

Un modelo de *self-management* para mejorar la eficiencia de la productividad en la administración de una base de datos

A self-management model to improve the efficiency of productivity in the administration of a database

Jorge L. del Mar Arzola

Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática. Lima, Perú
jorge.delmar@unmsm.edu.pe

Resumen

En la actualidad, las organizaciones enfrentan dos graves problemas con respecto a la gestión de base de datos. El primero de los problemas es el aumento en el volumen de los datos y la complejidad de los requerimientos de información, el segundo problema es que los administradores de base de datos con buenos conocimientos y amplia experiencia son un recurso cada vez más escaso y sus costos se han elevado. Debido a estos problemas, en los últimos años, los proveedores de software de base de datos buscan automatizar las funciones de la Administración de Base de Datos y existen diversos enfoques para dar solución a este problema. En el presente trabajo se propone un modelo para apoyar a la Administración de Base de Datos basado en el concepto de Self-Management, el modelo da énfasis a los requerimientos particulares de una organización buscando que la función de Administración de Base de Datos sea más efectiva. El modelo propuesto se ha implementado utilizando un software que permite reducir los costos que ocasionan los problemas en aplicaciones con base de datos. Además, para aplicar el modelo propuesto se realizaron pruebas en una organización de producción.

Palabras clave: Administración de Base de Datos; Data Base Management System; Self-Management.

Abstract

Currently, organizations face two serious problems with respect to database management. The first problem is the increase in the volume of data and the complexity of the information requirements, the second problem is that database administrators with good knowledge and extensive experience are an increasingly scarce resource and its costs they have risen. Due to these problems, in recent years, database software providers seek to automate the functions of Database Administration and there are various approaches to solve this problem. In the present work a model is proposed to support the Database Management based on the concept of Self-Management, the model emphasizes the particular requirements of an organization looking for the function of Database Management to be more effective. The proposed model has been implemented using software that allows reducing the costs caused by problems in database applications. In addition, to apply the proposed model tests were conducted in a production company.

Keywords: Database Administration; Data Base Management System; Self-Management.

Correspondencia:

Dirección: Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática Unidad de Posgrado (Oficina 205 – 2° Piso)
Ciudad Universitaria - Av. Germán Amézaga s/n (Puerta N° 7)

Recibido: 19/03/2018 - Aceptado: 21/03/2018

Citar como:

Del Mar J. Un modelo de Self-Management para mejorar la eficiencia de la productividad en la Administración de una Base de Datos. Revista Peruana de Computación y Sistemas 2018 1(1):23-34. <http://dx.doi.org/10.15381/xxxxxx>

© Los autores. Este artículo es publicado por la Revista Peruana de Computación y Sistemas de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos. Este es un artículo de acceso abierto, distribuido bajo los términos de la licencia Creative Commons Atribución - No Comercial_Compartir Igual 4.0 Internacional. (<http://creativecommons.org/licenses/by-nc-sa/4.0/>) que permite el uso no comercial, distribución y reproducción en cualquier medio, siempre que la obra original sea debidamente citada.

1. Introducción

Con el avance tecnológico a nivel de hardware y en el desarrollo de software, la demanda sobre los requerimientos funcionales de los Data Base Management System (DBMS) ha aumentado significativamente. Ante este escenario, los proveedores de DBMS han reaccionado a estos requerimientos agregando características adicionales a sus productos, haciéndonos más sofisticados para administrarlos.

El mercado de DBMS es muy amplio y está en constante crecimiento y evolución, en la Figura 1 se presenta el cuadrante mágico de DBMS operacionales desarrollado por la consultora Gartner [1]. El Cuadrante Mágico de Gartner presenta un análisis de los proveedores de DBMS operacionales, analizándolos desde el punto de vista de las innovaciones del mercado y comparándolos en función de lo que Gartner denomina “amplitud de visión” y “capacidad de ejecución”.

Debido al entrono descrito, es que la industria de productos de DBMS y las universidades en el mundo están desarrollando esfuerzo e investigaciones para automatizar las funciones de la Administración de la Base de Datos (ABD).



Figura 1. Cuadrante mágico de DBMS operacionales [1]

En la actualidad, solo una parte de las funciones de ABD se han automatizado y los enfoques que se han desarrollado generalmente buscan automatizar las funciones de ABD en forma aislada. Los problemas generados por una carencia de automatización integral hacen que la automatización de las funciones de ABD no se efectúe adecuadamente y a un costo razonable, lo que genera deterioro en la eficiencia de la productividad de las base de datos, lo que afecta el servicio de atención a los clientes y usuarios de las organizaciones.

El objetivo principal del presente trabajo es presentar un modelo basado en el enfoque de Self-Management, que busca integrar los componentes de un sistema de base de datos, permitiendo solucionar los requerimientos específicos de una organización.

La estructura del presente artículo es la siguiente: En la sección 2 se bosqueja el marco teórico conceptual en el que se describen los conceptos alrededor de la ABD y del Self-Management; en la sección 3 se presenta las principales características de los modelos de Self-Management existentes; en la sección 4 se presenta el modelo propuesto, sus fundamentos y componentes; mientras que en la sección 5 se describe el software desarrollado para implementar el enfoque de Self-Management en Base de Datos; finalmente en la sección 6 se brindan las conclusiones de los resultados encontrados y las recomendaciones sobre trabajos futuros.

2. Fundamentación Teórica

2.1. Administración de Base de Datos.

La ABD es una función de la Tecnología de la Información que tiene por objetivo la instalación, configuración, actualización, administración, monitoreo, mantenimiento y seguridad de las bases de datos de una organización [2]. La función de ABD involucra el diseño e implementación de estrategias de base de datos, monitoreo del sistema y mejorar el rendimiento y planificar para futuros requerimientos. Adicionalmente, puede planificar, coordinar e implementar las medidas de seguridad para proteger las bases de datos.

Las principales tareas que generalmente se realizan en la ABD son [3]:

- Instalar y actualizar con las nuevas versiones el DBMS y el software relacionado.
- Asignar el almacenamiento requerido para las bases de datos y planificar por los requerimientos de almacenamiento futuros de las bases de datos.
- Crear y modificar las estructuras de datos a partir de la información dada por los desarrolladores de aplicaciones.
- Generar los usuarios y mantener el sistema de seguridad relacionado.
- Controlar y monitorear el acceso de los usuarios.
- Monitorear y optimizar la performance.
- Gestionar el respaldo y recuperación de las bases de datos.
- Contactar con el proveedor del DBMS para obtener soporte técnico.

Las tareas en la ABD pueden generar diversos cargos como [4]:

- DBA de Sistema.
- Arquitecto de Base de Datos.
- Analista de Base de Datos.
- Diseñador de Base de Datos.
- DBA de Aplicaciones.
- DBA orientado a una tarea.
- DBA Analista de rendimiento (performance).
- DBA Datawarehouse.

Inclusive en cada uno de estos cargos se dan diferentes enfoques dependiendo de los requerimientos de las organizaciones, por ejemplo, para el DBA de Aplicaciones se puede estructurar sus funciones tal como se describe en la Figura 2.

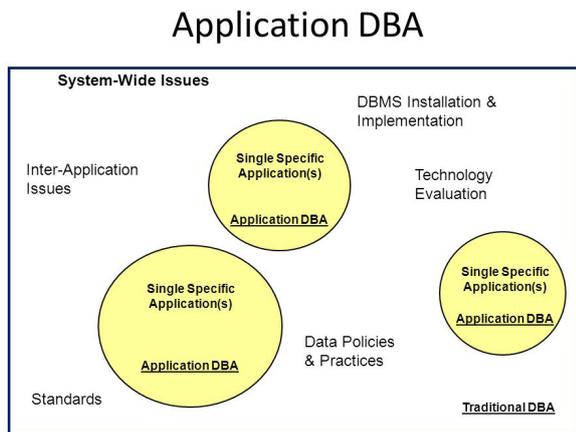


Figura 2. Funciones del DBA de Aplicaciones [4]

2.2. Software de Administración de Base de Datos.

El Software de Administración de Base de Datos o Data Base Management System (DBMS) es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, también se denomina Sistema de Gestión de Bases de Datos (SGBD) [5].

El objetivo fundamental de un DBMS es suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. Los programas de aplicación operan sobre los datos almacenados en la base utilizando las facilidades que brindan los DBMS, los que, en la mayoría de los casos, poseen lenguajes especiales de manipulación de la información que facilitan el trabajo de los usuarios [5].

En la Figura 3, se presenta los componentes funcionales de un DBMS que nos muestra la complejidad de la estructura interna de un DBMS, entre los componentes de un DBMS tenemos el gestor de archivos, el procesador de consultas, el optimizador de SQL, el gestor de accesos, etc.

En la actualidad, los DBMS se hacen más complejos y sofisticados debido a que tienden a dar soporte a una gran variedad de tecnologías como Inteligencia de Negocios, Big Data, Cloud, redes sociales, móviles, etc., y deben gestionar datos de cualquier tamaño, ya sean estructurados o sin estructurar, procesamiento de datos en batch y en línea.

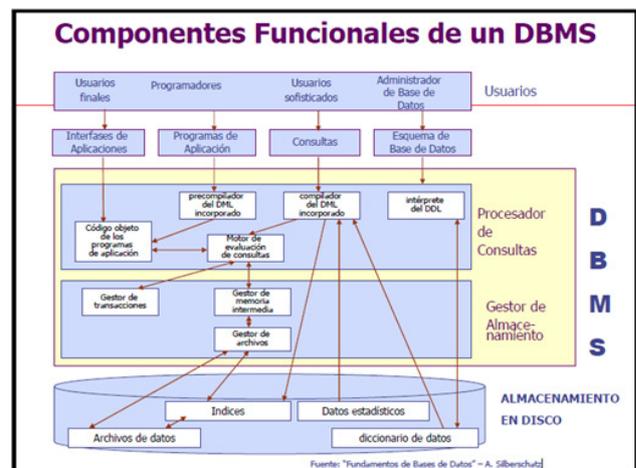


Figura 3. Componentes Funcionales de un DBMS [5]

2.3. Computación Automática.

En la actualidad, las personas y las organizaciones dependen de la Tecnología de la Información y las Comunicaciones (TIC). La TIC es una herramienta fundamental para nuestra vida cotidiana y para el desarrollo de las organizaciones. Tal como nos indica Hoyos y Valencia [21], cada vez son más evidentes las ventajas al adoptar TICs en las organizaciones, ya que desempeñan un rol fundamental en la generación de valor.

Uno de los problemas que enfrentan las organizaciones con las nuevas tendencias de TIC es la carencia de expertos de TIC. Debido a este factor, las organizaciones se ven afectadas con pérdidas económicas, tal como lo podemos ver en el reporte de Rockspace en colaboración con académicos del London School of Economics [22], es crítico para las organizaciones el problema de la escasez de expertos en TIC para su administración y funcionamiento.

El filósofo británico Alfred North Whitehead en su libro "Introducción a las Matemáticas", indico que la civilización avanza aumentando el número de operaciones importantes que podemos efectuar sin pensar en ellas. La enseñanza que nos dejó es que cuanto más podemos liberar nuestras mentes de tareas rutinarias, transfiriendo las tareas a asistentes tecnológicos, más poder mental podemos almacenar para las tareas más profundas y creativas de razonamiento [6].

Whitehead nos indicó que la solución a este problema está en la automatización de forma que los sistemas de TI puedan funcionar sin la intervención humana.

En el 2001, Paul Horn de IBM planteo un nuevo modelo de computación denominado Computación Automática o Automatic Computing, en su obra [7], indico que la complejidad creciente en los sistemas de TIC conjuntamente con una escasez de profesionales de TIC direcciona a una inevitable necesidad de automatizar muchas de las funciones asociados a los sistemas de TIC.

El modelo de Paul Horn, se basa en que los sistemas de TIC se auto regulen de la misma forma en que el sistema nervioso regula y protege el cuerpo humano en forma automática [8]. En la Figura 4 se presenta los componentes de la automatización, los gestores de la automatización y los elementos de la automatización.

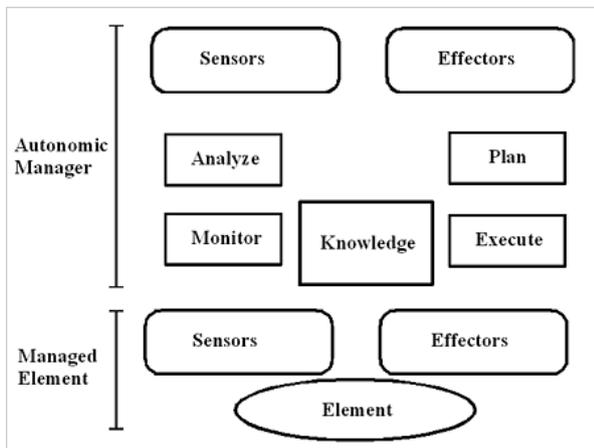


Figura 4. Componentes de la Automatización: Gestor y Elemento de Gestión [8]

2.4. Self-Management.

Self-Management (o administración autónoma), se ha convertido en un termino popular tanto en el estudio del comportamiento como en la Medicina [9], también es un termino muy utilizado en la Administración, donde el Self-Management consiste en brindar los elementos fundamentales de la Administración a la propia vida de los trabajadores, basándose en una premisa básica de que para dirigir a otros es preciso aprender a dirigirse a sí mismo, es decir, es el uso de cualquier método, habilidad o estrategia mediante la cual los participantes de un trabajo o actividad pueden lograr sus objetivos con autonomía [10]. El Self-Management se puede aplicar en los negocios, el trabajo y en la vida personal tal como se muestra en la Figura 5.

En el procesamiento de TIC, el Self-Management se aplica a la automatización de procesos tal como se plantea en la Computación Automática.



Figura 5. Self-Management en los negocios, el trabajo y la vida personal [22]

3. Estado del Arte del Self-Management en Base de Datos.

3.1. Tecnología de Self-Management en Base de Datos.

Hay dos tipos de productos de software que aplican el enfoque de Self-Management en base de datos:

3.1.1 Software de Self-Management fuera de línea

En los inicios del Self-Management en base de datos, los investigadores en esta área se orientaron a desarrollar software que permita buscar seleccionar los mejores índices para acceder a las tablas a partir de las sentencias SQL de la aplicación. Algunos productos que tienen esta característica son los siguientes:

- a. IBM DB2 Design Advisor.

El producto IBM DB2 Design Advisor permite seleccionar los índices más adecuados para mejorar la performance en las consultas a la base de datos de forma que los procesos tanto batch como en líneas puedan mejorar. IBM DB2 Design Advisor da recomendaciones a nivel del diseño físico de la base de datos [11].

- b. Microsoft SQL Server Database Tuning Advisor.

El producto Microsoft SQL Server Database Tuning Advisor al igual que IBM DB2 Design Advisor permite seleccionar los índices más adecuados a partir de las sentencias SQL, genera índices virtuales para tener sus índices candidatos a seleccionar y también da recomendaciones a nivel del diseño físico de la base de datos [12].

c. ORACLE SQL Tuning Advisor.

A diferencia de IBM DB2 Design Advisor y Microsoft SQL Server Database Tuning Advisor que dan recomendaciones a nivel de diseño físico, ORACLE SQL Tuning Advisor da recomendaciones a nivel de diseño físico, pero también da recomendaciones sobre estadísticas, planes y otras formas de replantear las consultas SQL [13].

3.1.2 *Software de Self-Management en línea.*

Si bien es cierto los productos de software de Self-Management fuera de línea fueron un buen aporte en el camino de automatización de los productos de software de base de datos, el ABD tenía que activar el software en determinado momento, es decir el ABD tenía que identificar en qué momento se producía un problema y debía activar el software de Self-Management. Sin embargo, el proceso de automatización todavía tenía deficiencias significativas. En el camino de cumplir los objetivos de automatización computacional, aparecen los productos de software de Self-Management en línea que están teniendo mucha aceptación por el mercado y por los proveedores de DBMS.

A continuación, presentaremos tres tareas en la que se aplican el enfoque de Self-Management mediante productos de software en línea [14]:

a. Gestión de memoria en línea.

En los DBMS hay una serie de áreas de memoria que se utilizan como las áreas de memoria asignada a los buffers del sistema, áreas de memoria asignada a ordenamiento de registros (sort), área de memoria asignada a los planes de ejecución y áreas de memoria asignada para los bloqueos (lock). Anteriormente estas áreas debían ser configuradas y reconfiguradas por el administrador de la base de datos, pero en las últimas versiones de los DBMS la configuración y reconfiguraciones las realizan dinámicamente los DBMS aplicando el enfoque de Self-Management.

b. Selección de índices en línea.

A la fecha se están realizando diversas investigaciones para automatizar la selección de índices en línea aplicando el enfoque de Self-Management

c. Recolección de estadísticas en línea.

El optimizador de planes de un DBMS genera los planes de acceso, para realizar adecuadamente esta tarea necesita de información, una de la cuales es la distribución de estadísticas sobre los datos. Los optimizadores de planes en los DBMS asumen algunas estadísticas que si son erradas pueden generar planes que no están bien optimizados, en la actualidad hay diversas investigaciones que han generado productos como LEarning Optimizer (LEO).

3.2. *Modelos de Self-Management en Base de Datos.*

De acuerdo a la literatura revisada en 3.1, vemos que en los últimos años los proveedores de DBMS y los investigadores del área de Base de Datos, buscan reducir los altos costos de mantenimiento en la ABD. El enfoque que se emplea para enfrentar este problema es aplicar los principios de computación automática [7] por lo que la tendencia es a desarrollar funciones de Self-Management en los DBMS que ayuden en la automatización de la ABD.

En esta parte del trabajo de investigación, se describen los modelos más recientes utilizados en Self-Management y los constructos que los componen. También se muestran algunos resultados obtenidos por los autores luego de las implementaciones que realizaron. En los DBMS se utilizan los siguientes modelos para aplicar el enfoque de Self-Management:

a. Modelo de Arquitectura DBMS tipo RISC.

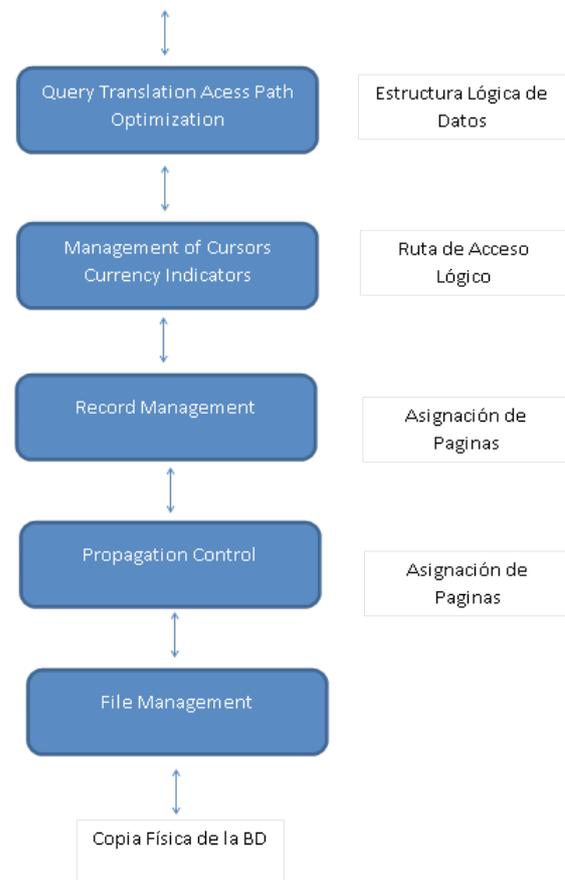


Figura 6. Estructura de un DBMS propuesto por Harder y Reuter [15]

En la implementación de un DBMS generalmente se realiza utilizando el enfoque por capas, tal como lo describe Harder y Reuter en [15]. Pero Weikum en [16], indica que la arquitectura de los actuales DBMS no está diseñada para una automatización total, por lo que plantea un cambio radical basado en los componentes de estilo RISC [17], en este enfoque, cada componente se supone que

tiene una función limitada, permitiendo predecir la performance del componente mediante modelos matemáticos. En el modelo resultante se supone que puede predecir cuantitativamente el comportamiento del componente RICS bajo todas las posibles configuraciones y características de carga de trabajo. Mediante la evaluación del modelo matemático, cada componente RICS puede realizar una optimización confiable (Self-Optimization).

Para poder implementar este modelo se requeriría una reingeniería total de un DBMS, inclusive las aplicaciones tendrían que rediseñarse para poder trabajar con el componente RICS. La implementación de este modelo puede ser una alternativa de solución a largo plazo, también hay que considerar que existen una gran cantidad de DBMS que deberían modificarse para poder implementar esta alternativa.

b. Modelo de Priorización de Consultas.

Con este modelo el DBMS puede internamente considerar las metas en cada estado del procesamiento de la sentencia. La estructura de trabajo descrita en [nmp+06] implica interceptar las consultas antes de que sean ejecutadas en el DBMS. Luego de interceptarlas son permitidas para ejecutar en un cierto punto en el tiempo basado en los costos estimados para su ejecución (esto es en un caso simple). En el caso de soluciones más sofisticadas en el que se requiere asignar recursos en consultas basadas en sus clases de servicio (por ejemplo, la cantidad de memoria en el buffer del sistema o la prioridad de los discos de lectura).

c. Modelo de Funciones Automáticas de Contexto Informado.

Los fabricantes de DBMS han diseñado un conjunto de manejadores que automatizan una tarea de administración particular para un componente del DBS. Para superar estos problemas, los manejadores automáticos tienen que estar informados del contexto donde trabajan.

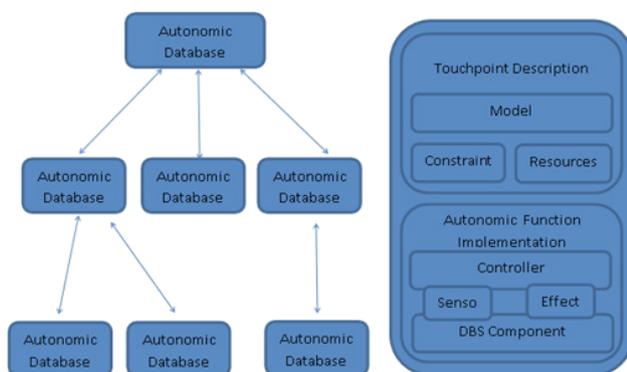


Figura 7. Jerarquía de Administradores Automáticos [18]

En el caso de las reconfiguraciones (sobre todo las que tienen efectos colaterales), tienen que haber acuerdos con las funciones automáticas que pueden ser afectadas por la reconfiguración. IBM propone la siguiente estructura [18], en el que cada función automática es controlada por su inmediato superior (dentro de la jerarquía), tal como se muestra en la Figura 7.

d. Modelo de Sistema Amplio de Self-Management.

El enfoque de sistema amplio de Self-Management [19] diseña una lógica única y centralizada de Self-Management, que tiene una visión integrada de todas las decisiones de Self-Management en el sistema de BD. La Figura 8 representa este enfoque, donde una única lógica de Self-Management monitorea el sensor de información brindado por el sistema de BD. El sensor de información esta compara las metas definidas por el DBA. Si el sensor de información requiere una configuración de BD, la lógica de Self-Management realiza un análisis ponderado de reconfiguración, que determina

los ajustes resultantes óptimos - considerando el estado actual, la carga de trabajo y la definición de metas. Durante el análisis de la reconfiguración la lógica Self-Management tiene que considerar todas las dependencias de los posibles cambios de configuración. Al igual que las funciones automáticas de contexto informado, el sistema amplio de Self-Management puede ser construido fuera del núcleo del DBMS y su lógica puede estar basado en los sensores existentes y los efectores del sistema de base de datos. Como la lógica de Self-Management opera en un nivel de sistema amplio, puede conectar los efectores individuales de configuración para predecir que también las metas de tiempo de respuesta, rendimiento, costos de recursos y disponibilidad.

De tal forma que los problemas de interacción e independencia de metas pueden ser resueltos con un recurso centralizado de Self-Management de sistema amplio. Adicionalmente, el modelo del sistema puede ser utilizado para almacenar modelos matemáticos cuantitativos del comportamiento del sistema de BD de forma que se pueda evitar reacciones excesivas.

Es importante indicar que todas las arquitecturas de Self-Management discutidas anteriormente son recursos validos hacia la meta de una automatización del DBMS verdadera. Sin embargo, el diseño general del recurso de sistema amplio de Self-Management esencialmente evita los problemas de interacción, desproporción e independencia de metas, por eso es que ha sido escogido como la guía para los conceptos de Self-Management desarrollados en este trabajo, pero incluso con estas guías muchos retos abiertos serán resueltos de forma de crear y automatizar un sistema de BD que cumpla los requerimientos hacia un sistema verdaderamente automático.

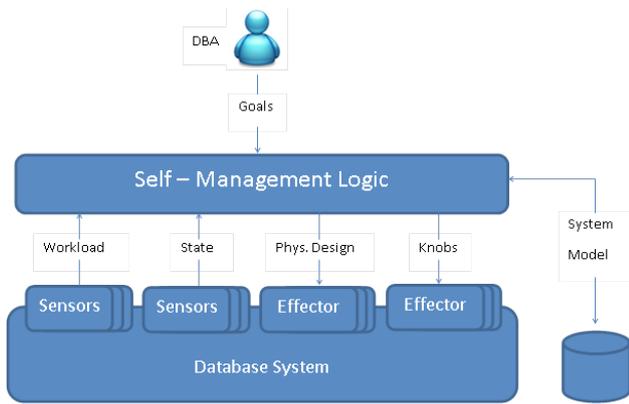


Figura 8. Modelo de Sistema Amplio de Self-Management [19]

4. Modelo de Self-Management en Base de Datos Propuesto.

Los modelos de Self-Management de la literatura y las implementadas (DB2, ORACLE, MS SQL Server, etc.) para la Administración de una Base de Datos se orientan a automatizar tareas comunes y rutinarias que realizan los ABD y las incorporan a los DBMS, como por ejemplo la actualización de estadísticas o la reorganización de tablas fragmentadas. Sin embargo, todas ellas no consideran problemas particulares de las empresas, pero que son críticas debido a que generan pérdidas económicas, de mercado, daña el servicio ofrecido, etc., por ejemplo las cancelaciones eventuales de procesos, es decir existen actividades que no están automatizadas y que no se pueden generalizar.

Las funciones básicas en la Administración General son [20]:

- Planificación
- Organización
- Monitoreo
- Análisis
- Transformación
- Evaluación

Siendo que las funciones de Planificar y Organizar están relacionadas más a la gestión, el modelo propuesto se fundamenta en las otras 4 funciones básicas (ver Figura 9).

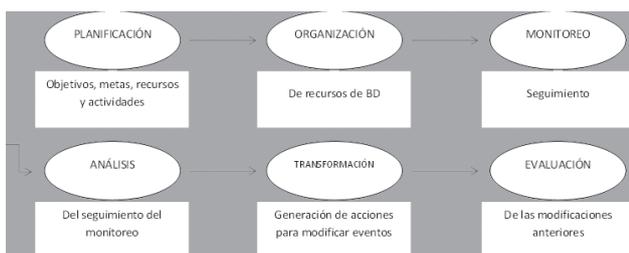


Figura 9. Funciones Básicas de la Administración General

Funciones consideradas en el modelo propuesto de Self-Management:

- La función de monitoreo consiste en el seguimiento de los procesos registrando datos con el fin de detectar alguna anomalía o interrupción del proceso.
- La función de análisis es la evaluación de los resultados del monitoreo para generar acciones, evaluando problemas o irregularidades que se encuentren. La función de análisis si encuentra un problema o irregularidad activara la ejecución de la función de transformación.
- La función de transformación permite realizar las acciones de base de datos requeridas para dar solución al incidente encontrado (problema o irregularidad).
- La función de evaluación nos permite controlar la correcta ejecución de las funciones de análisis y transformación.

Por otro lado, viendo el proceso de Self-Management en base de datos considerando el enfoque de la Administración General podríamos representarlo de la siguiente forma:

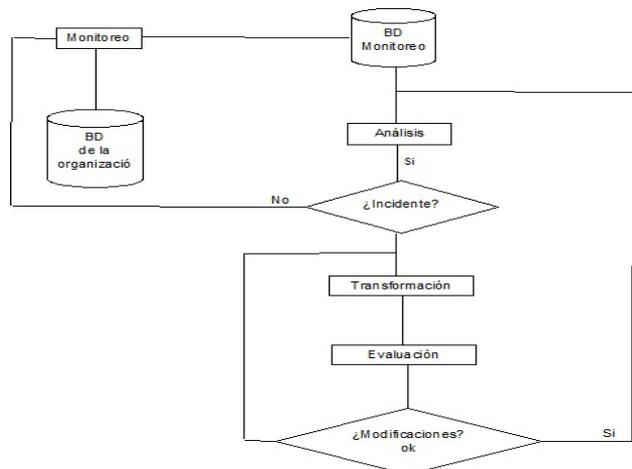


Figura 10. Proceso de Self-Management en Base de Datos

El modelo propuesto también se basa en la automatización de las 4 funciones, esto se realiza a través de un sistema experto. Las funciones de monitoreo y evaluación se automatizan mediante programas con instrucciones básicas. El monitoreo se hace registrando la información que se elige para detectar algún incidente o problema. En tanto que las funciones de análisis y transformación se realizan mediante un sistema experto, debido a que estas 2 funciones se pueden realizar mediante reglas dados por el especialista (experto). En tanto que las funciones de análisis y transformación se realizan mediante un sistema experto, debido a que estas 2 funciones se pueden realizar mediante reglas dados por el especialista (experto).

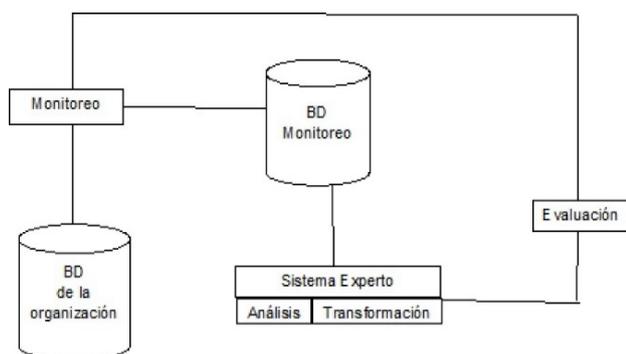


Figura 11. Modelo Propuesto de Self-Management en Base de Datos

En el caso de que ocurra una irregularidad no identificada, se requerirá de los especialistas para obtener las reglas que permita identificar dicha anomalía (análisis), y las reglas para corregirla (transformación)

Teniendo como objetivo encontrar el fundamento diferenciador de nuestro modelo, hemos buscado analizar el meta análisis y encontrar un aspecto en el podemos complementar el modelo planteado por Marc Holze [19] y uno de los puntos que consideramos importante es enfocar la función de Administración de Base de Datos desde el punto de la Administración General.

5. MATE.

MATE es la denominación del código que implementa el modelo propuesto, sus siglas provienen de: M monitoreo, A análisis, T transformación y E evaluación.

El desarrollo del software con enfoque de Self-Management se realizó en una empresa de producción, el software desarrollado permitió solucionar problemas en:

- La cancelación en la ejecución de procedimientos almacenados (stored procedures).
- El encolamiento de las transacciones de fraude.

5.1. Diseño de la Validación.

Con respecto al diseño de la validación para aplicar el enfoque de Self-Management en Base de Datos se dieron las siguientes consideraciones:

- La implementación del software de validación del modelo de Self-Management de base de datos se realizó en una de las principales empresas financieras del Perú, que por motivos de privacidad guardaremos en reserva su nombre comercial.
- Los casos de estudio que se consideraron para la validación fueron:

- Caso de estudio A: software de alerta en la ejecución de stored procedures.
 - Caso de estudio B: software de alerta en las transacciones de fraude.
- Los dos casos de estudio fueron desarrollados por el Administrador de la Base de Datos de la institución financiera para lo que realizó un monitoreo constante para determinar las ocurrencias del problema y el tiempo que le tomaba para solución estos problemas.

5.2. Casos de estudios.

Para la validación del modelo de Self-Management en base de datos se desarrollarán dos casos de estudios los que describiremos a continuación:

Caso de estudio A: software de alerta en la ejecución de stored procedures.

Uno de los problemas críticos que se presentaban en la organización de producción es que por limitación de recursos computacionales o por cancelación en la lógica de la aplicación se suspendía el procesamiento de transacciones debido a que un stored procedures quedaba en estado detenido, y ya no podía volverse a ejecutar el stored procedures hasta que se le daba un comando de reinicio.

El problema que el stored procedure quede detenido por cancelación es que el administrador de la base de datos tenía que estar constantemente monitoreando que los stored procedure estén activos y cuando uno de ellos estaba desactivado tenía que activarlo con el comando de reinicio. El problema es que el administrador de la base de datos tenía que estar alerta las 24 horas del día los 360 días del año.

Cuando por algún motivo cancelaba un stored procedure y el administrador de la base de datos no se percataba de este evento, la empresa de producción tenía pérdidas económicas y de imagen empresarial debido a que no se brindaba un servicio adecuado como esperaban los clientes. Los errores de los stored procedure generaban malestar en los clientes de la empresa de producción. En algunos casos las empresas clientes se retiraban de los servicios que tenían con la empresa de producción.

Este problema nos brindó la oportunidad de aplicar el concepto de Self-Management en base de datos, la solución que se implementó fue que cada minuto se active un proceso que mediante un comando revise el estado de los stored procedure, el resultado de esta revisión era leído por otro programa que identificaba que stored procedure estaban detenidos y generaba un proceso que generaba comandos para que se activen los stored procedures detenidos, en forma automática sin intervención manual.

Caso de estudio B: software de alerta en problemas con registros de operaciones.

En la empresa de producción en que realizamos este caso de estudio, algunas de las operaciones son pasadas a otro sistema que las analiza para ver si tienen algún problema.

En ese proceso se guardaban las operaciones y luego son pasadas al otro sistema. En un determinado momento la aplicación grababa registros y se demoraba para enviar al otro sistema que los analizaba y se producía un encolamiento de registros que no eran procesadas, lo que generaba problemas a la empresa de producción.

Inicialmente no se encontraba dónde estaba el error, no se sabía si el error estaba en la aplicación o debido a otro motivo. Para que no se continúe produciendo el encolamiento de nuevos registros se consideró generar un algoritmo para evitar el encolamiento, pero esta solución era manual y había que estar atento a que se produzca el encolamiento, lo que implicaba un monitoreo de 24 horas por 7 días a la semana y en muchas ocasiones se producían encolamiento en un descuido.

En este caso se implementa una solución aplicando el enfoque de Self-Management en que se define un rango máximo de encolamiento y cada dos minutos se activa un proceso que analizaba si había problemas y tomaba acción, los problemas de encolamiento se solucionaron con las acciones tomadas.

5.3. Implementación del Software de Validación.

La validación del modelo propuesto es necesaria ya que brinda un grado de confianza y seguridad en el modelo y en los resultados que se obtienen al aplicarlo. En el presente trabajo para realizar la validación se desarrollaron dos casos de estudio que fueron descritos en la sección anterior.

La implementación del software de validación del modelo propuesto de Self-Management de base de datos se desarrolló en el siguiente entorno:

a. Hardware.

La empresa de producción del caso de estudios tiene la siguiente arquitectura de hardware:

- Computador HP
- Discos Hitachi

b. Software Base.

Con respecto al software del sistema operativo y los productos de software relacionados a él, se tiene los siguientes productos:

- Sistema operativo Windows Server 2012
- DBMS MS SQL Server 2014.

c. Software de Aplicación.

- Software desarrollado por la empresa de producción.

- Software de análisis desarrollado por un proveedor de software.

d. Ambiente de Producción.

- Se tienen 89 bases de datos
- Hay 369 stored procedures utilizados por las aplicaciones.
- La tabla de registro de operaciones llega a tener hasta 4.5 millones de registros que se generan diariamente.

5.4. Métricas de Medición.

Las métricas de medición para la validación de la implementación del enfoque de Self-Management en base de datos se estructuraron desde dos puntos de vista, uno desde el punto de vista del sistema de base de datos otro desde el punto de vista del Administrador de la Base de Datos.

5.4.1 Métricas de Medición: Eficiencia del Sistema de Base de Datos.

Caso A: Software de alerta en la ejecución de stored procedures.

En este caso mediremos la eficiencia por la cantidad de veces que se presenta el problema de cancelación de stored procedures en un periodo de un mes.

Caso B: Software de alerta en las transacciones de operaciones.

En este caso mediremos la eficiencia por la cantidad de veces que se presenta el problema de encolamiento en un periodo de un mes.

5.4.2 Métricas de Medición: Eficiencia del Sistema Administrador de Base de Datos.

Caso A: Software de alerta en la ejecución de stored procedures.

En este caso mediremos la eficiencia por la cantidad de veces que se presenta el problema de cancelación de stored procedures y por el tiempo empleado por el administrador de base de datos en un periodo de un mes.

Caso B: Software de alerta en la ejecución de operaciones.

En este caso mediremos la eficiencia por la cantidad de veces que se presenta el problema de encolamiento y por el tiempo empleado por el administrador de base de datos en un periodo de un mes.

5.5. Resultados.

La ejecución de los casos de estudio se realizó en el ambiente de Producción de la empresa de producción

y se obtuvo una eficiencia de 100 por ciento aplicando el modelo propuesto mediante el software MATE.

6. Conclusiones y Trabajo Futuros.

El presente trabajo desarrolla un modelo para apoyar a la Administración de Base de Datos basado en el concepto de Self-Management. En la sección 6.1 se brinda un conjunto de conclusiones obtenidas en la realización del trabajo desarrollado. En la sección 6.2 se brinda una relación de estudios que se pueden desarrollar en el futuro para continuar el presente trabajo.

6.1. Conclusiones.

Al final del presente trabajo podemos indicar las siguientes conclusiones:

- Con el trabajo realizado, al modelo de Self-Management para base de datos desarrollado Marc Holze [19] se le ha agregado un componente en el que se le da énfasis a los requerimientos particulares de una organización con respecto a la Administración de Base de Datos, generándose soluciones basadas en procedimientos que pueden aplicarse o adaptarse a las organizaciones independientemente de los DBMS que tengan implementados.
- En el presente trabajo para probar la aplicación del enfoque de Self-Management, se utilizó procedimientos orientados a MS SQL Server, permitiendo en algunos casos solucionar en corto tiempo problemas que podrían haber tomado mucho tiempo encontrar una solución y evitando que se produzcan pérdidas que afecten a la organización.

6.2. Trabajos Futuros.

El objetivo del presente trabajo es aplicar conceptos de Self-Management en la Administración de una Base de Datos en una organización financiera con un alto volumen de transacciones en línea y procesos batch.

Las implementaciones desarrolladas existen como un complemento adicional a las investigaciones y desarrollo realizadas por Marc Holze [19] Pero aún faltan muchas áreas por investigar y desarrollar donde se requieren estudios adicionales y evaluaciones experimentales de forma que el modelo de Self-Management se considere e un nivel de madurez mayor, por lo que hemos considerado que en el futuro se podrían considerar los siguientes aspectos:

En el desarrollo de su trabajo, Marc Holze [19] no dio mucho énfasis a aspectos relacionados con la confiabilidad del servicio, es decir, que se asegure que las metras de alto nivel definido por el administrador de base de datos sean conocidos en todas las condiciones.

La importancia de este aspecto radica en que las violaciones a las metas sean detectadas lo más pronto

posible debido a que se activaran las reconfiguraciones necesarias para evitar los desvíos con respecto a las metas definidas.

En el trabajo que hemos desarrollado, hemos considerado niveles de fragmentación en los que se debe activar la reorganización. Pero faltan otros indicadores a los que se puede automatizar y que se puede continuar investigando.

Estos aspectos pueden ser detectados por tendencias o por otro tipo de técnicas de análisis, en un futuro se podría trabajar este aspecto para complementar el trabajo de Marc Holze [19].

El monitoreo anticipado tampoco fue considerado en el trabajo Marc Holze [19]. El objetivo del monitoreo es tener un monitoreo continuo que no genere sobrecarga al sistema.

En el trabajo desarrollado hemos utilizados algunas consultas a las tablas del catálogo del DBMS y eso nos ha permitido analizar y anticiparnos a los procesos de reorganización innecesarios por mantenimiento fijo y evitando que los problemas de se presenten (en forma proactiva).

En el presente trabajo se ha deseado complementar el trabajo de investigación de Marc Holze [19] y se ha aplicado utilizando MS SQL Server. Sería recomendable aplicar dichos conceptos en otros DBMS como ORACLE, DB2, etc. e inclusive en diversas versiones debido a que tienen otros componentes y parámetros de configuración. También es importante considerar la integración con productos de software que están relacionados conel DBMS como por ejemplo en el caso de MS SQL Server, Integration Services o Reporting Services.

Otro aspecto que se puede considerar en el futuro es la simulación de la carga de trabajo (workload), es decir se puede simular diferentes escenarios de carga para predecir el comportamiento del sistema de base de datos, eso nos permitirá definir los recursos necesarios en el futuro. También se puede simular cambios en la configuración, así como en las metas.

7. Referencias Bibliográficas.

- [1] Feinberg, Donald & Adrian, Mery & Heudecker, Nick, "Gartner Magic Quadrant for Operational Database Management Systems", Noviembre del 2017 G00317993, Stamford, Connecticut, USA.
- [2] Pratt, Philip & Adamski, Joseph, "Concepts of Database Management", Cengage Learning, 2011, USA.
- [3] Mullins, Craig, "Why DBAs Must Be Jacks-of-All-Trades", Pearson Education, Informit, 2018, USA.
- [4] Mullins, Craig "Database Administration: The Complete Guide to Practices and Procedures", Addison Wesley Professional, 2012, USA.
- [5] Elmasri, Ramez & Navathe, Shamkant "Fundamentals of Database System". Editorial Addison-Wesley, 6ta Edición, 2010, USA.

- [6] Whitehead, Alfred North, "Introducción a las Matemáticas", Editorial: Emecé, Biblioteca Emecé de Obras Universales, Sección II, Ciencias, Buenos Aires, 1949, Argentina.
- [7] Horn, P.: "Autonomic computing: IBM perspective on the state of information technology", USA, 2001.
- [8] "Migration to Autonomic Computing" <http://www.onjava.com/2005/11/30/what-is-on-demand-computing.html>
- [9] Lorig, Kate & Holman Halsted, "Self-Management Education: History, Definition, Outcomes, and Mechanisms", Stanford University School of Medicine, 2003, USA.
- [10] Kälkäjä, Maria, "Self-Management and its part in knowledge workers' experience of high performance", tesis de maestría Department of Management and International Business, OULU Business School, University of OULU, 2015, Finlandia.
- [11] Skelley, Alan, "DB2 Advisor: An Optimizer Smart Enough to Recommend its own Indexes", Proceeding ICDE '00 Proceedings of the 16th International Conference on Data Engineering, 2000, USA.
- [12] Bertucci, Paul & Silverstein, Alex & Gallelli, Chris & Rankins, Ray "Microsoft SQL Server 2014 Unleashed", Sams Publisher, 2015, USA.
- [13] Padfield, Bill; Alapati, Sam & Kuhn, Darl, "Expert Indexing in Oracle Database 11g: Maximum Performance for Your Database", Apress Media, 2011, USA.
- [14] Oracle, "Oracle Database 10g: The Self-Managing Database", An Oracle White Paper, Nov. 2003.
- [15] Harder, T; A. "Concepts for Implementing a Centralized Database Management System", Proceedings International Computing Symposium on Application Systems Development, 1983.
- [16] Weikum, Gerhard; Kumar, Sadish, "Self-Tuning Database Technology and Information Services, 2002.
- [17] Chen, Whei-Jen; Comeau, Bill; Ichikawa, Tomoko; Kumar; Sadish; Miskimen, Marcia, "DB2 Workload Manager for Linux, UNIX, and Windows, International Business Machine Corporation 2008, USA.
- [18] IBM, "IBM: An Architectural blueprint for automatic computing" 2005, USA.
- [19] Holze, Marc, Self-Management Concepts for Relational Database Systems. Tesis de Doctorado en Matemática, Informática y Ciencias Naturales, Universidad de Hamburg, 2012, Alemania.
- [20] Robbins, Stephen & Coulter, Mary, "Administración" 12ava edición, Pearson Educación de México, 2014, México
- [21] Hoyos Jonathan & Valencia Alejandro, "El papel de las TIC en el entorno organizacional de las PYMES", revista Trilogía número 7, 2012, Colombia.
- [22] Nicola Davies, "Your Self-Management Toolbox", Pain-Free Living <https://www.painfreelivinglife.com/tools-chronic-pain/emotional-health/self-management-toolbox/>, 2018, USA

