,
    entidad varchar(10),
    indicador varchar(10),
    segmento varchar(2),
    tipo char(1),
    valor float,
    fecha varchar(10),
    periodo smallint,
    calculado tinyint
)
CREATE INDEX ix_temp_agrega1 On #temporal_agrega (id_ent_arbol, indentacion, entidad, indicador,
segmento, tipo, periodo)
CREATE INDEX ix_temp_agrega2 On #temporal_agrega (id_ent_arbol, indentacion, indicador, segmento, tipo,
periodo)
CREATE INDEX ix_temp_agrega3 On #temporal_agrega (id_ent_arbol, indentacion, calculado)

CREATE TABLE #temporal_agrega_aux(
    id_ent_arbol tinyint,
    indentacion smallint,
    entidad varchar(10),
    indicador varchar(10),
    segmento varchar(2),
    tipo char(1),
    valor float,
    fecha varchar(10),
    periodo smallint,
    calculado tinyint
)
CREATE INDEX ix_temp_agrega_aux1 On #temporal_agrega_aux (id_ent_arbol, indentacion, entidad,
indicador, segmento, tipo, periodo)
CREATE INDEX ix_temp_agrega_aux2 On #temporal_agrega_aux (id_ent_arbol, indentacion, indicador,
segmento, tipo, periodo)
CREATE INDEX ix_temp_agrega_aux3 On #temporal_agrega_aux (id_ent_arbol, indentacion, calculado)

CREATE TABLE #temporal_entidades_faltantes(
    carga int,
    entidad varchar(10),
    indicador varchar(10),
    segmento varchar(2),
    tipo char(1),
    valor float,
    fecha varchar(10),
    periodo smallint,
    entidad_2 varchar(2)
)

IF NOT EXISTS (SELECT name FROM sysobjects WHERE name = 'bitacora_dif_agregacion' AND type = 'U')
BEGIN
    CREATE TABLE bitacora_dif_agregacion(
        carga int,
        entidad varchar(10),
        indicador varchar(10),
        segmento varchar(2),
        tipo char(1),
        periodo smallint,
        fecha varchar(10),
        valor_carg float,
        valor_calc float,
        responsable char(1)
    )
    CREATE INDEX ix_bitacora_dif_agregacion On bitacora_dif_agregacion (carga)
END
DELETE bitacora_dif_agregacion WHERE carga = @nCarga

CREATE TABLE #cat_niveles(
    id_ent_arbol tinyint,
    orden smallint, -- No se utilizará (Se requiere para el agrupamiento del query)
    indentacion smallint, -- No se utilizará (Se requiere para el agrupamiento del query)
    nivel_p char(1),
    nivel_h char(1),
    padre varchar(10),
    hijo varchar(10)
)
```

```

)
CREATE TABLE #cat_niveles_inf(
    id_ent_arbol tinyint,
    indentacion smallint,
    nivel_h char(1),
    hijo varchar(10),
    padre varchar(10) -- Cuando padre es nulo entonces el valor del hijo es la rama
inferior
)
-- Son las entidades que tienen hijos. Esta es la tabla que debe controlar las agregaciones.
CREATE TABLE #cat_niveles_sup(
    id_ent_arbol tinyint,
    indentacion smallint,
    padre varchar(10) -- Cuando padre es nulo entonces el valor del hijo es la rama
inferior
)
-- Se llena la tabla con los valores a partir de entidades_arbol
INSERT #cat_niveles
SELECT e.id_ent_arbol, min(e.orden), len(e.indentacion), e.nivel_p, e.nivel_h, e.padre, e.hijo
FROM entidades_arbol e WITH (NOLOCK), productos.dbo.cat_productos p WITH (NOLOCK)
WHERE e.id_ent_arbol = p.id_ent_arbol
AND e.padre IS NOT NULL
and e.id_ent_arbol = @nIdEntArbol
GROUP BY e.id_ent_arbol, len(e.indentacion), e.nivel_p, e.nivel_h, e.padre, e.hijo
ORDER BY e.id_ent_arbol ASC, len(e.indentacion) DESC, e.padre, e.hijo
IF (@@error <> 0)
RETURN -1
INSERT #cat_niveles_inf
SELECT distinct c1.id_ent_arbol, c1.indentacion, c1.nivel_h, c1.hijo, c2.padre
FROM #cat_niveles c1 WITH (NOLOCK), #cat_niveles c2 WITH (NOLOCK)
WHERE c1.id_ent_arbol *= c2.id_ent_arbol
AND c1.hijo *= c2.padre
INSERT #cat_niveles_sup
SELECT distinct c2.id_ent_arbol, c2.indentacion, c2.padre
FROM #cat_niveles c1 WITH (NOLOCK), #cat_niveles c2 WITH (NOLOCK)
WHERE c2.id_ent_arbol *= c1.id_ent_arbol
AND c2.padre *= c1.hijo
DELETE #cat_niveles_inf WHERE padre IS NOT NULL
-- Inicializa variables
IF @psPeriodicidad = 'S'
SELECT @sFecha = 'd.fecha', @sTabDatos = 'datos_sem_' + @psAño
ELSE IF @psPeriodicidad = 'M'
SELECT @sFecha = 'convert(varchar(10), d.fecha, 126)', @sTabDatos = 'datos_mes_' + @psAño
ELSE IF @psPeriodicidad = 'D'
SELECT @sFecha = 'convert(varchar(10), d.fecha, 126)', @sTabDatos = 'datos_dia_' + @psAño
SET @sNiveles = ''
IF @psPeriodicidad <> 'D'
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Obtiene el conjunto de indicadores que servirá como base')
-- Agrupamos los indicadores para que después se determine cuales son las entidades que participan
SELECT p.id_ent_arbol, t.indicador, t.segmento, t.tipo, t.fecha, t.periodo, n.nivel_h, n.padre, n.hijo,
max(indentacion) AS indentacion
INTO #universo_indica
FROM #temporal_proceso t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK),
productos.dbo.cat_productos p WITH (NOLOCK), #cat_niveles n WITH (NOLOCK),
#cat_tipos ct WITH (NOLOCK)
WHERE t.indicador = r.indicador
AND t.tipo = ct.tipo
AND ct.tipo_base = r.tipo
AND r.periodicidad = @psPeriodicidad
AND r.agregacion IN ('A', 'P')
AND r.producto = p.id_producto COLLATE SQL_Latin1_General_CP1_CI_AS
AND p.id_ent_arbol = n.id_ent_arbol
AND n.padre IS NOT NULL
GROUP BY p.id_ent_arbol, t.indicador, t.segmento, t.tipo, t.fecha, t.periodo, n.nivel_h, n.padre,
n.hijo
IF (@@error <> 0)
RETURN -2

```

Apéndice C. Código de los procesos

```
SELECT @nIndentacion = max(indentacion) FROM #cat_niveles_sup

IF @psPeriodicidad <> 'D'
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
    getdate(), 'Inserta en #temporal_agrega')

    INSERT #temporal_agrega (id_ent_arbol, indentacion, entidad, indicador, segmento, tipo, valor, periodo,
    fecha, calculado)
    SELECT u.id_ent_arbol, u.indentacion, t.entidad, t.indicador, t.segmento, t.tipo, t.valor, t.periodo,
    t.fecha, 0
    FROM #temporal_proceso t WITH (NOLOCK), #universo_indica u WITH (NOLOCK), #cat_niveles_inf c WITH
    (NOLOCK)
    WHERE t.indicador = u.indicador
    AND t.entidad = u.hijo
    AND t.segmento = u.segmento
    AND t.tipo = u.tipo
    AND t.periodo = u.periodo
    AND u.hijo = c.hijo
    IF (@@error <> 0)
        RETURN -3

IF @psPeriodicidad <> 'D'
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
    getdate(), 'Crea la tabla #entidades_faltantes')

    -- Crea el universo de datos, pero con valor nulo si no existe en los datos cargados
    SELECT u.id_ent_arbol, u.indentacion, u.hijo as entidad, u.indicador, u.segmento, u.tipo, a.valor,
    u.periodo,
    u.fecha, 0 as calculado
    INTO #entidades_faltantes
    FROM #universo_indica u WITH (NOLOCK), #temporal_agrega a WITH (NOLOCK), #cat_niveles_inf c WITH
    (NOLOCK)
    WHERE u.id_ent_arbol *= a.id_ent_arbol
    AND u.indentacion *= a.indentacion
    AND u.hijo *= a.entidad
    AND u.indicador *= a.indicador
    AND u.segmento *= a.segmento
    AND u.tipo *= a.tipo
    AND u.periodo *= a.periodo
    AND u.hijo = c.hijo

    -- Elimina los elementos que ya existen
    DELETE #entidades_faltantes WHERE valor IS NOT NULL

IF @psPeriodicidad <> 'D'
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
    getdate(), 'Obtendrá la información de datos sem o mes para las agregaciones')

    -- Obtiene los indicadores que se encuentran en datos_mmm_yyyy que no se encuentran en la carga
    SET @sSql = '
    INSERT #temporal_agrega (id_ent_arbol, indentacion, entidad, indicador, segmento, tipo, valor,
    periodo, fecha, calculado)
    SELECT e.id_ent_arbol, e.indentacion, d.entidad, d.indicador, d.segmento, d.tipo, d.valor,
    d.periodo, e.fecha, 0
    FROM ' + @sTabDatos + ' d WITH (NOLOCK), #entidades_faltantes e WITH (NOLOCK)
    WHERE d.entidad = e.entidad
    AND d.indicador = e.indicador
    AND d.segmento = e.segmento
    AND d.tipo = e.tipo
    AND d.periodo = e.periodo '
    EXEC(@sSql)
    IF (@@error <> 0)
        RETURN -4

    -- ===== INICIAN LOS CALCULOS DE LAS AGREGACIONES
    =====
IF @psPeriodicidad <> 'D'
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
    getdate(), 'Antes de realizar las operaciones.')

    DECLARE cursor_entidad_p CURSOR LOCAL FOR
    SELECT DISTINCT padre, indentacion FROM #cat_niveles_sup ORDER BY indentacion DESC
    OPEN cursor_entidad_p
    FETCH NEXT FROM cursor_entidad_p INTO @sEntidadP, @nIndentacion
    WHILE @@FETCH_STATUS = 0
    BEGIN
```

```

-- Genera las divisiones a partir de las áreas. Sólomente se calculan los de tipo agregación A =
suma
INSERT #temporal_agrega (id_ent_arbol, indentacion, entidad, indicador, segmento, tipo, valor,
periodo, fecha, calculado)
SELECT a.id_ent_arbol, @nIndentacion, e.padre, a.indicador, a.segmento, a.tipo,
CASE r.agregacion WHEN 'A' THEN sum(a.valor) WHEN 'P' THEN avg(a.valor) END,
a.periodo, a.fecha, 10
FROM #temporal_agrega a WITH (NOLOCK), entidades_arbol e WITH (NOLOCK), cat_responsabilidad r WITH
(NOLOCK),
#cat_tipos ct WITH (NOLOCK)
WHERE a.entidad = e.hijo
AND a.id_ent_arbol = e.id_ent_arbol
AND a.indicador = r.indicador
AND r.periodicidad = @psPeriodicidad
AND a.tipo = ct.tipo
AND r.tipo = ct.tipo_base
AND e.padre = @sEntidadP
GROUP BY a.id_ent_arbol, e.padre, a.indicador, a.segmento, a.tipo, a.periodo, a.fecha, r.agregacion
IF (@@error <> 0)
RETURN -7

-- Obtiene las entidades de nivel D que no se generarán por falta de área CASO (Telnor)
FETCH NEXT FROM cursor_entidad_p INTO @sEntidadP, @nIndentacion
END
CLOSE cursor_entidad_p
DEALLOCATE cursor_entidad_p

IF @psPeriodicidad <> 'D'
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Después de realizar las operaciones.')

-- ===== TERMINAN LOS CALCULOS DE LAS AGREGACIONES
=====

--===== INICIA OBTENCION DE DIFERENCIAS
=====
IF @psPeriodicidad <> 'D'
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Se almacenan las diferencias.')

-- Se insertan en la tabla de diferencias, los indicadores calculados y cargados para reportarlos
INSERT bitacora_dif_agregacion (carga, entidad, indicador, segmento, tipo, periodo, fecha, valor_carg,
valor_calc, responsable)
SELECT @nCarga as carga, p.entidad, p.indicador, p.segmento, p.tipo, p.periodo, p.fecha, p.valor,
a.valor,
CASE WHEN r.calc_agre = 1 THEN 'N' ELSE 'S' END as responsable
FROM #temporal_proceso p WITH (NOLOCK), #temporal_agrega a WITH (NOLOCK), cat_responsabilidad r WITH
(NOLOCK),
#cat_tipos ct WITH (NOLOCK)
WHERE p.entidad = a.entidad
AND p.indicador = a.indicador
AND p.segmento = a.segmento
AND p.tipo = a.tipo
AND p.periodo = a.periodo
AND abs(abs(a.valor) - abs(p.valor)) > 0.01
AND p.indicador = r.indicador
AND r.tipo = ct.tipo_base
AND p.tipo = ct.tipo
AND r.periodicidad = @psPeriodicidad
IF (@@error <> 0)
RETURN -10
IF @psPeriodicidad <> 'D'
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Pasó Se almacenan las diferencias.')

--===== TERMINA OBTENCION DE DIFERENCIAS
=====

-- ===== INICIA SECCION DE ENVIO DE CORREOS
=====
IF @pnuComparar = 0
BEGIN
IF EXISTS(SELECT * FROM bitacora_dif_agregacion WHERE carga = @nCarga AND responsable = 'S')
BEGIN
IF @psPeriodicidad <> 'D'
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga,

```

Apéndice C. Código de los procesos

```
@psPeriodicidad, getdate(), 'Envía correo de diferencias de agregaciones con responsabilidad del
usuario.')

-- Envía correo indicando que faltan algunas entidades para que esté completa la agregación
SELECT @sMensaje = 'Los indicadores agregados de la carga ' + @sCarga +
' tiene diferencias con los calculados por el sistema.' + char(13) +
' Verifica los datos del listado anexo.' + char(13) + char(13) +
' Recuerda que hasta el último día de diciembre se permitirá cargar los datos,
pero el ' +
' próximo año serán rechazados los indicadores agregados que presenten
diferencias con ' +
' respecto los calculados por el sistema.',
@ssubject = 'Cálculos SCI (' + @psPeriodicidad + ')_Informativo_Diferencias_' + 'Rechazo
por diferencias
en agregaciones de la carga ' + @sCarga,
@sSql = 'SELECT entidad, indicador, segmento, tipo, periodo, fecha, valor_carg, valor_calc
' +
' FROM sci.dbo.bitacora_dif_agregacion ' +
' WHERE carga = ' + @sCarga + ' AND responsable = ' + char(39) + 'S' + char(39)

EXECUTE master..xp_sendmail @recipients = @sMail,
@message = @sMensaje,
@query = @sSql,
@copy_recipients = 'jdelaveg@telmex.com; acelada@telmex.com;
psolis@telmex.com; jgteran@telmex.com',
@blind_copy_recipients = '',
@subject = @ssubject,
@attach_results = 'TRUE',
@width = 250

IF (@@error <> 0)
RETURN -11
END

IF EXISTS (SELECT * FROM bitacora_dif_agregacion WHERE carga = @nCarga AND responsable = 'N')
BEGIN
IF @psPeriodicidad <> 'D'
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga,
@psPeriodicidad, getdate(), 'Envía correo de diferencias de agregaciones sin responsabilidad del
usuario.')

SELECT @sMensaje = 'La carga ' + @sCarga + ' ha encontrado diferencias en los valores agregados.'
+ char(13) +
' Los valores calculados por el sistema sustituyen a los cargados.',
@ssubject = 'Cálculos SCI (' + @psPeriodicidad + ')_Informativo_Diferencias_' +
'Diferencias en agregaciones de la carga ' + @sCarga,
@sSql = 'SELECT entidad, indicador, segmento, tipo, periodo, fecha, valor_carg, valor_calc
' +
' FROM sci.dbo.bitacora_dif_agregacion ' +
' WHERE carga = ' + @sCarga + ' AND responsable = ' + char(39) + 'N' + char(39)

EXECUTE master..xp_sendmail @recipients = @sMail,
@message = @sMensaje,
@query = @sSql,
@copy_recipients = 'jdelaveg@telmex.com; acelada@telmex.com;
psolis@telmex.com; jgteran@telmex.com',
@blind_copy_recipients = '',
@subject = @ssubject,
@attach_results = 'TRUE',
@width = 250

IF (@@error <> 0)
RETURN -12
END
-- ===== TERMINA SECCION DE ENVIO DE CORREOS
=====

-- ===== INICIA SECCION DE TRANSFERENCIA DE DATOS A #temporal_proceso
=====
-- Eliminamos los agregados que se cargan para sustituirse por los calculados cuando cumplen
(calc_agre = 1)
DELETE #temporal_proceso
FROM #temporal_proceso t WITH (NOLOCK), #temporal_agrega a WITH (NOLOCK), cat_responsabilidad r
WITH (NOLOCK),
#cat_tipos ct WITH (NOLOCK)
WHERE t.entidad = a.entidad
AND t.indicador = a.indicador
AND t.segmento = a.segmento
AND t.tipo = a.tipo
AND t.periodo = a.periodo
AND a.calculado = 10
```

```

        AND a.indicador = r.indicador
        AND r.periodicidad = @psPeriodicidad
        AND r.tipo = ct.tipo_base
        AND t.tipo = ct.tipo
        AND r.calc_agre = 1
    IF (@@error <> 0)
        RETURN -13

    -- Eliminamos los agregados que se calcularon para dejar los cargados cuando cumplen (calc_agre = 0)
    DELETE #temporal_agrega
    FROM #temporal_proceso t WITH (NOLOCK), #temporal_agrega a WITH (NOLOCK), cat_responsabilidad r
WITH (NOLOCK),
        #cat_tipos ct WITH (NOLOCK)
    WHERE t.entidad = a.entidad
        AND t.indicador = a.indicador
        AND t.segmento = a.segmento
        AND t.tipo = a.tipo
        AND t.periodo = a.periodo
        AND a.calculado = 10
        AND a.indicador = r.indicador
        AND r.periodicidad = @psPeriodicidad
        AND r.tipo = ct.tipo_base
        AND t.tipo = ct.tipo
        AND r.calc_agre = 0
    IF (@@error <> 0)
        RETURN -14

    -- Pasa los indicadores calculados a #temporal_proceso
    INSERT #temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo,
tipo_calc)
    SELECT DISTINCT @nCarga, entidad, indicador, segmento, tipo, valor, fecha, periodo, 10
    FROM #temporal_agrega
    WHERE calculado = 10
    IF (@@error <> 0)
        RETURN -15

    DELETE #temporal_proceso WHERE valor IS NULL
END -- IF @pnuComparar = 0
ELSE IF @pnuComparar = 10 -- para auditoría de carga directa
BEGIN
    IF EXISTS(SELECT * FROM bitacora_dif_agregacion WHERE carga = @nCarga)
    BEGIN
    IF @psPeriodicidad <> 'D'
        insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga,
@psPeriodicidad, getdate(), 'Envía correo de diferencias de agregaciones con responsabilidad del
usuario.')

        -- Envía correo indicando que faltan algunas entidades para que esté completa la agregación
        SELECT @sMensaje = 'Los indicadores agregados de la carga ' + @sCarga +
            ' tiene diferencias con los calculados por el sistema.' + char(13) +
            'Verifica los datos del listado anexo.' + char(13) + char(13) +
            'Recuerda que hasta el último día de diciembre se permitirá cargar los datos,
pero el ' +
                'próximo año serán rechazados los indicadores agregados que presenten
diferencias con ' +
                    'respecto los calculados por el sistema.',
        @sSubject = 'Cálculos SCI (' + @psPeriodicidad + ')_Informativo_Diferencias_' + 'Rechazo
por diferencias en agregaciones de la carga ' + @sCarga,
        @sSql = 'SELECT entidad, indicador, segmento, tipo, periodo, fecha, valor_carg, valor_calc
' +
            'FROM sci.dbo.bitacora_dif_agregacion ' +
            'WHERE carga = ' + @sCarga

        EXECUTE master..xp_sendmail @recipients = @sMail,
            @message = @sMensaje,
            @query = @sSql,
            @copy_recipients = 'jdelaveg@telmex.com; acelada@telmex.com;
psolis@telmex.com; jgteran@telmex.com',
            @blind_copy_recipients = '',
            @subject = @sSubject,
            @attach_results = 'TRUE',
            @width = 250

    IF (@@error <> 0)
        RETURN -11

    END -- IF EXISTS(SELECT * FROM bitacora_dif_agregacion WHERE carga = @nCarga)
END -- ELSE (IF @pnuComparar = 1)
ELSE IF @pnuComparar = 20 -- Para auditoría y reagregaciones

```

```
BEGIN
-- Eliminamos los agregados que se calcularon para dejar los cargados cuando cumplen (calc_agre = 0)
DELETE #temporal_proceso
FROM #temporal_proceso t WITH (NOLOCK), #temporal_agrega a WITH (NOLOCK)
WHERE t.entidad = a.entidad
  AND t.indicador = a.indicador
  AND t.segmento = a.segmento
  AND t.tipo = a.tipo
  AND t.periodo = a.periodo
  AND a.calculado = 10
IF (@@error <> 0)
  RETURN -14

-- Pasa los indicadores calculados a #temporal_proceso
INSERT #temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo,
tipo_calc)
SELECT DISTINCT @nCarga, entidad, indicador, segmento, tipo, valor, fecha, periodo, 10
  FROM #temporal_agrega
 WHERE calculado = 10
IF (@@error <> 0)
  RETURN -15

DELETE #temporal_proceso WHERE valor IS NULL
END -- ELSE (IF @pnuComparar = 20)

--select * from bitacora_dif_agregacion
DELETE bitacora_dif_agregacion WHERE carga = @nCarga
IF (@@error <> 0)
  RETURN -16
-- ===== TERMINA SECCION DE TRANSFERENCIA DE DATOS A #temporal_proceso
=====
SELECT @nuProcesados = count(*) FROM #temporal_proceso WHERE tipo_calc = 10
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Se procesaron ' + convert(varchar(10), @nuProcesados) + ' elementos ')

-- select * from #temporal_proceso order by indicador, entidad

RETURN 0
END -- sp_Agregacion
```

C.3 Proceso de acumulación.

```

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Acumulacion' AND type = 'P')
  DROP PROCEDURE sp_Acumulacion
GO
--
*****
-- Nombre:          sp_Acumulacion
-- Descripción:     Obtiene el valor de los indicadores de tipo A a partir de los de tipo M actual y tipo A
del periodo
--                  anterior.
--
-- Parámetros:
-- Return:          -1:Error, 0:Éxito
--
*****
CREATE PROCEDURE sp_Acumulacion
  @psPeriodicidad char(1),
  @psAño          char(4),
  @pnuComparar   smallint -- 0: Proceso normal de carga
                        -- 10: Para auditoría de carga directa. NO se hace sustitución de la
información cargada
                        --                y se envía correo en caso de diferencias.
                        -- 20: Para la auditoría y reagregaciones. No toma en cuenta los correos
AS
BEGIN
  ----- INICIA SECCION DE DECLARACIONES -----
  -----
  DECLARE   @sSql          varchar(8000),
            @nError       int, @sError varchar(150),
            @sTabDatos    varchar(60), @sFecha varchar(100),
            @sAcumula     char(1), @sFormula varchar(100),
            @nPeriodoMin  smallint, @nPeriodoMax smallint, @nPeriodo smallint,
            @sMensaje     varchar(300),
            @sSubject     varchar(100),
            @sCarga       varchar(10),
            @sUsuario     varchar(15),
            @sMail        varchar(50),
            @nCarga       int,
            @nuProcesados int
  ----- TERMINA SECCION DE DECLARACIONES -----
  -----
  -- Establece el formato de fecha
  SET DATEFORMAT mdy

  CREATE TABLE #indicadores_sumados(
    entidad      varchar(10),
    indicador    varchar(10),
    segmento     varchar(2),
    tipo         char(1),
    valor        float,
    periodo     tinyint,
    fecha        varchar(10)
  )
  CREATE INDEX ix_indicadores_sumados1 On #indicadores_sumados (entidad, indicador, segmento, tipo,
periodo)
  CREATE INDEX ix_indicadores_sumados2 On #indicadores_sumados (entidad, indicador, segmento, tipo)

  -- Almacena los datos que servirán para realizar los cálculos
  CREATE TABLE #temporal_acumula(
    entidad      varchar(10),
    indicador    varchar(10),
    segmento     varchar(2),
    tipo         char(1),
    valor        float,
    periodo     tinyint,
    fecha        varchar(10)
  )
  CREATE INDEX ix_temporal_acumula1 On #temporal_acumula (entidad, indicador, segmento, tipo, periodo)
  CREATE INDEX ix_temporal_acumula2 On #temporal_acumula (entidad, indicador, segmento, tipo)

```

Apéndice C. Código de los procesos

```
-- Almacena los datos que servirán para realizar los cálculos
CREATE TABLE #temporal_acumula_2(
    entidad          varchar(10),
    indicador        varchar(10),
    segmento         varchar(2),
    tipo             char(1),
    valor            float,
    periodo          tinyint,
    fecha            varchar(10)
)
CREATE INDEX ix_temporal_acumula_21 On #temporal_acumula_2 (entidad, indicador, segmento, tipo,
periodo)
CREATE INDEX ix_temporal_acumula_22 On #temporal_acumula_2 (entidad, indicador, segmento, tipo)

CREATE TABLE #tipos(
    tipo_marg char(1),
    tipo_acum char(1)
)
INSERT INTO #tipos VALUES ('M', 'A')
INSERT INTO #tipos VALUES ('P', 'Q')

CREATE TABLE #tipos_responsabilidad(
    tipo_base char(1),
    tipo      char(1)
)
INSERT #tipos_responsabilidad VALUES ('M', 'M')
INSERT #tipos_responsabilidad VALUES ('M', 'A')
INSERT #tipos_responsabilidad VALUES ('P', 'P')
INSERT #tipos_responsabilidad VALUES ('P', 'Q')

CREATE TABLE #pivote(
    entidad          varchar(10),
    indicador        varchar(10),
    segmento         varchar(2),
    tipo             char(1)
)

CREATE TABLE #indicador_periodo_intermedio(
    entidad          varchar(10),
    indicador        varchar(10),
    segmento         varchar(2),
    tipo             char(1),
    periodo          smallint
)
CREATE INDEX ix_indicador_periodo_intermedio On #indicador_periodo_intermedio (entidad, indicador,
segmento, tipo, periodo)

IF NOT EXISTS (SELECT name FROM sysobjects WHERE name = 'bitacora_dif_acumulacion' AND type = 'U')
BEGIN
    CREATE TABLE bitacora_dif_acumulacion(
        carga          int,
        entidad        varchar(10),
        indicador      varchar(10),
        segmento       varchar(2),
        tipo           char(1),
        periodo        smallint,
        fecha          varchar(10),
        valor_carg     float,
        valor_calc     float,
        responsable    char(1)
    )
    CREATE INDEX ix_bitacora_dif_acumulacion On bitacora_dif_acumulacion (carga)
END

CREATE TABLE #cat_acumulacion(
    acumulacion char(1),
    formula      varchar(200)
)
INSERT INTO #cat_acumulacion VALUES('A', 'sum(t.valor), max(t.periodo), max(t.fecha)')
INSERT INTO #cat_acumulacion VALUES('P', 'avg(t.valor), max(t.periodo), max(t.fecha)')
INSERT INTO #cat_acumulacion VALUES('S', 't.valor, t.periodo, t.fecha')

SELECT TOP 1 @nCarga = carga FROM #temporal_proceso

SELECT @sUsuario = 'HPALAF0X', @sMail = 'hpalafox@telmex.com'
SET @sCarga = convert(varchar(10), @nCarga)
SELECT TOP 1 @sUsuario = usuario
FROM #temporal_proceso t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #tipos_responsabilidad
```

```

tr WITH (NOLOCK)
  WHERE t.indicador = r.indicador
  AND t.tipo = tr.tipo
  AND r.tipo = tr.tipo_base
  AND r.periodicidad = @psPeriodicidad

SELECT @sMail = e_mail FROM productos.dbo.cat_usuarios_producto WHERE id_responsable = @sUsuario

DELETE bitacora_dif_acumulacion WHERE carga = @nCarga

-- Obtiene el tipo de periodicidad y el nombre de tabla que tiene los datos con la que se tiene que
trabajar.
IF @psPeriodicidad = 'S'
  SELECT @sTabDatos = 'datos_sem_' + @psAño, @nPeriodoMax = 52, @sFecha = 'd.fecha'
ELSE
  SELECT @sTabDatos = 'datos_mes_' + @psAño, @nPeriodoMax = 12, @sFecha = 'convert(varchar(10),
d.fecha, 126)'

-- Obtiene los valores de #temporal_proceso que servirán de pivote
INSERT #pivote (entidad, indicador, segmento, tipo)
SELECT DISTINCT entidad, indicador, segmento, tipo FROM #temporal_proceso WITH (NOLOCK)

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Obtiene lista de indicadores con su periodo mínimo')
-- Obtiene una lista de los indicadores con el periodo mínimo que tengan valor cero, ya que se
comenzará a acumularse
-- a partir del periodo mínimo que sea distinto de cero.
SET @sSql = '
INSERT INTO #indicador_periodo_intermedio
SELECT p.entidad, p.indicador, p.segmento, p.tipo, isnull(min(d.periodo), 0)
FROM ' + @sTabDatos + ' d WITH (NOLOCK), #pivote p WITH (NOLOCK), cat_responsabilidad r WITH
(NOLOCK),
#tipos_responsabilidad tr WITH (NOLOCK)
WHERE p.entidad *= d.entidad
AND p.indicador *= d.indicador
AND p.segmento *= d.segmento
AND p.tipo *= d.tipo
AND p.tipo IN (' + char(39) + 'P' + char(39) + ', ' + char(39) + 'M' + char(39) + ')
AND p.indicador = r.indicador
AND p.tipo = tr.tipo
AND r.tipo = tr.tipo_base
AND r.periodicidad = ' + char(39) + @psPeriodicidad + char(39) + '
AND r.acumulacion IN (' + char(39) + 'P' + char(39) + ', ' + char(39) + 'A' + char(39) + ', ' +
char(39) + 'S' + char(39) + ')
AND d.valor <> 0
GROUP BY p.entidad, p.indicador, p.tipo, p.segmento '
SELECT @nError = -1, @sError = 'Error al insertar en la tabla #indicador_periodo_intermedio'
EXEC(@sSql)
IF (@@error <> 0)
  GOTO err_acumula

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Se pasan los indicadores a #temporal_acumula')
-- Pasa a la tabla #temporal_acumula los indicadores que intervendrán en el proceso de acumulación con
los
-- periodos que tiene #temporal_proceso
SET @sSql = '
INSERT INTO #temporal_acumula (entidad, indicador, segmento, tipo, valor, periodo, fecha)
SELECT t.entidad, t.indicador, t.segmento, t.tipo, t.valor, t.periodo, t.fecha
FROM #temporal_proceso t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK),
#tipos_responsabilidad tr WITH (NOLOCK)
WHERE t.indicador = r.indicador
AND t.tipo = tr.tipo
AND r.tipo = tr.tipo_base
AND r.periodicidad = ' + char(39) + @psPeriodicidad + char(39) + '
AND r.acumulacion IN (' + char(39) + 'P' + char(39) + ', ' + char(39) + 'A' + char(39) + ', ' +
char(39) + 'S' + char(39) + ') + '
AND t.tipo IN (' + char(39) + 'P' + char(39) + ', ' + char(39) + 'M' + char(39) + ') '
SELECT @nError = -2, @sError = 'Error al insertar en la tabla #temporal_acumula'
EXEC(@sSql)
IF (@@error <> 0)
  GOTO err_acumula
-- Se obtiene el periodo que se está cargando y se considera como el periodo mínimo.
SELECT @nPeriodoMin = min(periodo) FROM #temporal_acumula WITH (NOLOCK)

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Obtiene de datos_mmm_yyy los datos indicadores de los otros periodo')
-- Obtiene de datos_mmm_yyy los datos indicadores de los otros periodo para realizar las acumulaciones

```

Apéndice C. Código de los procesos

```
SET @sSql = '
INSERT #temporal_acumula (entidad, indicador, segmento, tipo, valor, periodo, fecha)
SELECT d.entidad, d.indicador, d.segmento, d.tipo, d.valor, d.periodo, ' + @sFecha + '
FROM ' + @sTabDatos + ' d WITH (NOLOCK), #indicador_periodo_intermedio i WITH (NOLOCK)
WHERE d.entidad = i.entidad
AND d.indicador = i.indicador
AND d.segmento = i.segmento
AND d.tipo = i.tipo
AND d.periodo >= i.periodo
AND rtrim(d.entidad) + rtrim(d.indicador) + rtrim(d.segmento) + d.tipo + convert(varchar(5),
d.periodo) NOT IN
(SELECT rtrim(entidad) + rtrim(indicador) + rtrim(segmento) + tipo + convert(varchar(5),
periodo)
FROM #temporal_acumula WITH (NOLOCK)) '
SET @sSql = '
INSERT #temporal_acumula_2 (entidad, indicador, segmento, tipo, valor, periodo, fecha)
SELECT d.entidad, d.indicador, d.segmento, d.tipo, d.valor, d.periodo, ' + @sFecha + '
FROM ' + @sTabDatos + ' d WITH (NOLOCK), #indicador_periodo_intermedio i WITH (NOLOCK)
WHERE d.entidad = i.entidad
AND d.indicador = i.indicador
AND d.segmento = i.segmento
AND d.tipo = i.tipo
AND d.periodo >= i.periodo
--AND rtrim(d.entidad) + rtrim(d.indicador) + rtrim(d.segmento) + d.tipo + convert(varchar(5),
d.periodo) NOT IN
-- (SELECT rtrim(entidad) + rtrim(indicador) + rtrim(segmento) + tipo + convert(varchar(5),
periodo)
-- FROM #temporal_acumula WITH (NOLOCK)) '
SELECT @nError = -3, @sError = 'Error al insertar en la tabla #temporal_acumula_2 de periodos
posteriores'
EXEC(@sSql)
IF (@@error <> 0)
GOTO err_acumula

SET @sSql = '
DELETE #temporal_acumula_2
FROM #temporal_acumula t1 WITH (NOLOCK), #temporal_acumula_2 t2 WITH (NOLOCK)
WHERE t1.entidad = t2.entidad
AND t1.indicador = t2.indicador
AND t1.segmento = t2.segmento
AND t1.tipo = t2.tipo
AND t1.periodo = t2.periodo '
SELECT @nError = -3, @sError = 'Error al eliminar en la tabla #temporal_acumula_2 de periodos
posteriores'
EXEC(@sSql)
IF (@@error <> 0)
GOTO err_acumula
SET @sSql = '
INSERT #temporal_acumula (entidad, indicador, segmento, tipo, valor, periodo, fecha)
SELECT entidad, indicador, segmento, tipo, valor, periodo, fecha
FROM #temporal_acumula_2 WITH (NOLOCK) '
EXEC(@sSql)
IF (@@error <> 0)
GOTO err_acumula

-- Se obtiene el máximo periodo después de que se han obtenido los indicadores de datos_mmm_yyyy.
SELECT @nPeriodoMax = max(periodo) FROM #temporal_acumula WITH (NOLOCK)

SELECT entidad, indicador, segmento, tipo, max(periodo) AS periodo_max
INTO #temporal_acumula_max
FROM #temporal_acumula WITH (NOLOCK)
GROUP BY entidad, indicador, segmento, tipo

SELECT @nError = -4, @sError = 'Error al insertar en la tabla #indicadores_sumados dentro del while'
-- Obtiene las acumulaciones de tipo A=suma, P=promedio S=saldo
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Antes de entrar al cursor cur_acumula ')
DECLARE cur_acumula CURSOR LOCAL FOR
SELECT acumulacion, formula FROM #cat_acumulacion
OPEN cur_acumula
FETCH NEXT FROM cur_acumula INTO @sAcumula, @sFormula
WHILE @@FETCH_STATUS = 0
BEGIN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Realiza cálculo de acumulación del tipo de acumulación ' + @sAcumula)
-- Se inicializa la variable periodo con el mínimo
SELECT @nPeriodo = @nPeriodoMin
```

```

-- Obtiene los datos acumulados hasta el periodo anterior
WHILE @nPeriodo <= @nPeriodoMax
BEGIN
    SET @sSql = '
    INSERT #indicadores_sumados (entidad, indicador, segmento, tipo, valor, periodo, fecha)
    SELECT t.entidad, t.indicador, t.segmento, ti.tipo_acum, ' + @sFormula + '
    FROM #temporal_acumula t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #tipos ti WITH
(NOLOCK),
        #tipos_responsabilidad tr WITH (NOLOCK), #temporal_acumula_max tm
    WHERE t.indicador = r.indicador
        AND t.tipo = tr.tipo
        AND r.tipo = tr.tipo_base
        AND r.periodicidad = ' + char(39) + @psPeriodicidad + char(39) + '
        AND r.acumulacion = ' + char(39) + @sAcumula + char(39) + '
        AND t.tipo = ti.tipo_marg
        AND t.tipo IN (' + char(39) + 'P' + char(39) + ', ' + char(39) + 'M' + char(39) + ')

        AND t.entidad = tm.entidad
        AND t.indicador = tm.indicador
        AND t.segmento = tm.segmento
        AND t.tipo = tm.tipo '
    IF @sAcumula = 'S'
        SET @sSql = @sSql + '
            AND t.periodo = ' + convert(varchar, @nPeriodo) + '
            AND t.periodo <= tm.periodo_max '
    ELSE
        SET @sSql = @sSql + '
            AND t.periodo <= ' + convert(varchar, @nPeriodo) + '
            AND t.periodo <= tm.periodo_max
        GROUP BY t.entidad, t.indicador, t.segmento, ti.tipo_acum '

--    print @sSql
EXEC(@sSql)
IF (@@error > 0)
    GOTO err_cursor

    SET @nPeriodo = @nPeriodo + 1
END -- WHILE @nPeriodo <= @nPeriodoMax

    FETCH NEXT FROM cur_acumula INTO @sAcumula, @sFormula
END
CLOSE cur_acumula
DEALLOCATE cur_acumula

-- Inserta los indicadores de los periodos posteriores a los de #temporal_proceso. Estos indicadores se
tomaron de
-- datos_mmm_yyyy. Se necesitan para realizar calculos, pero no se tienen que pasar a datos_mmm_yyyy.
Para no tomarlos
-- en cuenta en la carga se les pone tipo_calc = 1
----- INICIA OBTENCION DE DIFERENCIAS
=====
SELECT @nError = -5, @sError = 'Error al insertar en la tabla bitacora_diferencias'
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Registra en bitacora_diferencias las diferencias de los calculos realizados contra los
cargados')
-- BITACORA
-- Graba en bitacora los indicadores acumulados que tienen diferencias entre los calculados y los
cargados
INSERT bitacora_dif_acumulacion (carga, entidad, indicador, segmento, tipo, periodo, fecha, valor_carg,
valor_calc, responsable)
SELECT DISTINCT @nCarga, p.entidad, p.indicador, p.segmento, p.tipo, p.periodo, p.fecha, p.valor,
s.valor,
    CASE WHEN r.calc_acum = 1 THEN 'N' ELSE 'S' END
FROM #indicadores_sumados s WITH (NOLOCK), #temporal_proceso p WITH (NOLOCK), cat_responsabilidad r
WITH (NOLOCK),
    #tipos_responsabilidad tr WITH (NOLOCK)
WHERE s.entidad = p.entidad
    AND s.indicador = p.indicador
    AND s.segmento = p.segmento
    AND s.tipo = p.tipo
    AND s.periodo = p.periodo
    AND abs(abs(s.valor) - abs(p.valor)) > 0.01
    AND s.indicador = r.indicador
    AND s.tipo = tr.tipo
    AND r.tipo = tr.tipo_base
    AND r.periodicidad = @psPeriodicidad
IF (@@error > 0)

```

Apéndice C. Código de los procesos

```
GOTO err_acumula
----- TERMINA OBTENCION DE DIFERENCIAS
=====

----- INICIA TRATO DIFERENCIAS
=====

IF @pnuComparar = 0
BEGIN
-- Se rechaza porque hay diferencias y el responsable es el usuario
IF EXISTS (SELECT * FROM bitacora_dif_acumulacion WHERE carga = @nCarga AND responsable = 'S')
BEGIN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Envía correo de diferencias de acumulaciones con responsabilidad del usuario.')
SELECT @sMensaje = 'Los indicadores acumulados de la carga ' + @sCarga +
' tiene diferencias con los calculados por el sistema.' + char(13) +
' Verifica los datos del listado anexo.' + char(13) + char(13) +
'Recuerda que hasta el último día de diciembre se permitirá cargar los datos,
pero el ' +
' próximo año serán rechazados los indicadores acumulados que presenten
diferencias con ' +
' respecto los calculados por el sistema.',
@subject = 'Cálculos SCI (' + @psPeriodicidad + ')_Informativo_Diferencias_' + 'Rechazo
por diferencias en acumulaciones de la carga ' + @sCarga,
@sSql = 'SELECT entidad, indicador, segmento, tipo, periodo, fecha, valor_carg, valor_calc
' +
'FROM sci.dbo.bitacora_dif_acumulacion ' +
'WHERE carga = ' + @sCarga + ' AND responsable = ' + char(39) + 'S' + char(39)

EXECUTE master..xp_sendmail @recipients = @sMail,
@message = @sMensaje,
@query = @sSql,
@copy_recipients = 'jdelaveg@telmex.com; acelada@telmex.com;
psolis@telmex.com',
@blind_copy_recipients = '',
@subject = @subject,
@attach_results = 'TRUE',
@width = 250
--UPDATE bitacora_carga SET status_proceso = 'RU' WHERE carga = @nCarga
--RETURN -6
END -- IF EXISTS (SELECT * FROM bitacora_dif_acumulacion WHERE carga = @nCarga AND responsable =
'S')
IF EXISTS (SELECT * FROM bitacora_dif_acumulacion WHERE carga = @nCarga AND responsable = 'N')
BEGIN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Envía correo de diferencias de acumulaciones sin responsabilidad del usuario.')
SELECT @sMensaje = 'La carga ' + @sCarga + ' ha encontrado diferencias en los valores
acumulados.' + char(13) +
' Los valores calculados por el sistema sustituyen a los cargados.',
@subject = 'Cálculos SCI (' + @psPeriodicidad + ')_Informativo_Diferencias_' +
'Diferencias en acumulaciones de la carga ' + @sCarga,
@sSql = 'SELECT entidad, indicador, segmento, tipo, periodo, fecha, valor_carg, valor_calc
' +
'FROM sci.dbo.bitacora_dif_acumulacion ' +
'WHERE carga = ' + @sCarga + ' AND responsable = ' + char(39) + 'N' + char(39)

EXECUTE master..xp_sendmail @recipients = @sMail,
@message = @sMensaje,
@query = @sSql,
@copy_recipients = 'jdelaveg@telmex.com; acelada@telmex.com;
psolis@telmex.com',
@blind_copy_recipients = '',
@subject = @subject,
@attach_results = 'TRUE',
@width = 250

END
----- TERMINA TRATO DIFERENCIAS
=====

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Pasa datos a #temporal_proceso')
-- ===== INICIA SECCION DE TRANSFERENCIA DE DATOS A #temporal_proceso
=====
-- Eliminamos los acumulados que se cargan para sustituirse por los calculados cuando cumplen
(calc_acum = 1)
DELETE #temporal_proceso
FROM #temporal_proceso t WITH (NOLOCK), #indicadores_sumados s WITH (NOLOCK), cat_responsabilidad
r WITH (NOLOCK),
#tipos_responsabilidad tr WITH (NOLOCK)
```

```

WHERE t.entidad = s.entidad
AND t.indicador = s.indicador
AND t.segmento = s.segmento
AND t.tipo = s.tipo
AND t.periodo = s.periodo
AND s.indicador = r.indicador
AND s.tipo = tr.tipo
AND r.tipo = tr.tipo_base
AND r.periodicidad = @psPeriodicidad
AND r.calc_acum = 1
IF (@@error <> 0)
GOTO err_acumula

-- Eliminamos los acumulados que se calcularon para dejar los cargados cuando cumplen (calc_acum =
0)
DELETE #indicadores_sumados
FROM #temporal_proceso t WITH (NOLOCK), #indicadores_sumados s WITH (NOLOCK), cat_responsabilidad
r WITH (NOLOCK),
#tipos_responsabilidad tr WITH (NOLOCK)
WHERE t.entidad = s.entidad
AND t.indicador = s.indicador
AND t.segmento = s.segmento
AND t.tipo = s.tipo
AND t.periodo = s.periodo
AND s.indicador = r.indicador
AND s.tipo = tr.tipo
AND r.tipo = tr.tipo_base
AND r.periodicidad = @psPeriodicidad
AND r.calc_acum = 0
IF (@@error <> 0)
GOTO err_acumula

-- Pasa los indicadores calculados a #temporal_proceso
INSERT #temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo,
tipo_calc)
SELECT @nCarga, entidad, indicador, segmento, tipo, min(valor), fecha, periodo, 20
FROM #indicadores_sumados
GROUP BY entidad, indicador, segmento, tipo, fecha, periodo
IF (@@error <> 0)
GOTO err_acumula
END -- IF @pnuComparar = 0
ELSE IF @pnuComparar = 10 -- para auditoría de carga directa
BEGIN
IF EXISTS (SELECT * FROM bitacora_dif_acumulacion WHERE carga = @nCarga)
BEGIN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Envía correo de diferencias de acumulaciones sin responsabilidad del usuario.')
SELECT @sMensaje = 'La carga ' + @sCarga + ' ha encontrado diferencias en los valores
acumulados.' + char(13),
@sSubject = 'Cálculos SCI (' + @psPeriodicidad + ')_Informativo_Diferencias_' +
'Diferencias en acumulaciones de la carga ' + @sCarga,
@sSql = 'SELECT entidad, indicador, segmento, tipo, periodo, fecha, valor_carg, valor_calc
' +
'FROM sci.dbo.bitacora_dif_acumulacion ' +
'WHERE carga = ' + @sCarga

EXECUTE master..xp_sendmail @recipients = @sMail,
@message = @sMensaje,
@query = @sSql,
@copy_recipients = 'jdelaveg@telmex.com; acelada@telmex.com;
psolis@telmex.com',
@blind_copy_recipients = '',
@subject = @sSubject,
@attach_results = 'TRUE',
@width = 250
END -- IF EXISTS (SELECT * FROM bitacora_dif_acumulacion WHERE carga = @nCarga)
END -- ELSE IF @pnuComparar = 10
ELSE IF @pnuComparar = 20 -- para auditoría y reacumulación
BEGIN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Pasa datos a #temporal_proceso')
-- ===== INICIA SECCION DE TRANSFERENCIA DE DATOS A #temporal_proceso
=====
-- Eliminamos los acumulados que se cargan para sustituirse por los calculados cuando cumplen
(calc_acum = 1)
DELETE #temporal_proceso
FROM #temporal_proceso t WITH (NOLOCK), #indicadores_sumados s WITH (NOLOCK)
WHERE t.entidad = s.entidad

```

```
        AND t.indicador = s.indicador
        AND t.segmento = s.segmento
        AND t.tipo = s.tipo
        AND t.periodo = s.periodo
    IF (@@error <> 0)
        GOTO err_acumula

    -- Pasa los indicadores calculados a #temporal_proceso
    INSERT #temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo,
tipo_calc)
    SELECT @nCarga, entidad, indicador, segmento, tipo, min(valor), fecha, periodo, 20
    FROM #indicadores_sumados
    GROUP BY entidad, indicador, segmento, tipo, fecha, periodo
    IF (@@error <> 0)
        GOTO err_acumula
    END -- ELSE IF @pnuComparar = 20
    DELETE bitacora_dif_acumulacion WHERE carga = @nCarga
    -- ===== TERMINA SECCION DE TRANSFERENCIA DE DATOS A #temporal_proceso
    =====
    SELECT @nuProcesados = count(*) FROM #temporal_proceso WHERE tipo_calc = 20
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'Se procesaron ' + convert(varchar(10), @nuProcesados) + ' elementos ')

    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), 'TERMINA ACUMULACIONES')

    return 0

err_acumula:
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), @sError)
    RETURN @nError
err_cursor:
    CLOSE cur_acumula
    DEALLOCATE cur_acumula
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @psPeriodicidad,
getdate(), @sError)
    RETURN @nError
END -- sp_Acumulacion
```

C.4 Principal de cálculos.

```

IF EXISTS (SELECT name FROM sysobjects WITH (NOLOCK) WHERE name = 'sp_Principal_Calculos' AND type = 'P')
  DROP PROCEDURE sp_Principal_Calculos
GO
--
*****
*****
-- Nombre:          sp_Principal_Calculos
-- Descripción:     A partir de una carga de datos, hace el llamado del proceso de generación de valores de
los
--                  indicadores calculados. Además agrupa el resultado de los calculados que pertenecen
al mismo
--                  producto para que se inserten en datos_mmm_yyyy. Los indicadores que no pertenecen al
mismo producto
--                  se les genera un archivo para ser insertados como nueva carga.
--
-- Parámetros:
-- Return:          -1:Error, 0:Éxito
--
*****
*****
CREATE PROCEDURE sp_Principal_Calculos
AS
BEGIN
  ----- INICIA SECCION DE DECLARACIONES -----
  -----
  DECLARE   @sSql          varchar(4000),
            @sError       varchar(150),
            @sPeriodicidad char(1),
            @sAño         char(4),
            @nPeriodoMin  smallint,
            @result       int,
            @sProducto    varchar(15), @sProducto_2 varchar(15), @sArchivo varchar(100),
            @sFecha       varchar(10),
            @sTabDatos    varchar(30),
            @nCarga       int,
            @nCargaNueva  int,
            @sUsuario     varchar(10),
            @sp_recur_sivo int,
            @sPeriodo1    varchar(10),
            @sPeriodo2    varchar(10),
            @nEnviarEMail bit,
            @nuProcesados int
  DECLARE   @sSubject     varchar(100),
            @sMail        varchar(50),
            @sMensaje     varchar(200)
  ----- TERMINA SECCION DE DECLARACIONES -----
  -----
  -- Establece el formato de fecha
  SET DATEFORMAT mdy
  --
  =====
  -- SECCION DE CONSTRUCCION DE TABLAS
  -----
  -- Tipos que se manejan en cat_responsabilidad
  CREATE TABLE #cat_tipos(
    tipo_base char(1),
    tipo      char(1)
  )
  INSERT #cat_tipos VALUES('M', 'M')
  INSERT #cat_tipos VALUES('M', 'A')
  INSERT #cat_tipos VALUES('P', 'P')
  INSERT #cat_tipos VALUES('P', 'Q')

  -- Tabla de trabajo
  CREATE TABLE #temporal_proceso(
    carga      int,

```

Apéndice C. Código de los procesos

```
entidad      varchar(10),
indicador    varchar(10),
segmento     varchar(2),
tipo         char(1),
valor        float,
fecha        varchar(10),
periodo      smallint,
tipo_calc    tinyint -- 0 = cargado, 1 = auxiliar, 10 = agregado, 20 = acumulado, 30 = calculado
)
CREATE INDEX ix_temp_proceso1 On #temporal_proceso (entidad, indicador, segmento, tipo, periodo, valor)
CREATE INDEX ix_temp_proceso2 On #temporal_proceso (entidad, indicador, segmento, tipo, periodo, fecha)
CREATE INDEX ix_temp_proceso3 On #temporal_proceso (entidad, indicador, segmento, tipo, periodo)
CREATE INDEX ix_temp_proceso4 On #temporal_proceso (entidad, indicador, segmento, periodo)
CREATE INDEX ix_temp_proceso5 On #temporal_proceso (indicador, tipo)
CREATE INDEX ix_temp_proceso6 On #temporal_proceso (entidad)
CREATE INDEX ix_temp_proceso7 On #temporal_proceso (tipo)

-- Tabla que almacena el total de indicadores calculados del mismo producto
CREATE TABLE #temporal_proceso_total(
    carga      int,
    entidad    varchar(10),
    indicador  varchar(10),
    segmento   varchar(2),
    tipo       char(1),
    valor      float,
    fecha      varchar(10),
    periodo    smallint,
    tipo_calc  tinyint, -- 0 = cargado, 1 = auxiliar, 10 = agregado, 20 = acumulado, 30 = calculado
    nivel_calc smallint
)
CREATE INDEX ix_temp_proceso_total_1 On #temporal_proceso_total (entidad, indicador, segmento, tipo,
periodo, valor)
CREATE INDEX ix_temp_proceso_total_2 On #temporal_proceso_total (entidad, indicador, segmento, tipo,
periodo, fecha)
CREATE INDEX ix_temp_proceso_total_3 On #temporal_proceso_total (entidad, indicador, segmento, tipo,
periodo)
CREATE INDEX ix_temp_proceso_total_4 On #temporal_proceso_total (entidad, indicador, segmento, periodo)
CREATE INDEX ix_temp_proceso_total_5 On #temporal_proceso_total (indicador, tipo)
CREATE INDEX ix_temp_proceso_total_6 On #temporal_proceso_total (entidad)
CREATE INDEX ix_temp_proceso_total_7 On #temporal_proceso_total (tipo)

-- Tabla auxiliar para manipular los datos
-- Tabla que almacenará los indicadores que provienen de datos_sem, ya que estos no se tienen que
cargar nuevamente.
CREATE TABLE #temporal_datos(
    entidad    varchar(10),
    indicador  varchar(10),
    segmento   varchar(2),
    tipo       char(1),
    valor      float,
    fecha      varchar(10),
    periodo    smallint
)
CREATE INDEX ix_temporal_datos1 On #temporal_datos (entidad, indicador, segmento, tipo, periodo, fecha)
CREATE INDEX ix_temporal_datos2 On #temporal_datos (entidad, indicador, segmento, tipo, periodo)
CREATE INDEX ix_temporal_datos3 On #temporal_datos (entidad, indicador, segmento, periodo)
CREATE INDEX ix_temporal_datos4 On #temporal_datos (indicador, tipo)
CREATE INDEX ix_temporal_datos5 On #temporal_datos (entidad)
CREATE INDEX ix_temporal_datos6 On #temporal_datos (tipo)

CREATE TABLE #tipos_formula(
    tipo_formula char(1),
    tipo_real    char(1)
)
INSERT #tipos_formula VALUES ('X', 'M')
INSERT #tipos_formula VALUES ('X', 'A')
INSERT #tipos_formula VALUES ('X', 'P')
INSERT #tipos_formula VALUES ('X', 'Q')
INSERT #tipos_formula VALUES ('M', 'M')
INSERT #tipos_formula VALUES ('A', 'A')
INSERT #tipos_formula VALUES ('P', 'P')
INSERT #tipos_formula VALUES ('Q', 'Q')
INSERT #tipos_formula VALUES ('T', 'T')
INSERT #tipos_formula VALUES ('U', 'U')

CREATE TABLE #insertados(
    carga      int,
    entidad    char(10),
```

```

        indicador char(10),
        segmento char(2),
        tipo char(1),
        valor float,
        fecha char(10),
        año smallint,
        periodo smallint,
        nivel char(1)
    )

CREATE TABLE #modificados(
    carga int,
    entidad char(10),
    indicador char(10),
    segmento char(2),
    tipo char(1),
    valor float,
    fecha char(10),
    año smallint,
    periodo smallint,
    nivel char(1),
    valor_anterior float
)

--
=====
-- SECCION CARGA INICIALIZACION DE VALORES Y CARGA DE DATOS
-----

SELECT @nCarga = isnull(min(carga), 0) FROM bitacora_carga WITH (NOLOCK) WHERE status_proceso = 'CC'
SELECT @sPeriodicidad = periodicidad FROM bitacora_carga WITH (NOLOCK) WHERE carga = @nCarga

IF @nCarga > 0
BEGIN
    SELECT @sError = 'Error al actualizar el estado CR en bitacora_carga'
    -- Actualiza estado para indicar que ya se están procesando los cálculos
    UPDATE bitacora_carga SET status_proceso = 'CR' WHERE carga = @nCarga
    IF (@@error <> 0)
        GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Inserta en #temporal_proceso_aux el producto ' + @sProducto)
SELECT @sError = 'Error al insertar #temporal_proceso_aux para el producto ' + @sProducto
-- Carga los datos en la tabla de trabajo (tipo_calc = 0 significa que no son calculados)
INSERT #temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo,
tipo_calc)
SELECT carga, entidad, indicador, segmento, tipo, valor, fecha, periodo, 0
FROM calculos_insertados WITH (NOLOCK)
WHERE carga = @nCarga
UNION
SELECT carga, entidad, indicador, segmento, tipo, valor, fecha, periodo, 0
FROM calculos_modificados WITH (NOLOCK)
WHERE carga = @nCarga
IF (@@error <> 0)
    GOTO err_calc

SELECT TOP 1 @sAño = año
FROM calculos_insertados WITH (NOLOCK)
WHERE carga = @nCarga
IF (@@error <> 0)
    GOTO err_calc

IF @sAño IS NULL
BEGIN
    SELECT TOP 1 @sAño = año
    FROM calculos_modificados WITH (NOLOCK)
    WHERE carga = @nCarga
    IF (@@error <> 0)
        GOTO err_calc
END

SELECT @sTabDatos = 'datos_' + CASE @sPeriodicidad WHEN 'M' THEN 'mes_' WHEN 'S' THEN 'sem_' END +
@sAño

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Obtiene producto inicial')
SELECT @sError = 'Error al obtener el producto'

```

Apéndice C. Código de los procesos

```
SELECT TOP 1 @sProducto = r.producto, @sUsuario = r.usuario
  FROM #temporal_proceso t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #cat_tipos ct WITH
(NOLOCK)
  WHERE t.indicador = r.indicador
        AND t.tipo = ct.tipo
        AND r.tipo = ct.tipo_base
        AND r.periodicidad = @sPeriodicidad
IF (@@error <> 0)
  GOTO err_calc

-- Obtiene el correo electrónico del usuario responsable
SELECT @sMail = e_mail FROM productos.dbo.cat_usuarios_producto WITH (NOLOCK) WHERE id_responsable =
@sUsuario

-- Inicialización de valores
SELECT @sError = 'Error al obtener el año'
SELECT --@sAño = CASE @sPeriodicidad WHEN 'M' THEN left(fecha, 4) WHEN 'S' THEN '20' + left(fecha,
2) END,
        @nPeriodoMin = min(periodo)
  FROM #temporal_proceso WITH (NOLOCK) GROUP BY fecha
IF (@@error <> 0)
  GOTO err_calc

--
=====
----
-- SECCION DE CALCULOS
-----
-- Ejecuta el proceso de CALCULOS
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'ENTRA a sp_calculo_recurativo_2' )

SELECT @sError = 'Error al ejecutar sp_calculo_recurativo_2'
EXEC @sp_recurativo = sp_calculo_recurativo @nCarga, @sPeriodicidad, @sProducto, @sAño, 1
IF (@@error <> 0 OR @sp_recurativo < 0)
  GOTO err_calc
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'SALIO de todos los sp_calculo_recurativo_2' )

--
=====
----
-- Se eliminan los indicadores calculados duplicados de menor nivel. Estos se generan cuando una
carga tiene un
-- elemento y busca el otro en la base de datos, pero este último resulta que se calcula en otro
nivel, por lo
-- tanto vuelve a generar otro calculado y por eso se duplican
SELECT entidad, indicador, segmento, tipo, periodo, max(nivel_calc) as nivel_calc
  INTO #temporal_proceso_total_pivote
  FROM #temporal_proceso_total WITH (NOLOCK)
  GROUP BY entidad, indicador, segmento, tipo, periodo
IF (@@error <> 0)
  GOTO err_calc

-- Se genera una tabla que contendrá los valores finales
SELECT DISTINCT t.entidad, t.indicador, t.segmento, t.tipo, t.valor, t.periodo, t.fecha
  INTO #temporal_proceso_final
  FROM #temporal_proceso_total t WITH (NOLOCK), #temporal_proceso_total_pivote p WITH (NOLOCK)
  WHERE t.entidad = p.entidad
        AND t.indicador = p.indicador
        AND t.segmento = p.segmento
        AND t.tipo = p.tipo
        AND t.periodo = p.periodo
        AND t.nivel_calc = p.nivel_calc
IF (@@error <> 0)
  GOTO err_calc

-- Inicializa variable diciendo que enviará correo electrónico
SET @nEnviarEMail = 1
--
=====
----
IF EXISTS(SELECT * FROM sysobjects WITH (NOLOCK) WHERE name = 'bcp_calculados' and type = 'U')
  DROP TABLE bcp_calculados

CREATE TABLE bcp_calculados(
  entidad      varchar(10),
  indicador    varchar(10),
```

```

        segmento    varchar(2),
        tipo        char(1),
        valor       float,
        fecha       varchar(10)
    )
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Antes de entrar al cursor de productos' )
-- Se generan los archivos de los indicadores calculados cuyo producto es distinto al inicial
DECLARE cursor_producto CURSOR LOCAL FOR
-- Se obtiene los productos de los indicadores calculados
SELECT DISTINCT r.producto, min(t.periodo)
FROM #temporal_proceso_final t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK),
#cat_tipos ct WITH (NOLOCK)
WHERE t.indicador = r.indicador
AND t.tipo = ct.tipo
AND r.tipo = ct.tipo_base
AND r.periodicidad = @sPeriodicidad
AND r.producto <> @sProducto
GROUP BY r.producto
OPEN cursor_producto
FETCH NEXT FROM cursor_producto INTO @sProducto_2, @nPeriodoMin
WHILE @@FETCH_STATUS = 0
BEGIN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Genera el archivo para el producto ' + @sProducto_2 )
SELECT @sError = 'Error al truncar la tabla bcp_calcula
-- Se eliminan todos los datos para procesar otro producto
TRUNCATE TABLE bcp_calculados
IF (@@error <> 0)
GOTO err_cursor

SELECT @sError = 'Error al obtener el nombre de archivo'
-- Obtiene el nombre del archivo del producto
SELECT @sArchivo = rtrim(nombre_archivo) +
CASE @sPeriodicidad WHEN 'M' THEN '_' WHEN 'S' THEN '_s' END +
right(@sAño, 2) + right('00' + convert(varchar(5), @nPeriodoMin), 2) +
'@' + convert(varchar(10), @nCarga) + '.' + rtrim(extension)
FROM productos.dbo.cat_productos WITH (NOLOCK)
WHERE id_producto = @sProducto_2
IF (@@error <> 0)
GOTO err_cursor

SELECT @sError = 'Error al insertar datos en bcp_calculados'
-- Genera tabla auxiliar para generar archivo, ya que no se puede directamente desde una temporal
INSERT bcp_calculados (entidad, indicador, segmento, tipo, valor, fecha)
SELECT DISTINCT rtrim(t.entidad), rtrim(t.indicador), rtrim(t.segmento), t.tipo, t.valor,
rtrim(t.fecha)
FROM #temporal_proceso_final t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #cat_tipos
ct WITH (NOLOCK)
WHERE t.indicador = r.indicador
AND t.tipo = ct.tipo
AND r.tipo = ct.tipo_base
AND r.producto = @sProducto_2
AND r.periodicidad = @sPeriodicidad
IF (@@error <> 0)
GOTO err_cursor

-- Manda generar los periodos de control de los productos
SET @sPeriodo1 = right(@sAño, 2) + right('00' + convert(varchar(5), @nPeriodoMin), 2)
SET @sPeriodo2 = 'S' + right(@sAño, 2) + right('00' + convert(varchar(5), @nPeriodoMin), 2)
IF @sPeriodicidad = 'M'
BEGIN
EXEC srvscil.productos.dbo.sp_crea_periodo_mensual_SCI @sProducto_2, @sPeriodo1
IF (@@error <> 0)
GOTO err_cursor
END
ELSE IF @sPeriodicidad = 'S'
BEGIN
EXEC srvscil.productos.dbo.sp_crea_periodo_semanal_SCI @sProducto_2, @sPeriodo2
IF (@@error <> 0)
GOTO err_cursor
END

SELECT @sError = 'Error al ejecutar el BCP'

-- Realiza el BCP para generar el archivo
SET @sSql = 'BCP "SCI.dbo.bcp_calculados" OUT "' + '\\Srvscil\Repositorio\Usuarios\hpalafox\scp\'
+

```

Apéndice C. Código de los procesos

```
                @sArchivo + '" -Ssrvscil -Usa -Pcontraseña -c '
EXEC @result = master.dbo.xp_cmdshell @sSql, NO_OUTPUT
IF (@@error <> 0)
    GOTO err_cursor

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Generó el archivo ' + @sArchivo )
-- Indica que no tendrá que enviar correo electrónico
SET @nEnviarEMail = 1

        FETCH NEXT FROM cursor_producto INTO @sProducto_2, @nPeriodoMin
END -- WHILE @@FETCH_STATUS = 0
CLOSE cursor_producto
DEALLOCATE cursor_producto

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Salió del cursor de productos' )

        SELECT @sError = 'Error al eliminar los indicadores de diferente producto de la tabla
#temporal_proceso_final'
-- Se eliminan de la tabla de trabajo los indicadores que sean de productos diferentes al inicial,
porque ya se
-- generaron sus respectivos archivos
DELETE #temporal_proceso_final
        FROM #temporal_proceso_final t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #cat_tipos ct
WITH (NOLOCK)
        WHERE t.indicador = r.indicador
        AND t.tipo = ct.tipo
        AND r.tipo = ct.tipo_base
        AND r.periodicidad = @sPeriodicidad
        AND r.producto <> @sProducto
IF (@@error <> 0)
    GOTO err_calc

--
=====
-- SE GENERAN TABLAS #MODIFICADOS E #INSERTADOS PARA USARLAS EN LA GENERACION DE LAS CIFRAS DE
CONTROL
-----

-- Se eliminan los indicadores cuyo valor es el mismo al que se quiere insertar
SET @sSql = '
DELETE #temporal_proceso_final
        FROM ' + @sTabDatos + ' d WITH (NOLOCK), #temporal_proceso_final t WITH (NOLOCK)
        WHERE d.entidad = t.entidad
        AND d.indicador = t.indicador
        AND d.segmento = t.segmento
        AND d.tipo = t.tipo
        AND d.periodo = t.periodo
        AND round(d.valor, 2) = round(t.valor, 2) '
EXEC(@sSql)
IF (@@error <> 0)
    GOTO err_calc

-- Se crea una tabla auxiliar que contenga todos los registros de #temporal_proceso_final
SELECT * INTO #temporal_proceso_final_aux FROM #temporal_proceso_final WITH (NOLOCK)
IF (@@error <> 0)
    GOTO err_calc

-- Se generar los registros a modificar en la tabla #modificados
SET @sSql = '
INSERT #modificados (carga, entidad, indicador, segmento, tipo, valor, fecha, año, periodo,
nivel, valor_anterior)
        SELECT ' + convert(varchar(10), @nCarga) + ', t.entidad, t.indicador, t.segmento, t.tipo,
t.valor,
                t.fecha, ' + @sAño + ', t.periodo, e.nivel, d.valor
        FROM ' + @sTabDatos + ' d WITH (NOLOCK), #temporal_proceso_final_aux t WITH (NOLOCK),
        cat_entidades e WITH (NOLOCK)
        WHERE d.entidad = t.entidad
        AND d.indicador = t.indicador
        AND d.segmento = t.segmento
        AND d.tipo = t.tipo
        AND d.periodo = t.periodo
        AND round(d.valor, 2) = round(t.valor, 2)
        AND d.entidad = e.entidad '
EXEC(@sSql)
IF (@@error <> 0)
```

```

GOTO err_calc

-- Se eliminan de #temporal_proceso_final_aux los registros que se generaron como modificables
DELETE #temporal_proceso_final_aux
FROM #temporal_proceso_final_aux t WITH (NOLOCK), #modificados m WITH (NOLOCK)
WHERE t.entidad = m.entidad
      AND t.indicador = m.indicador
      AND t.segmento = m.segmento
      AND t.tipo = m.tipo
      AND t.periodo = m.periodo
IF (@@error <> 0)
GOTO err_calc

-- Se insertan los datos nuevos en #insertados
INSERT #insertados (carga, entidad, indicador, segmento, tipo, valor, fecha, año, periodo, nivel)
SELECT @nCarga, t.entidad, t.indicador, t.segmento, t.tipo, t.valor, t.fecha, @sAño, t.periodo,
e.nivel
FROM #temporal_proceso_final_aux t WITH (NOLOCK), cat_entidades e WITH (NOLOCK)
WHERE t.entidad = e.entidad
IF (@@error <> 0)
GOTO err_calc
--
=====
=====

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Inicia TRANSACCION final' )
--
=====
=====

-- SE ACTUALIZAN LAS TABLAS DONDE SE OBTUVIERON LOS DATOS Y SE CARGAN LOS CALCULADOS
-----
-----
IF EXISTS (SELECT * FROM #temporal_proceso_final WITH (NOLOCK) )
BEGIN
BEGIN TRAN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Elimina de la tabla datos_mmm_yyyy' )
SELECT @sError = 'Error al eliminar los indicadores de la tabla datos_mmm_yyyy'
-- Elimina los datos que se van a insertar
SET @sSql = '
DELETE ' + @sTabDatos + '
FROM ' + @sTabDatos + ' d WITH (NOLOCK), #temporal_proceso_final t WITH (NOLOCK)
WHERE d.entidad = t.entidad
      AND d.indicador = t.indicador
      AND d.segmento = t.segmento
      AND d.tipo = t.tipo
      AND d.periodo = t.periodo '
EXEC(@sSql)
IF (@@error <> 0)
GOTO err_tran

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Inserta en bitacora_carga AA' )
SELECT @sError = 'Error al generar número de carga en la tabla bitacora_carga'
-- Genera un nuevo número de carga
INSERT bitacora_carga (producto, fecha_carga, archivo, periodicidad, status_proceso, usuario,
enviar_email)
VALUES(@sProducto, getdate(), '@@CALCULADOS de la carga ' + convert(varchar(10), @nCarga),
@sPeriodicidad, 'AA', @sUsuario, @nEnviarEMail)
IF (@@error <> 0)
GOTO err_tran

SELECT @sError = 'Error al obtener el nuevo número de carga'
-- Obtiene el nuevo número de carga
SELECT @nCargaNueva = max(carga)
FROM bitacora_carga WITH (NOLOCK)
WHERE periodicidad = @sPeriodicidad
      AND status_proceso = 'AA'
      AND archivo = '@@CALCULADOS de la carga ' + convert(varchar(10), @nCarga)
IF (@@error <> 0)
GOTO err_tran

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Inserta en datos_mmm_yyyy' )
SELECT @sError = 'Error al insertar los indicadores calculados en datos_mmm_yyyy'
-- Se insertan los indicadores calculados del mismo producto que el inicial en la tabla
datos_mmm_yyyy

```

Apéndice C. Código de los procesos

```
SET @sSql = '
    INSERT ' + @sTabDatos + ' (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo)
    SELECT DISTINCT ' + convert(varchar(10), @nCargaNueva) + ', entidad, indicador, segmento,
tipo,
        valor, fecha, periodo
    FROM #temporal_proceso_final WITH (NOLOCK) '
EXEC(@sSql)
IF (@@error <> 0)
    GOTO err_tran

COMMIT TRAN

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Inserta en temporal_proceso' )
SELECT @sError = 'Error al insertar los indicadores calculados en temporal_proceso'
-- Se insertan los indicadores calculados del mismo producto que el inicial en la tabla
temporal_proceso_total
-- para que funcione el llamado de la generación de vistas.
INSERT temporal_proceso (carga, entidad, indicador, segmento, tipo, valor, fecha, año, periodo,
nivel)
SELECT @nCargaNueva, t.entidad, t.indicador, t.segmento, t.tipo, t.valor, t.fecha, @sAño,
t.periodo, e.nivel
FROM #temporal_proceso_final t WITH (NOLOCK), cat_entidades e WITH (NOLOCK)
WHERE t.entidad = e.entidad
IF (@@error <> 0)
    GOTO err_calc

--
=====
-- SECCION DE CIFRAS DE CONTROL
-----

SET @sError = 'Error al insertar en la tabla cifras_control'
INSERT cifras_control (carga, año, periodicidad, periodo, indicador, tipo, nivel,
insertados, modificados, nullos, ceros, total, renglones)
SELECT @nCargaNueva, @sAño, @sPeriodicidad, t.periodo, t.indicador, t.tipo, t.nivel,
(SELECT IsNull(Count(*), 0) FROM #insertados WITH (NOLOCK)
WHERE t.carga = carga AND t.indicador = indicador AND t.tipo = tipo AND t.nivel = nivel
AND t.periodo = periodo),
(SELECT IsNull(Count(*), 0) FROM #modificados WITH (NOLOCK)
WHERE t.carga = carga AND t.indicador = indicador AND t.tipo = tipo AND t.nivel = nivel
AND t.periodo = periodo),
(SELECT IsNull(Count(*), 0) FROM temporal_proceso WITH (NOLOCK)
WHERE valor IS NULL AND t.carga = carga AND t.indicador = indicador AND t.tipo = tipo
AND t.nivel = nivel AND t.periodo = periodo),
(SELECT IsNull(Count(*), 0) FROM temporal_proceso WITH (NOLOCK)
WHERE valor = 0 AND t.carga = carga AND t.indicador = indicador AND t.tipo = tipo AND
t.nivel = nivel AND t.periodo = periodo),
(SELECT IsNull(Count(*), 0) FROM temporal_proceso WITH (NOLOCK)
WHERE t.carga = carga AND t.indicador = indicador AND t.tipo = tipo AND t.nivel = nivel
AND t.periodo = periodo),
(SELECT IsNull(Count(*), 0) FROM temporal_proceso WITH (NOLOCK)
WHERE t.carga = carga)
FROM temporal_proceso t WITH (NOLOCK)
WHERE t.carga = @nCargaNueva
GROUP BY t.carga, t.indicador, t.tipo, t.nivel, t.periodo
ORDER BY t.carga, t.indicador, t.tipo, t.nivel, t.periodo
IF @@ERROR <> 0
    GOTO err_calc

--
=====

SELECT @sError = 'Error al actualizar bitacora_carga con CV de la nueva carga'
-- Actualiza el estado de la bitacora carga
UPDATE bitacora_carga SET status_proceso = 'CV' WHERE carga = @nCargaNueva
IF (@@error <> 0)
    GOTO err_calc
END -- IF EXISTS (SELECT * FROM #temporal_proceso_final)

SELECT @sError = 'Error al actualizar bitacora_carga con CV de la carga original'
-- Actualiza el estado de la bitacora carga
UPDATE bitacora_carga SET status_proceso = 'CV' WHERE carga = @nCarga
IF (@@error <> 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Elimina de la tabla calculos_insertados' )
SELECT @sError = 'Error al eliminar datos de calculos_insertados'
```

```

-- Elimina los indicadores de la carga procesada
DELETE FROM calculos_insertados WHERE carga = @nCarga
IF (@@error <> 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Elimina de la tabla calculos_modificados' )
SELECT @sError = 'Error al eliminar datos de calculos_modificados'
DELETE FROM calculos_modificados WHERE carga = @nCarga
IF (@@error <> 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Elimina de la tabla indicadores_recalculados' )
SELECT @sError = 'Error al eliminar datos de indicadores_recalculados'
DELETE indicadores_recalculados WHERE carga = @nCarga
IF (@@error <> 0)
    GOTO err_calc
--
=====
=====

SELECT @nuProcesados = count(*) FROM #temporal_proceso_final WITH (NOLOCK)
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Se procesaron ' + convert(varchar(10), @nuProcesados) + ' elementos ' )

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'TERMINA sp_Principal_Calculos' )

-- Verifica si la carga procesada era de algún producto de LATAM
IF EXISTS (SELECT *
           FROM #temporal_proceso_final t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK),
           cat_productos_latam p WITH (NOLOCK)
           WHERE t.indicador = r.indicador AND t.tipo = r.tipo AND r.periodicidad = @sPeriodicidad
           AND r.producto = p.producto AND p.activo = 1 )
BEGIN
    INSERT bitacora_productos_latam (carga, producto, fecha_proceso, periodicidad, año, status)
    VALUES (@nCargaNueva, @sProducto, getdate(), @sPeriodicidad, @sAño, 1)
END

-- ===== INICIA SECCION DE ENVIO DE CORREOS
=====
IF EXISTS (SELECT * FROM bitacora_diferencias_calc_msd WITH (NOLOCK) WHERE carga = @nCarga AND
periodicidad = @sPeriodicidad)
BEGIN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), 'Envía correo de diferencias de cálculos.')
SELECT @sMensaje = 'La carga ' + convert(varchar(10), @nCarga) + ' ha encontrado diferencias en
los indicadores calculados.' + char(13) +
                'Los valores calculados por el sistema fueron sustituidos por los cargados.',
@sSubject = 'Cálculos SCI (' + @sPeriodicidad + ')_Informativo_Diferencias_' +
'Diferencias en cálculos de la carga ' + convert(varchar(10), @nCarga),
@sSql = 'SELECT entidad, indicador, segmento, tipo, periodo, fecha, valor_carg, valor_calc
' +
                'FROM sci.dbo.bitacora_diferencias_calc_msd WITH (NOLOCK) ' +
                'WHERE carga = ' + convert(varchar(10), @nCarga) + ' AND periodicidad = ' +
char(39) + @sPeriodicidad + char(39)

SELECT @sError = 'Error al enviar el correo electrónico'
EXECUTE master..xp_sendmail @recipients = @sMail,
                            @message = @sMensaje,
                            @query = @sSql,
                            @copy_recipients = 'jdelaveg@telmex.com; acelada@telmex.com;
psolis@telmex.com',
                            @blind_copy_recipients = '',
                            @subject = @sSubject,
                            @attach_results = 'TRUE',
                            @width = 250

IF (@@error <> 0)
    GOTO err_calc

SELECT @sError = 'Error al eliminar datos de bitacora_diferencias_calc_msd'
DELETE bitacora_diferencias_calc_msd WHERE carga = @nCarga AND periodicidad = @sPeriodicidad
IF (@@error <> 0)
    GOTO err_calc
END
-- ===== TERMINA SECCION DE ENVIO DE CORREOS
=====

```

Apéndice C. Código de los procesos

```
END -- IF @nCarga > 0

RETURN 0
err_cursor:
    CLOSE cursor_producto
    DEALLOCATE cursor_producto
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), @sError )
    EXEC sp_sendMail_error_2 @nCarga, @sError, 'jdelaveg@telmex.com', 'psolis@telmex.com;
acelada@telmex.com; jgteran@telmex.com', 'Cálculo'
    RETURN -1
err_tran:
    ROLLBACK TRAN
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), @sError )
    EXEC sp_sendMail_error_2 @nCarga, @sError, 'jdelaveg@telmex.com', 'psolis@telmex.com;
acelada@telmex.com; jgteran@telmex.com', 'Cálculo'
    RETURN -2
err_calc:
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@nCarga, @sPeriodicidad,
getdate(), @sError )
    EXEC sp_sendMail_error_2 @nCarga, @sError, 'jdelaveg@telmex.com', 'psolis@telmex.com;
acelada@telmex.com; jgteran@telmex.com', 'Cálculo'
    RETURN -3

END -- CREATE PROCEDURE sp_Principal_Calculos
```

C.5 Recursividad de cálculos.

```

IF EXISTS (SELECT name FROM sysobjects WITH (NOLOCK) WHERE name = 'sp_calculo_recursivo' AND type = 'P')
  DROP PROCEDURE sp_calculo_recursivo
GO
--
*****
-- Nombre:          sp_calculo_recursivo
-- Descripción:     A partir de una carga de datos, hace el llamado del proceso de generación de valores de
los
--                  indicadores calculados. Además agrupa el resultado de los calculados que pertenecen
al mismo
--                  producto para que se inserten en datos_mmm_yyyy. Los indicadores que no pertenecen al
mismo producto
--                  se les genera un archivo para ser insertados como nueva carga.
--
-- Parametros:
-- Return:          -1:Error, 0:Exito
--
*****
CREATE PROCEDURE sp_calculo_recursivo
  @pnCarga          int,
  @psPeriodicidad  char(1),
  @psProducto       varchar(15),
  @psAño            char(4),
  @pnRecursividad  smallint
AS
BEGIN
  DECLARE   @sError      varchar(150), @nError int,
            @nAcumula    smallint,
            @nCalcula    smallint,
            @nRecursivo  smallint

  -- Verifica si hay indicadores raices cuyo tipo y periodicidad pertenezca a una fórmula
  WHILE EXISTS (SELECT *
                FROM #temporal_proceso t WITH (NOLOCK), cat_calculado_raiz_anidado c WITH (NOLOCK)
                WHERE t.indicador = c.indicador_raiz
                   AND t.tipo = c.tipo_raiz
                   AND c.periodicidad = @psPeriodicidad )
  BEGIN
    -- Obtiene los indicadores que son del producto
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Entra al proceso cálculos')
    SELECT @sError = 'Error al ejecutar sp_Calculos'
    EXEC @nCalcula = sp_Calculos @pnCarga, @psPeriodicidad, @psAño, 0
    SET @nError = @nCalcula
    IF (@nError <> 0)
      GOTO err_recursivo

    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Eliminando #temporal_proceso donde tipo_calc = 0')
    SELECT @sError = 'Error al eliminar los registros de #temporal_proceso donde tipo_calc = 0'
    -- Elimina los indicadores raiz y deja los calculados
    DELETE #temporal_proceso WHERE tipo_calc = 0
    IF (@@error <> 0)
      GOTO err_recursivo

    SELECT @sError = 'Error al actualizar los calculados de #temporal_proceso con tipo_calc = 0'
    -- Actualiza el tipo de cálculo a cero para que después realice la agregación y la acumulación
    UPDATE #temporal_proceso SET tipo_calc = 0
    IF (@@error <> 0)
      GOTO err_recursivo

    IF @psPeriodicidad <> 'D'
    BEGIN
      SELECT @sError = 'Error en la ejecución de sp_Acumulacion'
      -- Ejecuta el proceso de ACUMULACIONES
      insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Entra al proceso de acumulación de calculados')
      EXEC @nAcumula = sp_Acumulacion @psPeriodicidad, @psAño, 0
      IF (@@error <> 0 OR @nAcumula < 0)

```

Apéndice C. Código de los procesos

```
        GOTO err_recur_sivo
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'sale del proceso de acumulación de calculados')
END

-- Se verifica si se generaron indicadores calculados
IF EXISTS (SELECT * FROM #temporal_proceso WITH (NOLOCK))
BEGIN
    -- Se eliminan los valores existentes para sustituirse por los nuevos valores generados.
    DELETE #temporal_proceso_total
    FROM #temporal_proceso_total tt WITH (NOLOCK), #temporal_proceso t WITH (NOLOCK)
    WHERE tt.entidad = t.entidad
        AND tt.indicador = t.indicador
        AND tt.segmento = t.segmento
        AND tt.tipo = t.tipo
        AND tt.periodo = t.periodo

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Insertando en #temporal_proceso_total')
    SELECT @sError = 'Error al insertar calculados en #temporal_proceso_total'
    -- Todos los indicadores (mismo producto o diferente) se pasan a la tabla #temporal_proceso_total
    INSERT #temporal_proceso_total (carga, entidad, indicador, segmento, tipo, valor, fecha, periodo,
tipo_calc, nivel_calc)
    SELECT carga, entidad, indicador, segmento, tipo, valor, fecha, periodo, tipo_calc,
@pnRecur_sividad
    FROM #temporal_proceso t WITH (NOLOCK)
    IF (@@error <> 0)
        GOTO err_recur_sivo

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Eliminando de #temporal_proceso los indicadores de producto distinto al original')
    SELECT @sError = 'Error eliminar de #temporal_proceso los calculados cuyo producto es distinto al
original'
    -- Se eliminan los indicadores de diferentes productos al inicial para que procese el siguiente
nivel de anidamiento
    DELETE #temporal_proceso
    FROM #temporal_proceso t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #cat_tipos ct WITH
(NOLOCK)
    WHERE t.indicador = r.indicador
        AND t.tipo = ct.tipo
        AND r.tipo = ct.tipo_base
        AND r.periodicidad = @psPeriodicidad
        AND r.producto <> @psProducto
    IF (@@error <> 0)
        GOTO err_recur_sivo

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Eliminando de #temporal_proceso los indicadores de tipos distintos de M y P')
    SELECT @sError = 'Error eliminar de #temporal_proceso los indicadores de tipo distintos de M y P'
    -- Se eliminan los acumulados para que en la siguiente pasada no se tomen en cuenta dentro de los
cálculos
    DELETE #temporal_proceso WHERE tipo NOT IN ('M', 'P')
    IF (@@error <> 0)
        GOTO err_recur_sivo

    SELECT @sError = 'Error eliminar de #temporal_proceso los indicadores de tipo_calc = 0'
    -- Actualiza el estado para que se procese nuevamente los cálculos
    UPDATE #temporal_proceso SET tipo_calc = 0
    IF (@@error <> 0)
        GOTO err_recur_sivo

-----
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), convert(varchar(5), @pnRecur_sividad) + ' veces entrando a sp_calculo_recur_sivo')
    SELECT @sError = 'Error al ejecutar sp_calculo_recur_sivo'
    -- Ejecuta recursivamente el proceso de calculos anidados.
    SET @pnRecur_sividad = @pnRecur_sividad + 1
    EXEC @nRecur_sivo = sp_calculo_recur_sivo @pnCarga, @psPeriodicidad, @psProducto, @psAño,
@pnRecur_sividad
    IF (@@error <> 0 OR @nRecur_sivo < 0)
        GOTO err_recur_sivo
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), ' Saliendo de ' + convert(varchar(5), @pnRecur_sividad) + ' sp_calculo_recur_sivo')
-----

END -- IF EXISTS (SELECT * FROM #temporal_proceso)

END -- WHILE EXISTS (SELECT * FROM #temporal_proceso WHERE tipo_calc = 30)
```

```
RETURN 0

err_recur_sivo:
    SELECT TOP 1 @sError = @sError + ' (' + isnull(description, '') + ') . ' FROM master.dbo.sysmessages
WITH (NOLOCK) WHERE error = @nError
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), @sError )
EXEC sp_sendMail_error_2 @pnCarga, @sError, 'jdelaveg@telmex.com', 'psolis@telmex.com;
acelada@telmex.com', 'Cálculo'
RETURN -1

END -- CREATE PROCEDURE sp_calculo_recur_sivo
```

C.6 Proceso de cálculos.

```

IF EXISTS (SELECT name FROM sysobjects WITH (NOLOCK) WHERE name = 'sp_Calculos' AND type = 'P')
    DROP PROCEDURE sp_Calculos
GO
--
*****
-- Nombre:          sp_Calculos
-- Descripción:     Obtiene el valor de los indicadores calculados a partir de los indicadores raíz. Funciona
para
--                 indicadores raiz del mismo tipo y periodo que el calculado, así como indicadores raíz de
tipo y
--                 periodo distintos que el calculado.
-- Parametros:
-- Return:          -1:Error, 0:Exito
--
*****
CREATE PROCEDURE sp_Calculos
    @pnCarga          int,
    @psPeriodicidad  char(1),
    @psAño            char(4),
    @pbComparar      bit -- 1: Solamente compara los valores cargados contra los calculados sin
sustituir los valores cargados
AS
BEGIN
    ----- INICIA SECCION DE DECLARACIONES -----
    -----
    DECLARE @indicador_calc  varchar(10), @tipo_calc char(1), @tipo_indicador char(1),
            @nCont smallint, @sCondicion varchar(100), @sIndicador varchar(10),
            @sCampos varchar(8000), @sCampos2 varchar(8000), @sFrom varchar(8000), @sWhere varchar(8000),
            @sFrom2 varchar(8000), @sFrom3 varchar(8000), @sFrom4 varchar(8000),
            @sIncador_calculado varchar(20), @sTabConstraint varchar(100),
            @sFecha varchar(200)

    DECLARE @año_calculado smallint, @periodo_calculado smallint, @entidad_calculado varchar(10),
            @tipo_calculado char(1), @segmento_calculado char(2), @fecha_calculado char(10)
    DECLARE @indicador varchar(25), @indicadores varchar(3000)
    DECLARE @calculado varchar(25)
    DECLARE @valor float, @valores varchar(8000)
    DECLARE @sSql varchar(8000)
    DECLARE @nError int, @sError varchar(150)
    DECLARE @nGetDatos smallint

    DECLARE @sCarga varchar(10),
            @sUsuario varchar(15),
            @sSubject varchar(100),
            @sMail varchar(50),
            @sMensaje varchar(200)

    DECLARE @col varchar(8000), @col2 varchar(8000), @columns varchar(8000), @columns2 varchar(8000),
            @columns3 varchar(8000), @columns4 varchar(8000), @columns5 varchar(8000)

    DECLARE @nuNivel smallint
    ----- TERMINA SECCION DE DECLARACIONES -----
    -----

    -- Establece el formato de fecha
    SET DATEFORMAT mdy

    CREATE TABLE #temporal_calculo(
        indicador_calc          char(10),
        tipo_calc               char(1),
        entidad_calc            char(10),
        segmento_calc           char(2),
        año_calc                 smallint,
        periodo_calc            smallint,
        fecha_calc              char(10),
        indicador_raiz          char(10),
        tipo_raiz                char(1),
        periodo_relativo_raiz   smallint,
        valor_raiz               float,

```

```

        nivel                smallint
    )
    CREATE UNIQUE INDEX ix_temporal_calculo On #temporal_calculo
        (indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc, periodo_calc,
        indicador_raiz, tipo_raiz, periodo_relativo_raiz)

-- Lista de indicadores que se tienen que calcular
CREATE TABLE #calculados(
    indicador_calc          char(10),
    tipo_calc              char(1),
    entidad_calc           char(10),
    segmento_calc          char(2),
    año_calc               smallint,
    periodo_calc           smallint,
    fecha_calc             char(10),
    nivel                  smallint
)

-- Lista de indicadores que se tienen que calcular
CREATE TABLE #calculados_aux(
    indicador_calc          char(10),
    tipo_calc              char(1),
    entidad_calc           char(10),
    segmento_calc          char(2),
    año_calc               smallint,
    periodo_calc           smallint,
    fecha_calc             char(10)
)

-- Lista de indicadores raiz que se necesitan para el calculo
CREATE TABLE #raices(
    indicador_calc          char(10),
    tipo_calc              char(1),
    indicador_raiz          char(10),
    tipo_raiz              char(1),
    periodo_relativo_raiz  smallint,
    entidad                 char(10),
    año                     smallint,
    periodo                 smallint,
    segmento                 char(2),
    fecha                   char(10),
    nivel                   smallint
)

CREATE TABLE #raices_fijos(
    indicador_calc          char(10),
    tipo_calc              char(1),
    indicador_raiz          char(10),
    tipo_raiz              char(1),
    periodo_relativo_raiz  smallint,
    entidad                 char(10),
    año                     smallint,
    periodo                 smallint,
    segmento                 char(2),
    fecha                   char(10),
    nivel                   smallint
)

-- Tabla que almacena los resultados de los indicadores calculados
CREATE TABLE #indicadores_calculados(
    entidad                varchar(10),
    indicador               varchar(10),
    segmento                varchar(2),
    tipo                    char(1),
    valor                   float,
    fecha                   varchar(10),
    periodo                 smallint
)
CREATE INDEX ix_indica_calculados1 On #indicadores_calculados (entidad, indicador, segmento, tipo,
periodo, fecha)
CREATE INDEX ix_indica_calculados2 On #indicadores_calculados (entidad, indicador, segmento, tipo,
periodo)
CREATE INDEX ix_indica_calculados3 On #indicadores_calculados (entidad, indicador, segmento, periodo)
CREATE INDEX ix_indica_calculados4 On #indicadores_calculados (indicador, tipo)
CREATE INDEX ix_indica_calculados5 On #indicadores_calculados (entidad)
CREATE INDEX ix_indica_calculados6 On #indicadores_calculados (tipo)

CREATE TABLE #tabla_pivote(

```

Apéndice C. Código de los procesos

```
        año          smallint      NOT NULL,
        entidad      varchar(10)   NOT NULL,
        tipo         char(1)     NOT NULL,
        periodo      smallint     NOT NULL,
        segmento     char(2)      NOT NULL,
        fecha        char(10)    NOT NULL
    )
    CREATE INDEX ix_tabla_pivote1 On #tabla_pivote (año, entidad, tipo, periodo, segmento, fecha)
    CREATE INDEX ix_tabla_pivote2 On #tabla_pivote (año, entidad, tipo, periodo, segmento)

    IF NOT EXISTS (SELECT name FROM sysobjects WITH (NOLOCK) WHERE name = 'bitacora_diferencias_calc_msd'
    AND type = 'U')
    BEGIN
        CREATE TABLE bitacora_diferencias_calc_msd(
            carga          int,
            periodicidad   char(1),
            entidad        varchar(10),
            indicador      varchar(10),
            segmento       varchar(2),
            tipo           char(1),
            periodo        smallint,
            fecha          varchar(10),
            tipo_calc       varchar(8),
            valor_carg     float,
            valor_calc     float
        )
        CREATE INDEX ix_bitacora_diferencias_calc_msd On bitacora_diferencias_calc_msd (carga)
    END
    DELETE bitacora_diferencias_calc_msd WHERE carga = @pnCarga AND periodicidad = @psPeriodicidad

    --
    =====
    ===
    -- Obtiene datos para realizar calculos
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
    getdate(), 'Entra a sp_getRootForCalc')
    SELECT @nError = -1, @sError = 'Error dentro del procedimiento almacenado sp_getRootForCalc'

    EXEC @nGetDatos = sp_getRootForCalc @pnCarga, @psPeriodicidad, @psAño
    SET @nError = @@error
    IF (@nError <> 0 OR @nGetDatos < 0)
        GOTO err_error

    --
    =====
    ===

    IF EXISTS (SELECT * FROM #temporal_calculo WITH (NOLOCK))
    BEGIN
        --
        =====
        ===
        ----- INICIA CREACION DE TABLA TEMPORAL PARA CALCULOS -----
        -----
        -----
        CREATE TABLE #tabla_datos(
            col          varchar(7800),
            col2         varchar(100),
            sIndicador   varchar(10),
            tipo_indicador varchar(1),
            sCondicion   varchar(100)
        )

        -- Crea la tabla dinámica que realizará los cálculos
        SET @sCarga = convert(varchar(10), @pnCarga)
        -- Obtiene el usuario responsable
        SELECT TOP 1 @sUsuario = usuario
        FROM #temporal_proceso t WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK), #cat_tipos ct WITH
        (NOLOCK)
        WHERE t.indicador = r.indicador
            AND t.tipo = ct.tipo
            AND r.tipo = ct.tipo_base
            AND r.periodicidad = @psPeriodicidad

        -- Obtiene el correo electrónico del usuario responsable
        SELECT @sMail = e_mail FROM productos.dbo.cat_usuarios_producto WITH (NOLOCK) WHERE id_responsable =
        @sUsuario
```

```

-- Se utiliza al momento de pasar los datos a #temporal_calculos. La fecha que se debe poner es la
fecha del
-- calculado, para esto se debe sumar el periodo relativo a la fecha del resultado. Con esto nos da
la fecha
-- del calculado
IF @psPeriodicidad = 'M'
    SELECT @sFecha = 'convert(varchar(10), dateadd(dd, -1, dateadd(mm, 1, ' + char(39) + @psAño + '-'
+ char(39) + ' + convert(varchar(5), periodo) + ' + char(39) + '-01' + char(39) + ')), 126) AS fecha_calc'
ELSE IF @psPeriodicidad = 'S'
    SELECT @sFecha = 'fecha as fecha_calc'
ELSE IF @psPeriodicidad = 'D'
    SELECT @sFecha = 'fecha as fecha_calc'

--
=====
----
-- Se crea un cursor para los calculados (se creará una tabla por calculado)
-----

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Crear cursor cursor_calc')
DECLARE cursor_calc CURSOR LOCAL FOR
    SELECT DISTINCT rtrim(indicador_calc), tipo_calc, nivel
    FROM #temporal_calculo WITH (NOLOCK) ORDER BY nivel, rtrim(indicador_calc)
OPEN cursor_calc
FETCH NEXT FROM cursor_calc INTO @indicador_calc, @tipo_calc, @nuNivel
WHILE @@FETCH_STATUS = 0
BEGIN
    IF EXISTS(SELECT * FROM sysobjects WITH (NOLOCK) WHERE type = 'U' AND name =
't_calculos_indicadores_' + @sCarga + @psPeriodicidad)
    BEGIN
        SET @sSql = 'DROP TABLE t_calculos_indicadores_' + @sCarga + @psPeriodicidad
        EXEC(@sSql)
        SET @nError = @@error
        IF (@nError <> 0)
            GOTO err_cursor_calc
    END
    IF EXISTS(SELECT * FROM sysobjects WITH (NOLOCK) WHERE type = 'K' AND name = 'pk_t_calculos_' +
@sCarga + @psPeriodicidad)
    BEGIN
        SELECT @sTabConstraint = o2.name FROM sysobjects o1 WITH (NOLOCK) , sysobjects o2 WITH
(NOLOCK) WHERE o1.type = 'K' AND o1.parent_obj = o2.id AND o1.name = 'pk_t_calculos_' + @sCarga +
@psPeriodicidad
        SET @sSql = 'ALTER TABLE ' + char(39) + @sTabConstraint + char(39) + 'DROP CONSTRAINT
pk_t_calculos_' + @sCarga + @psPeriodicidad
        EXEC(@sSql)
        SET @nError = @@error
        IF (@nError <> 0)
            GOTO err_cursor_calc
    END
    SELECT @nError = -2, @sError = 'Error al crear la table t_calculos_indicadores_' + @sCarga +
@psPeriodicidad
    SET @sSql =
    CREATE TABLE t_calculos_indicadores_' + @sCarga + @psPeriodicidad + ' (
        [año]                smallint          NOT NULL,
        [entidad]            varchar(10)       NOT NULL,
        [tipo]                char(1)          NOT NULL,
        [periodo]            smallint         NOT NULL,
        [segmento]           char(2)          NOT NULL,
        [fecha]              char(10)         NOT NULL ) '
    EXEC(@sSql)
    SET @nError = @@error
    IF (@nError <> 0)
        GOTO err_cursor_calc

-- INICIA NOMBRES DE INDICADORES CALCULADOS
=====
-- Se crean las columnas calculadas de esta manera, debido a que si se crea dinámicamente la
tabla,
-- después SQL no reconoce a la tabla.

----- INICIA NOMBRES DE INDICADORES RAIZ -----
---
```

Apéndice C. Código de los procesos

```
-- Obtiene el nombre del indicador raiz, cuando es del mismo tipo y periodo que el calculado
TRUNCATE TABLE #tabla_datos

INSERT #tabla_datos (col, col2, sIndicador, tipo_indicador, sCondicion)
SELECT DISTINCT '[' + rtrim(indicador_raiz) + ']' float NULL, '[' + rtrim(indicador_raiz) +
']',
    rtrim(indicador_raiz), 'R', ' AND periodo_relativo_raiz = 0 AND tipo_calc = tipo_raiz'
FROM cat_calculado_raiz_anidado WITH (NOLOCK)
WHERE periodo_relativo_raiz = 0 AND tipo_calculado = tipo_raiz AND indicador_calculado =
@indicador_calc
AND tipo_calculado = @tipo_calc AND periodicidad = @psPeriodicidad
UNION
-- Obtiene el nombre del indicador raiz, cuando es de periodo distinto que el calculado _PN:
Cuando es
-- negativo _PP: Cuando se trata de un periodo fijo, por ejemplo 12(diciembre). Los nombres inician
con
-- 5 caracteres que lo identifican como periodo distinto. Ejem: _PN01 (periodo anterior). _PN03
-- (tres periodos anteriores)
SELECT DISTINCT CASE WHEN periodo_relativo_raiz < 0 THEN '['_PN' ELSE '['_PP' END +
    replace(space(2 - len(abs(periodo_relativo_raiz))) +
    convert(varchar(2), abs(periodo_relativo_raiz)), ' ', '0') + '_' +
    rtrim(indicador_raiz) + ']' float NULL, '['_PN' ELSE '['_PP' END +
    replace(space(2 - len(abs(periodo_relativo_raiz))) +
    convert(varchar(2), abs(periodo_relativo_raiz)), ' ', '0') + '_' +
    rtrim(indicador_raiz) + ']', rtrim(indicador_raiz), 'R',
    ' AND periodo_relativo_raiz <> 0 AND tipo_calc = tipo_raiz'
FROM cat_calculado_raiz_anidado WITH (NOLOCK)
WHERE periodo_relativo_raiz <> 0 AND tipo_calculado = tipo_raiz AND indicador_calculado =
@indicador_calc
AND tipo_calculado = @tipo_calc AND periodicidad = @psPeriodicidad
UNION
-- Obtiene el nombre del indicador raiz, cuando es de tipo distinto que el calculado. Los nombres
inician con 5
-- caracteres que lo identifican como periodo distinto. Ejem: _TIPA (tipo A)
SELECT DISTINCT '['_TIP' + tipo_raiz + '_' + rtrim(indicador_raiz) + ']' float NULL, '['_TIP' + tipo_raiz + '_' + rtrim(indicador_raiz) +
']', rtrim(indicador_raiz),
'R', ' AND periodo_relativo_raiz = 0 AND tipo_calc <> tipo_raiz'
FROM cat_calculado_raiz_anidado WITH (NOLOCK)
WHERE periodo_relativo_raiz = 0 AND tipo_calculado <> tipo_raiz AND indicador_calculado =
@indicador_calc
AND tipo_calculado = @tipo_calc AND periodicidad = @psPeriodicidad
UNION
----- TERMINA NOMBRES DE INDICADORES RAIZ -----
-----
----- INICIA NOMBRES DE INDICADORES CALCULADOS -----
-----
-- Obtiene el nombre y fórmula del indicador calculado cuando la fórmula es específica para cada
tipo. La ~ lo
-- identifica como formula específica para cierto tipo
SELECT DISTINCT '['~' + f.tipo + '_' + rtrim(f.indicador) + ']' AS '[' + f.formula_corta + ', ',
    '['~' + f.tipo + '_' + rtrim(f.indicador) + ']', null, 'C', ''
FROM cat_calculado_raiz_anidado c WITH (NOLOCK), cat_formulas f WITH (NOLOCK)
WHERE c.periodicidad = f.periodicidad AND c.indicador_calculado = f.indicador AND
c.tipo_calculado = f.tipo
AND c.periodicidad = @psPeriodicidad AND c.indicador_calculado = @indicador_calc
AND c.tipo_calculado = @tipo_calc
----- TERMINA NOMBRES DE INDICADORES CALCULADOS -----
-----

-- Limpia tabla
TRUNCATE TABLE #tabla_pivote
-- Inserta datos pivote en la tabla
INSERT #tabla_pivote (año, entidad, tipo, periodo, segmento, fecha)
SELECT DISTINCT año_calc, entidad_calc, tipo_calc, periodo_calc, segmento_calc, fecha_calc
FROM #temporal_calculo WITH (NOLOCK)
WHERE indicador_calc = @indicador_calc
AND tipo_calc = @tipo_calc
GROUP BY año_calc, entidad_calc, tipo_calc, periodo_calc, segmento_calc, fecha_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Crea cursor Columnas_Cursor')
SET @columns = 'ALTER TABLE t_calculos_indicadores_' + @sCarga + @psPeriodicidad + ' ADD' +
char(13)
SELECT @columns2 = ' ', @columns3 = ' ', @columns4 = ' ', @columns5 = ' '
DECLARE Columnas_Cursor CURSOR LOCAL FOR
SELECT DISTINCT col, col2, sIndicador, tipo_indicador, sCondicion
```

```

FROM #tabla_datos WITH (NOLOCK)
ORDER BY col2 DESC
OPEN Columnas_Cursor

SET @nCont = 1
FETCH NEXT FROM Columnas_Cursor INTO @col, @col2, @sIndicador, @tipo_indicador, @sCondicion
WHILE @@FETCH_STATUS = 0
BEGIN
    IF len(@columns) + len(@columns2) + len(@columns3) + len(@columns4) + len(@columns5) +
len(isnull(@col, '')) < 7900
        SET @columns = @columns + isnull(@col, '') + char(13) + ' '
    ELSE IF len(@columns) + len(@columns2) + len(@columns3) + len(@columns4) + len(@columns5) +
len(isnull(@col, '')) < 15900
        SET @columns2 = @columns2 + isnull(@col, '') + char(13) + ' '
    ELSE IF len(@columns) + len(@columns2) + len(@columns3) + len(@columns4) + len(@columns5) +
len(isnull(@col, '')) < 23900
        SET @columns3 = @columns3 + isnull(@col, '') + char(13) + ' '
    ELSE IF len(@columns) + len(@columns2) + len(@columns3) + len(@columns4) + len(@columns5) +
len(isnull(@col, '')) < 31900
        SET @columns4 = @columns4 + isnull(@col, '') + char(13) + ' '
    ELSE IF len(@columns) + len(@columns2) + len(@columns3) + len(@columns4) + len(@columns5) +
len(isnull(@col, '')) < 39900
        SET @columns5 = @columns5 + isnull(@col, '') + char(13) + ' '

-- Se crea una tabla por cada indicador raiz
IF @tipo_indicador = 'R'
BEGIN
--Arma la consulta de los indicadores pero de forma horizontal
IF @nCont = 1
BEGIN
    SET @sCampos = '
        SELECT t1.año, t1.entidad, t1.tipo, t1.periodo, t1.segmento, t1.fecha '
    SET @sCampos2 = 'año, entidad, tipo, periodo, segmento, fecha '
    SET @sFrom = '
        FROM #tabla_pivote t1 WITH (NOLOCK) '
    SET @sWhere = '
        WHERE t1.año = ' + @psAño
    END
    SET @nCont = @nCont + 1

    SELECT @nError = -3, @sError = 'Error al crear la tabla t' + @psPeriodicidad + '_' +
convert(varchar(5), @nCont)
    SET @sSql = '
        IF EXISTS(SELECT * FROM sysobjects WITH (NOLOCK) WHERE name = ' + char(39) + 't' +
@psPeriodicidad + '_' + convert(varchar(5), @nCont) + char(39) + ' and type = ' + char(39) + 'U' +
char(39) + ')
        DROP TABLE t' + @psPeriodicidad + '_' + convert(varchar(5), @nCont) + '
        SELECT año_calc, entidad_calc, tipo_calc, periodo_calc, segmento_calc, fecha_calc,
valor_raiz
        INTO t' + @psPeriodicidad + '_' + convert(varchar(5), @nCont) + '
        FROM #temporal_calculo WITH (NOLOCK)
        WHERE indicador_raiz = ' + char(39) + @sIndicador + char(39) + '
        AND indicador_calc = ' + char(39) + @indicador_calc + char(39) + '
        AND tipo_calc = ' + char(39) + @tipo_calc + char(39) + @sCondicion
diferente tipo IF charindex('[_TIP', @col2) > 0 -- Se utiliza para las fórmulas que tienen indicadores de
        SET @sSql = @sSql + '
        AND tipo_raiz = ' + char(39) + substring(@col2, 6, 1) + char(39)
diferente tipo IF charindex('[_PN', @col2) > 0 -- Se utiliza para las fórmulas que tienen indicadores de
        SET @sSql = @sSql + '
        AND periodo_relativo_raiz = -' + substring(@col2, 5, 2)
diferente tipo IF charindex('[_PP', @col2) > 0 -- Se utiliza para las fórmulas que tienen indicadores de
        SET @sSql = @sSql + '
        AND periodo_relativo_raiz = ' + substring(@col2, 5, 2)
EXEC(@sSql)
SET @nError = @@error
IF (@nError <> 0)
    GOTO err_Columnas_Cursor

    SET @sCampos = @sCampos + ', ' + ' t' + convert(varchar(5), @nCont) + '.valor_raiz AS ' +
@col2
    SET @sCampos2 = @sCampos2 + ', ' + @col2
    IF len(@sFrom) + len(@sFrom2) + len(@sFrom3) + len(@sFrom4) < 7000
        SET @sFrom = @sFrom +

```

```

' LEFT JOIN t' + @psPeriodicidad + '_' + convert(varchar(5), @nCont) + ' t' +
convert(varchar(5), @nCont) + ' WITH (NOLOCK) ' +
' ON t1.año = ' + ' t' + convert(varchar(5), @nCont) + '.año_calc ' +
' AND t1.entidad = ' + ' t' + convert(varchar(5), @nCont) + '.entidad_calc ' +
' AND t1.tipo = ' + ' t' + convert(varchar(5), @nCont) + '.tipo_calc ' +
' AND t1.segmento = ' + ' t' + convert(varchar(5), @nCont) + '.segmento_calc ' +
' AND t1.periodo = ' + ' t' + convert(varchar(5), @nCont) + '.periodo_calc '
ELSE IF len(@sFrom) + len(@sFrom2) + len(@sFrom3) + len(@sFrom4) < 14000
SET @sFrom2 = @sFrom2 +
' LEFT JOIN t' + @psPeriodicidad + '_' + convert(varchar(5), @nCont) + ' t' +
convert(varchar(5), @nCont) + ' WITH (NOLOCK) ' +
' ON t1.año = ' + ' t' + convert(varchar(5), @nCont) + '.año_calc ' +
' AND t1.entidad = ' + ' t' + convert(varchar(5), @nCont) + '.entidad_calc ' +
' AND t1.tipo = ' + ' t' + convert(varchar(5), @nCont) + '.tipo_calc ' +
' AND t1.segmento = ' + ' t' + convert(varchar(5), @nCont) + '.segmento_calc ' +
' AND t1.periodo = ' + ' t' + convert(varchar(5), @nCont) + '.periodo_calc '
ELSE IF len(@sFrom) + len(@sFrom2) + len(@sFrom3) + len(@sFrom4) < 21000
SET @sFrom3 = @sFrom3 +
' LEFT JOIN t' + @psPeriodicidad + '_' + convert(varchar(5), @nCont) + ' t' +
convert(varchar(5), @nCont) + ' WITH (NOLOCK) ' +
' ON t1.año = ' + ' t' + convert(varchar(5), @nCont) + '.año_calc ' +
' AND t1.entidad = ' + ' t' + convert(varchar(5), @nCont) + '.entidad_calc ' +
' AND t1.tipo = ' + ' t' + convert(varchar(5), @nCont) + '.tipo_calc ' +
' AND t1.segmento = ' + ' t' + convert(varchar(5), @nCont) + '.segmento_calc ' +
' AND t1.periodo = ' + ' t' + convert(varchar(5), @nCont) + '.periodo_calc '
ELSE IF len(@sFrom) + len(@sFrom2) + len(@sFrom3) + len(@sFrom4) < 28000
SET @sFrom4 = @sFrom4 +
' LEFT JOIN t' + @psPeriodicidad + '_' + convert(varchar(5), @nCont) + ' t' +
convert(varchar(5), @nCont) + ' WITH (NOLOCK) ' +
' ON t1.año = ' + ' t' + convert(varchar(5), @nCont) + '.año_calc ' +
' AND t1.entidad = ' + ' t' + convert(varchar(5), @nCont) + '.entidad_calc ' +
' AND t1.tipo = ' + ' t' + convert(varchar(5), @nCont) + '.tipo_calc ' +
' AND t1.segmento = ' + ' t' + convert(varchar(5), @nCont) + '.segmento_calc ' +
' AND t1.periodo = ' + ' t' + convert(varchar(5), @nCont) + '.periodo_calc '
END -- IF @tipo_indicador = 'R'
ELSE
BEGIN
SET @sIncador_calculado = @col2
END

FETCH NEXT FROM Columnas_Cursor INTO @col, @col2, @sIndicador, @tipo_indicador, @sCondicion
END
CLOSE Columnas_Cursor
DEALLOCATE Columnas_Cursor

SET @columns5 = @columns5 + 'CONSTRAINT pk_t_calculos_' + @sCarga + @psPeriodicidad + ' PRIMARY
KEY (entidad, tipo, periodo, fecha)'

SELECT @nError = -4, @sError = 'Error al realizar ALTER TABLE a t_calculos_indicadores_' +
@sCarga + @psPeriodicidad

EXEC(@columns + @columns2 + @columns3 + @columns4 + @columns5)
SET @nError = @@error
IF (@nError <> 0)
GOTO err_cursor_calc
-- TERMINA NOMBRES DE INDICADORES CALCULADOS
=====

SELECT @nError = -5, @sError = 'Error al insertar t_calculos_indicadores_' + @sCarga +
@psPeriodicidad + '(' + @sFrom + ')

print 'INSERT t_calculos_indicadores_' + @sCarga + @psPeriodicidad + '(' + @sCampos2 + ')' + @sCampos +
@sFrom + @sFrom2 + @sFrom3 + @sFrom4 + @sWhere
EXEC('INSERT t_calculos_indicadores_' + @sCarga + @psPeriodicidad + '(' + @sCampos2 + ')' +
@sCampos + @sFrom + @sFrom2 + @sFrom3 + @sFrom4 + @sWhere)
SET @nError = @@error
IF (@nError <> 0)
GOTO err_cursor_calc

SELECT @nError = -6, @sError = 'Error al insertar en #indicadores_calculados (' + @sFrom + ')'
SET @sSql =
INSERT #indicadores_calculados (entidad, indicador, segmento, tipo, valor, fecha, periodo)
SELECT DISTINCT entidad, ' + char(39) + @indicador_calc + char(39) + ', segmento, tipo, ' +
@sIncador_calculado + ', ' +
@sFecha + ', periodo
FROM t_calculos_indicadores_' + @sCarga + @psPeriodicidad + ' WITH (NOLOCK)
WHERE ' + @sIncador_calculado + ' IS NOT NULL
EXEC(@sSql)

```

```

        SET @nError = @@error
        IF (@nError <> 0)
            GOTO err_cursor_calc

        FETCH NEXT FROM cursor_calc INTO @indicador_calc, @tipo_calc, @nuNivel
    END
    CLOSE cursor_calc
    DEALLOCATE cursor_calc

-- Se eliminan las tablas auxiliares
WHILE @nCont <= 1
    BEGIN
        SET @sError = 'Error al eliminar las tablas de apoyo para la generar los indicadores
horizontalmente.'
        SET @sSql = 'DROP TABLE t' + @psPeriodicidad + '_' + convert(varchar(5), @nCont)
        EXEC(@sSql)
        SET @nError = @@error
        IF (@nError <> 0)
            GOTO err_error

        SET @nCont = @nCont - 1
    END
    --
=====
=====
-- BITACORA
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Inserta diferencias de calculos en bitacora_diferencias_calc_msd')
SELECT @nError = -7, @sError = 'Error al insertar datos a la tabla bitacora_diferencias_calc_msd'
-- Graba en bitacora los indicadores acumulados que tienen diferencias entre los calculados y los
cargados
INSERT bitacora_diferencias_calc_msd (carga, periodicidad, entidad, indicador, segmento, tipo,
periodo, fecha, tipo_calc, valor_carg, valor_calc)
SELECT DISTINCT @pnCarga, @psPeriodicidad, p.entidad, p.indicador, p.segmento, p.tipo, p.periodo,
p.fecha, 'CALC', p.valor, c.valor
FROM #indicadores_calculados c WITH (NOLOCK), #temporal_proceso p WITH (NOLOCK)
WHERE c.entidad = p.entidad
AND c.indicador = p.indicador
AND c.segmento = p.segmento
AND c.tipo = p.tipo
AND c.periodo = p.periodo
AND abs(abs(c.valor) - abs(p.valor)) > 0.01
SET @nError = @@error
IF (@nError <> 0)
    GOTO err_error

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Inserta los indicadores calculados en #temporal_proceso')
IF @pbComparar <> 1
    BEGIN
        SELECT @nError = -10, @sError = 'Error al insertar datos a la tabla #temporal_proceso'
        -- Pasa a #temporal_proceso los indicadores calculados que no fueron cargados
        -- 1. Primero se eliminan de #indicadores_calculados los indicadores que son iguales en
#indicadores_calculados y
-- #temporal_proceso
DELETE #temporal_proceso
FROM #indicadores_calculados i WITH (NOLOCK), #temporal_proceso t WITH (NOLOCK)
WHERE i.entidad = t.entidad
AND i.indicador = t.indicador
AND i.segmento = t.segmento
AND i.tipo = t.tipo
AND i.periodo = t.periodo
SET @nError = @@error
IF (@nError <> 0)
    GOTO err_error

-- 2. Se insertan en #temporal_proceso todos los datos que se en #indicadores_calculados
INSERT #temporal_proceso (entidad, indicador, segmento, tipo, valor, fecha, periodo, tipo_calc)
SELECT DISTINCT entidad, indicador, segmento, tipo, valor, fecha, periodo, 30
FROM #indicadores_calculados WITH (NOLOCK)
WHERE valor IS NOT NULL
SET @nError = @@error
IF (@nError <> 0)
    GOTO err_error
END -- IF @pbComparar <> 1

SELECT @nError = -11, @sError = 'Error al dropear la tabla t_calculos_indicadores_' + @sCarga +

```

```
@psPeriodicidad
    SET @sSql = 'DROP TABLE t_calculos_indicadores_' + @sCarga + @psPeriodicidad
    EXEC (@sSql)
    SET @nError = @@error
    IF (@nError <> 0)
        GOTO err_error
    END -- IF EXISTS (SELECT * FROM #temporal_calculo WITH (NOLOCK))

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'TERMINA PROCEDIMIENTO ALMACENADO sp_Calculos')

    RETURN 0

err_error:
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), @sError)
    RETURN @nError

err_cursor_calc:
    CLOSE cursor_calc
    DEALLOCATE cursor_calc
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), @sError)
    RETURN @nError

err_Columnas_Cursor:
    CLOSE Columnas_Cursor
    DEALLOCATE Columnas_Cursor
    CLOSE cursor_calc
    DEALLOCATE cursor_calc
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), @sError)
    RETURN @nError

END -- PROC
```

C.7 Obtención de datos.

```

IF EXISTS (SELECT name FROM sysobjects WITH (NOLOCK) WHERE name = 'sp_getRootForCalc' AND type = 'P')
    DROP PROCEDURE sp_getRootForCalc
GO
--
*****
-- Nombre:          sp_getRootForCalc
-- Descripción:     De los indicadores raices cargados, se obtienen los valores de todos los indicadores
raices que
--                 intervienen en las fórmulas para obtener los indicadores calculados
-- Parametros:     @psPeriodicidad: Periodicidad que se está cargando
--                 @psAño:          Año que se está cargando
-- Return:         -1:Error, 0:Exito
--
*****
CREATE PROC sp_getRootForCalc
    @pnCarga          int,
    @psPeriodicidad  char(1),
    @psAño            char(4)
AS
BEGIN
    DECLARE    @sAñoAnterior          char(4),          -- Año anterior del que se está cargando
               @sTabDatos            varchar(20),      -- Nombre de la tabla de [datos_periodes]
               @sTabDatosAnt         varchar(20),      -- Nombre de la tabla de [datos_periodes] del año
               @sSql                  varchar(8000),    -- Cadena que contiene las sentencias SQL
               @nPeriodoMaxAnt        smallint,        -- Periodo máximo de acuerdo a la periodicidad del año
               @nPeriodoMax           smallint,        -- Periodo máximo de acuerdo a la periodicidad
               @nPeriodo              smallint,        -- Periodo que se está cargando
               @nUltimoPeriodo        smallint,        -- Último periodo que ha sido cargado antes del proceso
               @nCont                  smallint,        -- Recorre cada uno de los periodos posteriores al que
se está cargado
               @sCont                  varchar(5),      -- El valor de la variable @nCont pero como cadena
               @sFecha                varchar(500),    -- Contiene la instrucción que arma la fecha semanal
               @sFechaCierreAñoAnt    varchar(100),
               @nError                 smallint,        -- Nos indica el lugar del código donde fue producido
               @sError                 varchar(150),    -- Descripción de la parte donde ocurrió el error
               @sIndica_calculado      varchar(10),
               @nuNivel                 smallint,
               @nuCont                 smallint

    -- Establece el formato de fecha
    SET DATEFORMAT mdy

    -- Se utiliza para indicar cuales son los indicadores calculados que se tienen que procesar para una
carga posterior
    CREATE TABLE #calc_posterior(
        indicador          varchar(10),
        tipo               char(1)
    )

    -- Lista de indicadores que se tienen que calcular
    CREATE TABLE #calculados_pivote(
        indicador_calc     char(10),
        tipo_calc          char(1),
        entidad_calc       char(10),
        segmento_calc      char(2),
        año_calc            smallint,
        periodo_calc       smallint,
        fecha_calc          char(10),
        nivel              smallint
    )

    CREATE TABLE #temporal_calculo_aux(
        consecutivo        smallint,
        indicador_calc      char(10),
        tipo_calc           char(1),

```

Apéndice C. Código de los procesos

```
        entidad_calc          char(10),
        segmento_calc        char(2),
        año_calc              smallint,
        periodo_calc         smallint,
        fecha_calc           char(10),
        indicador_raiz       char(10),
        tipo_raiz            char(1),
        periodo_relativo_raiz smallint,
        valor_raiz           float,
        nivel                 smallint
    )
    CREATE UNIQUE INDEX ix_temporal_calculo On #temporal_calculo_aux
        (consecutivo, indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
periodo_calc,
        indicador_raiz, tipo_raiz, periodo_relativo_raiz )

    CREATE TABLE #entidades_requeridas(
        indicador varchar(10),
        entidad   varchar(10)
    )

    SET @sAñoAnterior = @psAño - 1
    -- Asigna el nombre de la tabla a una variable
    IF @psPeriodicidad = 'M'
        SELECT @sTabDatos = 'datos_mes_' + @psAño, @sTabDatosAnt = 'datos_mes_' + @sAñoAnterior,
@nPeriodoMax = 12, @nPeriodoMaxAnt = 12
    ELSE IF @psPeriodicidad = 'S'
        SELECT @sTabDatos = 'datos_sem_' + @psAño, @sTabDatosAnt = 'datos_sem_' + @sAñoAnterior,
@nPeriodoMax = 52, @nPeriodoMaxAnt = 52
    ELSE
        SELECT @sTabDatos = 'datos_dia_' + @psAño, @sTabDatosAnt = 'datos_dia_' + @sAñoAnterior,
@nPeriodoMax = datepart(dy, @psAño + '-12-31'), @nPeriodoMaxAnt = datepart(dy, @sAñoAnterior + '-12-31')

    -----
    -- PASO 1) Obtener los nombres de indicadores calculados que se tienen que generar con base a los
    indicadores
    --          raíz que se están cargado.
    -----

    -- Obtiene todos los indicadores calculados donde, por lo menos, tiene un elemento raíz
    insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Inserta datos en la tabla #calculados')
    SELECT @nError = -1, @sError = 'Error al insertar en la tabla #calculados'
    INSERT #calculados_aux (indicador_calc,      tipo_calc,      entidad_calc,      segmento_calc,
        año_calc,      periodo_calc,      fecha_calc)
    SELECT DISTINCT cr.indicador_calculado, cr.tipo_calculado, t.entidad,      t.segmento,
        @psAño,      t.periodo,      t.fecha
    FROM cat_calculado_raiz_anidado cr WITH (NOLOCK), #temporal_proceso t WITH (NOLOCK)
    WHERE t.indicador      = cr.indicador_raiz
    AND t.tipo              = cr.tipo_raiz
    AND cr.periodicidad    = @psPeriodicidad
    IF (@@error <> 0)
        GOTO err_calc

    -- Verifica si es un recalculado. Cuando es recalculado existe en la tabla indicadores_recalculados
    IF EXISTS (SELECT * FROM indicadores_recalculados WITH (NOLOCK) WHERE carga = @pnCarga)
        DELETE #calculados_aux WHERE indicador_calc NOT IN (SELECT indicador FROM indicadores_recalculados
WITH (NOLOCK) WHERE carga = @pnCarga)

    --
    =====
    =====
    -- Se eliminan las entidades de niveles que no debe tener. Esto se presenta cuando los indicadores
    raices tienen
    -- mas entidades que el calculado. Además pueden generarse valores que no estén asociados a una entidad
    cuyo árbol
    -- esté relacionado.
    INSERT #entidades_requeridas
    SELECT DISTINCT r.indicador AS indicador, e.hijo AS entidad
    FROM productos.dbo.cat_productos p WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK),
        entidades_arbol e WITH (NOLOCK), #calculados_aux c WITH (NOLOCK)
    WHERE r.producto = p.id_producto
    AND p.id_ent_arbol = e.id_ent_arbol
    AND r.indicador = c.indicador_calc
    AND e.hijo = c.entidad_calc
```

```

AND e.hijo IS NOT NULL
UNION
SELECT DISTINCT r.indicador AS indicador, e.padre AS entidad
FROM productos.dbo.cat_productos p WITH (NOLOCK), cat_responsabilidad r WITH (NOLOCK),
entidades_arbol e WITH (NOLOCK), #calculados_aux c WITH (NOLOCK)
WHERE r.producto = p.id_producto
AND p.id_ent_arbol = e.id_ent_arbol
AND r.indicador = c.indicador_calc
AND e.padre = c.entidad_calc
AND e.padre IS NOT NULL
UNION
SELECT DISTINCT a.indicador AS indicador, e.padre AS entidad
FROM cat_arbol_si a WITH (NOLOCK), entidades_arbol e WITH (NOLOCK), #calculados_aux c WITH (NOLOCK)
WHERE a.id_ent_arbol = e.id_ent_arbol
AND a.indicador NOT IN ('TITULO', 'BLANCO', 'PORDEFINIR')
AND a.indicador = c.indicador_calc
AND e.padre = c.entidad_calc
AND e.padre IS NOT NULL
UNION
SELECT DISTINCT a.indicador AS indicador, e.hijo AS entidad
FROM cat_arbol_si a WITH (NOLOCK), entidades_arbol e WITH (NOLOCK), #calculados_aux c WITH (NOLOCK)
WHERE a.id_ent_arbol = e.id_ent_arbol
AND a.indicador NOT IN ('TITULO', 'BLANCO', 'PORDEFINIR')
AND a.indicador = c.indicador_calc
AND e.hijo = c.entidad_calc
AND e.hijo IS NOT NULL
ORDER BY indicador, entidad

SELECT c.indicador_calc, c.entidad_calc, e.indicador, e.entidad
INTO #entidades_por_eliminar
FROM #calculados_aux c WITH (NOLOCK), #entidades_requeridas e WITH (NOLOCK)
WHERE c.indicador_calc *= e.indicador
AND c.entidad_calc *= e.entidad

DELETE #calculados_aux
FROM #calculados_aux c WITH (NOLOCK), #entidades_por_eliminar e WITH (NOLOCK)
WHERE c.indicador_calc = e.indicador_calc
AND c.entidad_calc = e.entidad_calc
AND e.entidad IS NULL

----- ORDERNARA LA LISTA DE CALCULADOS EN FORMA ANIDADADA -----
-- 1) Obtenemos el subconjunto de indicadores a procesar de cat_calculado_raiz_anidado.
SELECT DISTINCT cl.*
INTO #cat_calculado_raiz_anidado
FROM cat_calculado_raiz_anidado cl WITH (NOLOCK), #calculados_aux c WITH (NOLOCK)
WHERE cl.periodicidad = @psPeriodicidad
AND cl.indicador_calculado = c.indicador_calc
AND cl.tipo_calculado = c.tipo_calc
-- 2) Obtenemos un nuevo subconjunto donde los indicadores raíz tambien son calculado. Esto se necesita
para que
-- posteriormente obtengamos el indicador origen de todos los demás calculados.
SELECT *
INTO #cat_calculado_raiz_anidado_h
FROM #cat_calculado_raiz_anidado WITH (NOLOCK)
WHERE indicador_raiz in (select indicador_calculado from #cat_calculado_raiz_anidado WITH (NOLOCK))
-- 3) Obtenemos una copia del subconjunto
SELECT *
INTO #cat_calculado_raiz_anidado_p
FROM #cat_calculado_raiz_anidado_h WITH (NOLOCK)
-- 4) Ligamos ambas tablas, pero se ligan el indicador padre con el hijo. La lista donde sea
indicador_calculado
-- son los indicadores que son el origen de todos los demás.
SELECT DISTINCT p.indicador_calculado AS indicador_calculado_2,
h.indicador_calculado, h.tipo_calculado, h.indicador_raiz, 1 as nivel
INTO #cat_calculado_raiz_anidado_aux
FROM #cat_calculado_raiz_anidado_p p WITH (NOLOCK), #cat_calculado_raiz_anidado_h h WITH (NOLOCK)
WHERE p.indicador_calculado =* h.indicador_raiz
AND p.tipo_calculado =* h.tipo_calculado
-- 5) Se eliminan todos los indicadores cuyo campo indicador_calculado_2 es NO NULO. Con esto obtenemos
el origen
-- de los indicadores calculados
DELETE #cat_calculado_raiz_anidado_aux WHERE indicador_calculado_2 IS NOT NULL

-- 6) Se almacenarán los datos en #calculados pero con el orden de anidamiento requerido. Para esto se
usará un loop
SELECT @nuNivel = 1
WHILE @nuNivel > 0

```

Apéndice C. Código de los procesos

```
BEGIN
  INSERT #calculados (indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
    periodo_calc, fecha_calc, nivel)
  SELECT DISTINCT indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc, periodo_calc,
    fecha_calc, @nuNivel
  FROM #calculados_aux WITH (NOLOCK)
  WHERE indicador_calc IN
    (SELECT indicador_raiz FROM #cat_calculado_raiz_anidado_aux WITH (NOLOCK) WHERE nivel =
@nuNivel)

  INSERT #cat_calculado_raiz_anidado_aux (indicador_calculado_2, indicador_calculado, tipo_calculado,
    indicador_raiz, nivel)
  SELECT DISTINCT NULL, r.indicador_calculado, r.tipo_calculado, r.indicador_raiz, @nuNivel + 1
  FROM #cat_calculado_raiz_anidado r WITH (NOLOCK), #cat_calculado_raiz_anidado_aux c WITH (NOLOCK)
  WHERE c.indicador_calculado = r.indicador_raiz
  AND c.tipo_calculado = r.tipo_calculado
  AND c.nivel = @nuNivel

  DELETE #cat_calculado_raiz_anidado
  FROM #cat_calculado_raiz_anidado c WITH (NOLOCK), #cat_calculado_raiz_anidado_aux a WITH (NOLOCK)
  WHERE c.indicador_calculado = a.indicador_raiz
  AND c.tipo_calculado = a.tipo_calculado
  AND a.nivel = @nuNivel

  IF NOT EXISTS (SELECT * FROM #cat_calculado_raiz_anidado_aux WHERE nivel = @nuNivel)
  BEGIN
    INSERT #calculados (indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
    periodo_calc,
      fecha_calc, nivel)
    SELECT DISTINCT c.indicador_calc, c.tipo_calc, c.entidad_calc, c.segmento_calc, c.año_calc,
    c.periodo_calc,
      c.fecha_calc, @nuNivel
    FROM #calculados_aux c WITH (NOLOCK), #cat_calculado_raiz_anidado a WITH (NOLOCK)
    WHERE c.indicador_calc = a.indicador_calculado
    AND c.tipo_calc = a.tipo_calculado

    SELECT @nuNivel = -1
  END

  SELECT @nuNivel = @nuNivel + 1
END

-- Se crea una tabla auxiliar para determinar cual es el nivel mínimo de cada indicador
-- (para los indicadores repetidos debido a que tienen diferentes niveles)
INSERT #calculados_pivote (indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
    periodo_calc, fecha_calc, nivel)
  SELECT indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc, periodo_calc, fecha_calc,
    max(nivel) as nivel
  FROM #calculados
  GROUP BY indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc, periodo_calc, fecha_calc
  HAVING count(*) > 1

-- Se eliminan los indicadores repetidos debido a que un mismo indicador presente diferentes niveles
DELETE #calculados
  FROM #calculados c WITH (NOLOCK), #calculados_pivote p WITH (NOLOCK)
  WHERE c.indicador_calc = p.indicador_calc
  AND c.tipo_calc = p.tipo_calc
  AND c.entidad_calc = p.entidad_calc
  AND c.segmento_calc = p.segmento_calc
  AND c.año_calc = p.año_calc
  AND c.periodo_calc = p.periodo_calc
  AND c.nivel = p.nivel
--
=====
=====
-----
-----
-- PASO 2) Obtener todos los nombres de indicadores raices que intervienen en las fórmulas de los
calculados
-----
-----
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Inserta datos en la tabla #raices')
  SELECT @nError = -2, @sError = 'Error al insertar en la tabla #raices'
  INSERT #raices (indicador_calc, tipo_calc, indicador_raiz,
    tipo_raiz,
    periodo_relativo_raiz, entidad, segmento,
```

```

año,                periodo,                fecha,
SELECT DISTINCT c.indicador_calc,                c.tipo_calc,                cr.indicador_raiz,
                cr.tipo_raiz,
                cr.periodo_relativo_raiz,        c.entidad_calc,            c.segmento_calc,
                c.año_calc,                    c.periodo_calc,            c.fecha_calc,
                c.nivel
FROM #calculados c WITH (NOLOCK), cat_calculado_raiz_anidado cr WITH (NOLOCK)
WHERE c.indicador_calc = cr.indicador_calculado
AND cr.periodicidad = @psPeriodicidad
AND cr.tipo_calculado = c.tipo_calc
IF (@@error <> 0)
GOTO err_calc

-- ----- Separación de los periodos relativos fijos del año anterior -----
-- Para darle un tratamiento distinto a los indicadores de periodos relativos fijos se pasarán a la
-- tabla
-- #raices_fijos
INSERT #raices_fijos (indicador_calc, tipo_calc, indicador_raiz, tipo_raiz, periodo_relativo_raiz,
entidad,
segmento, año, periodo, fecha, nivel)
SELECT DISTINCT indicador_calc, tipo_calc, indicador_raiz, tipo_raiz, periodo_relativo_raiz, entidad,
segmento, año,
periodo, fecha, nivel
FROM #raices WITH (NOLOCK)
WHERE periodo_relativo_raiz > 0

-- Se eliminan de #raices para que los valores se integren por separado #raices y #raices_fijos
DELETE #raices WHERE periodo_relativo_raiz > 0

-----

-----

-- PASO 3) Se obtienen los datos de las tablas datos_mes o datos_sem del año correspondiente.
-- Se utiliza un cursor para realizar las búsquedas correspondientes en las tablas del año
adecuado.

-----

IF EXISTS (SELECT * FROM #raices WITH (NOLOCK))
BEGIN
SELECT @nCont = 0, @sCont = '0'

SELECT @nPeriodo = min(periodo), @nUltimoPeriodo = max(periodo)
FROM #raices WITH (NOLOCK) WHERE periodo_relativo_raiz = 0

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Obtiene los valores de los indicadores de los periodos posteriores del ' + convert(varchar(5),
@nPeriodo) + ' al periodo ' + convert(varchar(5), @nUltimoPeriodo))
-- Asigna la cadena cuya instrucción obtiene la fecha correspondiente para cuando se trata de
periodicidad
-- mensual o semanal.
IF @psPeriodicidad = 'S'
SET @sFecha = char(39) + right(@psAño, 2) + char(39) + ' + right(' + char(39) + '0' + char(39)
+
' + convert(varchar(5), r.periodo + ' + @sCont + '), 2)' +
' AS fecha, '
ELSE IF @psPeriodicidad = 'M'
SET @sFecha = 'convert(varchar(10), r.fecha, 126), '
ELSE IF @psPeriodicidad = 'D'
SET @sFecha = 'convert(varchar(10), dateadd(d, r.periodo - 1 + ' + @sCont + ', ' + char(39) +
@psAño + '-01-01' + char(39) + '), 126) AS fecha, '

-- Query que obtiene los datos de la tabla del año que se está cargando
SET @sSql = '
INSERT #temporal_calculo_aux
(indicador_calc,                tipo_calc,                entidad_calc,
segmento_calc,                año_calc,                periodo_calc,
fecha_calc,                    indicador_raiz,            tipo_raiz,
periodo_relativo_raiz,        valor_raiz,                consecutivo,
nivel)
SELECT DISTINCT
r.indicador_calc,                r.tipo_calc,                r.entidad,
r.segmento,                ' + @psAño + ' AS año,        r.periodo, ' +
@sFecha + '                r.indicador_raiz,            r.tipo_raiz,
r.periodo_relativo_raiz,        d.valor,                1,

```

Apéndice C. Código de los procesos

```
        r.nivel
    FROM #raices r WITH (NOLOCK), ' + @sTabDatos + ' d WITH (NOLOCK)
    WHERE r.periodo + r.periodo_relativo_raiz > 0
        AND r.periodo + r.periodo_relativo_raiz <= ' + convert(varchar(5), @nPeriodoMax) + '
        AND r.año
            = ' + @psAño + '
        AND r.indicador_raiz
            = d.indicador
        AND r.entidad
            = d.entidad
        AND r.segmento
            = d.segmento
        AND r.tipo_raiz
            = d.tipo
        AND r.periodo + r.periodo_relativo_raiz = d.periodo '
actual'
SELECT @nError = -4, @sError = 'Error al insertar en la tabla #temporal_calculo_aux para el año

EXEC (@sSql)
IF (@@error <> 0)
    GOTO err_calc

IF @psPeriodicidad = 'M'
BEGIN
    SET @sSql = '
        INSERT #temporal_calculo_aux
            (indicador_calc,          tipo_calc,          entidad_calc,
             segmento_calc,         año_calc,          periodo_calc,
             fecha_calc,            indicador_raiz,    tipo_raiz,
             periodo_relativo_raiz, valor_raiz,        consecutivo,
             nivel)
        SELECT DISTINCT
            r.indicador_calc,          r.tipo_calc,          r.entidad,
            r.segmento,                ' + @psAño + ' AS año,  r.periodo, ' +
            @sFecha + '                r.indicador_raiz,    r.tipo_raiz,
            r.periodo_relativo_raiz,  d.valor,              1,
            r.nivel
        FROM #raices r WITH (NOLOCK), datos_mes_adic d WITH (NOLOCK)
        WHERE r.periodo + r.periodo_relativo_raiz > 0
            AND r.periodo + r.periodo_relativo_raiz <= ' + convert(varchar(5), @nPeriodoMax) + '
            AND r.año
                = ' + @psAño + '
            AND r.indicador_raiz
                = d.indicador
            AND r.entidad
                = d.entidad
            AND r.segmento
                = d.segmento
            AND r.tipo_raiz
                = d.tipo
            AND r.periodo + r.periodo_relativo_raiz = datepart(mm, d.fecha)
            AND datepart(yy, d.fecha) = ' + @psAño
        SELECT @nError = -5, @sError = 'Error al insertar en la tabla #temporal_calculo_aux para el
año actual de datos_mes_adic'
        EXEC (@sSql)
        IF (@@error <> 0)
            GOTO err_calc
    END

IF EXISTS (SELECT * FROM sysobjects WITH (NOLOCK) WHERE name = @sTabDatosAnt AND type = 'U')
BEGIN
    -- Query que obtiene los datos de la tabla del año anterior
    SET @sSql = '
        -- Cuando se trata de periodos negativos (PN12)
        INSERT #temporal_calculo_aux
            (indicador_calc,          tipo_calc,          entidad_calc,
             segmento_calc,         año_calc,          periodo_calc,
             fecha_calc,            indicador_raiz,    tipo_raiz,
             periodo_relativo_raiz, valor_raiz,        consecutivo,
             nivel)
        SELECT r.indicador_calc,          r.tipo_calc,          r.entidad,
            r.segmento,                ' + @psAño + ' AS Año,  r.periodo, ' +
            @sFecha + '                r.indicador_raiz,    r.tipo_raiz,
            r.periodo_relativo_raiz,  d.valor,              1,
            r.nivel
        FROM #raices r WITH (NOLOCK), ' + @sTabDatosAnt + ' d WITH (NOLOCK)
        WHERE r.periodo + r.periodo_relativo_raiz <= 0
            AND r.año
                = ' + @psAño + '
            AND r.indicador_raiz
                = d.indicador
            AND r.entidad
                = d.entidad
            AND r.segmento
                = d.segmento
            AND r.tipo_raiz
                = d.tipo
            AND d.periodo
                = ' + convert(varchar(5), @nPeriodoMaxAnt) +
                ' + (r.periodo + r.periodo_relativo_raiz)

        UNION
        -- Cuando se trata de periodos fijos (PP12)
        SELECT r.indicador_calc,          r.tipo_calc,          r.entidad,
            r.segmento,                ' + @psAño + ',          r.periodo, ' +

```

```

                @sFecha + '                r.indicador_raiz,                r.tipo_raiz,
                r.periodo_relativo_raiz,  d.valor,                1,
                r.nivel
        FROM #raices r WITH (NOLOCK), ' + @sTabDatosAnt + ' d WITH (NOLOCK)
        WHERE r.periodo_relativo_raiz > 0
        AND r.año                                = ' + @psAño + '
        AND r.indicador_raiz                    = d.indicador
        AND r.entidad                            = d.entidad
        AND r.segmento                          = d.segmento
        AND r.tipo_raiz                        = d.tipo
        AND d.periodo                            = r.periodo_relativo_raiz '
anterior' SELECT @nError = -6, @sError = 'Error al insertar en la tabla #temporal_calculo_aux del año
        EXEC (@sSql)
        IF (@@error <> 0)
            GOTO err_calc

        END -- IF EXISTS (Select * From dbo.sysobjects )

--Inserta los registros sin repetir en la tabla temporal_calculo
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Inserta de la tabla #temporal_calculo_aux a la tabla #temporal_calculo')
SELECT @nError = -8, @sError = 'Error al insertar en la tabla #temporal_calculo'
INSERT #temporal_calculo
        (indicador_calc,                tipo_calc,                entidad_calc,
        segmento_calc,                año_calc,                periodo_calc,
        fecha_calc,                indicador_raiz,                tipo_raiz,
        periodo_relativo_raiz,                valor_raiz,                nivel)
SELECT DISTINCT
        indicador_calc,                tipo_calc,                entidad_calc,
        segmento_calc,                año_calc,                periodo_calc,
        fecha_calc,                indicador_raiz,                tipo_raiz,
        periodo_relativo_raiz,                valor_raiz,                nivel
        FROM #temporal_calculo_aux WITH (NOLOCK)
IF (@@error <> 0)
    GOTO err_calc

--
=====
-- SE COMPLEMENTAN LOS DATOS. ESTE PROCESO SE REALIZA CUANDO NO TODOS LOS VALORES SE ENCUENTRAN EN
DATOS_MMM_YYYY.
-----
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se inserta de la tabla #temporal_calculo a la tabla #tmp_aux_tot')
-----
SELECT @nError = -4, @sError = 'Error al insertar en la tabla #temporal_calculo_aux para el año
actual'
-- Query que obtiene los datos de la tabla del año que se está cargando
SELECT DISTINCT
        r.indicador_calc,                r.tipo_calc,                r.entidad as entidad_calc,
        r.segmento as segmento_calc,    @psAño as año_calc,                r.periodo as periodo_calc,
        d.fecha as fecha_calc,                r.indicador_raiz,                r.tipo_raiz,
        r.periodo_relativo_raiz,                d.valor,                r.nivel
        INTO #tmp_aux_tot
        FROM #raices r WITH (NOLOCK), #temporal_proceso_total d WITH (NOLOCK)
        WHERE r.año                                = @psAño
        AND r.indicador_raiz                    = d.indicador
        AND r.entidad                            = d.entidad
        AND r.segmento                          = d.segmento
        AND r.tipo_raiz                        = d.tipo
        AND r.periodo + r.periodo_relativo_raiz = d.periodo
        IF (@@error <> 0)
            GOTO err_calc
-----
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se eliminan registros de la tabla #temporal_calculo')
SELECT @nError = -10, @sError = 'Error al eliminar en la tabla #temporal_calculo'
-- 2. Elimina los indicadores de #temporal_calculo
DELETE #temporal_calculo
        FROM #temporal_calculo c WITH (NOLOCK), #temporal_proceso_total p WITH (NOLOCK)
        WHERE c.entidad_calc = p.entidad
        AND c.indicador_raiz = p.indicador
        AND c.tipo_raiz = p.tipo

```

Apéndice C. Código de los procesos

```
AND c.segmento_calc = p.segmento
AND c.periodo_calc + c.periodo_relativo_raiz = p.periodo
IF (@@error <> 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se insertan registros en la tabla #tmp_aux_2_tot')
SELECT @nError = -11, @sError = 'Error al Inserta los datos de #tmp_aux a #temporal_calculo'
-- 3. Inserta los datos de #tmp_aux a #temporal_calculo. Lo realizaremos en dos pasos.
-- 3.1 Crearemos una tabla auxiliar por medio de un outer join
SELECT a.indicador_calc, a.tipo_calc, a.entidad_calc, a.segmento_calc, a.año_calc, a.periodo_calc,
a.fecha_calc,
    a.indicador_raiz, a.tipo_raiz, a.periodo_relativo_raiz, a.valor, c.indicador_calc AS
indicador_calc_2,
    a.nivel
INTO #tmp_aux_2_tot
FROM #tmp_aux_tot a WITH (NOLOCK), #temporal_calculo c WITH (NOLOCK)
WHERE a.indicador_calc *= c.indicador_calc
    AND a.tipo_calc *= c.tipo_calc
    AND a.entidad_calc *= c.entidad_calc
    AND a.segmento_calc *= c.segmento_calc
    AND a.periodo_calc *= c.periodo_calc
    AND a.indicador_raiz *= c.indicador_raiz
    AND a.tipo_raiz *= c.tipo_raiz
    AND a.periodo_relativo_raiz *= c.periodo_relativo_raiz
IF (@@error <> 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se insertan registros de la tabla #tmp_aux_2_tot a #temporal_calculo')
-- 3.1 Pasamos los datos a #temporal_calculo pero que cumplan cierta condición
INSERT #temporal_calculo(indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
periodo_calc, fecha_calc,
    indicador_raiz, tipo_raiz, periodo_relativo_raiz, valor_raiz, nivel)
SELECT DISTINCT indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc, periodo_calc,
fecha_calc,
    indicador_raiz, tipo_raiz, periodo_relativo_raiz, valor, nivel
FROM #tmp_aux_2_tot WITH (NOLOCK)
WHERE indicador_calc_2 IS NULL
IF (@@error <> 0)
    GOTO err_calc

-----
-----

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se inserta de la tabla #temporal_calculo a la tabla #tmp_aux')
SELECT @nError = -9, @sError = 'Error al insertar en la tabla #tmp_aux'
-- Se actualizan de #temporal_calculo los indicadores que existen en #temporal_proceso. Se hace en 3
pasos
-- 1. Obtiene los registro repetidos con los valores de #temporal_proceso (no importa si es el mismo
valor o no)
SELECT DISTINCT c.indicador_calc, c.tipo_calc, c.entidad_calc, c.segmento_calc, c.año_calc,
c.periodo_calc,
    c.fecha_calc, c.indicador_raiz, c.tipo_raiz, c.periodo_relativo_raiz, p.valor, c.nivel
INTO #tmp_aux
FROM #temporal_calculo c WITH (NOLOCK), #temporal_proceso p WITH (NOLOCK)
WHERE c.entidad_calc = p.entidad
    AND c.indicador_raiz = p.indicador
    AND c.tipo_raiz = p.tipo
    AND c.segmento_calc = p.segmento
    AND c.periodo_calc + c.periodo_relativo_raiz = p.periodo
IF (@@error <> 0)
    GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se eliminan registros de la tabla #temporal_calculo')
SELECT @nError = -10, @sError = 'Error al eliminar en la tabla #temporal_calculo'
-- 2. Elimina los indicadores de #temporal_calculo
DELETE #temporal_calculo
FROM #temporal_calculo c WITH (NOLOCK), #temporal_proceso p WITH (NOLOCK)
WHERE c.entidad_calc = p.entidad
    AND c.indicador_raiz = p.indicador
    AND c.tipo_raiz = p.tipo
    AND c.segmento_calc = p.segmento
    AND c.periodo_calc + c.periodo_relativo_raiz = p.periodo
IF (@@error <> 0)
```

```

GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se insertan registros de en la tabla #tmp_aux_2')
SELECT @nError = -11, @sError = 'Error al Inserta los datos de #tmp_aux a #temporal_calculo'
-- 3. Inserta los datos de #tmp_aux a #temporal_calculo. Lo realizaremos en dos pasos.
-- 3.1 Crearemos una tabla auxiliar por medio de un outer join
SELECT a.indicador_calc, a.tipo_calc, a.entidad_calc, a.segmento_calc, a.año_calc, a.periodo_calc,
a.fecha_calc,
a.indicador_raiz, a.tipo_raiz, a.periodo_relativo_raiz, a.valor, c.indicador_calc AS
indicador_calc_2,
a.nivel
INTO #tmp_aux_2
FROM #tmp_aux a WITH (NOLOCK), #temporal_calculo c WITH (NOLOCK)
WHERE a.indicador_calc *= c.indicador_calc
AND a.tipo_calc *= c.tipo_calc
AND a.entidad_calc *= c.entidad_calc
AND a.segmento_calc *= c.segmento_calc
AND a.periodo_calc *= c.periodo_calc
AND a.indicador_raiz *= c.indicador_raiz
AND a.tipo_raiz *= c.tipo_raiz
AND a.periodo_relativo_raiz *= c.periodo_relativo_raiz
IF (@@error <> 0)
GOTO err_calc

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se insertan registros de la tabla #tmp_aux_2 a #temporal_calculo')
-- 3.1 Pasamos los datos a #temporal_calculo pero que cumplan cierta condición
INSERT #temporal_calculo(indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
periodo_calc, fecha_calc,
indicador_raiz, tipo_raiz, periodo_relativo_raiz, valor_raiz, nivel)
SELECT indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc, periodo_calc, fecha_calc,
indicador_raiz, tipo_raiz, periodo_relativo_raiz, valor, nivel
FROM #tmp_aux_2 WITH (NOLOCK)
WHERE indicador_calc_2 IS NULL
IF (@@error <> 0)
GOTO err_calc

-----
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'Se insertan registros de la tabla #raices a #temporal_calculo')
SELECT @nError = -12, @sError = 'Error al Inserta los indicadores que intervienen en la formula que
existen en #temporal_proceso, pero que no existen en #temporal_calculo'
-- Inserta los indicadores que intervienen en la formula que existen en #temporal_proceso, pero que
no existen
-- en #temporal_calculo. Con esto se tiene el total de indicadores para realizar la formula. Lo
haremos en dos pasos
-- 4.1 Crearemos una tabla auxiliar por medio de un outer join
SELECT r.indicador_calc, r.tipo_calc, r.entidad, r.segmento, @psAño AS año, r.periodo, p.fecha,
r.indicador_raiz, r.tipo_raiz, r.periodo_relativo_raiz, p.valor, c.indicador_calc AS
indicador_calc_2,
r.nivel
INTO #tmp_aux_3
FROM #raices r WITH (NOLOCK), #temporal_proceso p WITH (NOLOCK), #temporal_calculo c WITH (NOLOCK)
WHERE r.año = @psAño
AND r.indicador_raiz = p.indicador
AND r.entidad = p.entidad
AND r.segmento = p.segmento
AND r.tipo_raiz = p.tipo
AND r.periodo + r.periodo_relativo_raiz = p.periodo
AND r.indicador_calc *= c.indicador_calc
AND r.tipo_calc *= c.tipo_calc
AND r.entidad *= c.entidad_calc
AND r.segmento *= c.segmento_calc
AND r.periodo *= c.periodo_calc
AND r.indicador_raiz *= c.indicador_raiz
AND r.tipo_raiz *= c.tipo_raiz
AND r.periodo_relativo_raiz *= c.periodo_relativo_raiz
IF (@@error <> 0)
GOTO err_calc

SELECT @nError = -13, @sError = 'Error al Inserta los indicadores en #temporal_calculo'
INSERT #temporal_calculo (indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
periodo_calc, fecha_calc,
indicador_raiz, tipo_raiz, periodo_relativo_raiz, valor_raiz, nivel)
SELECT DISTINCT indicador_calc, tipo_calc, entidad, segmento, año, periodo, fecha,
indicador_raiz, tipo_raiz, periodo_relativo_raiz, valor, nivel

```

```

FROM #tmp_aux_3 WITH (NOLOCK)
WHERE indicador_calc_2 IS NULL
IF (@@error <> 0)
    GOTO err_calc

DELETE #temporal_calculo WHERE periodo_calc > @nPeriodoMax

-----
-- -----
-- Se integra los datos de los periodos relativos fijos del año anterior
IF EXISTS (SELECT * FROM sysobjects WITH (NOLOCK) WHERE name = @sTabDatosAnt AND type = 'U')
BEGIN
    SELECT @nError = -14, @sError = 'Error al insertar en la tabla #temporal_calculo_aux de
#raices_fijo'
    SET @sSql = '
INSERT #temporal_calculo (indicador_calc, tipo_calc, entidad_calc, segmento_calc, año_calc,
periodo_calc,
                fecha_calc, indicador_raiz, tipo_raiz, periodo_relativo_raiz, valor_raiz, nivel)
SELECT DISTINCT r.indicador_calc, r.tipo_calc, r.entidad, r.segmento, ' + @psAño + ' AS Año,
r.periodo, ' +
                @sFecha + ' r.indicador_raiz, r.tipo_raiz, r.periodo_relativo_raiz, d.valor, nivel
FROM #raices_fijos r WITH (NOLOCK), ' + @sTabDatosAnt + ' d WITH (NOLOCK)
WHERE r.periodo_relativo_raiz = d.periodo
AND r.año = ' + @psAño + '
AND r.indicador_raiz = d.indicador
AND r.entidad = d.entidad
AND r.segmento = d.segmento
AND r.tipo_raiz = d.tipo '
    EXEC (@sSql)
    IF (@@error <> 0)
        GOTO err_calc
END
-----
--
=====
END -- IF EXISTS (SELECT count(*) FROM #raices)

insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), 'TERMINA sp_getRootForCalc')

RETURN 0

err_calc:
insert bitacora_proceso_calc(carga, periodicidad, hora, descripcion) values (@pnCarga, @psPeriodicidad,
getdate(), @sError)
RETURN @nError

```

