



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

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

TESIS

Para optar el Grado Académico de Magíster en Ingeniería de
Sistemas e Informática con mención en Dirección y Gestión de
Tecnología de Información

AUTOR

Jorge Luis DEL MAR ARZOLA

ASESOR

David Santos MAURICIO SÁNCHEZ

Lima, Perú

2020



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Del Mar, J. (2020). *Un modelo de self-management para mejorar la eficiencia de la productividad en la administración de una base de datos*. Tesis para optar el grado de Magíster en Ingeniería de Sistemas e Informática con mención en Dirección y Gestión de Tecnología de Información. Unidad de Posgrado, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

Información complementaria

Código ORCID del asesor	0000-0001-9262-626X
Autor DNI (Obligatorio) Pasaporte /carnet de extranjería (sólo extranjeros)	08232375
Asesor DNI (Obligatorio)	6445495
Código ORCID del autor	0000-0002-3218-9229
Grupo de investigación	Inteligencia Artificial
Financiamiento	_____
Ubicación geográfica donde se desarrolló la investigación (incluir localidades y/o coordenadas geográficas).	12.0464° S. 77.0428° W
Año o rango de años que la investigación abarcó.	2017 - 2020
Disciplinas OCDE	Ingeniería de Sistemas y Comunicaciones http://purl.org/pe-repo/ocde/ford#2.02.04



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

**ACTA DE SUSTENTACIÓN VIRTUAL DE TESIS PARA OPTAR EL GRADO
ACADÉMICO DE MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON
MENCION EN DIRECCIÓN Y GESTIÓN DE TECNOLOGÍA DE INFORMACIÓN**

A los ventidos (22) días del mes de octubre del 2020, siendo las dieciocho horas, se reunieron en la sala virtual <https://meet.google.com/shv-gojm-yck>, el Jurado de Tesis conformado por los siguientes docentes:

Mg. Juan Carlos Gonzales Suárez (Presidente)
Mg. Marco Antonio Coral Ygnacio (Miembro)
Dr. Willy Ugarte Rojas (Miembro)
Dr. David Santos Mauricio Sánchez (Asesor)

Se inició la Sustentación invitando al candidato a Magíster **Jorge Luis Del Mar Arzola**, para que realice la exposición oral y virtual de la tesis para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Dirección y Gestión de Tecnología de Información, siendo la Tesis intitulada:

“Un Modelo de Self-Management para Mejorar la Eficiencia de la Productividad en la Administración de una Base de datos”

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por la graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

16 (Dieciséis) Bueno

Por tanto el Presidente del Jurado, de acuerdo al Reglamento General de Estudios de Posgrado, otorga al Bachiller **Jorge Luis Del Mar Arzola** el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Dirección y Gestión de Tecnología de Información.

Siendo las...19:00..... horas, el Presidente del Jurado de Tesis da por concluido el acto académico de Sustentación de Tesis.

Mg. **Juan Carlos Gonzales Suárez**
(Presidente)

Dr. **Willy Ugarte Rojas**
(Miembro)

 Firmado digitalmente por CORAL
YGNACIO Marco Antonio FAU
20148092282 soft
Motivo: Soy el autor del documento
Fecha: 23.10.2020 15:19:24 -05:00

Mg. **Marco Antonio Coral Ygnacio**
(Miembro)

Dr. **David Santos Mauricio Sánchez**
(Asesor)

DEDICATORIA

A Dios por guiarme en mi camino y ser mi compañía en todas las etapas de mi vida y porque me ha dado la fortaleza y el apoyo para superar las dificultades, terminar mis estudios y trabajos de investigación.

A mi madre Ana y a mi padre Jorge por su constante amor, por haber sembrado en mí la semilla del esfuerzo, la constancia, la dedicación, el honor, la lealtad, la verdad y porque han sido los mejores maestros de mi vida con su ejemplo y guía permanente.

A mi esposa Teresa y a mis hijos Anita y Jorgito por ser el motor que me impulsa a seguir perfeccionándome, por el apoyo y dedicación que me brindaron para mi desarrollo personal y profesional, y porque siempre me brindaron su confianza y amor.

A los estudiantes, profesores, investigadores y a todas las personas que siguen el camino del desarrollo de la automatización de procesos para brindarnos un mundo mejor

AGRADECIMIENTOS

A mi asesor de tesis y profesor Dr. David Mauricio Sánchez, por su apoyo, orientación, paciencia, revisiones y motivación para el desarrollo del presente trabajo de investigación.

A todos los directores y profesores de la Unidad de Posgrado de la Facultad de Ingeniería de Sistemas e Informática de la UNMSM durante el tiempo que estude, por el apoyo brindado.

A mis compañeros de la primera promoción de ingresantes a la maestría de Ingeniería de Sistemas e Informática de la UNMSM, en especial a Jorge Díaz, Pablo Romero, Carlos Canepa, William Enríquez y Juan José Zamudio por el apoyo brindado durante los años de estudio.

A mi esposa Teresa y a mis hijos Anita y Jorgito por su amor, apoyo, paciencia, motivación y por compartir el ideal de que con sacrificio, esfuerzo y dedicación nuestros sueños se vuelven alcanzables.

A mi madre Ana y a mi padre Jorge por su amor y enseñanzas y por desear siempre lo mejor para mi vida, gracias por ser mi fuente inagotable de aliento para seguir adelante.

A Dios, a la Virgen del Carmen y a la Virgen de Guadalupe porque todo es posible con su apoyo y bendición.

“Lo que no se puede medir no se puede controlar; lo que no se puede controlar no se puede gestionar; lo que no se puede gestionar no se puede mejorar”.

PETER DRUCKER (1909-2005)

(Prominente intelectual, sus estudios e investigaciones son trascendentales en el desarrollo de los conceptos fundamentales de la Administración)

ÍNDICE GENERAL

AGRADECIMIENTOS	- 4 -
LISTA DE TABLAS	- 9 -
CAPÍTULO 1: INTRODUCCIÓN	- 13 -
1.1 Antecedentes.	- 13 -
1.2 Definición del Problema.	- 18 -
1.3 Objetivos.	- 19 -
1.3.1 Objetivo General.	- 19 -
1.3.2 Objetivo Específicos.	- 20 -
1.4 Justificación.....	- 21 -
1.4.1 Justificación Práctica.....	- 21 -
1.4.2 Justificación Teórica.	- 21 -
1.5 Propuesta de la Tesis.	- 22 -
1.6 Organización de la Tesis.	- 23 -
CAPÍTULO 2: MARCO TEORICO	- 26 -
2.1 Administración de Base de Datos.	- 26 -
2.2 Software de Administración de Base de Datos.	- 29 -
2.4 Computación Automática.	- 33 -
2.5 Automatización de Procesos.	- 35 -
2.6 Automatización de Procesos en Base de Datos.....	- 37 -
2.7 Sistema Experto.	- 38 -
2.7.1 Características de un Sistema Experto.	- 38 -
2.7.2 Ventajas de un Sistema Experto.....	- 39 -
2.7.3 Componentes de un Sistema Experto.....	- 40 -
2.8 Self-Management.	- 43 -
2.9 Resumen del Capítulo.	- 45 -
CAPÍTULO 3: ESTADO DEL ARTE.....	- 46 -
3.1 Evolución de la Administración de Base de Datos.	- 46 -
3.2 Modelos en Base de Datos.	- 61 -
3.2.1 Modelo de Arquitectura DBMS tipo RISC.....	- 62 -
3.2.2 Modelo de Priorización de Consultas.	- 64 -
3.2.3 Modelo de Funciones Automáticas de Contexto Informado.....	- 64 -
3.2.4 Modelo de Sistema Total de Self-Management.....	- 65 -
3.3 Software de Self-Management en Base de Datos.	- 67 -
3.3.1 Software de Self-Management fuera de línea.	- 67 -
3.3.2 Software de Self-Management en línea.	- 70 -
3.4 Resumen del Capítulo.	- 72 -
CAPÍTULO 4: MODELO PROPUESTO DE SELF-MANAGEMENT EN BASE DE DATOS	- 73 -
4.1 Motivación del Modelo Propuesto	- 74 -
4.2 Fundamentos del Modelo Propuesto.....	- 75 -
4.3 Modelo Propuesto.	- 77 -
4.4 Descripción Específica del Modelo Propuesto.	- 78 -
4.4.1 Monitoreo.....	- 79 -
4.4.2 Análisis.....	- 79 -
4.4.3 Transformación.	- 80 -
4.4.4 Evaluación.....	- 81 -

4.5 Restricciones del Modelo.....	- 83 -
4.6 Resumen del Capítulo.	- 83 -
CAPÍTULO 5: SOFTWARE DESARROLLADO PARA IMPLEMENTAR.....	- 84 -
SELF-MANAGEMENT EN BASE DE DATOS	- 84 -
5.1 Arquitectura del Sistema.	- 84 -
5.2 Componentes.....	- 88 -
5.2.1 Base de Hechos.	- 88 -
5.2.2 Modulo de Monitoreo.	- 89 -
5.2.3 Modulo de Análisis.	- 89 -
5.2.4 Modulo de Transformación.....	- 89 -
5.2.5 Modulo de Evaluación.	- 90 -
5.2.6 Interface de Usuario.	- 90 -
5.3 Resumen del Capítulo.	- 90 -
CAPÍTULO 6: VALIDACION DEL MODELO PROPUESTO DE SELF-	
MANAGEMENT EN BASE DE DATOS.....	- 91 -
6.1 Diseño de la Validación.	- 91 -
6.2 Casos de estudio.....	- 92 -
6.3 Implementación del Software de Validación.	- 95 -
6.4 Métricas de Medición.....	- 98 -
6.4.1 Métricas de Medición: Eficiencia del Sistema de Base de Datos.	- 98 -
6.4.2 Métricas de Medición: Eficiencia del Sistema del Administrador de Base de Datos.	- 100 -
6.5 Resumen del Capítulo.	- 105 -
CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS	- 106 -
7.1 Conclusiones.	- 106 -
7.1.1 Conclusión General.....	- 106 -
7.1.2 Conclusiones Especificas.	- 106 -
7.2 Limitaciones.....	- 108 -
7.3 Trabajos Futuros.	- 108 -
REFERENCIAS BIBLIOGRÁFICAS.....	- 110 -

LISTA DE FIGURAS

<i>Figura 1.</i> Edgar F. Codd (Edgar, Codd & Turing, 2012, p. 12).	- 16 -
<i>Figura 2.</i> Cuadrante mágico de DBMS operacionales (Feinberg, Adrian & Heudecker, 2017, p. 57)	- 17 -
<i>Figura 3.</i> Como trabaja el afinamiento automático en SQL Azure (Popovic, Ljepava & Raveler, 2018, p. 26)	- 19 -
<i>Figura 4.</i> Actualización automática de parches de MS SQL Server (Kuhnash, & Salyer, 2013, p. 88).	- 22 -
<i>Figura 5.</i> Organización de la Tesis	- 25 -
<i>Figura 6.</i> Funciones del Administrador de Base de Datos de Aplicaciones (Mullins, 2015, p. 32)	- 28 -
<i>Figura 7.</i> Componentes Funcionales de un DBMS	- 30 -
<i>Figura 8.</i> Componentes de la Automatización: Gestor y Elemento de Gestión [8] (Horn 2001).	- 35 -
<i>Figura 9.</i> BPM 360 grados (Club BMP,2009).....	- 36 -
<i>Figura 10.</i> Self-Management en los negocios, el trabajo	- 44 -
<i>Figura 11.</i> Historia de los DBMS (Shah, 2013, p. 92).	- 47 -
<i>Figura 12.</i> Datawarehouse (PowerData, 2002, p. 48).....	- 49 -
<i>Figura 13.</i> Proyecto AutoAdmin	- 51 -
<i>Figura 14.</i> Producto Database Engine Tuning Advisor.....	- 52 -
<i>Figura 15.</i> Administración de Memoria Automática DB2.....	- 55 -
<i>Figura 16.</i> Self- Management en ORACLE 11g	- 57 -
<i>Figura 17.</i> Self-Tuning ORACLE 9i	- 59 -
<i>Figura 18.</i> Desfragmentación Automática en MS SQL Server 2008 R2.....	- 61 -
<i>Figura 19.</i> Estructura de un DBMS propuesto por Harder y Reuter (Harder & Reuter 1983, p. 95).	- 63 -
<i>Figura 20.</i> Jerarquía de Administradores Automáticos (IBM, 2005, p. 93).....	- 65 -
<i>Figura 21.</i> Modelo de Sistema Amplio de Self-Management (Holze, 2012, p. 71).....	- 66 -
<i>Figura 22.</i> IBM DB2 Design Advisor (Quest Software Inc., 2018, p. 67).....	- 68 -
<i>Figura 23.</i> Microsoft SQL Server Database Tuning Advisor (Bertucci, Silverstein, Gallelli & Rankins, 2015, p. 33)	- 69 -
<i>Figura 24.</i> ORACLE SQL Tuning Advisor (ORACLE, 2018, p. 93).....	- 70 -
<i>Figura 25.</i> Funciones Básicas de la Administración General (Robbins & Coulter, 2014, p. 94)	- 75 -
<i>Figura 26.</i> Proceso de Self-Management en Base de Datos	- 77 -
<i>Figura 27.</i> Modelo Propuesto de Self-Management en Base de Datos	- 78 -
<i>Figura 28.</i> Modelo Propuesto de Self-Management de Base de Datos	- 82 -

LISTA DE TABLAS

Tabla 1 <i>Registro de Incidentes 1</i>	- 101 -
Tabla 2 <i>Registro de Incidentes 2</i>	- 102 -
Tabla 3 <i>Registro de Incidentes 3</i>	- 103 -
Tabla 4 <i>Registro de Incidentes 4</i>	- 104 -
Tabla 5 <i>Resumen de los Resultados</i>	- 105 -

TABLA DE ACRÓNIMOS

ACRONIMO	DESCRIPCION O SIGNIFICADO
ABD	Administración de Base de Datos
BPA	Business Process Automation
BPM	Business Process Management
DBMS	Data Base Management System o Sistema de Administración de Base de Datos
DTA	Database Engine Tuning Advisor
IBM	International Business Machine
ITW	Index Tuning Wizard
OLAP	OnLine Analytical Processing
OLTP	OnLine Transaction Processing
SBD	Sistema de Base de Datos
SE	Sistema Experto
SGA	System Global Area
SGBD	Sistema de Gestión de Base de Datos
PGA	Program Global Area
RICS	Reduced Instruction Set Computer
TCO	Costo total de la propiedad
TIC	Tecnología de la Información y las Comunicaciones

UN MODELO DE SELF-MANAGEMENT PARA MEJORAR LA EFICIENCIA DE LA PRODUCTIVIDAD EN LA ADMINISTRACIÓN DE UNA BASE DE DATOS

RESUMEN

En los últimos años, las organizaciones enfrentan dos graves problemas con respecto a la Administración de Base de Datos. El primero de los problemas es un mayor incremento en el volumen de datos que se almacenan en una base de datos y en la complejidad en 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 dos problemas críticos, los proveedores de software de base de datos buscan automatizar las funciones de Administración de Base de Datos y se han desarrollado diversos enfoques de automatización, sin embargo, no han considerado los problemas críticos específicos de Administración de Base de Datos que tienen las organizaciones.

En el presente trabajo de tesis se propone un modelo para apoyar a la Administración de Base de Datos, que está basado en el concepto de Self-Management, el modelo propuesto da énfasis a los requerimientos por problemas críticos específicos de una organización, buscando que la función de Administración de Base de Datos sea más eficiente. El modelo propuesto se ha implementado utilizando un software que permite reducir los costos que ocasionan los problemas en las aplicaciones con base de datos. Para validar el modelo de Self-Management propuesto se realizaron pruebas en una empresa financiera de nuestro país, obteniendo como resultado solucionar problemas particulares generados por las aplicaciones, una disminución en los costos y un aumento en la productividad en la Administración de Base de Datos.

Palabras Claves: Self-Management, Administración de Base de Datos, Data Base Management System, Productividad, Sistema Experto.

A MODEL OF SELF-MANAGEMENT TO IMPROVE PRODUCTIVITY EFFICIENCY OF DATABASE ADMINISTRATION

ABSTRACT

In recent years, organizations face two serious problems with respect to Database Administration. The first problem is a greater increase in the volume of data stored in a database and in 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 their costs have risen. Due to these two critical problems, the database software providers seek to automate the functions of Database Administration and various automation approaches have been developed, however, they have not considered the specific critical problems of Database Administration. that the organizations have.

In this thesis work we propose a model to support Database Management, which is based on the concept of Self-Management, the proposed model emphasizes the requirements for specific critical problems of an organization, looking for the Database Management function is more efficient. The proposed model has been implemented using software that reduces the costs caused by problems in database applications. To validate the proposed Self-Management model, tests were conducted in a financial company in our country, resulting in solving particular application-generated problems, lower costs, and increased productivity in Database Management.

Keywords: Self-Management, Database Administration, Data Base Management System, Productivity, Expert System.

CAPÍTULO 1:

INTRODUCCIÓN

1.1 Antecedentes.

Los avances desarrollados en la Tecnología de la Información y las Comunicaciones han transformado el modo de vida de nuestra sociedad. El desarrollo en la Ciencia de los Datos, las comunicaciones, la Internet, el correo electrónico, las redes sociales, la Inteligencia Artificial, el comercio electrónico, las plataformas virtuales en la Educación, los robots, etc., han generado grandes cambios en las organizaciones, han aparecido nuevas formas de realizar negocios, de aprender y se han dado cambios en los hábitos de consumo de las personas (Gil, O. & Rosas, 2012, p. 65).

Como consecuencia del avance tecnológico se ha generado nuevos retos y nuevas oportunidades para todos y la información ha tomado gran importancia, en la Nueva Economía quien tenga información (confiable, actualizada y oportuna) tendrá el poder y podrá dominar el mercado (Hoyos & Valencia, 2012, p. 71).

El impacto del poder de la información se hace notorio en el ámbito de las organizaciones debido a que buscan la continuidad, desarrollo y la sobrevivencia o si no se enfrentan a la desaparición (Hoyos & Valencia, 2012, p. 72).

Con la tecnología de Big Data, la cantidad de datos que se generan y almacenan es impresionante y crece a una velocidad nunca vista, Bernard Marr un prestigioso consultor nos brinda algunas estadísticas que son interesantes de tener en cuenta (Sánchez, 2013, p. 23):

- a) Se estima que para el año 2020 cada persona generará 1.7 megabytes de datos por segundo y que el parque de datos pasará de 4.4 zettabytes que existen actualmente a 44 zettabytes (44 billones de gigabytes).

- b) En los últimos años, se realizan 40.000 búsquedas en Google por segundo, es decir, 3.5 millones de búsquedas por día, lo que equivale a 1.2 billones de búsquedas por año.

- c) En lo que respecta a Facebook, en agosto del 2015, se conectaron más de 1,000 millones de personas por día, lo que significa un incremento del 3,37% con respecto al mes de agosto del 2014 y se envían 31.25 millones de mensajes y son vistos 2.77 millones de vídeos por minuto.

- d) Con respecto a YouTube, por minuto se sube un promedio de 300 horas de vídeos.

- e) Se estima que para el 2020 habrá más de 6,100 millones de usuarios de smartphones en el mundo. En los próximos años habrá más de 50.000 millones de dispositivos inteligentes conectados en el mundo, todos desarrollados para recolectar, analizar y compartir datos.

- f) En el 2020, por lo menos la tercera parte de los datos pasarán por la nube.

- g) Por ahora, solo se analiza y utiliza el 0.5% de los datos que se disponen, es decir, existe un gran potencial en este mercado.

En la actualidad, la organización que desea mantenerse exitosamente en el mercado competitivo será aquella que disponga de información con una buena calidad sobre sus clientes, competidores, productos o servicios que brinda.

Tomando como base la premisa anterior, las organizaciones implementan una base de datos debido a que les permite administrar los datos que le brindara soporte en el proceso de toma de decisiones y para implementar sus estrategias de forma que

puedan aumentar la productividad obteniendo una mayor rentabilidad y hacerlas más competitivas.

Cuando se implementa una base de datos se espera que tenga las siguientes características:

- Un buen diseño.
- Una eficacia plena.
- Disponibilidad del cien por ciento.
- Satisfaga los requerimientos de información.
- Buena adaptación a los cambios en los requerimientos de información (Pratt & Adamski, 2011, p. 43).

Un aspecto importante en la implementación de una base de datos es una adecuada eficiencia, de forma que se brinde la información de acuerdo a los requerimientos de información en el tiempo requerido y al costo adecuado para la organización.

Uno de los modelos más populares implementados en los Data Base Management System (DBMS) es el modelo relacional, desarrollado por Edgar F. Codd (Figura 1), una característica del modelo relacional es que brinda independencia física de los datos. En la implementación de una base de datos, uno de los factores críticos de éxitos es el diseño de los modelos conceptual, lógico y físico de una base de datos, en la búsqueda de la eficiencia en la ejecución de consultas a la base de datos para satisfacer los requerimientos de información (Horn, 2001, p. 55).

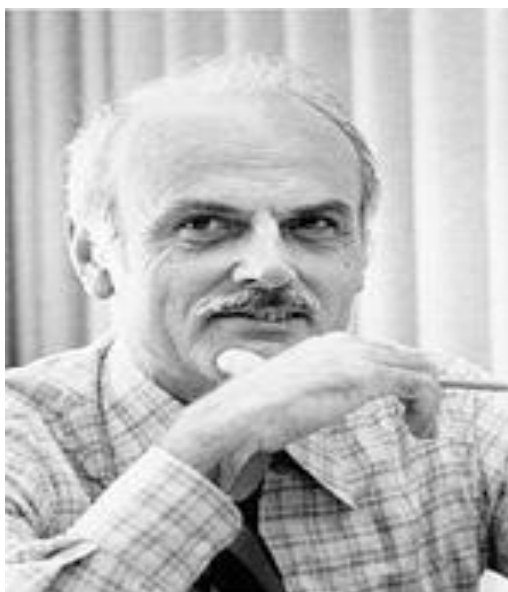


Figura 1. Edgar F. Codd (Edgar, Codd & Turing, 2012, p. 12).

En los inicios de la automatización del diseño físico de la base de datos, el objetivo principal estaba orientado a la optimización de las consultas a la base de datos, la primera generación de DBMS relacionales estaban enfocados para el procesamiento Online Transaction Processing (OLTP) y la elección de un índice correcto era un factor crítico en la ejecución eficiente de una consulta SQL (Horn, 2001, p. 87).

Con los avances tecnológicos a nivel de hardware y de software, la demanda sobre los requerimientos funcionales de los DBMS ha aumentado significativamente en el mercado, lo que ha obligado a los proveedores de DBMS a agregar nuevas características funcionales a los DBMS, lo que género que dichos productos sean más sofisticados y complejos en cuanto a su administración y manejo.

En ese escenario, el mercado de DBMS es muy diverso, competitivo, constante crecimiento y evolución, en la Figura 2 está el cuadrante mágico de DBMS operacionales que ha sido investigado y analizado por la prestigiosa consultora Gartner (Feinberg, Adrian & Heudecker, 2017, p. 57). 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 criterios de evaluación que denominan “amplitud de visión” y "capacidad de ejecución".



Figura 2. Cuadrante mágico de DBMS operacionales (Feinberg, Adrian & Cook, 2019, p. 2)

Como consecuencia del entorno descrito, en los últimos años, los proveedores de DBMS y las universidades en el mundo están desarrollando esfuerzos e investigaciones para automatizar las tareas de Administración de Base de Datos (ABD) de forma que sea más eficiente la productividad que se realiza en una organización.

En los últimos años, los DBMS tienen mayores capacidades funcionales que son más complejas de administrar, por lo que se requiere mayor conocimiento y experiencia en su administración, lo que genera mayor dificultad en su administración y aumento en los costos asociados.

Tal como se indica en (Elnaffar & Powler & Benoit & Martin 2003, p. 8), la necesidad del Self-Management en el área de Base de Datos es cada vez mayor debido a que hay mayor necesidad en los enfoques que brinda como:

- La auto configuración (Self-Configuration).
- La auto optimización (Self-Optimization)
- La auto protección (Self-Protection).
- La auto curación (Self-Healing).

En la actualidad, solo una parte de las funciones de ABD se han automatizado y los enfoques que se han desarrollado generalmente tienden a automatizar las funciones de ABD en forma aislada. Los problemas generados por una falta de automatización integral hacen que la automatización de las funciones de ABD no se efectúe adecuadamente y a un costo adecuado, lo que genera problemas en la eficiencia de la productividad de la ABD, lo que afecta a las organizaciones.

El objetivo principal del presente trabajo de tesis es desarrollar 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, para que sean más eficiente los procesos de ABD en una organización.

1.2 Definición del Problema.

Con los avances tecnológicos a nivel de hardware y de software y las nuevas tendencias que han aparecido en los últimos años (Inteligencia de Negocios, Big Data, etc.) la demanda sobre los requerimientos funcionales de los DBMS ha aumentado significativamente. En este escenario, los proveedores de DBMS han reaccionado a estos requerimientos, agregando características adicionales a sus productos, haciéndolos más sofisticados en lo que respecta a la ABD. Lo que genera mayor complejidad en su administración.

Debido al entorno 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 ABD.

En la Figura 3 se presenta como trabaja el afinamiento automático de MS SQL Server Azure (Popovic, Ljepava & Raveler, 2018, p. 26) que utiliza Inteligencia Artificial y Machine Learning para constantemente monitorear las consultas ejecutadas en una base de datos (queries) y optimizarlas automáticamente, mejorando su rendimiento (performance).

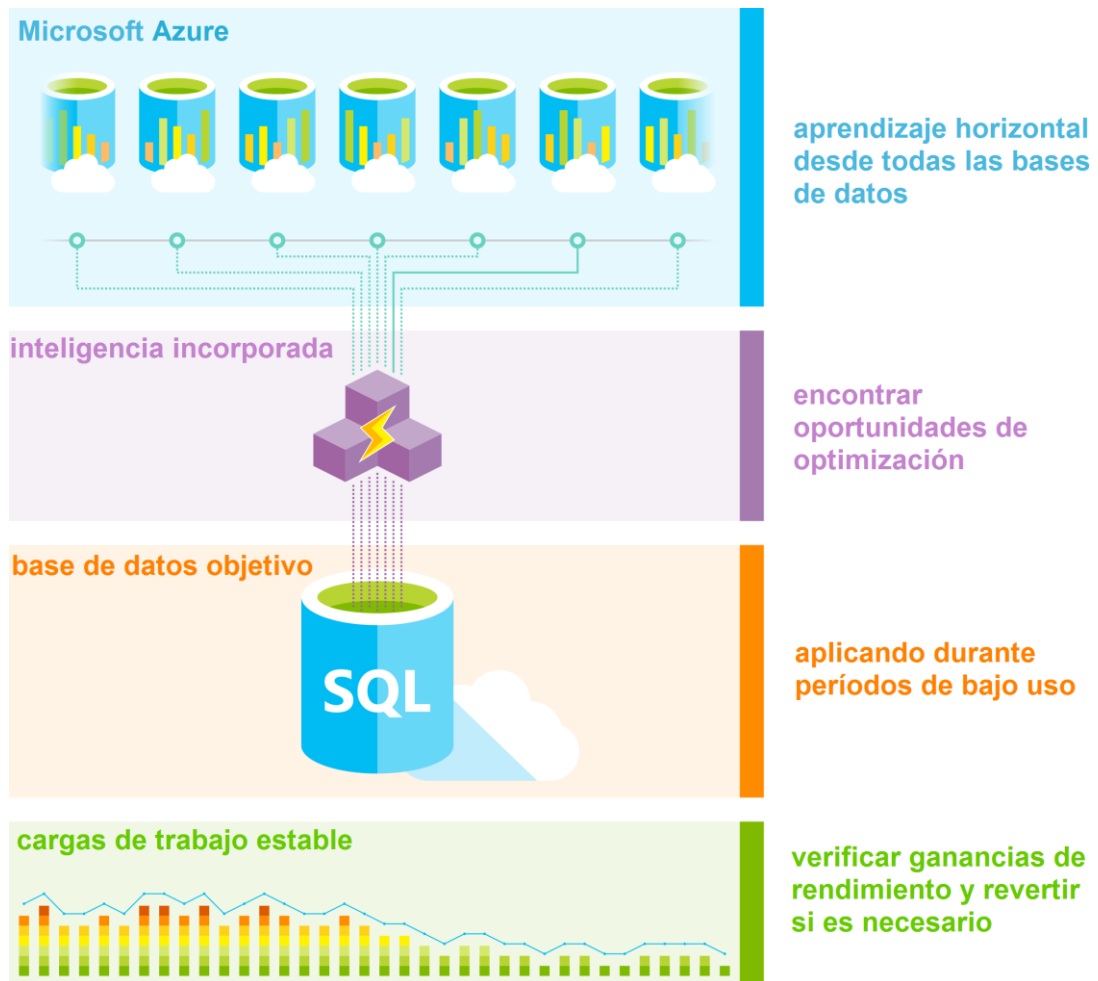


Figura 3. Como trabaja el afinamiento automático en SQL Azure (Popovic, Ljepava & Raveler, 2018, p. 26)

1.3 Objetivos.

1.3.1 Objetivo General.

El objetivo general del presente trabajo de tesis es desarrollar un modelo de Self-Management en base de datos con el enfoque de integrar los componentes de un

sistema de base de datos basándonos en la investigación de Marc Holze en (Holze, 2012, p. 74) y que este sustentado en las funciones de la Administración General, de forma que permita solucionar problemas críticos específicos que tienen las organizaciones.

El modelo propuesto, permite que el proceso de ABD sea más eficiente y que se reduzcan los costos que se generan por la realización de dicho proceso en forma tradicional, lo que brinda una mejora en la eficiencia de la productividad en la ABD, generando una ventaja competitiva a la organización.

La implementación del modelo propuesto debe permitir:

- a) Automatizar procesos específicos que utilizan las aplicaciones de base de datos de una organización, de forma que permita aumentar la eficiencia de la productividad de la ABD.
- b) Generar automáticamente comandos que permitan tomar acción en el caso de problemas específicos y que permitan reducir el tiempo empleado por el Administrador de Base de Datos cuando los atiende manualmente.

1.3.2 Objetivo Específicos.

Asimismo, a partir del objetivo general, los objetivos específicos del presente trabajo son:

- OE1: Revisar la literatura especializada, buscando modelos y enfoques que permitan aplicar el enfoque de Self-Management en la ABD.
- OE2: Diseñar un modelo de Self-Management que considere requerimientos específicos de ABD y que permita solucionar problemas críticos en las organizaciones.
- OE3: Desarrollar un software que permita aplicar los conceptos desarrollados en el modelo de Self-Management propuesto.

- OE4: Realizar dos casos de estudio para comprobar que el modelo de Self-Management propuesto brinda mejoras en la eficiencia de la productividad de la ABD.

1.4 Justificación.

1.4.1 Justificación Práctica.

Las razones sobre la utilización y aplicación de los resultados del presente trabajo de tesis son:

- Consideramos que no es suficiente el enfoque de Self-Management que se aplica en los de DBMS, sino que este proceso se debería complementar de forma que se aplique para solucionar problemas particulares que tienen las organizaciones de forma que brinde una mejor productividad en la ABD.
- Las organizaciones buscan realizar los procesos de tecnología de información con el objetivo de implementar sus estrategias de negocios y ser más competitivas. Si el proceso de ABD se desarrolla en forma equivocada, se generan sobrecostos en los procesos, lo que trae como consecuencia pérdidas financieras y de competitividad para las organizaciones.

1.4.2 Justificación Teórica.

Los motivos teóricos que consideráramos como justificación teórica para realizar el presente trabajo son:

- Proponer un valor adicional al enfoque de Self-Management en base de datos que permita reducir los costos del proceso de la ABD y mejorar la eficiencia en la productividad de la ABD.
- Por la carencia de un enfoque metodológico que permita que sea más eficiente la implementación del enfoque de Self-Management de base de datos y que permita guiar y hacer más efectivo su implementación. Los

esquemas de trabajo y metodologías existentes para dicho proceso son superficiales o confusas en su utilización.

- En los últimos años en el área de base de datos se está realizando muchas investigaciones alrededor del Self-Management y los proveedores de DBMS están aplicando dichos conceptos en sus productos como las actualizaciones automáticas de parches de mejoras (upgrades) en MS Server, Figura 4.

Automatización de VMware para aplicaciones de nivel 1

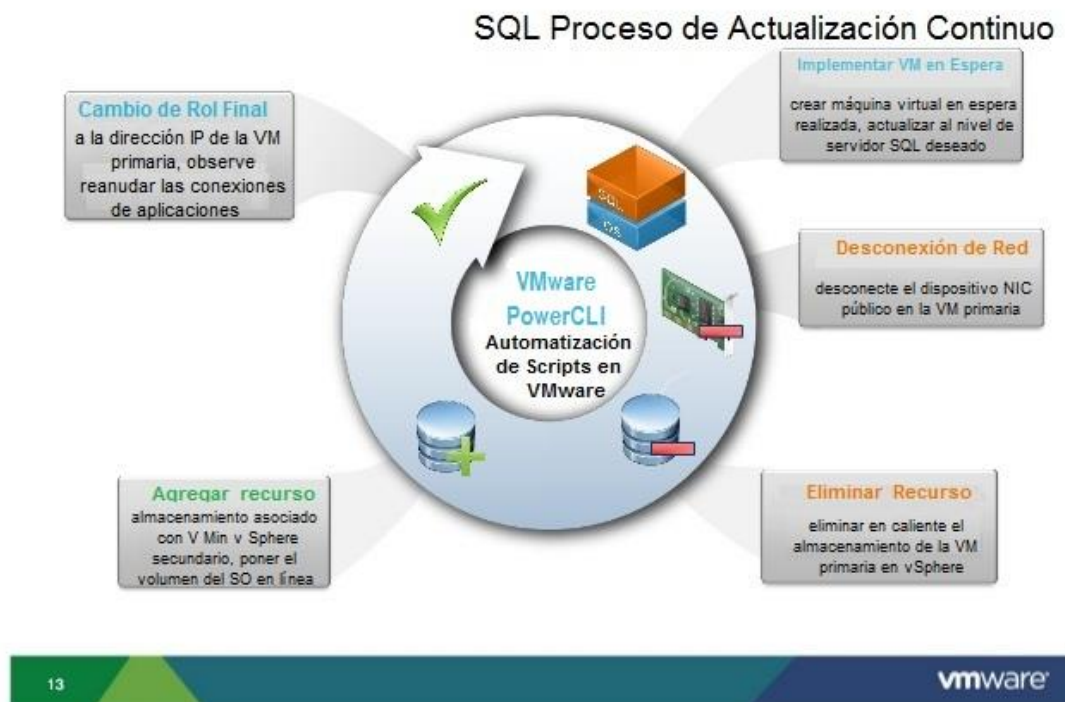


Figura 4. Actualización automática de parches de MS SQL Server (Kuhnash, & Salyer, 2013, p. 88).

1.5 Propuesta de la Tesis.

En la presente tesis proponemos un modelo de Self-Management que tomado como base el trabajo desarrollado por Marc Holze (Holze, 2012, p. 64) adicionalmente considere los requerimientos particulares sobre problemas críticos de base de datos que puede tener una organización.

Para alcanzar el objetivo propuesto, se considera las siguientes actividades:

- a) Un estudio del estado del arte de la tecnología de Self-Management de base de datos.
- b) Un estudio del estado del arte de los modelos de Self-Management en base de datos.
- c) Desarrollar una revisión de los principales aportes de investigaciones que se han desarrollado en el área de Self-Management en base de datos.
- d) Desarrollar un modelo propuesto de Self-Management en base de datos que permita mejorar la eficiencia de la productividad de la ABD.
- e) Elaborar un software de aplicación que permita aplicar el modelo de Self-Management en base de datos propuesto.
- f) Analizar los resultados obtenidos en la aplicación del software desarrollado para aplicar el modelo de Self-Management en base de datos propuesto.

1.6 Organización de la Tesis.

El presente trabajo de tesis está organizado en siete capítulos, tal como se puede observar en la Figura 5.

En el Capítulo 1, se realiza la introducción del trabajo de investigación, definiendo el problema que se desea investigar, el objetivo, la justificación y la propuesta del trabajo de tesis.

En el Capítulo 2, se desarrolla el marco teórico conceptual en el que se describen los conceptos alrededor de la Administración de Base de Datos, Sistemas Expertos y Self-Management.

En el Capítulo 3, se revisan las principales características de los modelos de Self-Management de base de datos existentes, dando énfasis al modelo de Self-Management propuesto en la tesis de doctorado de Marc Holze (Holze, 2012, p. 64) que es un trabajo de investigación muy interesante sobre Self-Management en base de datos relacionales.

En el Capítulo 4, se describe el aporte teórico del presente trabajo de tesis, donde se presenta el modelo de Self-Management propuesto. En la primera parte, se da una descripción de la motivación para el desarrollo del modelo propuesto. En la segunda

parte, se brinda los fundamentos en el que se basa el modelo de Self-Management propuesto. En la tercera parte del capítulo, se da una descripción del modelo propuesto. En la cuarta parte, se describe los componentes del modelo de Self-Management propuesto. Finalmente, en la quinta parte del capítulo se describen las limitaciones que tiene el modelo de Self-Management propuesto.

En el Capítulo 5, se presenta el software desarrollado para implementar el modelo de Self-Management en base de datos propuesto; en esta parte del presente trabajo se describe cómo se ha construido este software y se explica cómo se utiliza.

En el Capítulo 6 se describe el aporte práctico del presente trabajo, donde se presenta dos casos de estudio. El primer caso de estudio trata sobre un software de alerta en la ejecución de procedimientos almacenados (procedimiento almacenados), que se desarrolló para una empresa financiera de nuestro país. El segundo caso de estudio trata sobre una alerta cuando hay problemas con determinadas transacciones, que se desarrolló para la misma empresa financiera del primer caso de estudio. Primero se aplicará métricas en cada caso de estudio y luego se aplicará métricas utilizando la implementación del modelo de Self-Management propuesto y finalmente se compararán los resultados obtenidos.

Finalmente, en el Capítulo 7, se presentan las conclusiones de los resultados encontrados en la investigación del presente trabajo, indicando las conclusiones obtenidos en los resultados del trabajo desarrollado y describiendo cada uno de los objetivos del trabajo de tesis y se termina con las recomendaciones y trabajos futuros.

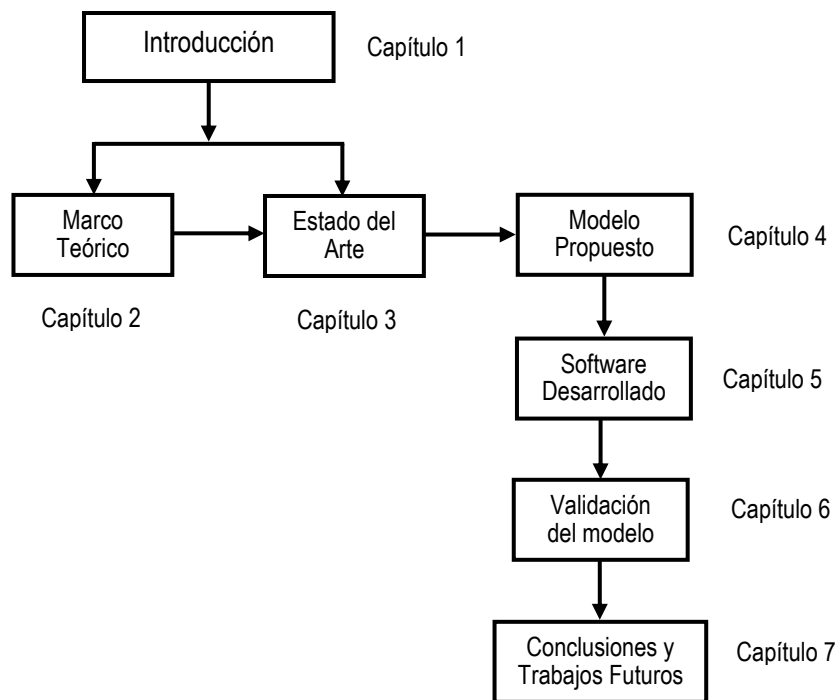


Figura 5. Organización de la Tesis

CAPÍTULO 2:

MARCO TEORICO

En este capítulo se describen conceptos que se utilizan en el presente trabajo de tesis como: Administración de Base de Datos, Software de Administración de Base de Datos, Características Generales del Software de Administración de Base de Datos, Computación Automática, Automatización de Procesos, Automatización de Procesos en Base de Datos, Sistema Experto y Self-Management.

2.1 Administración de Base de Datos.

Pratt & Adamaski describen la ABD en su obra (Pratt & Adamski, 2011, p. 36) indicando que la ABD es una de las funciones básicas de la Tecnología de la Información que tiene por objetivo planificar, desarrollar e implementar las siguientes tareas en las bases de datos de una organización:

- Instalación.
- Configuración.
- Actualización.
- Monitoreo.
- Mantenimiento.
- Optimización.
- Gestión de la seguridad.

Concordamos con el autor de que la función de ABD tiene como una de sus tareas más importantes la planificación de las estrategias para implementar las bases de datos de la organización, para que luego del diseño de las bases de datos, se

implementen y posteriormente se busque su optimización. Adicionalmente, se debe planificar, coordinar e implementar los procedimientos y restricciones de seguridad para proteger las bases de datos.

Mullins en su obra (Mullins, 2018, p. 71), indica que las principales tareas que generalmente se realizan en la ABD son:

- Instalar y actualizar las nuevas versiones del DBMS y el software de base de datos relacionado.
- Asignar el espacio en disco requerido para las bases de datos y planificar los requerimientos de espacio en disco futuros.
- Crear y modificar la definición de las estructuras de datos mediante el Lenguaje de Definición de Datos (Data Definition Language) a partir de la información brindada por los desarrolladores de aplicaciones.
- Definir y mantener los usuarios y perfiles que se requerirán para los accesos a las bases de datos.
- Monitorear y supervisar el acceso de los usuarios.
- Monitorear y optimizar el rendimiento (performance) de las aplicaciones que acceden a las bases de datos.
- Administrar el respaldo y recuperación de las bases de datos.
- Mantener los contactos requeridos con el proveedor del DBMS para tener soporte técnico.

Mullins en su obra (Mullins, 2015, p. 32) indica que en la interacción con las bases de datos se pueden generar varios cargos como:

- El Administrador de Base de Datos de Sistema (orientado a las tareas relacionadas con el sistema operativo).
- Arquitecto de Base de Datos (orientado a la arquitectura alrededor de la base de datos como es el hardware, software, comunicaciones, evaluación de tecnologías, etc.).
- Analista de Base de Datos (orientado a definir los requerimientos de datos con los usuarios).
- Diseñador de Base de Datos (orientado al modelamiento de los datos).

- Administrador de Base de Datos de Aplicaciones (orientado al desarrollo, implementación y optimización de los accesos a la base de datos por las aplicaciones).
- Administrador de Base de Datos DBA (orientado a las tareas propias de la ABD).
- Administrador de Base de Datos Analista de rendimiento (orientado a la búsqueda de la optimización del rendimiento de las aplicaciones (performance)).
- Administrador de Data Warehouse (orientado a las aplicaciones y software relacionadas con la Inteligencia de Negocios).

Concordamos con el Mullins en que para cada uno de estos cargos tienen diferentes enfoques dependiendo de los requerimientos propios de cada organización, por ejemplo, para el DBA de Aplicaciones se puede estructurar sus funciones tal como lo indica el autor en su obra en la Figura 6.

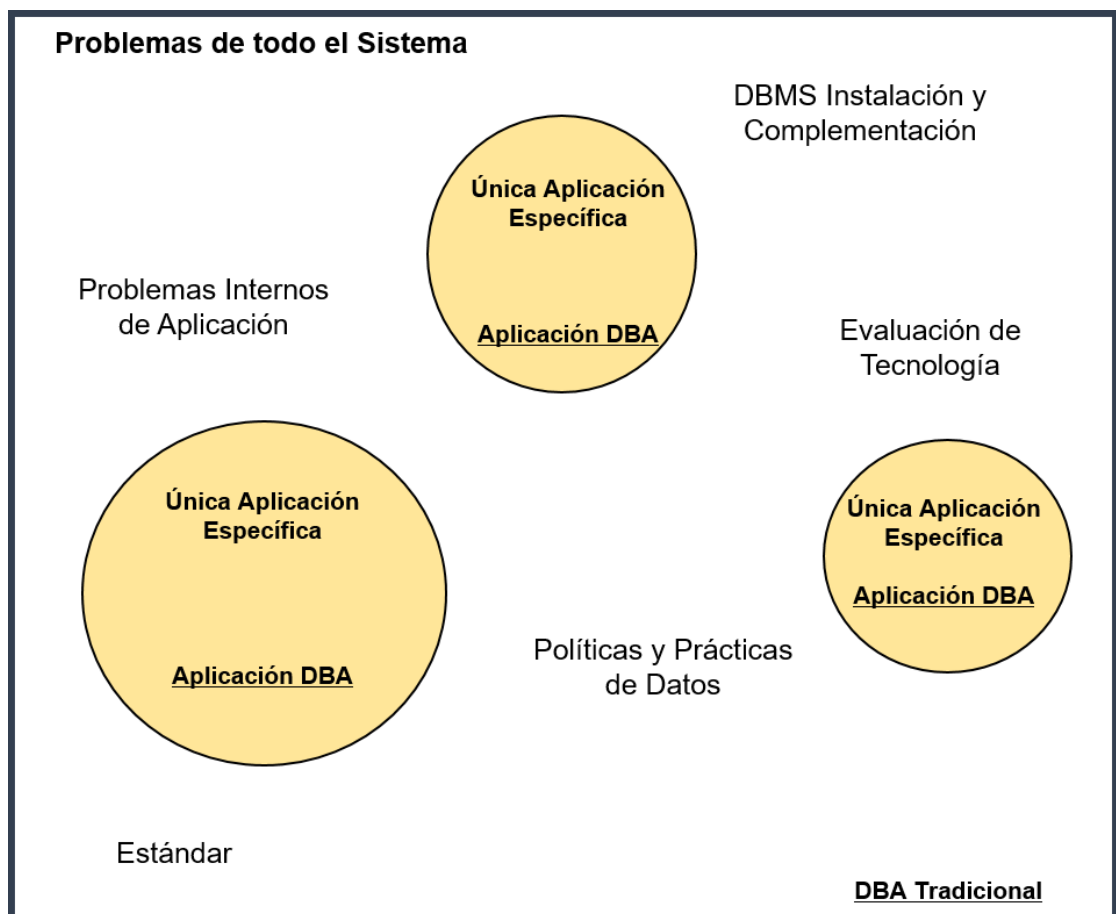


Figura 6. Funciones del Administrador de Base de Datos de Aplicaciones (Mullins, 2015, p. 32)

2.2 Software de Administración de Base de Datos.

Para poder implementar el enfoque de base de datos es necesario emplear un software, el Software de Administración de Base de Datos o Data Base Management System (DBMS) o Sistema de Gestión de Base de datos (SGBD).

Un DBMS es el software que se utiliza para desarrollar la ABD, permite la creación, el mantenimiento y el acceso a los datos almacenados en las bases de datos por los usuarios de una organización, un DBMS es un conjunto de programas que permiten administrar las bases de datos de una organización (Elmasri & Navathe, 2010, p.44).

El objetivo fundamental de un DBMS es brindarle al usuario las facilidades que le permitan manipular los datos de una base de datos, de forma que le sea transparente saber cómo es el almacenamiento de los datos en la computadora, tampoco requiere saber el método de acceso utilizado. Los sistemas de aplicación acceden y manipulan los datos almacenados en las bases de datos utilizando las facilidades que brindan los DBMS, los que disponen de lenguajes de manipulación de los datos que facilitan el trabajo de los usuarios y que los apoyan en el proceso de toma de decisiones (Elmasri & Navathe, 2010, p.45).

En la Figura 7, se presentan los componentes funcionales de un DBMS que nos presenta 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 dan soporte a una gran variedad de tecnologías como la Inteligencia de Negocios, el Big Data, Cloud, redes sociales, móviles, etc., y deben administrar bases de datos de diferentes volúmenes, manejando datos estructurados o sin estructurar, o mediante procesamiento por lotes (batch) o en línea.

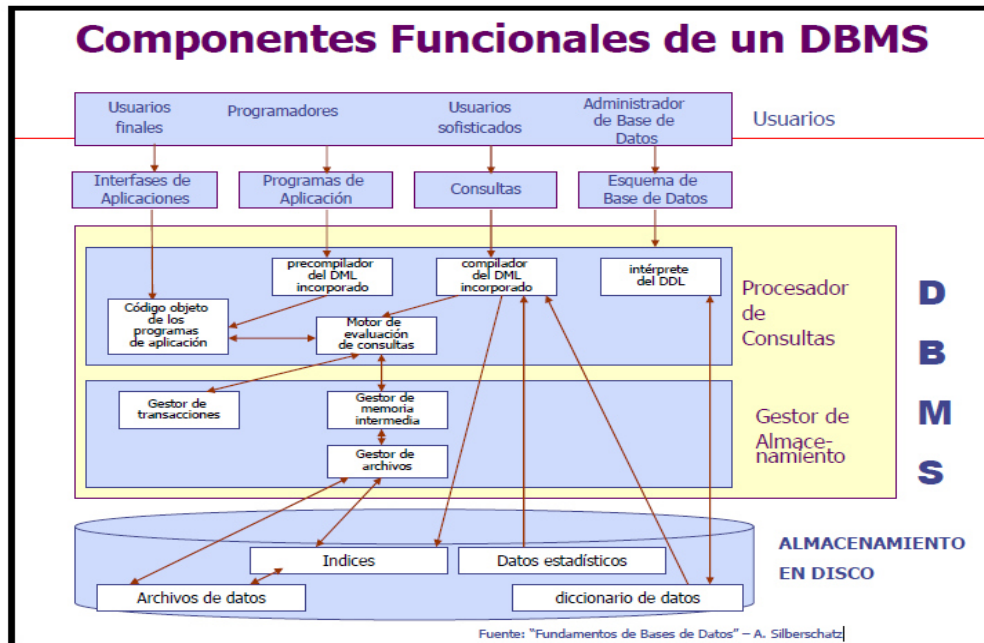


Figura 7. Componentes Funcionales de un DBMS

2.3 Características Generales del Software de Administración de Base de Datos.

Tal como se indica en (Gomez & Regalado & Gutierrez 2017), los DBMS tienen un conjunto de características generales que permiten facilitar el diseño de las aplicaciones, de forma que los procesos en las organizaciones sean eficientes y productivos, brindando una mayor flexibilidad a los usuarios finales, entre las principales características generales que brindan los DBMS tenemos:

a. La independencia de los datos.

Con el procesamiento de los archivos tradicionales, la lógica de la aplicación tiene que considerar la organización de los archivos y el método de acceso a los datos. En el caso de que por razones de eficiencia se utiliza un archivo secuencial, el programa de aplicación debe tener en cuenta la existencia de los índices y la secuencia del archivo, por lo que es imposible modificar la estructura de almacenamiento o la estrategia de acceso a los datos sin afectar al programa de aplicación, es decir hay una dependencia de cómo se procesan los archivos, lo que es ajeno al problema real que el programa de aplicación quiere solucionar. En el entorno de desarrollo de una

aplicación que utiliza un DBMS, no existe la dependencia de aplicaciones y datos, porque las aplicaciones pueden requerir diferentes características de los mismos datos (decimal o binario). En este entorno se puede modificar la estructura de almacenamiento o los métodos de acceso a los datos sin que sea necesario modificar los programas de aplicación, lo cual genera mayor eficiencia en la productividad de los desarrolladores de aplicaciones.

La independencia de datos permite que no sea necesario modificar las aplicaciones frente a los cambios en la estructura de almacenamiento y en la estrategia de acceso a los datos, lo cual es una de las premisas básicas del enfoque de base de datos.

b. Reducción de la redundancia en los datos.

En los archivos tradicionales es muy frecuente y necesario que haya redundancia en los datos. En la utilización de los DBMS en el enfoque de base de datos, uno de los objetivos es reducir la redundancia de los datos. No es que se busque eliminar la redundancia de datos redundantes, en realidad puede existir redundancia en un grado controlado y no significativo por motivos de performance, para disminuir los tiempos de acceso a los datos por motivos de rendimiento (performance), lo que se busca es eliminar la redundancia superflua o no controla.

c. Integración y sincronización de las bases de datos.

La integración de datos en una base de datos busca garantizar una respuesta a los requerimientos de los usuarios de una organización, de forma que a pesar de que la base de datos almacene los datos con una determinada estructura y un tipo de representación, se debe entregar al programa de aplicación los datos que solicita y en la forma en que lo solicita. La integración de los datos está relacionada con la sincronización de los datos, que consiste en la necesidad de garantizar el acceso concurrente a la base de datos, de forma que los datos pueden compartirse por diferentes usuarios simultáneamente. Ambos conceptos están relacionados, debido a que diferentes usuarios requieran diferentes enfoques sobre los mismos datos, pero desde diferentes puntos de vista.

d. Integridad de los datos.

La integridad de los datos garantiza la consistencia de los datos almacenados en una base de datos, de forma tal que, en cualquier momento del tiempo, los datos almacenados sean correctos, es decir, que no haya inconsistencias entre los datos. Una característica de los datos es de que, a menor redundancia de los datos, es más fácil implementar la integridad de los datos.

e. Protección y seguridad de los datos.

La protección de los datos tiene por objetivo garantizar el acceso autorizado a los datos, de forma que se evite cualquier tipo de acceso no autorizado, ya sea por un error del usuario o debido a mala una intención.

La seguridad de los datos tiene por objetivo que el sistema de bases de datos tenga procedimientos que garanticen la recuperación de las bases de datos al producirse algún error o falla técnica.

f. Facilidad en la manipulación de los datos.

Los usuarios pueden tener acceso a una base de datos a partir de requerimientos que les brinde apoyo en el proceso de toma de decisiones o para administrar un sistema de información. El DBMS brinda facilidades de búsquedas rápidas por diferentes criterios, y permite que los usuarios puedan realizarlo en un estilo simple, sin la complejidad de los enfoques de procesamientos de datos tradicionales. Los DBMS tienen lenguajes de manipulación de datos tanto para el especialista de TI como para el usuario que no es un especialista de TI de forma que ambos puedan manipular los datos en un estilo simple de forma que aumente su productividad en la organización.

g. Control centralizado.

Uno de los objetivos más importantes de los DBMS es garantizar el control centralizado de los datos, es decir, controlar de manera estructurada y única los datos que se almacenan en la base de datos, así como el acceso a ella. Para alcanzar esta

característica, debe existir una persona o conjunto de personas que tenga la responsabilidad la administración centralizada de la base de datos y que utilice el software de base de datos para implementar dicha centralización. Las tareas que se deben realizar para tener un control centralizado, podemos mencionar las siguientes:

- Definir el contenido de las bases de datos que satisfagan los requerimientos de información de la organización.
- Definir las estructuras de datos y las estrategias de acceso a la base de datos.
- Garantizar el acceso de los usuarios.
- Definir los procedimientos de autorización de validación.
- Definir los procedimientos de monitoreo de los procesos y transacciones que interactúan con la base de datos.
- Definir los procedimientos de reorganización de las bases de datos para mejorar el rendimiento (performance) de los sistemas de información.

2.4 Computación Automática.

En nuestra sociedad, las personas y las organizaciones dependemos fuertemente de la Tecnología de la Información y las Comunicaciones (TIC). La TIC es una herramienta indispensable para nuestra vida cotidiana y para la implementación de las estrategias y el desarrollo de las organizaciones. Como nos indica Hoyos y Valencia (Hoyos & Valencia, 2012, p. 27), las ventajas que las organizaciones tienen con las TIC en las organizaciones son evidentes ya que desempeñan un rol importante en la generación de valor y en la competitividad.

Pero, uno de los problemas críticos que enfrentan las organizaciones con las TIC es la carencia de especialistas de TIC. Debido a este problema, las organizaciones sufren pérdidas económicas, tal como lo indica el reporte de Rockspace en colaboración con académicos del London School of Economics (Rockspace & London School Economic, 2017, p. 58), es un problema crítico para las organizaciones la carencia de especialistas en TIC para su administración y operatividad.

Alfred North Whitehead en su libro “Introducción a las Matemáticas”, señaló que la sociedad avanza en función del aumento del número de procesos importantes que podemos efectuar sin pensar en ellos. El mensaje de Alfred North Whitehead es mientras más podemos liberar nuestras mentes de tareas repetitivas y rutinarias, transfiriendo esas tareas para que lo realicen procesos tecnológicos, más poder mental podemos separar para las tareas más complejas y creativas en nuestro trabajo o tareas que realizamos (Whitehead, 1949, p. 73).

El filósofo británico Whitehead nos señaló que la solución a este problema está en la automatización, de forma que los sistemas de TIC pueden funcionar sin la intervención del ser humano.

En el 2001, Paul Horn de la compañía International Business Machine (IBM) desarrollo un nuevo modelo de computación llamado Computación Automática (Automatic Computing), en su obra (Horn 2001, p.xx), indico que la complejidad creciente en los sistemas de TIC y la escasez de profesionales de TIC apuntan a la necesidad de automatizar muchas de las funciones relacionadas a los sistemas de TIC.

El modelo desarrollado por Paul Horn, se basa en la premisa de que los sistemas de TIC se auto regulen en forma similar en que el sistema nervioso de una persona puede regular y proteger el cuerpo humano en forma automática (Horn 2001). En la Figura 8 se muestran los componentes de la automatización, los gestores de la automatización y los elementos de la automatización del modelo de Paul Horn.

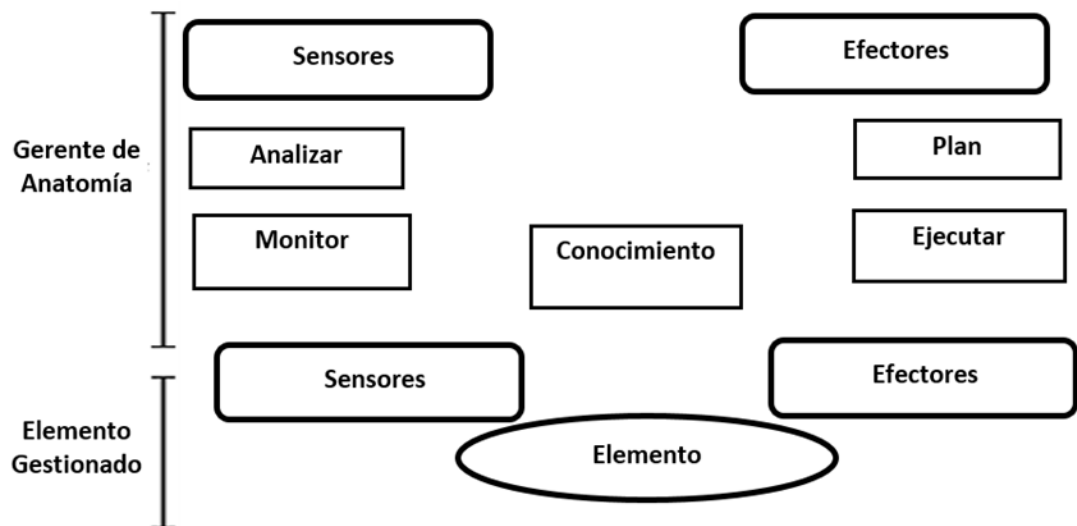


Figura 8. Componentes de la Automatización: Gestor y Elemento de Gestión [8] (Horn 2001).

2.5 Automatización de Procesos.

En la Administración de Procesos en los Negocios o Business Process Management (BPM), uno de los conceptos fundamentales es la Automatización de Procesos en los Negocios o Business Process Automation (BPA). El BPA se aplica para el control y la maximización de la eficiencia de las organizaciones que buscan asegurar la calidad en todos los niveles. La definición del significado y las implicancias del BPA requiere comprender primero en forma clara el significado de BPM.

BPM es un enfoque gerencial holístico dirigido a la parte operativa de una organización que tiene como objetivo tener una organización completamente eficiente, ágil e innovadora y que supere los logros que nos brindaban ls enfoques de administración tradicionales 30] Falta referencia bibliográfica, Figura 9.

Por lo tanto, el objetivo fundamental del BPM es el de maximizar el desempeño de la organización, lo que genera la creación de valor tanto para los clientes como para los accionistas de las organizaciones.

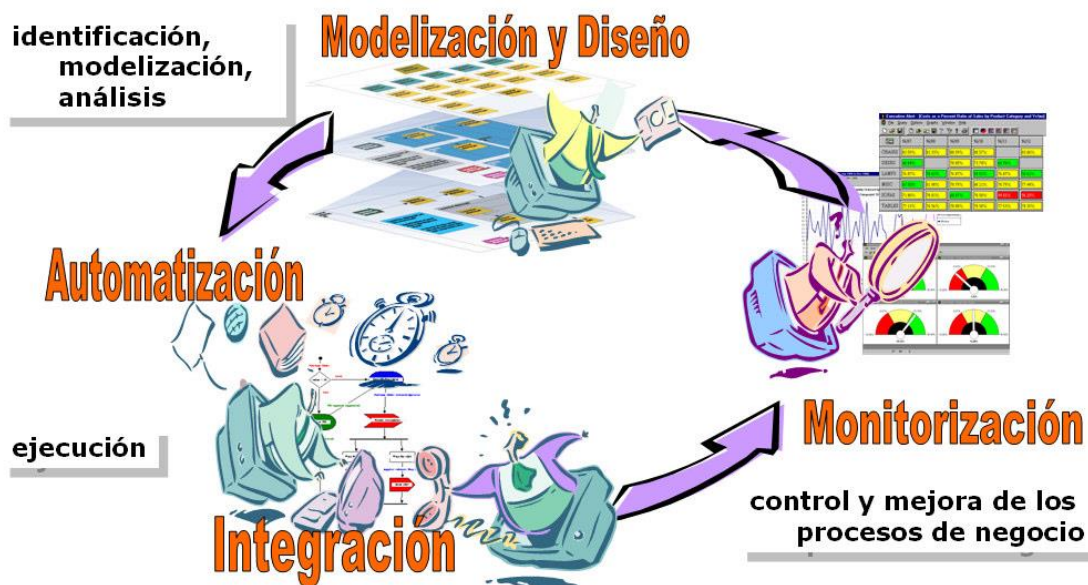


Figura 9. BPM 360 grados (Club BMP,2009).

El BPA implica la utilización de sistemas de TIC para automatizar las actividades y/o servicios de una función o unidad específica de la organización. Por lo tanto, los procesos de la organización como los que realizan las áreas de ventas, administración, operaciones, abastecimiento y distribución, cobranzas, recursos humanos o TI pueden ser automatizados mediante la aplicación de sistemas de información especializados para desarrollar tal proceso. Como una consecuencia, el BPA libera al personal de la organización de tareas rutinarias para que puedan desarrollar actividades que maximicen el valor agregado de la función que desempeñan.

Todos los procesos de la organización que sea iniciado por una tarea rutinaria, como, por ejemplo, el llenado de un formulario, el que debe seguir una serie de procedimientos predefinidos dentro de un flujo determinado, y que finaliza con la aprobación de dicha tarea, puede ser automatizado utilizando BPA. El objetivo del BPA no solo es automatizar los procesos de la organización, sino el simplificar y mejorar los flujos de los procesos en su contexto integral.

En el BPM, la automatización de procesos es administrada en forma integral con el objetivo de mejorar el flujo del trabajo en general de la organización de forma que se maximizar la eficiencia y eficacia, y a su vez brindar flexibilizar y una mejor adaptabilidad a los frecuentes cambios en los requerimientos de las organizaciones,

lo que brinda como beneficio la reducción de los errores humanos comúnmente generados por el exceso de tareas que el proceso requiere para su aprobación o termino. Adicionalmente, el personal se beneficia al tener roles, funciones y responsabilidades mejor definidas y en las que sus habilidades y experiencias son utilizadas en forma mucho más eficientes, tanto en su desarrollo personal como en el de la organización.

El BPA puede considerarse como un subconjunto del BPM, y que brinda soluciones que impactan positivamente tanto a la organización en la parte operativa como al personal en la parte motivacional y productiva.

El resultado de la automatización de procesos de negocio es la optimización de las tareas operativas de los diferentes componentes de la organización, lo que brinda una mayor eficiencia y calidad en todos los niveles de la organización.

2.6 Automatización de Procesos en Base de Datos.

Cada vez más nuestra sociedad depende de la TIC tanto para las actividades personales, como de trabajo, los negocios, toma de decisión, etc. Pero, para que las TIC brinden un mayor valor agregado, se tiene que buscar reducir la complejidad en su implementación y utilización. Si bien es cierto que en los últimos años el costo del hardware y el software se ha reducido (en algunos casos la reducción ha sido dramáticamente), el costo total de propiedad (TCO) de las TIC está cada vez alto debido a los costos de los recursos humanos principalmente por los especialistas en determinadas áreas del conocimiento como es en la ABD.

Debido a la premisa anterior, los principales proveedores de software de base de datos le dan énfasis al problema de TCO y buscan que sus productos sean más automatizados. El enfoque de la auto gestión (Self-Management) es fácil de comprenderlo, es difícil de diseñar y construir un sistema de información completamente auto gestionado. En la actualidad, se han desarrollado una gama de soluciones y técnicas para mejorar la auto gestión de los sistemas información de forma que se reduzca el TCO. También se está avanzando en muchos frentes para mejorar la capacidad de auto gestión en los DBMS (Sánchez, 2013, p. 17).

2.7 Sistema Experto.

Un sistema experto (SE), es una de las aplicaciones interesantes de la Inteligencia Artificial que tiene por objetivo la utilización del conocimiento especializado para resolver algún problema al mismo nivel que lo haría una persona experta o especialista en el área de resolución del problema (Giattarrano & Ryley, 2004, p. 89).

2.7.1 Características de un Sistema Experto.

Como se hace referencia en (Castillo, 1998), (Giattarrano, 1998) y (Wilensky, 1983), cuando se diseña un SE, uno de los objetivos es que tenga las siguientes características:

- Buen rendimiento (performance).
El SE debe tener la capacidad de brindar una respuesta de calidad, de forma que iguale o supere la respuesta que brindaría un experto especialista en el tema.
- Tiempo de respuesta adecuado.
El SE debe brindar una respuesta en un tiempo menor o igual al que le tomaría a un experto en seleccionar una alternativa o para brindar opinión o decisión.
- Buen nivel de confiabilidad.
El SE debe tener la capacidad de no tener fallas y brindar confianza en su utilización.
- Que pueda justificar sus decisiones.
El SE debe estar preparado para poder describir paso a paso el algoritmo que utilizo para obtener la solución, de forma que el sistema no solo brinde una solución y actúe como una “caja negra”. Esta capacidad es muy importante debido a que permite que el SE justifique sus conclusiones y en el caso de una depuración del SE, se pueda verificar el que realizo una correcta deducción o razonamiento.
- Flexibilidad para el cambio.
Debido a la gran cantidad de conocimiento que dispone el SE, debe tener un mecanismo eficiente para su mantenimiento (insertar, eliminar o actualizar).

2.7.2 Ventajas de un Sistema Experto.

De acuerdo con (Castillo, 1998), (Giattarrano, 1998) y (Wilensky, 1983), los SE brindan las siguientes ventajas:

- Reducción de costos.
Un atractivo de los SE es que el costo de brindar respuestas por usuario es drásticamente bajo.
- Disminución del riesgo.
Un SE puede ser utilizado en un ambiente que es peligroso para los seres humanos.
- Son Permanentes.
Los seres humanos estamos expuestos a renunciar, padecer de enfermedades o fallecer, los conocimientos que brindan un SE siempre están disponibles.
- Conocimientos variados.
Especialización múltiple. El conocimiento de muchos expertos humanos puede estar contenida dentro de un SE, aumentando la capacidad del dominio de conocimiento disponible.
- Incrementar la confiabilidad.
Un SE puede comprobar las decisiones que han sido tomadas por otro sistema, de forma que incrementa la confiabilidad de que una decisión ha sido correcta.
- Explicación.
Un SE puede explicar en un estilo detallado el razonamiento que ha seguido para llegar a una conclusión. En cambio, un experto humano podría ser que no explique cómo llego a una conclusión debido a falta de tiempo, cansancio, etc.
- Respuesta continúa.
En entornos en que se requiere una respuesta inmediata (on-line) un SE tiene la ventaja sobre el experto humano de que no sufre de stress o de cansancio y siempre está disponible.
- Tutor inteligente.

El SE puede utilizarse como un instructor inteligente permitiendo que los alumnos puedan aprender de como el SE ha llegado a una solución y de esa manera aprenden de un experto.

- Base de datos inteligente.

El SE puede utilizarse para acceder a una base de datos y recuperar información en un estilo inteligente como lo hace un experto humano.

2.7.3 Componentes de un Sistema Experto.

Un SE tiene los componentes (Castillo, 1998).

a) Componente Humano.

El componente humano de un SE está formado por un grupo de expertos que brindan sus conocimientos en un determinado tema o área, también se puede tener un grupo de ingenieros del conocimiento que toman los conocimientos de los expertos y lo transforman a un lenguaje al que el SE pueda entenderlo e interactuar con los usuarios finales. La participación de este grupo humano es considerada como el evento más importante en el diseño, desarrollo e implementación de un SE.

b) Base de Conocimiento.

La base de conocimientos de un SE es un conjunto de datos estructurado que son brindados por los expertos humanos y que tiene los conocimientos que se requieren para el desarrollo de un SE. Una base de datos de conocimientos es un tipo de base de datos que almacena datos para la gestión del conocimiento por un SE.

Cuando nos referimos al conocimiento se consideran el conjunto de reglas que utiliza el SE y que se almacenan permanentemente en la base de conocimiento y que permiten representar conocimiento de los expertos para que el SE pueda generar más conocimiento o resolver un problema. Hay que tener en cuenta que los datos de un SE son los hechos o evidencias que se almacenan en las bases de datos de las aplicaciones. Generalmente la base de conocimiento es de tipo declarativo e interactúa con el motor de inferencia para resolver un problema determinado.

c) Subsistema de Adquisición de Conocimiento.

El subsistema de adquisición de conocimiento de un SE es el componente que permite incorporar los nuevos conocimientos que fluyen desde el experto o grupo de expertos humanos a la base de conocimiento para su almacenamiento. El proceso de adquisición de conocimiento en el desarrollo de un SE es continuo y constante debido a que siempre se presentan nuevos procesos, productos, problemas, etc. que modifican al SE.

d) Control de la Coherencia.

El control de la coherencia es un componente fundamental de un SE que permite controlar la consistencia de la base de conocimientos para evitar que un conocimiento inconsistente brindado por un experto humano ingrese al SE.

Un experto por un error involuntario podría brindar información inconsistente como reglas inconsistentes o combinaciones de hechos que no son factibles. El objetivo principal del control de coherencia es brindar información sobre las restricciones para que el conocimiento sea coherente con lo definido en la base de conocimiento, de forma que apoye al experto humano en brindar información consistente y estable tanto en los procesos de construcción de la base de conocimientos como en la adquisición de conocimiento.

e) Motor de Inferencia.

El motor de inferencia es el componente fundamental de un SE que permite obtener conclusiones o un diagnóstico aplicando el conocimiento almacenado en la base de conocimiento y a partir de los hechos. El objetivo principal del motor de inferencia es seleccionar las reglas que permitan resolver el problema u obtener una conclusión para lo que aplica estrategias sistemáticas o heurísticas.

Las conclusiones que obtiene el motor de inferencia de un SE pueden basarse en conocimiento determinista o en conocimiento probabilístico. En el caso del conocimiento probabilístico, la incertidumbre le permite al motor de inferencia obtener conclusiones bajo incertidumbre.

f) Subsistema de Adquisición de Información.

El subsistema de adquisición de información de un SE permite obtener información del usuario que es utilizada por el motor de inferencia del SE como un conocimiento

adicional para continuar con el proceso de inferencia y llegar a una conclusión, generalmente esto se aplica cuando el conocimiento inicial del SE es muy limitado. Dicha información es obtenida mediante el componente de la interfase de usuario y debe comprobarse la coherencia de la misma antes de que sea almacenada en la base de conocimiento.

g) Interface de Usuario.

La interface de usuario es el nexo entre el SE y el usuario que mediante procedimientos eficientes o diálogos permite al SE interactuar con el usuario para presentar los resultados u obtener datos en un estilo sencillo y agradable.

h) Subsistema de Ejecución de Órdenes.

El subsistema de ejecución de órdenes es el componente del SE que permite que inicie acciones basadas en las conclusiones que obtiene el motor de inferencia. El SE brinda al usuario la explicación de porqué se inician las acciones mediante el subsistema de explicación.

i) Subsistema de Explicación.

El subsistema de explicación es el componente del SE que explica el proceso que ha realizado el motor de inferencia o el subsistema de ejecución de órdenes.

La explicación de lo realizado por el motor de inferencia o el subsistema de ejecución se justifica en algunos casos debido a los riesgos que conlleva las acciones a ejecutar por el SE.

j) Subsistema de Aprendizaje.

El subsistema de aprendizaje es un componente importante ya que un SE puede tener la capacidad de aprender. El aprendizaje que maneja un SE puede ser estructural o paramétrico. El aprendizaje estructural está relacionado a los aspectos de la estructura del conocimiento (reglas, distribuciones de probabilidad, etc.), por ejemplo, la inclusión de una nueva regla en la base de conocimiento del SE. El aprendizaje paramétrico tiene por objetivo estimar los parámetros necesarios para construir la base de conocimiento.

Los sistemas expertos pueden también obtener experiencia a partir de los datos disponibles. Estos datos pueden ser obtenidos por expertos y no expertos y pueden

utilizarse por el subsistema de adquisición de conocimiento y por el subsistema de aprendizaje.

2.8 Self-Management.

El Self-Management o auto administración, es un enfoque relacionado con la utilización de algún procedimiento, método, habilidad o estrategia mediante el cual los participantes de una actividad pueden orientar el logro de sus objetivos con autonomía en la administración de los recursos.

El Self-Management se realiza por medio del establecimiento de metas, planificación, programación, seguimiento de tareas, autoevaluación, auto intervención y autodesarrollo, Figura 10.

El Self-Management busca el apoderamiento de los individuos para que alcancen sus objetivos por sí mismos, por ejemplo, considera varios aspectos de la organización, como la preparación personal para asumir competencias y el liderazgo en los equipos o grupos de trabajo (Foote, 2017, p. 28).

El Self-Management se puede aplicar en varias áreas:

- En el desarrollo personal, se aplica a los métodos, habilidades y estrategias por los que las personas pueden dirigir sus propias actividades en forma eficaz buscando alcanzar sus objetivos.
- En el trabajo, se aplica en la administración organizacional basado en una autodirección en los procesos del trabajo.
- En la Economía, se aplica en un sistema basado en el enfoque de la autoadministración empresarial.
- En la Ingeniería de Sistemas, se aplica en el proceso por el que un centro de cómputo puede administrar su propia operación.
- En la salud, se aplica en el desarrollo de las capacidades por las que los pacientes con una invalidez o enfermedad pueden cuidarse por sí mismos (Foote, 2017, p. 28).

Habilidades y Consejos de Autogestión

Manejo del estrés

- ejercita tu cuerpo;
- diviértete, descansa lo más regularmente posible;
- ir a vacaciones y días festivos con las personas que amas;
- pasear en la naturaleza;
- hacer pasatiempos.



Automotivación

- encuentra y enumera tus motivos (necesidades y deseos);
- encontrar diferentes fuentes de motivación e inspiración (música, libros, actividades);
- piensa pensamientos expansivos;
- vive completamente en el momento presente;
- querido tener grandes sueños;
- sueña a menudo, todos los días;
- recuerda que nada es imposible.



Auto confianza

- deshazte de los pensamientos negativos en tu cabeza;
- piensa positivamente;
- se feliz con cosas pequeñas;
- no olvides decirte a ti mismo que eres inteligente;
- chatear con personas positivas.



Gestión del tiempo y habilidades de organización

- obtener y priorizar sus objetivos;
- crea un horario;
- hacer listas de tareas;
- utilizar diferentes herramientas de optimización;
- divide tareas grandes en tareas más pequeñas.



Tener un estilo de vida saludable y una dieta balanceada

- Las comidas saludables y las actividades físicas son extremadamente importantes cuando se trata de aumentar el poder mental y corporal.

www.businessphrases.net

Figura 10. Self-Management en los negocios, el trabajo

2.9 Resumen del Capítulo.

En el presente capítulo se ha descrito el marco conceptual del trabajo de tesis desarrollado. Se ha descrito la función de ABD en función de las tareas básicas que debe realizar el Administrador de la Base de Datos, esta función es la que se busca automatizar aplicando conceptos de Self-Management.

Luego se describió el software de administración de base de datos y sus características principales que brindan a las organizaciones. El software de base de datos es el que brinda apoyo a la función de ABD para realizar sus tareas principales.

Después se ha presentado los conceptos de Computación Automática, Automatización de Procesos y Automatización de Procesos en base de datos como un preámbulo al concepto de Self-Management.

Debido a que el software desarrollado para validar el modelo propuesto esta basado en un SE, se presenta dicho concepto. Finalmente se presenta el concepto de Self-Management, que es el concepto que en el presente trabajo se busca aplicar en el área de base de datos.

En el siguiente capítulo presentaremos la revisión del estado del arte relacionado con el presente trabajo y para mostrar la evolución del concepto de automatización en los sistemas de base de datos.

CAPÍTULO 3:

ESTADO DEL ARTE

En el presente capítulo, se muestran los resultados obtenidos de la revisión de artículos de la literatura de Ingeniería de Sistemas e Informática de los últimos años, relacionados a la ABD, dando énfasis al enfoque de Self-Management en el área de base de datos. Los artículos revisados son presentados en un orden cronológico, describiendo de forma breve los enfoques realizados respecto al Self-Management en Base de Datos y los aportes que nos brindan. Además, se describen los modelos de Self-Management aplicados en base de datos que se han utilizado en las investigaciones desarrolladas en el área de base de datos.

3.1 Evolución de la Administración de Base de Datos.

A inicios de los 60's, los discos de almacenamiento magnético se hicieron más populares y utilizados por las organizaciones, desplazando a las cintas magnéticas. Las cintas y los discos magnéticos fueron muy utilizados con sistemas de archivos tradicionales. En 1963 se empieza a escuchar el termino base de datos y en 1964 aparecen los primeros DBMS, buscando reemplazar a los sistemas de archivos tradicionales, que estaban muy ligados a la estructura de almacenamiento físico de los datos. Los DBMS generaron el concepto de ABD, y se desarrollan los conceptos de integración de datos, independencia física, esquemas, integridad, métodos de acceso y organización física sobre los datos, lo que permitió que se consigan valores agregados como: administración de usuarios, seguridad, independencia lógica y física de los datos.

El producto Integrated Data Stored (IDS) fue el primer DBMS comercial implementado que estaba basado en el modelo de datos en red. Luego se desarrolló el DBMS Information Management System (IMS), producto de IBM que estaba basado en el modelo de datos jerárquico.

En los 60's, los primeros DBMS permitían acceder a las bases de datos mediante lenguajes de programación como Assembler, Cobol, PL/I, etc., lo cual hacía que la generación y el mantenimiento de las aplicaciones fueran controladas, pero muy complejas.

History of DBMS

- ▶ **1960** – First DBMS designed by Charles Bachman at GE. Integrated Data source(IBM).
- ▶ **1970** – Codd introduced IMS. IBM's Information Management System (IMS)
- ▶ **1980** – Relational model became popular and accepted as the main database paradigm. SQL, ANSI SQL, etc.
- ▶ **1980 to 1990** – New data models, powerful query languages, etc. Popular vendors are Oracle, SQL Server, IBM's DB2, Informix, etc.



Figura 11. Historia de los DBMS (Shah, 2013, p. 92).

A inicio de los 70' se da un hito histórico, se genera un concepto innovador, donde las bases de datos pueden almacenar los datos de una organización basado en el concepto de estructura tabular llamada relación o tabla. Dicho concepto fue desarrollado en la tesis de doctorado “Un modelo relacional de datos para grandes bancos de datos compartidos” desarrollada por Edgar F. Codd, matemático de profesión y que trabajo en los laboratorios de investigación de IBM, Figura 11.

El modelo relacional está basado en la teoría de conjuntos, su estructura fundamental son las tablas o relaciones que están compuestas por filas y columnas. En los años 70s aparece el DBMS INGRES un DBMS relacional desarrollado con enfoque académico. Luego, en 1983 aparece el DBMS relacional DB2 de IBM, siendo el primer DBMS relacional comercial. La popularidad y la acogida del modelo

relacional hace que se desarrollen otros DBMS relacionales como ORACLE, INFORMIX, SYBASE, Microsoft SQL Server, etc.

Los DBMS relacionales ofrecen una característica fundamental en el enfoque de base de datos, la independencia física de los datos, que permite que las estructuras físicas de las tablas como los índices se puedan modificar sin afectar el resultado de las consultas (o queries). Además, el modelo relacional permite aumentar la productividad en el desarrollo de aplicaciones y en la utilización por los usuarios finales, mientras que el diseño físico de una base de datos es un componente fundamental que permite optimizar la eficiencia de una consulta en un DBMS.

A fines de los años 70s y a inicios de los años 80s. comienzan a desarrollarse teorías e investigaciones que dan mucha importancia al diseño físico de una base de datos, entre dichas investigaciones se tiene la investigación de Stonebraker (Stonebraker, M.,1989) que presenta un modelo paramétrico de carga de trabajo y la investigación de Hammer y Chan (Hammer, M. & Chan, A., 1976) que presenta un modelo predictivo. Posteriormente, aparecen otras investigaciones que le dan mucha importancia al diseño físico de una base de datos.

En los últimos años el aumento en el volumen de los datos en las bases de datos ha permitido que se desarrollen sofisticadas técnicas de búsquedas con índices que permiten la adaptación del diseño físico de una base de datos a los requerimientos de información de los usuarios de una organización.

En 1986, se desarrolla el enfoque de Inteligencia de Negocios y se da mucho énfasis al desarrollo de los Data Warehouse, lo que trae como consecuencia que los DBMS tengan que mejorar para se pueda dar soporte a la toma de decisiones, denominado On-Line Analytical Processing (OLAP), en el que una característica fundamental es el manejo de grandes volúmenes de datos.



Figura 12. Datawarehouse (PowerData, 2002, p. 48).

Con el avance en las TIC, los requerimientos sobre las bases de datos han aumentado exponencialmente, Figura 12, lo que ha generado que los proveedores de DBMS hayan agregado nuevas capacidades funcionales a los DBMS, como una consecuencia de este cambio, se ha hecho más compleja la ABD debido a que es más sofisticada la configuración de un DBMS y se tienen más opciones para el diseño físico. Adicionalmente, se le ha dado poco énfasis al desarrollo de interfaces en los DBMS para la ABD y las nuevas capacidades que tienen. A este panorama, hay que agregar que los Administradores de Base de Datos son cada vez más un recurso de un costo elevado para las organizaciones que requieren que sea altamente capacitados y experimentados, en algunas plataformas tecnológicas son un recurso difícil de encontrar.

Debido al entorno descrito, las organizaciones de desarrollo de DBMS y las organizaciones de investigación están orientando sus capacidades y esfuerzos en la aplicación del enfoque de Computación Automática a las bases de datos, de forma que los DBMS y los sistemas que utilizan base de datos puedan operar y mantenerse por sí mismo, inclusive que automáticamente se puedan adaptar a los cambios en los requerimientos utilizando patrones.

Mientras que los costos de hardware y software se han ido reduciendo (por los avances tecnológicos y de economía de escala), los costos propietarios en el entorno de los DBMS se han incrementado debido a los costos del personal que lo desarrolla y administran, como por ejemplo los Administradores de Base de Datos, se han convertido en un recurso más escaso y costoso. En la actualidad, las bases de datos están creciendo cada vez más en complejidad y en volumen, por eso la tendencia es la de automatizar los procesos de ABD aplicando el concepto de Self-Management.

A continuación, describiremos la revisión que hemos realizado sobre los estudios de investigación que se han desarrollado aplicando el enfoque de Self-Management en base de datos.

En el trabajo “Self-Tuning Database Systems: A Decade of Progress” (Chaudhuri, S. & Narasayya, V, 2007), se describe como el grupo de desarrollo e investigación del producto SQL Server de Microsoft asume la difícil e innovadora tarea de rediseñar el producto para el lanzamiento de la siguiente versión (Microsoft SQL Server 7.0) de forma que sea fácil su utilización, tenga nuevas capacidades funcionales y que se eliminen los errores que tenía el producto en las versiones anteriores. Microsoft a este nuevo reto lo denominó el proyecto AutoAdmin.

Hay que recordar que los orígenes de MS SQL Server están en el DBMS Sybase (ahora el DBMS se llama SAP Adaptive Server Enterprise), debido a que Microsoft necesitaba un DBMS y le compró el código de su DBMS en versión Windows a la empresa Sybase.

En ese momento, en el área de Base de Datos se estaban desarrollando investigaciones sobre las técnicas de Análisis de Datos y Minería de Datos (Datamining), que más adelante serían muy utilizadas para productos de Inteligencia de Negocios. El grupo de desarrollo e investigación del proyecto AutoAdmin de Microsoft decidió aplicar las técnicas de Análisis de Datos y Minería de Datos en la automatización de las tareas administrativas y de optimización de MS SQL Server 7.0, Figura 13.

El objetivo principal del grupo AutoAdmin fue orientar la automatización alrededor del diseño físico de la base de datos, que en esa época era un problema crítico y que estaba relacionado al procesamiento de consultas SQL (queries).

Los autores de dicho artículo participaron en el proyecto AutoAdmin y describen sus experiencias en la elaboración de la primera versión que encontraron para automatizar el diseño físico de una base de datos.

AutoAdmin: Self-Tuning Database

Initial thought ...
Focus on automating the physical design of the relational database

The outcomes ...

- Tuning technology in Microsoft SQL server
- Self-tuning Histograms
- Monitoring infrastructure

The Physical Database Design "Problem" – Given the storage bounds , optimizer-estimation costs and the workload – which is the best physical design?

Choice of physical design = f (usage profile of DB server, optimization overhead, storage)

Challenges:

- Multiple physical design features
- Interaction due to updates and storage
- Scaling challenges

AutoAdmin's Ideas:

- Recognize the frequent item-sets
- Workload compression
- Merge and Reduce
- Enumeration: *top-down, bottom-up*

4

Figura 13. Proyecto AutoAdmin

Un aspecto interesante en dicho trabajo es que describen algunos de los trabajos de investigación realizados en la década de los años 1997 a 2007 en el área de Auto afinamiento (Self-Tuning), entre los más importantes avances que obtuvieron tenemos los siguientes:

- ✓ En los primeros años de la automatización del diseño físico de base de datos, el objetivo principal estaba orientado a la optimización de las consultas SQL. Los primeros DBMS relacionales estaban orientados al procesamiento transaccional en línea, en inglés, OnLine Transaction Processing (OLTP) y un aspecto crítico era a la selección del índice adecuado para la ejecución eficiente de una consulta SQL.

- ✓ En 1998, SQL Server 7.0 de Microsoft fue el primer DBMS comercial que tenía una facilidad para el diseño físico de base de datos denominado Index Tuning Wizard (ITW). En Microsoft SQL Server 2000, el ITW fue mejorado porque se le agregó la capacidad funcional de brindar recomendaciones en la selección de índices y vistas materializadas.
- ✓ En el 2005, Microsoft lanza al mercado SQL Server 2005, que era un producto rediseñado con significativas mejoras funcionales, en dicha versión ITW fue remplazado por el producto Database Engine Tuning Advisor (DTA) que brindaba recomendaciones para la selección de índices, vistas materializadas y el particionamiento de tablas.

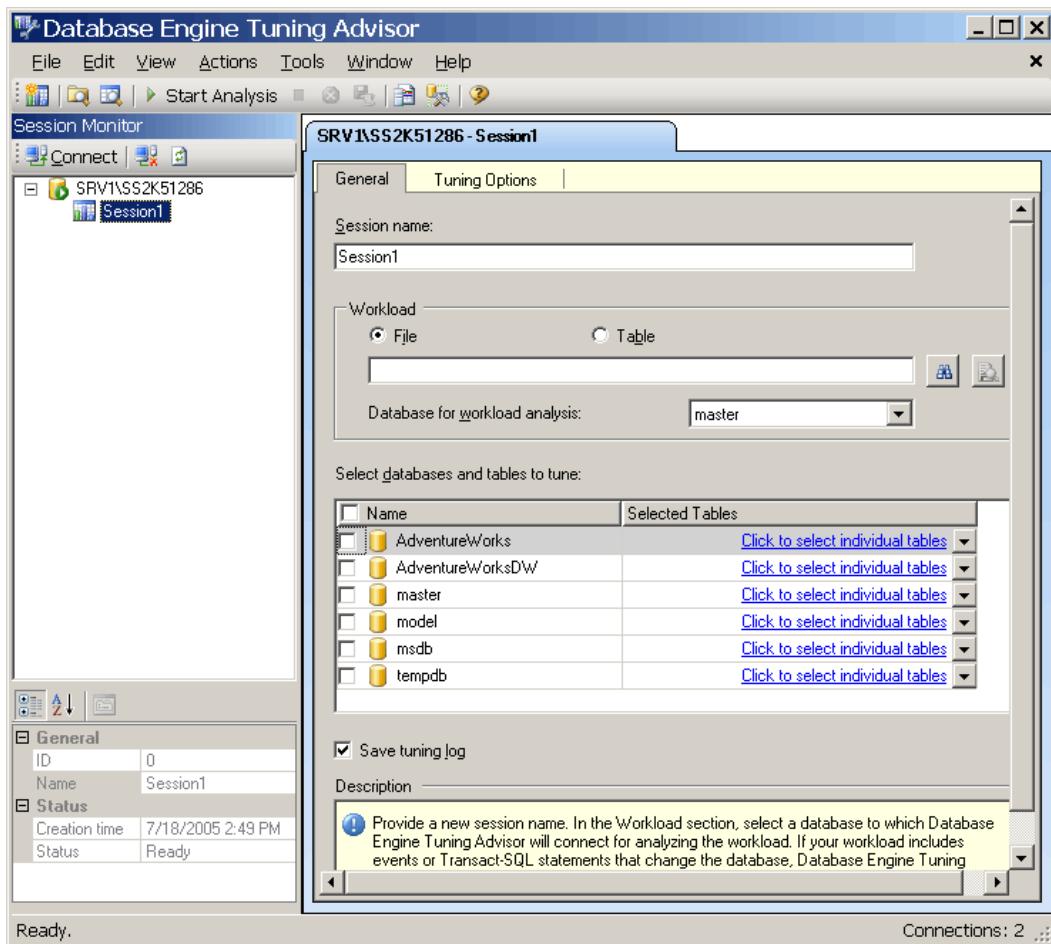


Figura 14. Producto Database Engine Tuning Advisor

- ✓ En el 2001, los investigadores Sam S. Lightstone & Guy Lohman & Danny Zilio, en (Zilio, D. & Lightstone, S. & Guy, K.,2001) presentan un trabajo en el que se aplica el enfoque de Self-Management al producto DB2 de IBM en las plataformas UNIX y Windows (DB2 Universal Database). En el trabajo se presenta como el enfoque de Self-Management puede apoyar para reducir el costo total de propiedad de los DBMS y mejorar el rendimiento del sistema.

DB2 Universal Database tiene buenas características del enfoque de Self-Management, dispone de varias características de autoadministración automáticas en el soporte para asegurar la integridad, diseño físico y optimización de la base de datos.

- ✓ En el 2006, el trabajo “Adaptive Self-Tuning Memory in DB2” (Storm, A. & García-Arellano, C. & Lightstone, S., 2006) se describe la tecnología desarrollada por el DBMS DB2 V9.1 de IBM para plataformas Linux, UNIX y Windows que simplifica el afinamiento de la memoria de la base de datos al utilizar el enfoque de Self-Management.

Por muchos años uno de los temas críticos en los DBMS fue la administración de la memoria, con DB2 V9.1 se hizo un cambio significativo en el diseño de las bases de datos y en el afinamiento de los sistemas computacionales grandes (mainframes). Hay que tener en cuenta que IBM ha sido una de las empresas que ha impulsado la Computación Automatizada que brinda varios aportes de Self-Management como:

- Afinamiento automático de la base de datos
- Diseño físico automatizado
- Auto curación (Self-Healing)
- Auto configuración (Self-Configuring).

En este artículo, se describe como IBM DB2 V9.1 para Linux, UNIX y Windows automatiza y simplifica el afinamiento de la memoria de la base de datos apoyando a al ABD en los siguientes aspectos:

- Deficiencias en el conocimiento de la utilización de la memoria en el DBMS.
- Manejo de los requerimientos de memoria excepcionales para poder soportar una sobrecarga de trabajos que no están considerados previamente.
- Variaciones en el comportamiento de la carga de trabajo.
- Afinamiento de la memoria.

En este trabajo de investigación, se comenta sobre la aplicación de Self-Tuning Memory Manager (STMM) que permite combinar la simulación de sistemas con la teoría de control de recursos para optimizar la asignación de memoria de un DBMS relacional como DB2 de IBM.

STMM ha sido probado en escenarios en los que los requerimientos de memoria para la carga de trabajo pueden variar drásticamente en el tiempo. IBM ha desarrollado pruebas internas con STMM que demuestran su efectividad en el afinamiento de la memoria de la base de datos para lo que ha aplicado dicho enfoque con fuerte carga de trabajo en entornos con problemas de rendimiento, y obteniéndose un mejor afinamiento que los propuestos por los Administradores de Base de Datos.

Adaptive Self Tuning Memory Management

- DB2 9 introduced a revolutionary memory tuning system called the Self Tuning Memory Manager (STMM)
 - Works on main database memory parameters
 - Sort, locklist, package cache, buffer pools, and total database memory
 - Hands-off online memory tuning
 - Requires no DBA intervention
 - Senses the underlying workload and tunes the memory based on need
 - Adapts quickly to workload shifts that require memory redistribution
 - Adapts tuning frequency based on workload

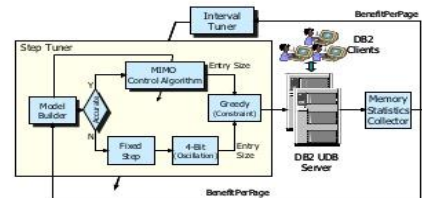
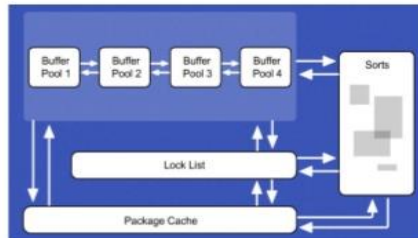


Figura 15. Administración de Memoria Automática DB2

- En el 2007, Armando Barreto & Ben Wongsaroj & Tariq M. King & Tuan Cameron & Suzette V. Diaz en su trabajo (Barreto, A. & Wongsaroj, B. & King, T., 2007) plantean que los computadores crecen en tamaño y cada vez tienen más funciones con una mayor sofisticación y complejidad, lo que genera dificultad para su administración y aumentan sus costos. Ante este problema, los autores proponen la necesidad de disponer de sistemas computacionales que se administren por sí mismo de acuerdo con un nivel de métricas.

Generalmente los grandes sistemas computacionales tienen base de datos con grandes cantidades de datos, por lo que se requiere que los DBMS incorporen características de auto administración como son: Self-Configuration, Self-Optimization, Self-Protection y Self-Healing.

En este trabajo de investigación, se presentan criterios para evaluar características de Self-Management en un DBMS y los aplican para evaluar productos de amplia trayectoria como: IBM DB2, Microsoft SQL Server, ORACLE, Teradata y SAP Sybase ASE.

Los criterios de evaluación se aplican a cada uno de los DBMS mencionados y se generan comparaciones entre ellos, como resultado se obtiene que los DBMS evaluados tienen características muy similares en cuanto a Self-Management (ORACLE 23%, DB2 21%, Microsoft SQL Server 19%, Teradata 19%, SAP Sybase ASE 18%).

Hay que tener en cuenta que en el proceso de evaluación los DBMS fueron evaluados en una determinada versión del producto. Una de las conclusiones interesante de dicho trabajo es que uno de los puntos débiles de los DBMS evaluados es el Self-Protection que tiene por objetivo principal brindar soporte en los cambios a la privacidad y a la integridad de los datos.

- En el 2011, en el artículo “A Decade of ORACLE Database Manageability” (Belknap, P. & Beresniewicz, P. & Dageville, 2011), se describe como el DBMS Oracle en las versiones 10g y 11g aplica el enfoque de Self-Management en base de datos para incrementar la productividad en la ABD.

Oracle busco desarrollar su DBMS de forma que no solo se oriente al Self-Tuning, sino que considere otras características de Self-Management de forma que sea más productivo, se le dieron características a su producto para facilitar la auto configuración, reducir la complejidad de conceptos y tareas que debía aplicar el Administrador de la Base de Datos, simplificar el monitoreo y utilizar el análisis de datos para mejorar el rendimiento.

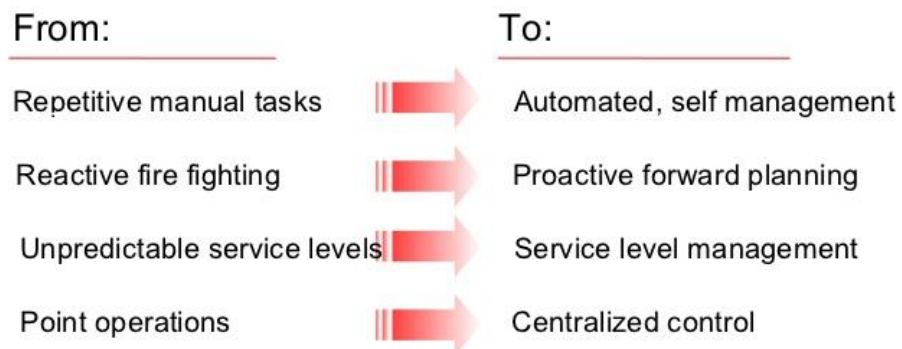
Oracle enfoco el Self-Management en base de datos basado en una arquitectura de 3 niveles:

- Nivel de estadísticas y reportes en el entorno de base de datos, frecuentemente se recolectan estadísticas del rendimiento y los datos relacionados a la ejecución de los procesos que accedan a la base de datos, los cuales son almacenados en un repositorio de Self-Management y que el usuario puede revisar mediante reportes y una interfaz gráfica.

- Nivel de asesoría, en el que los datos de rendimiento (performance) son analizados en periodos de tiempo basados en una metodología en que se hace observaciones en un periodo de tiempo y se brindan recomendaciones sobre los problemas de rendimiento que se han encontrado, las causas que lo producen y las oportunidades de mejora.
- Nivel de implementación automática, las recomendaciones que se brindaron en el nivel anterior son implementadas en forma automática con un riesgo bajo y con la conformidad de los Administradores de Base de Datos.

Why Oracle Database 11g?

For manageability and change



ORACLE

Figura 16. Self- Management en ORACLE 11g

Adicionalmente, en este trabajo de investigación, se describe como ORACLE ha desarrollado una tecnología para aplicar auto curación (Self-Healing) y auto diagnostico (Self- Self-Diagnosing) para errores y defectos en el DBMS. Oracle brinda una interfaz denominada Enterprise Manager que se utiliza para administrar la base de datos y que brinda facilidades para el Administrador de la Base de Datos de forma que no tenga que memorizar

secuencias lógicas para completar una tarea simple y brinda interfaces de usuario interactivas para que pueda revisar la gran variedad de datos recolectados en tiempo real. El DBMS ORACLE tiene un software denominado Tuning Advisor que brinda recomendaciones y en ciertos casos automáticamente identifica y corrige el problema por sí solo.

Las facilidades de Self-Management que tiene ORACLE brindan apoyo al Administrador de la Base de Datos, permitiendo reducir la curva de aprendizaje y dándole un nivel similar a un especialista con muchos años de experiencia.

- En el 2011, en el trabajo de investigación “Self-Tuning of ORACLE Database Using SQL Scripts” (Rani, P. & Kumar, Singh, P. & Sharma, H., 2011) , se describe que en el ambiente de Oracle se debe controlar en forma precisa la cantidad de memoria RAM que es utilizado por la instancia de la base de datos. La memoria asignada al Oracle está compuesta por Area Global del Sistema o System Global Area (SGA) y Area Global de Programa o Program Global Area (PGA). El proceso de afinar la memoria puede ser bastante complejo, pero a partir de Oracle 9i se ofrece al Administrador de la Base de Datos facilidades para dimensionar el SGA y el PGA.

En este trabajo de investigación, se presenta como ORACLE aplica el Self-Tuning en la administración de la memoria. ORACLE utiliza scripts de SQL para mantener la demanda de memoria RAM tanto de la base datos como en las conexiones de la base de datos en un nivel por debajo de la cantidad de memoria RAM física.

ORACLE brinda un enfoque de Self-Tuning para administrar la memoria RAM de forma de que se pueda reconfigurar dinámicamente la memoria RAM asignada y eso permite que otros parámetros que dependan de la memoria se puedan reconfigurar automáticamente.

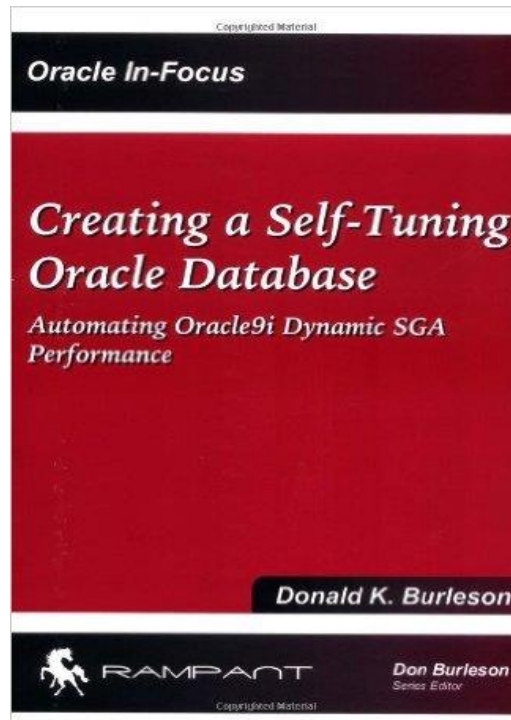


Figura 17. Self-Tuning ORACLE 9i

- En el 2011, en el artículo de investigación “Auto Admin Project at Microsoft Research: Lessons Learned” (Bruno, N. & Chaudhuri S. & Konig, A., 2011), los autores señalan que inicialmente su objetivo era automatizar el diseño físico de una base de datos, pero que luego cambiaron su objetivo por desarrollar un DBMS que aplique el enfoque de Self-Management en base de datos.

El artículo describe como el grupo Auto Admin del centro de investigación de Microsoft desarrollo trabajos para:

- Automatizar el diseño físico.
- Automatizar la parte de monitoreo.
- Desarrollar facilidades para la optimización de consultas.
- Automatizar la desfragmentación de índices.

Como resultado de las investigaciones que desarrollo el centro de investigación de Microsoft se implementaron varios productos con enfoque de Self-Management en base de datos como Index Tunning Wizard (ITW) y el Database Engine Tuning Advisor (DTA), que son productos para mejorar

el rendimiento en las consultas a la base de datos optimizando índices y particiones.

- En el 2013, el trabajo de investigación “Automatic Workload Driven Index Defragmentation” (Narasayya, V. & Park, H. & Syamala, M. (2011), plantean que en muchos casos las consultas a las tablas de una base de datos involucran buscar en grandes cantidades de datos y que la eficiencia en estas consultas generalmente depende de los índices.

Los índices cuando se crean y se empiezan a utilizar, no tienen fragmentación y las búsquedas son rápidas, pero con el paso del tiempo debido a las actualizaciones en los datos, se presenta fragmentación en los índices generando lentitud en las consultas, por lo que es necesario aplicar la desfragmentación a los índices, lo que tiene un costo asociado.

En el trabajo de investigación, se describe como el centro de investigación de Microsoft abordó una solución para:

- Desfragmentar un índice a un costo bajo y que se obtenga un alto nivel de eficiencia.
- Desarrollar facilidades para desfragmentar un índice B-tree por rango en línea.
- Administrar la desfragmentación de un índice sin la intervención del Administrador de la Base de Datos.

El equipo del centro de investigación de Microsoft desarrolló técnicas para solucionar los problemas que se generan a partir de la desfragmentación de un índice utilizando un prototipo elaborado con Microsoft SQL Server 2008 R2, lo que permite detectar la fragmentación y brindar facilidades para desfragmentar los índices por rango y en línea con eficiencia.

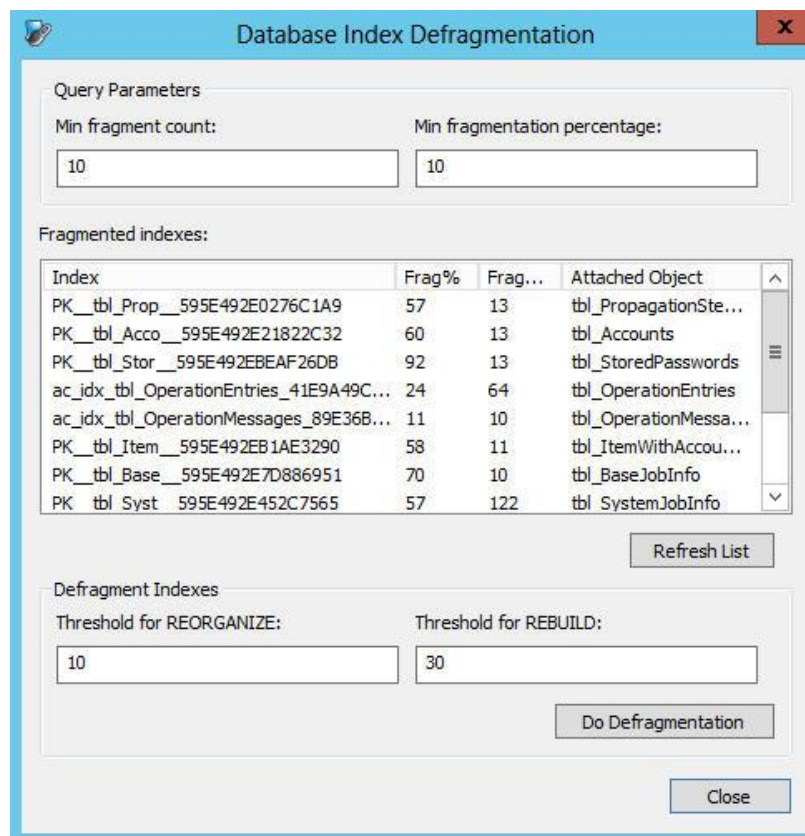


Figura 18. Desfragmentación Automática en MS SQL Server 2008 R2

3.2 Modelos en Base de Datos.

De acuerdo con la literatura revisada en la sección 3.1 del presente trabajo de tesis, podemos ver que en los últimos años los proveedores de DBMS y los investigadores del área de Base de Datos buscan reducir los altos costos desarrollo y de mantenimiento en la ABD. El enfoque que se utiliza para enfrentar este problema es aplicar los principios de la Computación Automática (Horn, 2001) por lo que la tendencia es desarrollar funciones de Self-Management en los DBMS que ayuden en la automatización de la ABD.

En esta parte del trabajo de tesis, se describen los modelos utilizados en el área de Base de Datos, los constructos que los componen y se presentan algunos resultados obtenidos en las implementaciones que se realizaron. En los DBMS se utilizan los siguientes modelos:

3.2.1 Modelo de Arquitectura DBMS con enfoque RISC.

En el diseño y desarrollo de un DBMS generalmente se utiliza el enfoque por capas, tal como lo describe Harder y Reuter en (Harder & Reuter, 1983) Figura 19. Pero Weikum en (Weikum & otros, 2002), indica que la arquitectura de los DBMS tradicionales no está diseñada para una automatización total, por lo que plantea un cambio radical basado en los componentes de estilo RISC [25]. En este enfoque, cada componente se supone que tiene una función limitada, permitiendo predecir la performance del componente mediante modelos matemáticos. En el modelo de arquitectura tipo RISC se puede predecir cuantitativamente el comportamiento de cada componente RICS bajo todas las posibles configuraciones y características de carga de trabajo. A partir de la evaluación del modelo matemático, cada componente RICS se puede optimizar adecuadamente (Self-Optimization).

La implementación de este modelo requiere una reingeniería total de un DBMS, inclusive los sistemas de información tendrían que rediseñarse para poder trabajar con el enfoque de componente RICS. La implementación de este modelo tomaría un largo plazo, con un costo alto, y habría que modificar una gran cantidad de productos de DBMS y de sistemas de información, lo que hace que esta alternativa sería inviable.

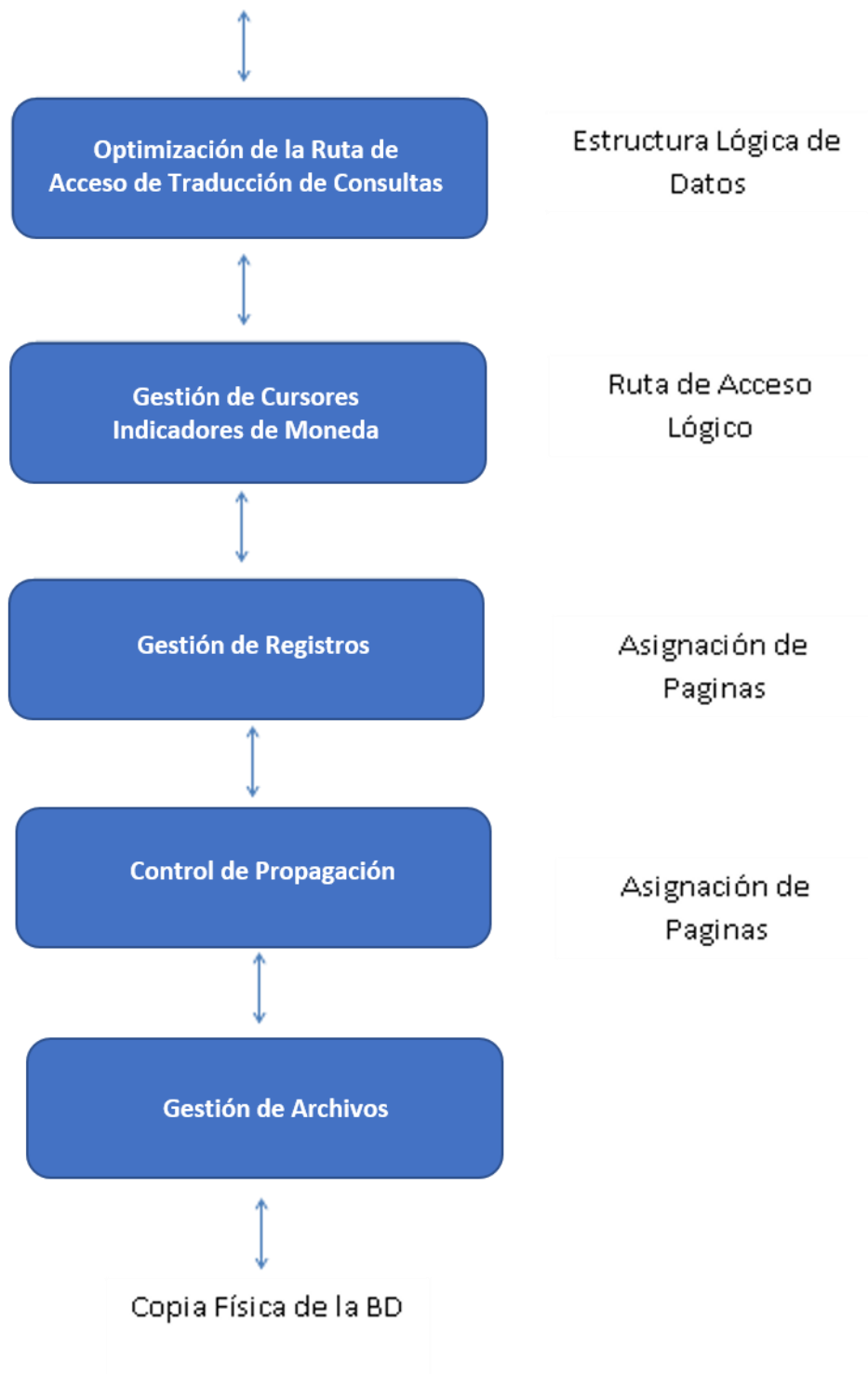


Figura 19. Estructura de un DBMS propuesto por Harder y Reuter (Harder & Reuter 1983, p. 95).

3.2.2 Modelo de Priorización de Consultas.

Con el modelo de priorización de consultas, el DBMS internamente considera la priorización de las consultas en función de cada clase de servicio antes de que se ejecuten las consultas (como se utiliza en DB2 Workload Manager), con este enfoque el DBMS tiene que considerar las metas que tienen las clases de servicios en procesamiento de las sentencias SQL. El DBMS considerando los costos estimados de las consultas, debe decidir en qué orden debe ejecutarse.

La implementación de este modelo, descrita en (Niu & otros, 2006) implica interceptar las consultas antes de que sean ejecutadas en el DBMS. Luego de interceptarlas se evaluara su ejecución tomando como base los costos estimados de ejecución, en casos simples no es complicado esta alternativa de solución, pero 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), lo que implicaría una reingeniería de los DBMS para poder aplicar este modelo, lo que hace que sea inviable.

3.2.3 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 SBD. Para superar estos problemas, los manejadores automáticos tienen que estar informados del contexto donde trabajan.

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 (IBM, 2005), en el que cada función automática es controlada por su inmediata superior (dentro de la jerarquía), Figura 20.

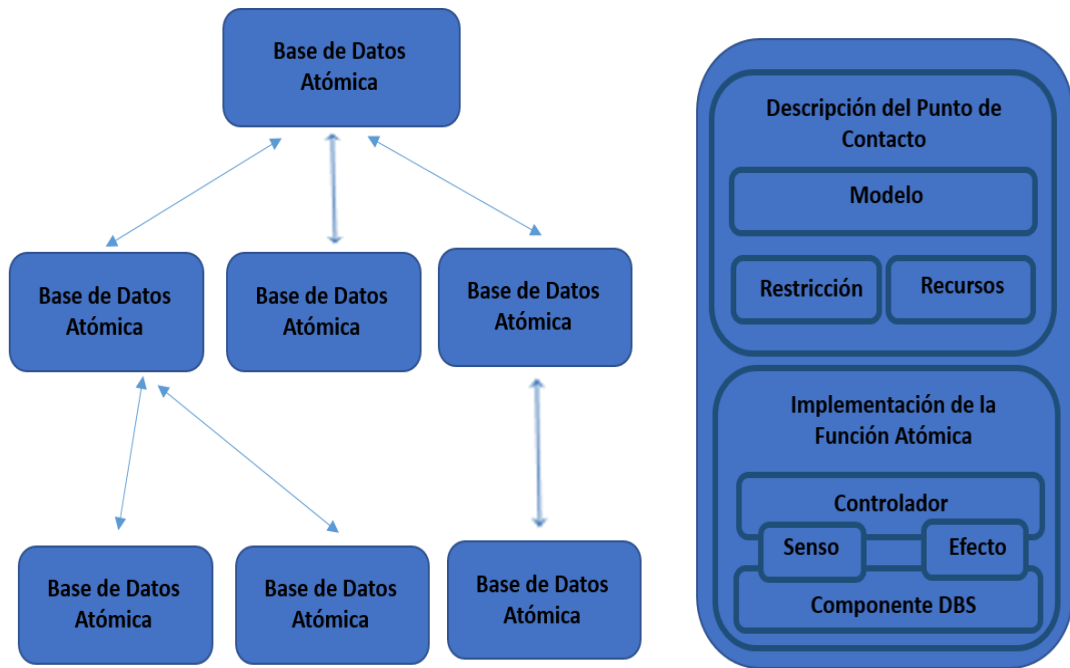


Figura 20. Jerarquía de Administradores Automáticos (IBM, 2005, p. 93).

3.2.4 Modelo de Sistema Total de Self-Management.

El modelo de sistema total de Self-Management (Holze, 2012) se caracteriza porque tiene una lógica centralizada de Self-Management que brinda una visión integrada de las decisiones que se tomen sobre Self-Management en el sistema de base de datos. La Figura 21 representa el modelo de sistema total de Self-Management, que se caracteriza porque tiene un repositorio de lógica de Self-Management que monitorea el sensor de información del sistema de base de datos y que está constantemente comparándolo con las métricas de las metas definidas por el Administrador de la Base de Datos.

En el caso de que el sensor requiera de una configuración diferente, la lógica que se utiliza para implementar Self-Management realiza un análisis para decidir si es necesario una reconfiguración y evalúa y determina los ajustes óptimos que se requieran (considerando el estado actual y la carga de trabajo del sistema de base de datos y teniendo como premisa las metas definidas). En el análisis de la reconfiguración la lógica Self-Management debe considerar todas las dependencias de las alternativas de cambios de configuración a realizarse.

El modelo de sistema total de Self-Management puede ser implementado en forma externa al DBMS (no tiene por qué ser un componente de la parte interna del DBMS)

y su lógica puede basarse en los sensores y los efectores del SBD. Una característica interesante del modelo de sistema total de Self-Management es como se implementa con un enfoque integral, puede identificar los efectores individuales de configuración para poder predecir los tiempos de respuesta, rendimiento, costos de recursos y disponibilidad.

La utilización de un recurso centralizado como lo plantea el modelo de sistema total de Self-Management permite que los problemas de interacción e independencia de metas pueden ser resueltos. Además, el modelo de sistema total de Self-Management puede complementarse con la utilización de modelos matemáticos cuantitativos del comportamiento del sistema de base de datos de forma que se pueda evitar decisiones que generen sobrecargas.

Los enfoques para poder implementar Self-Management en base de datos buscan la automatización de las funciones de un DBMS de forma que pueda funcionar por sí mismo. Una característica importante del modelo de sistema total de Self-Management es que su diseño de una lógica centralizada permite evitar los problemas de interacción, desproporción e independencia de metas.

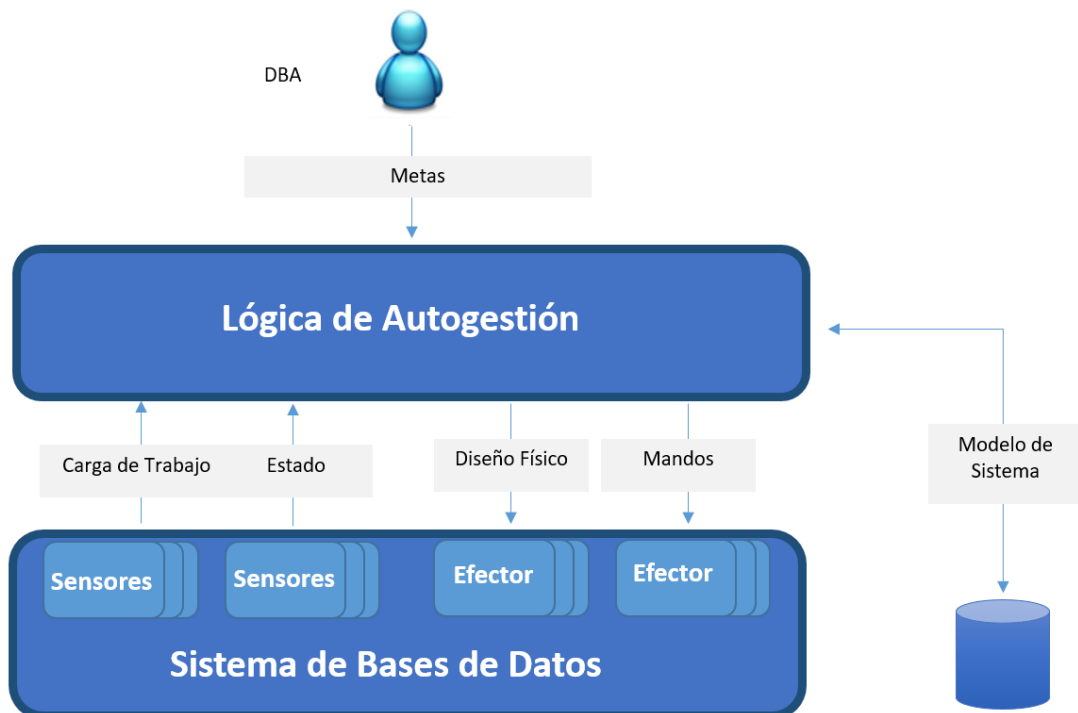


Figura 21. Modelo de Sistema Amplio de Self-Management (Holze, 2012, p. 71).

3.3 Software de Self-Management en Base de Datos.

Los productos de software de base de datos que aplican el enfoque de Self-Management en base de datos se pueden agrupar en dos categorías:

- Software que aplican Self-Management fuera de línea.
- Software que aplican Self-Management en línea.

A continuación, describiremos cada uno de estos tipos de categoría de productos de software que aplican el enfoque de Self-Management en base de datos:

3.3.1 Software de Self-Management fuera de línea.

En los inicios del Self-Management en base de datos, los investigadores se orientaron a desarrollar software que permita seleccionar los índices más adecuados para acceder a las tablas y optimizar las sentencias SQL que utilizan las aplicaciones, productos que tienen estas características son:

3.3.1.1 IBM DB2 Design Advisor.

El software IBM DB2 Design Advisor, es un producto que permite seleccionar los índices más adecuados para mejorar el rendimiento en las consultas SQL de forma que los procesos tanto batch como en líneas puedan mejorar sus tiempos de proceso. IBM DB2 Design Advisor brinda recomendaciones a nivel del diseño físico de la base de datos (Quest Software Inc., 2018), Figura 22.

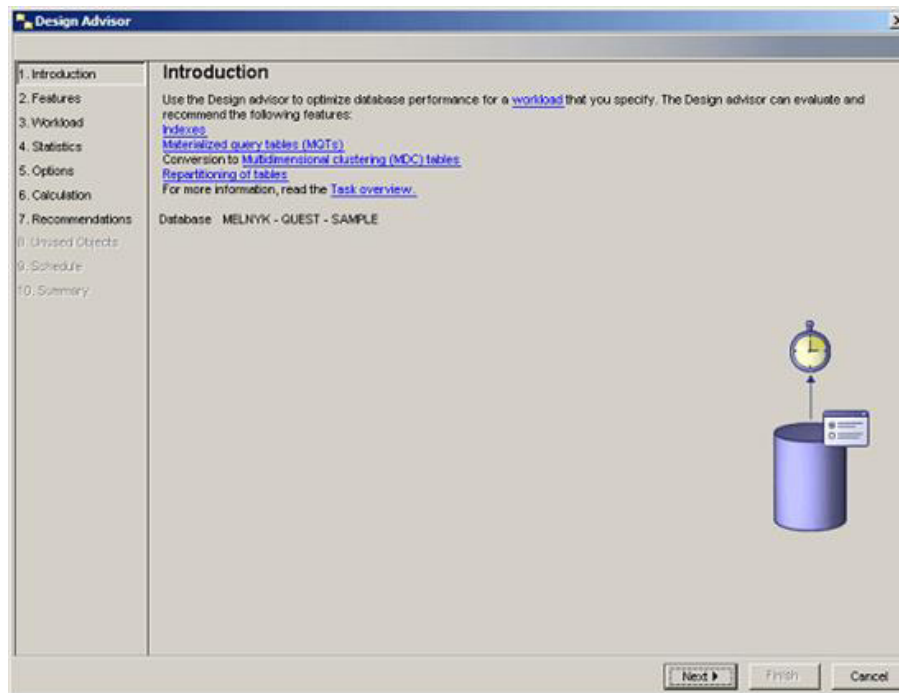


Figura 22. IBM DB2 Design Advisor (Quest Software Inc., 2018, p. 67)

3.3.1.2 Microsoft SQL Server Database Tuning Advisor.

El producto Microsoft SQL Server Database Tuning Advisor permite seleccionar los índices más adecuados a partir de las sentencias SQL, internamente genera índices virtuales para tener sus índices candidatos para evaluar y seleccionar la mejor alternativa y brinda recomendaciones a nivel del diseño físico de la base de datos (Bertucci, Silverstein, Gallelli & Rankins, 2015, p. 33), Figura 23.

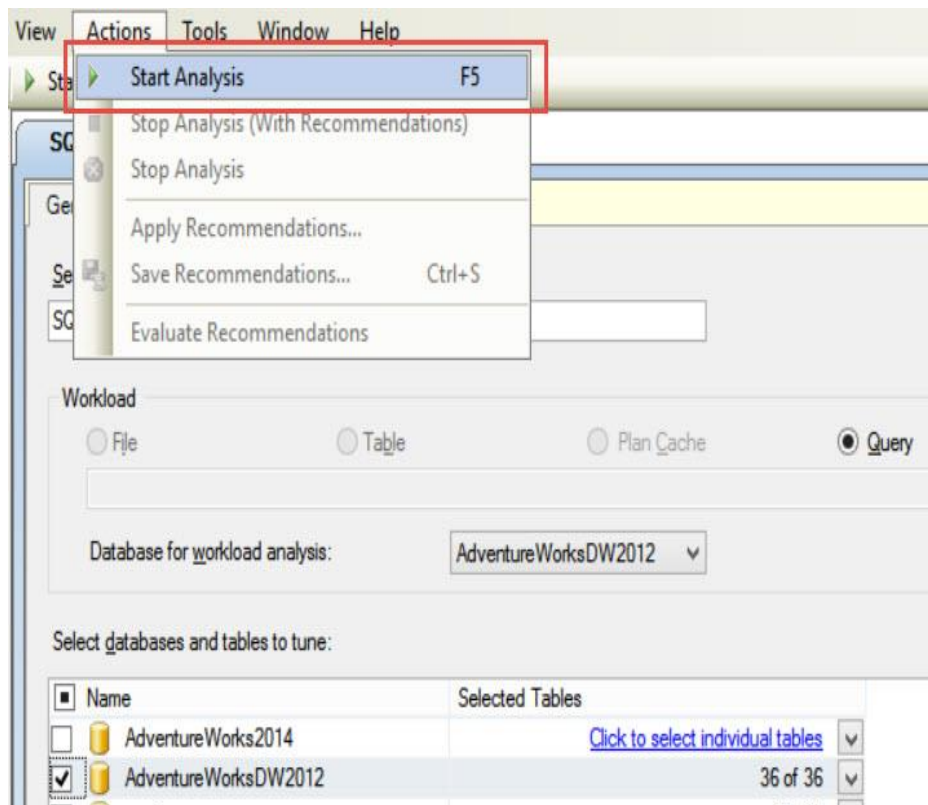


Figura 23. Microsoft SQL Server Database Tuning Advisor (Bertucci, Silverstein, Gallelli & Rankins, 2015, p. 33)

3.3.1.3 ORACLE SQL Tuning Advisor.

A diferencia de los dos productos mencionados previamente, que brindan recomendaciones a nivel de diseño físico, ORACLE SQL Tuning Advisor brinda recomendaciones a nivel de diseño físico y adicionalmente brinda recomendaciones sobre estadísticas, planes y otras formas de replantear las consultas SQL (ORACLE, 2018), Figura 24.

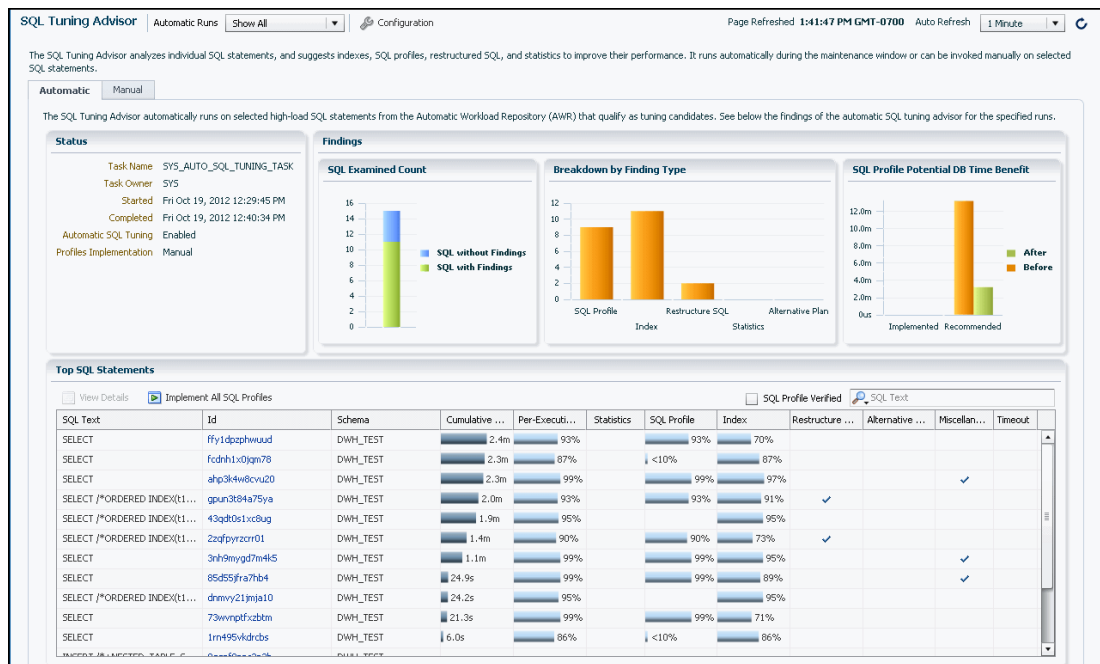


Figura 24. ORACLE SQL Tuning Advisor (ORACLE, 2018, p. 93).

3.3.2 Software de Self-Management en línea.

Los productos de software de Self-Management fuera de línea fueron un buen aporte en la implementación del Self-Management en base de datos, sin embargo, la activación del software en determinado momento tenía que hacerlo el Administrador de la Base de Datos, es decir, tenía que identificar en qué momento se producía un problema y debía activar el software de Self-Management para que el software pueda brindar alternativas de solución.

La activación de los productos de software de Self-Management fuera de línea por el Administrador de Base de Datos hace que se pierda uno de los objetivos principales del Self-Management en base de datos que es el de la auto administración.

Para llenar ese vacío en la automatización computacional, aparecen los productos de software de Self-Management en línea que tiene mucha aceptación en el mercado de TI y por los fabricantes de DBMS, continuación, presentaremos tres tareas en la que se aplican el enfoque de Self-Management utilizando productos de software en línea:

3.3.2.1 Administración de memoria en línea.

En un DBMS hay una serie de áreas de memoria que se utilizan como, por ejemplo:

- Las áreas de memoria asignada a los buffers de datos.
- Las áreas de memoria asignada al ordenamiento de registros (sort).
- Las áreas de memoria asignada a los planes de ejecución.
- Las áreas de memoria asignada para los bloqueos (lock).
- Las áreas de memoria asignada a los procedimientos almacenados (procedimiento almacenados).

En la ABD una de las tareas rutinarias que tenía que realizar el Administrador de Base de Datos es la configuración, monitoreo y reconfiguración de estas áreas, pero en las últimas versiones de los DBMS la configuración, monitoreo y reconfiguraciones las realizan dinámicamente los DBMS aplicando el enfoque de Self-Management. En este tema se han hecho diversos estudios desarrollados por los proveedores de DBMS como el desarrollado por IBM en (Storm & otros, 2006)

3.3.2.2 Selección de índices en línea.

En el área de Base de Datos, se han desarrollado diversos estudios que permiten monitorear la carga de trabajo de un SBD para mejorar el diseño físico de una base de datos principalmente buscando la efectividad en la utilización de los índices como los desarrollados en (Kolaczowski Piotr & Rybinski Henryk, 2011).

Muchas de estas investigaciones utilizan las estadísticas de distribución de los datos y de rendimiento en diversos niveles de detalle y luego aplicar algoritmos de afinamiento que permita a los DBMS relacionales tener alternativas de configuración de índices, logrando afinar el acceso a las tablas.

En la actualidad, se disponen de productos que realizan la selección de índices en línea aplicando el enfoque de Self-Management y se realizan muchas investigaciones para mejorarlo.

3.3.2.3 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, como la sentencia SQL a optimizar, las estadísticas de distribución de los datos y la estructura de la base de datos. En el caso de las estadísticas de distribución de los datos, los optimizadores de planes 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 ORACLE 12C Release 1 que aplican la recolección de estadísticas en línea (ORACLE, 2019).

3.4 Resumen del Capítulo.

En el presente capítulo se ha descrito el estado del arte relacionado con el presente trabajo de investigación. Para una mejor presentación, el capítulo se organizó en 3 secciones.

En la primera sección del capítulo se describe en forma general la evolución en el tiempo de la ABD, se revisan los avances en el área de Base de Datos hacia la tecnología de Self-Management en base de datos.

En la siguiente sección, se describen los modelos en la ABD producto de la revisión de la literatura utilizada en la presente investigación.

En la tercera sección, se describen los productos de software de base de datos que aplican el enfoque de Self-Management en base de datos.

En el siguiente capítulo del presente trabajo de investigación, se describe el modelo propuesto de Self-Management en base de datos.

CAPÍTULO 4:

MODELO PROPUESTO DE SELF-MANAGEMENT EN BASE DE DATOS

En el presente capítulo se describe el modelo propuesto que está basado en el enfoque de Self-Management y que permite mejorar la eficiencia de la productividad de una base de datos en una organización. En la primera parte, presentamos la motivación de desarrollar un modelo de Self-Management a partir de los beneficios y aportes encontrados en documentos de investigación producidos en los últimos años y se resume lo que aún falta desarrollar o profundizar en lo que respuesta a Self-Management en Base de Datos, tanto en la parte de desarrollo como en la investigación en el área de Base de Datos. En la segunda parte, describimos en forma general los fundamentos en que se basa el modelo de Self-Management propuesto. En la tercera parte, se presenta como está estructurado el Modelo de Self-Management propuesto. En la cuarta parte, se describe en detalle cada uno de los componentes del modelo de Self-Management propuesto. Finalmente, en la quinta parte del presente trabajo, se indican las restricciones del modelo propuesto.

4.1 Motivación del Modelo Propuesto.

La realización del presente trabajo de investigación se vio motivada por los aportes o bondades encontradas en los trabajos científicos de investigación que en los últimos años se han desarrollado alrededor de la aplicación del Self-Management en el área de Base de Datos.

En particular, nuestra motivación aumento cuando encontramos el apoyo que brindan las principales empresas líderes en software de base de datos (IBM, ORACLE, MICROSOFT, etc.) en el desarrollo, implementación e investigación del enfoque de Self-Management. En los últimos años, dichas empresas aplican el enfoque de Self-Management en sus productos de software de base de datos, sin embargo, las investigaciones desarrolladas son escasas en determinados puntos. Debido a que existen escasas investigaciones en mejorar la eficiencia de la productividad en la ABD hemos dado énfasis en la búsqueda sobre los enfoques que son aplicados a automatizar una función particular de ABD o al desarrollo de los productos que se orientan a un DBMS en particular, también hemos considerado los enfoques que son integrales.

Los modelos de Self-Management de la literatura se orientan a automatizar tareas comunes y rutinarias que realizan los Administradores de Base de Datos y las incorporan a los DBMS, por ejemplo, la actualización de estadísticas o la reorganización de tablas fragmentadas. Sin embargo, no consideran problemas particulares de las organizaciones que son críticos debido a que generan pérdidas económicas, de mercado, afectan el servicio ofrecido, etc., por ejemplo, las cancelaciones eventuales de procesos que utilizan Base de Dato, es decir existen actividades que no están automatizadas y que no se pueden generalizar.

Teniendo como objetivo encontrar el fundamento diferenciador de nuestro modelo, hemos analizado los modelos desarrollados en investigaciones previas y encontramos que podemos complementar el modelo de Self-Management en Base de Datos planteado por Marc Holze (Holze,2012), considerando que es importante enfocar la función de ABD desde el punto de la Administración General y teniendo en cuenta

que los requerimientos particulares de ABD en una organización pueden automatizarse con la utilización de un sistema experto.

4.2 Fundamentos del Modelo Propuesto.

Tomando como premisa que las tareas de la ABD se basan en las funciones de la Administración General, dimos ese enfoque para llevarlo al desarrollo de nuestro modelo de Self-Management desarrollado.

Las funciones básicas en la Administración General según (Robbins, S. & Coulter, M., 2014) son.

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

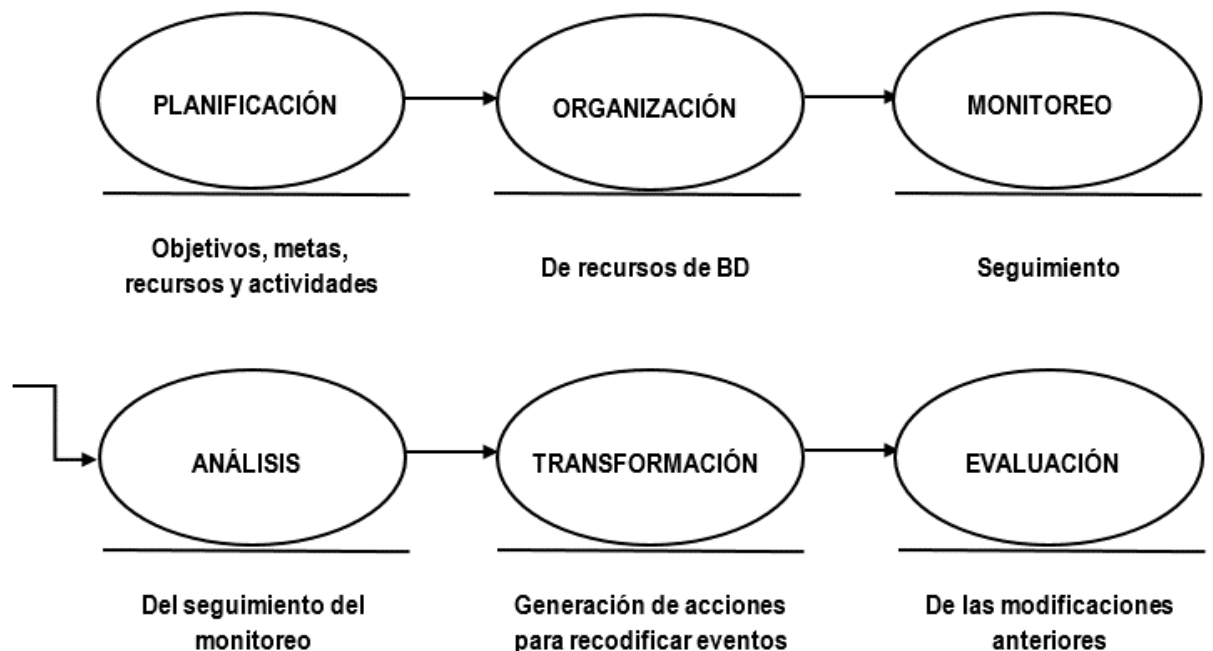


Figura 25. Funciones Básicas de la Administración General (Robbins & Coulter, 2014, p. 94)

Siendo que las funciones de Planificar y Organizar están relacionadas más a la gestión y requieren un desarrollo especializado y no es posible automatizar en su total dimensión, por lo que en el presente trabajo se fundamenta en las siguientes funciones básicas:

- Monitoreo
- Análisis
- Transformación
- Evaluación

Con respecto a las funciones básicas de Administración General, en el modelo de Self-Management propuesto consideramos lo siguiente:

- La función de monitoreo que 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.

El proceso de Self-Management en Base de Datos considerando el enfoque de la Administración General podríamos representarlo como en la Figura 26.

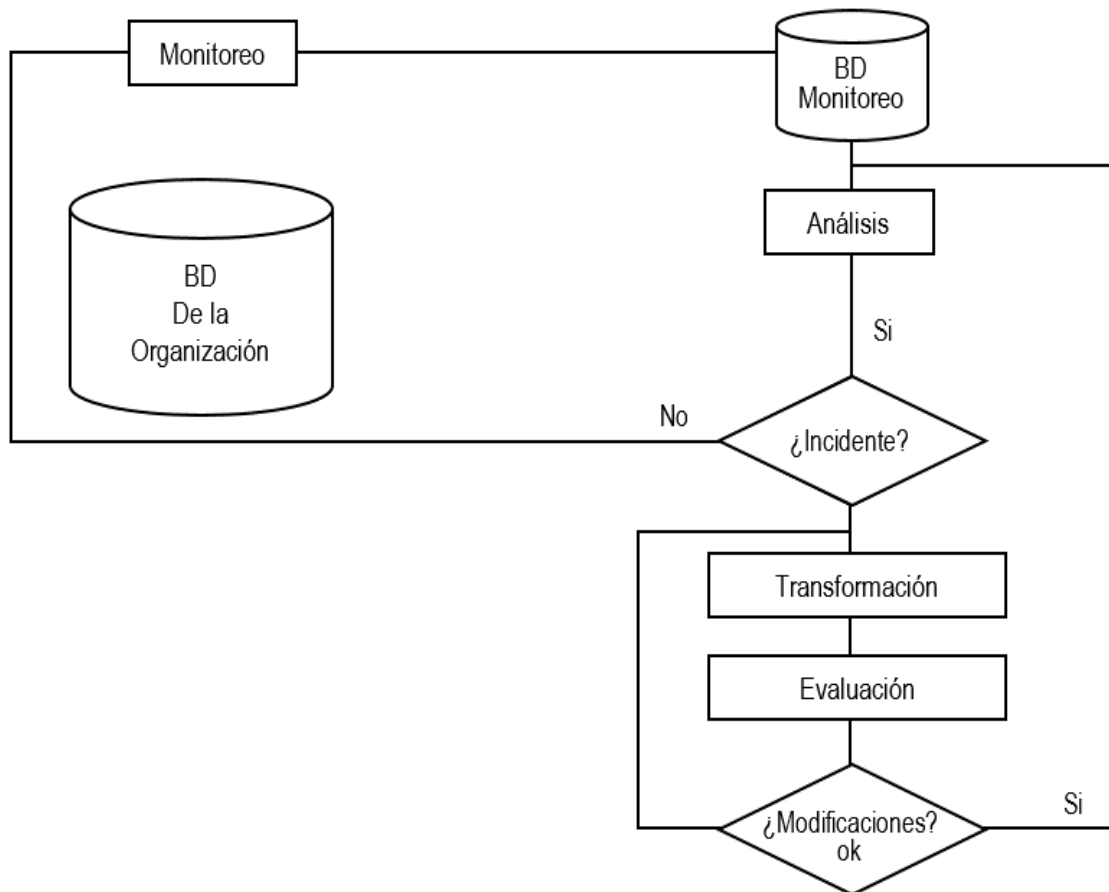


Figura 26. Proceso de Self-Management en Base de Datos

4.3 Modelo Propuesto.

El modelo propuesto se basa en la automatización de las 4 funciones de la Administración General (Monitoreo, Análisis, Transformación y Evaluación). 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 dadas por un especialista (experto), tal como se representa en la Figura 27.

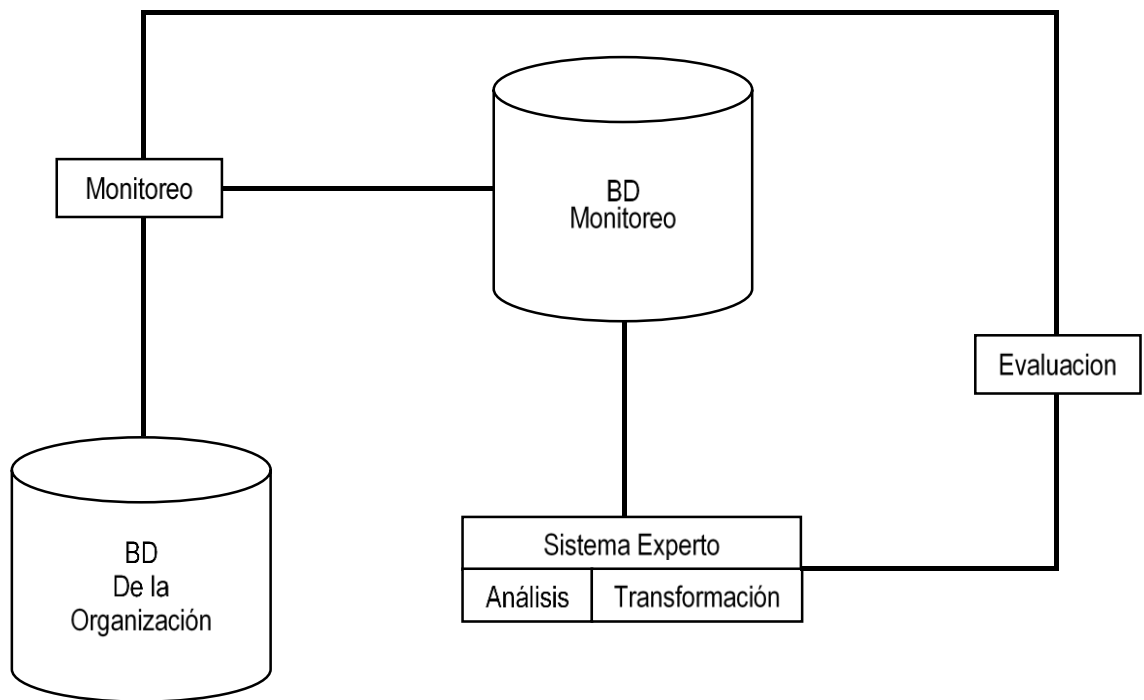


Figura 27. Modelo Propuesto de Self-Management en Base de Datos

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).

4.4 Descripción Específica del Modelo Propuesto.

Previo a la aplicación del modelo propuesto se debe definir el problema de ABD al que va a ser aplicado. El desarrollo de la solución del problema de ABD debe considerar la estructura del modelo de Self-Management a aplicar, el que esta basado en las siguientes funciones:

- Monitoreo
- Análisis
- Transformación
- Evaluación

A continuación, describiremos las funciones del modelo propuesto.

4.4.1 Monitoreo.

Una de las tareas de la ABD es el monitoreo del sistema de BD para saber cómo están funcionando, vigilando y estar alerta de su funcionamiento. El monitoreo nos permitirá planificar futuras modificaciones y cambios para optimizar el funcionamiento del sistema de BD.

Consideramos que existen dos tipos de monitoreo:

a) Monitoreo específico.

Se monitorea para tareas específicas como la determinación de un problema o si se realizan pruebas. El monitoreo específico se utiliza cuando queremos determinar una situación puntual en la que buscamos encontrar una explicación a un evento, problema o cambio realizado.

b) Monitoreo preventivo.

Se monitorea interrupciones de servicios y alerta sobre posibles problemas lo que nos permitirá descubrir oportunidades de cambios en nuestro sistema de BD que provocan o podrían provocar problemas en un futuro.

En el modelo propuesto, la función de monitoreo registra datos de los eventos de los procesos, datos extraídos del monitor del DBMS, datos del catálogo del DBMS y datos generados por las aplicaciones.

La función de monitoreo constantemente está registrando en una base de datos los eventos que pueden generar una alerta, las reglas que controlan lo que se debe registrar en el monitoreo dependen de una base de conocimientos.

A partir de dichos datos se generan archivos que serán utilizados por la función de análisis.

4.4.2 Análisis.

Los datos registrados en la base de datos de monitoreo constantemente están siendo analizados para detectar si se ha producido un incidente o un desvío significativo (se tiene una tarea que se activa en un periodo de tiempo determinado para que se active

la función de análisis), para lo cual se apoya en las reglas que están especificadas en la base de datos de conocimiento de la función de análisis. En el caso de se encuentre un incidente o desvió significativo, se genera una alerta de nivel crítico.

Los eventos que generan alertas son aquellos que han sido identificados como un problema crítico para la organización y que no tienen una solución convencional. Por ejemplo, se dejó de grabar una cantidad mayor a un umbral determinado de registros en una tabla que almacena las transacciones procesadas, eso significa que no se están procesando transacciones porque se están encolando.

Se revisa el archivo generado en la función de monitoreo y si se determina que existe un incidente, se envía una alerta.

Las categorías de alerta están dadas por colores que indican el nivel o intensidad del riesgo para que se tomen las medidas necesarias.

Las categorías de alerta son:

- Alerta roja, cuando se ha llegado a un umbral o a un estado crítico en el que se pone en riesgo la continuidad del servicio y se tiene que tomar una acción.
- Alerta amarilla, indica que hay un aumento de riesgo o de intensidad que todavía no es crítico, pero si continua pudiera generar un incidente.
- Alerta verde, que indica que hay una continuidad del servicio en forma normal sin problema en la continuidad del servicio.

A partir de las alertas se generan correos que avisan sobre la alerta crítica (roja) y se generan archivos con datos para la función de transformación, luego se activa la función de transformación.

4.4.3 Transformación.

En el caso de la función de análisis detecte un incidente se realizarán las acciones que la función de transformación genere para corregir la desviación producida por el incidente.

Las acciones que resultan de la función de transformación están asociadas a jobs que se generan a partir de modelos de jobs predefinidos y que a partir de los datos pasados en la función de análisis se adecuan al job para que sean ejecutados y que permitan revertir y superar el incidente.

La función de transformación utiliza una base de conocimiento para construir los jobs que permiten revertir y superar el incidente.

Una vez que se han ejecutado todas las acciones necesarias para superar la alerta rojas, se continua con la función de evaluación.

4.4.4 Evaluación.

La función de evaluación debe determinar si el incidente se resolvió, para lo que debe aplicar las reglas definidas en su base de datos de conocimiento.

Para poder determinar si el incidente se superó, la función de evaluación utiliza datos extraídos del monitor del DBMS, datos del catálogo del DBMS y datos generados por las aplicaciones.

Si la función de evaluación determina que el incidente se resolvió se envía una alerta avisando que el incidente ha sido superado y finaliza el procesamiento para el incidente.

En la figura 28 se presenta el diagrama de flujo del modelo propuesto de Self-Management de base de datos.

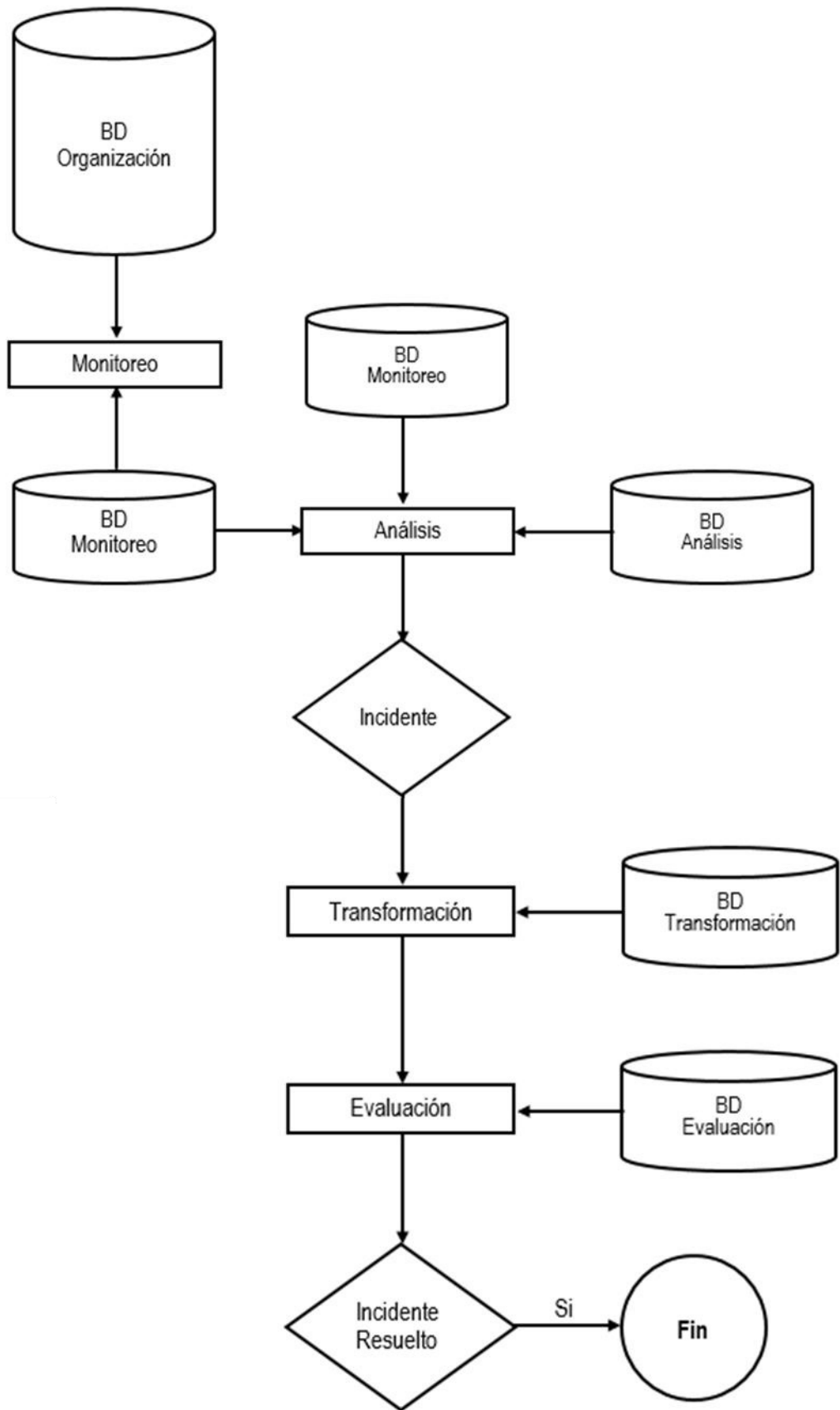


Figura 28. Modelo Propuesto de Self-Management de Base de Datos

4.5 Restricciones del Modelo.

En el modelo de Self-Management propuesto se tienen las siguientes restricciones:

- a) En el desarrollo actual del modelo de Self-Management propuesto no se consideran las funciones de Planificar y Organizar porque están relacionadas más a la gestión y requieren un desarrollo especializado y no es posible automatizar en su total dimensión.
- b) Se consideran requerimientos identificados, en el caso de requerimientos no identificados se requerirá de los especialistas para obtener las reglas que permita identificar dicha anomalía (análisis), y las reglas para corregirla (transformación).

4.6 Resumen del Capítulo.

En el capítulo 4 se ha descrito el modelo de Self-Management propuesto que es el aporte del presente trabajo de investigación, el capítulo se organizó en 5 secciones.

En la primera sección del capítulo se describe la motivación que nos llevó a desarrollar un modelo de Self-Management para base de datos, nuestra motivación se debe a la cantidad de apoyo en el desarrollo, implementación e investigación al enfoque de Self-Management en base de datos que brindan las principales empresas líderes en software para base de datos.

En la siguiente sección, se describió los fundamentos del modelo propuesto de Self-Management en base de datos, haciendo énfasis que el modelo se basa en enfocar la función de ABD desde el punto de la Administración General.

En la tercera sección, se presenta el modelo propuesto de Self-Management en base de datos.

En el siguiente capítulo del presente trabajo de investigación, se describe el modelo de Self-Management en base de datos propuesto.

CAPÍTULO 5:

SOFTWARE DESARROLLADO PARA IMPLEMENTAR SELF-MANAGEMENT EN BASE DE DATOS

En el presente capítulo, se describe el software desarrollado para implementar Self-Management en base de datos. En la primera parte se describe la arquitectura alrededor del sistema experto desarrollado. En la segunda parte se describe los componentes que conforman el software desarrollado para implementar Self-Management en base de datos.

5.1 Arquitectura del Sistema.

La arquitectura de la solución para el software desarrollado para implementar Self-Management en base de datos se muestra en la Figura 5.1.

El sistema experto utiliza el conocimiento especializado de los expertos en la ABD con los siguientes objetivos:

- Que la resolución de problemas de base de datos específicos sea más rápida.
- Reducir las pérdidas que se generan para la organización por dichos problemas
- Buscar una mejor utilización del tiempo del personal humano.
- Que la ABD sea más eficiente.
- Utilizar el conocimiento especializado de los sistemas expertos para buscar que la ABD sea más eficiente en la resolución de problemas de base de datos específicos.

El sistema experto consta de cuatro bases de conocimiento:

- Base de conocimiento de Monitoreo.

- Base de conocimiento de Análisis.
- Base de conocimiento de Transformación.
- Base de conocimiento de Evaluación.

Además, el sistema experto consta de cuatro módulos de procesamiento:

- Módulo de Monitoreo.
- Módulo de Análisis.
- Módulo de Transformación.
- Módulo de Evaluación.

Cada módulo utiliza su propia base de conocimiento, por ejemplo, el módulo de análisis utiliza la base de conocimiento de análisis y una interfaz de usuario.

El sistema experto recibe datos de la base de datos de la organización que tiene información sobre procesos, transacciones y eventos alrededor del sistema de la base de datos, como por ejemplo las transacciones que han cancelado, etc. A través del módulo de monitoreo que usa la base de conocimiento genera la base de datos de monitoreo que tiene datos sobre los procesos que más demoran, que cancelan, etc. De ahí los datos de la base de datos de monitoreo es enviada al módulo de análisis que trabaja con una base de datos.

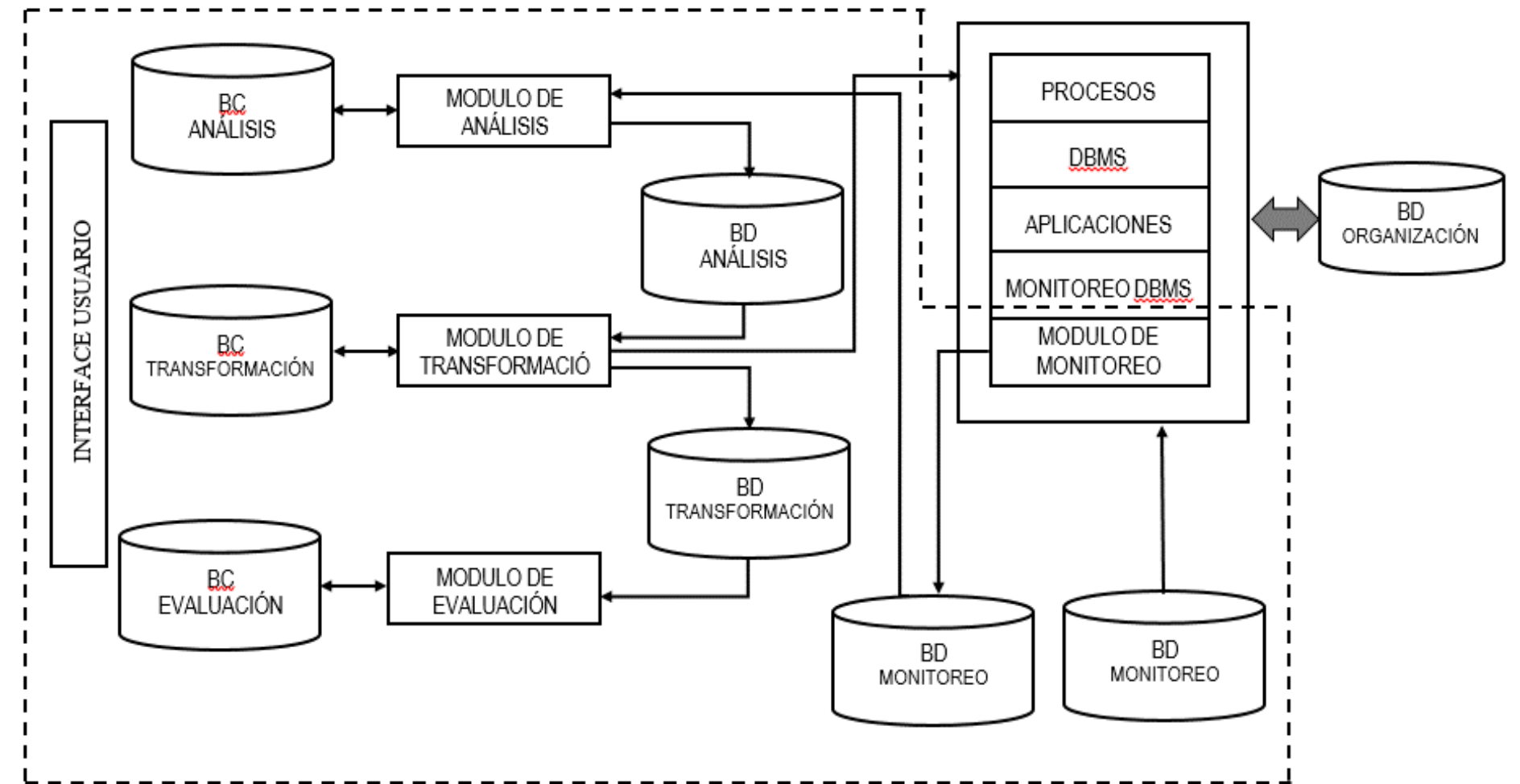


Figura. 29. Arquitectura de Sistema

Cuando un proceso del negocio utiliza la base de datos generalmente lo hace mediante un programa de aplicación que interactúa con el DBMS, mientras sucede esto, el sistema experto recolecta datos de monitoreo para lo que tiene una base de datos de conocimiento en el que tiene información de sobre los datos que se deben registrar en la base de datos de monitoreo que está en un servidor. Luego, la información de monitoreo se analiza para determinar si se produce un incidente, para lo que se utiliza la base de datos de conocimiento de análisis.

En la implementación del sistema se ha considerado que el módulo de monitoreo está en el servidor principal debido a que recolecta datos de los procesos en línea. Y los modelos de Análisis, Transformación y Evaluación están en el ambiente distribuido debido a que sus procesos son un análisis posterior.

En el caso de que el módulo de análisis determina que no hay incidente se continúa monitoreando debido a que no hay ningún incidente.

a) Modulo de Self-Management en el servidor principal.

El módulo de Self-Management en el servidor principal, es el módulo que interactúa con los procesos que se ejecutan en el servidor principal (batch, on-line, procesos enviados desde el distribuido), también interactúa con el DBMS, librerías relacionadas a las aplicaciones y el monitor del DBMS.

En este módulo, constantemente se recolecta datos de monitoreo para determinar si se produce un incidente. El módulo de Self-Management en el servidor principal trabaja con la base de datos de conocimiento de monitoreo para identificar lo que se debe monitorear.

b) Modulo de Self-Management en el Servidor Distribuido.

El módulo de Self-Management en el servidor distribuido, es el módulo que se encarga de interactuar con los módulos de Análisis, Transformación y Evaluación para la gestión de los incidentes.

En el caso de que el módulo de Análisis determina que se produce un incidente:

- El módulo de Transformación envía a ejecutar una o más acciones para lo que utiliza su base de datos de conocimiento de transformación.
- El resultado de la ejecución de una o más tareas generadas por el módulo de transformación es evaluado por el módulo de evaluación para ver si se ha corregido el incidente para lo que utiliza la base de datos de conocimiento de evaluación.
- En el caso de que el módulo de evaluación indique que el incidente fue resuelto, se termina con el proceso, en el caso de que el incidente no se haya resuelto, se reporta para ver si hay algo que modificar en el sistema experto o si no puede resolver el incidente.

5.2 Componentes.

Una vez que hemos presentado la arquitectura de la solución, describiremos los componentes que conforman el software desarrollado para implementar Self-Management en base de datos.

5.2.1 Base de Hechos.

La base de hechos está conformada por las tablas generadas para almacenar datos o características particulares de la operación inusual obtenida de las tablas operacionales del sistema de base de datos de la organización, por ejemplo, para el caso desarrollado se guarda información sobre:

- Cantidad de transacciones no procesadas.
- Procedimientos almacenados en estado detenido.

La información anterior se encuentra almacenada en tablas de la base de datos, la cual es consultada cada vez que se ejecuta la aplicación y cuyo mantenimiento es responsabilidad de la organización.

5.2.2 Modulo de Monitoreo.

El módulo de monitoreo periódicamente registra datos en la base de datos de monitoreo a partir de la base de conocimiento de monitoreo determina que datos debe registrar. En el caso desarrollado, por ejemplo, para la cantidad de transacciones no procesadas, cada cinco minutos se registra la cantidad de transacciones no procesadas, cantidad de transacciones procesadas y otros tipos de transacciones, estos datos posteriormente servirán de información para el módulo de análisis.

5.2.3 Modulo de Análisis.

El módulo de análisis periódicamente revisa la base de datos de monitoreo y a partir de la base de conocimiento de análisis determina si es que se está produciendo un problema que requiere que se genere una acción correctiva. En el caso desarrollado, por ejemplo, para la cantidad de transacciones no procesadas, si la cantidad excede las 15,000 transacciones, es que se produciendo un problema de encolamiento, esta información le servirá al módulo de transformación para que realice una acción correctiva.

5.2.4 Modulo de Transformación.

El módulo de transformación cuando recibe información de hay que realizar una acción correctiva, genera un conjunto de acciones para corregir el problema que se ha generado en el sistema de la base de datos. En el caso desarrollado, por ejemplo, si se ha producido un encolamiento de transacciones, se generan sentencias SQL que modifican (UPDATE) el estado de la transacción, de forma que ya no figuran como no procesadas y posteriormente se regresan a su estado de no procesadas (en un estado del sistema en que no se produce encolamiento de transacciones).

5.2.5 Modulo de Evaluación.

Cuando se ha producido un incidente, el módulo de evaluación verifica si es que se ha superado el incidente, para lo cual utiliza su base de conocimiento de evaluación. En el caso de que no se haya superado el incidente, se continúa con la ejecución del módulo de transformación para generar acciones correctivas. En el caso desarrollado, por ejemplo, si aún se tiene un encolamiento de más de 15,000 transacciones, se continúa con la generación de sentencias SQL que modifican (UPDATE) el estado de la transacción. Una vez que el incidente se ha superado, se elimina la alerta de incidente.

5.2.6 Interface de Usuario.

El usuario interactúa con el software desarrollado mediante pantallas que le muestran el monitoreo que se va realizando y alerta si se producido una incidencia.

5.3 Resumen del Capítulo.

En el capítulo 5 se ha descrito el software desarrollado para implementar el modelo desarrollado de Self-Management en base de datos, el capítulo 5 se organizó en 2 secciones:

En la primera sección, se describió la arquitectura del sistema experto desarrollado, el que está organizado en cuatro módulos de procesamiento: monitoreo, análisis, transformación y evaluación que se basan en cuatro de las funciones básicas de la Administración General.

En la segunda sección, se describió los componentes que conforman el software desarrollado para implementar Self-Management en base de datos, haciendo énfasis en la base de hechos y en los módulos desarrollados.

CAPÍTULO 6:

VALIDACION DEL MODELO PROPUESTO DE SELF-MANAGEMENT EN BASE DE DATOS

En el presente capítulo se describe en detalle las pruebas realizadas con el objetivo de validar el software desarrollado aplicando el enfoque de Self-Management en base de datos y confirmar el cumplimiento de los objetivos del trabajo de tesis. En la primera sección del capítulo se describe el proceso de diseño de la validación del software desarrollado. En el segundo capítulo se describen los dos casos de estudio empleados para la validación. En la tercera sección se describen la implementación en la empresa financiera del software desarrollado para aplicar el enfoque de Self-Management. En la cuarta sección se describe la forma como se implementó el modelo de Self-Management propuesto en los dos casos de estudio realizados. En la quinta sección se describen las métricas empleadas para cuantificar y sustentar la validación del modelo propuesto. En la sexta sección del capítulo se presentan los resultados obtenidos en los casos de estudios. Finalmente, en la séptima sección se hace un resumen del presente capítulo.

6.1 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:

- a) 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, cuyo nombre comercial guardaremos en reserva por motivos de privacidad.

- b) Los casos de estudio que se consideraron para la validación fueron:
- Caso de estudio A: software de alerta en la ejecución de procedimientos almacenados.
 - Caso de estudio B: software de alerta en las transacciones de fraude.
- c) Los dos casos de estudio fueron desarrollados por el ABD 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.

6.2 Casos de estudio.

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

Caso de estudio A: software de alerta en la ejecución de procedimientos almacenados.

Uno de los problemas críticos que se presentaban en la institución financiera es que por limitación de recursos computacionales o por cancelación en la lógica de la aplicación (procedimientos almacenados) se suspendía el procesamiento de transacciones debido a que el procedimiento almacenado quedaba en estado detenido (stopped), y ya no podía volverse a ejecutar el procedimiento almacenado hasta que se le daba un comando de reinicio.

El problema que el procedimiento almacenado quede detenido por cancelación es que el ABD tenía que estar constantemente monitoreando que los procedimientos almacenados estén activos y cuando uno de ellos estaba desactivado tenía que activarlo con el comando de reinicio. El problema es que el ABD tenía que estar alerta las 24 horas del día los 360 días del año.

Cuando un procedimiento almacenado cancelaba debido a problemas como: en la lógica de la aplicación o por pasar el límite de tiempo de procesamiento o por errores internos de la versión del DBMS y el ABD no se percataba de este evento, la institución financiera tenía pérdidas económicas y de imagen empresarial debido a que no se brindaba un servicio adecuado como esperan los clientes. Por ejemplo, el pago de planillas de empresas, en una oportunidad un procedimiento almacenado quedo detenido debido a una cancelación y se dejó de pagar la planilla de una empresa que implicaba 840.000 nuevos soles. En otra oportunidad se canceló el procedimiento almacenado y se dejó de pagar la planilla a más de 200 empresas. Los errores de procedimientos almacenados generaban malestar en los clientes de la institución financiera debido a que habían hecho desembolso para el pago de la planilla y los trabajadores de la empresa cliente no tenían el depósito de sus sueldos en sus cuentas quejándose a la empresa cliente, luego la empresa cliente se quejaba a la institución financiera. En algunos casos las empresas clientes se retiraban de los servicios de pago de planilla o de todos los servicios que tenían con la institución financiera.

Este problema nos brindó la oportunidad de aplicar el concepto de Self-Management en base de datos, la solución que se implemento fue que cada minuto se active un JOB que mediante un comando revise el estado de procedimientos almacenados, el resultado de esta revisión era leído por otro programa que identificaba que procedimientos almacenados estaban detenidos y generaba un correo que era enviado a la casilla de correos del Centro de Computo para que los operadores activen el procedimientos almacenados detenido mediante un comando. En una segunda versión del software en lugar de enviar un correo se genera el comando para que el procedimiento almacenado se active automáticamente sin intervención manual.

Caso de estudio B: software de alerta en las transacciones de fraude.

Uno de los requerimientos regulatorios que deben cumplir las empresas financieras es que las transacciones de depósito que pasan de un monto determinado deben informarse y analizarse para ver sino están involucradas en algún fraude.

En la institución financiera en que realizamos este caso de estudio, las transacciones de depósito son pasadas a otro sistema que lo analiza para ver si tienen un origen fraudulento.

En ese proceso se graban las transacciones en una tabla relacional y luego son extraídas y pasadas al sistema de fraudes para su análisis. En un determinado momento la aplicación grababa registro en la tabla relacional y se demoraba para hacer enviado al sistema que los analizaba y se producía un encolamiento de registros que no eran procesados por el sistema de análisis de fraude, inclusive en una oportunidad se llegó a encolar más de un millón de registros lo que generaba problemas a la institución financiera.

Inicialmente no se encontraba dónde estaba el error, no se sabía si el error estaba en la aplicación o en el DBMS o debido a otro motivo. Para que no se continúe produciendo el encolamiento de nuevos registros se consideró generar un nuevo estado de registro (99) y los registros encolados se les ponían este valor y luego poco a poco se les cambiaba el estado de 99 a 00 (en proceso).

Pero esta solución era manual y había que estar atento a que se produzca 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 Job que envía una consulta a la tabla que registra transacciones y otro programa genera un gráfico alertando si hay encolamiento.

En una segunda versión del software, pero cuando sucede encolamiento dentro de un rango determinado se activa un job que pasa las transacciones encoladas a modo 99 y luego cada 5 minutos genera comandos que pasan las transacciones 99 a modo 00 para que se vayan procesando hasta que se normalice y no haya encolamiento.

Luego se determinó que el procesamiento de la aplicación generaba el encolamiento y se informó para que modifiquen la aplicación, los problemas de encolamiento se redujeron con las acciones tomadas.

6.3 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 de tesis 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 financiera 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 de pago de planillas desarrollado por la institución financiera.
- Software de monitoreo de transacciones de fraude desarrollado por un proveedor de software.

d) Ambiente de Producción.

- Se tienen 39 bases de datos
- Hay 235 procedimientos almacenados utilizados por las aplicaciones.
- La tabla de monitoreo llega a tener hasta 5.5 millones de registros que se generan diariamente y se guarda la información del día (a excepción de los fines de semana que se guarda del sábado, Domingo y lunes, de igual manera se consideran los feriados).

6.4 Implementación del Modelo Propuesto.

En los casos desarrollados se implementó el modelo de Self-Management propuesto de la siguiente forma:

6.4.1 En el caso A.

Monitoreo.

En el DBMS utilizado se dispone de un comando que presenta información estadística de los procedimientos almacenados, como:

- Estado actual del procedimiento almacenado
Activo, puede procesarse en cualquier momento
- Detenido, detenido por una cancelación previa del procedimiento almacenado, las ejecuciones del procedimiento almacenado son rechazadas
- Detenido con cola, detenido pero las invocaciones al procedimiento almacenado forman una cola de espera
- Detenido por comando, las invocaciones al procedimiento almacenado son rechazadas
- Total de veces que el procedimiento almacenado ha estado activo
- Total de instancias actuales que el procedimiento almacenado está en espera
- Total de veces que el procedimiento almacenado que han estado en espera
- Total de veces que el procedimiento almacenado que han cancelado (abend)
- Total de veces que el procedimiento almacenado que han cancelado por tiempo de espera (timeout)

- Total de veces que el procedimiento almacenado ha cancelado desde el ultimo inicio

El resultado de dicho comando generaba un reporte

Análisis.

Se tiene una tarea que se activa cada minuto, el objetivo de dicha tarea es verificar el estado de los procedimientos almacenados, dicha tarea revisaba el reporte generado en el paso anterior y si encontraba un procedimiento almacenado detenido envía una alerta dando información del procedimiento almacenado en estado detenido.

Transformación.

Recibe una alerta con información del procedimiento almacenado detenido y genera una tarea (un job) que permite activar el procedimiento almacenado detenido mediante comando de inicio de procedimiento almacenado.

Se envía a ejecutar el job generado para que active el procedimiento almacenado detenido

Evaluación.

Determina si el procedimiento almacenado que estaba detenido ahora está activo. También se registra información histórica

6.4.2 En el caso B.

Monitoreo.

Cada 2 minutos se ejecuta un job que consulta a una tabla de transacciones procesadas, dicho job genera un reporte identificado:

La cantidad de registros procesados (que ha pasado por la validación de fraude).

La cantidad de registros no procesados (que no han pasado por la validación de fraude).

La cantidad de registros que ha sido enviado a procesarse (que ha sido enviado a la validación de fraude).

Análisis.

A continuación, se ejecuta un job que revisa el reporte anterior y en el caso de que encuentra que la cantidad de registros no procesados es mayor a 15,000 transacciones, envía una alerta

Transformación.

Recibe una alerta con la cantidad de registros que no han sido procesados.

A dichos registros les cambia el estado de procesamiento a un estado 99 (estado con problema), según el siguiente procedimiento:

- Revisar si la cantidad de registros no procesados es mayor a 15,000 transacciones
THEN
Seguir al inicio del procedimiento
ELSE
Modificar estado de registro de 99 a estado de enviado a procesarse
Ir al inicio del procedimiento

Evaluación.

Determinar que no hay más de 15,000 registros encolados (que no han pasado por la validación de fraude)

6.5 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 SBD y otro desde el punto de vista del Administrador de la Base de Datos.

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

Caso A: Software de alerta en la ejecución de procedimientos almacenados.

En este caso mediremos la eficiencia como la cantidad de veces que se presenta el problema de la cancelación de procedimientos almacenados en un periodo de un mes, esto debido:

- Cuando un procedimiento almacenado se detiene se deja de brindar un servicio o no se ejecuta una transacción por lo que se afecta la atención a los clientes generando malestar, lo que afecta a la institución financiera. La medición de la cantidad de veces que se presenta el problema de la cancelación de procedimientos almacenados nos permite cuantificar la cantidad de errores que se presentan en los procedimientos almacenados.
- El periodo de un mes se eligió porque la mayoría de los procesos se realizan de forma repetitiva mensualmente, por ejemplo, pagos de planilla, pagos de tarjetas de créditos, pagos de servicios de luz, agua, etc.

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

En este caso mediremos la eficiencia por la cantidad de veces que se presenta el problema de encolamiento en un periodo de un mes, esto debido a:

- Cuando se produce un encolamiento de transacciones de depósito que van a ser enviadas a analizar por el sistema de fraudes, se presentan demoras en la detección de una operación fraudulenta lo que podría generar que no se detecten fraudes debido a que el sistema de fraudes lo detecta después de mucho tiempo cuando ya han retirado el dinero.
- En la institución financiera en que realizamos el caso de estudio, las transacciones de depósito se realizan todos los días del año, dentro de un mes los días de mayor cantidad de movimiento son la quincena y fin de mes. Para este caso no se consideró los días previos a un depósito especial como las gratificaciones o los meses con mayor cantidad de depósitos.
- En el caso de encolamiento de transacciones fraudulentas, se eligió el periodo de un mes porque las transacciones de depósito se encolaban principalmente en la quincena o para fin de mes o en días en que había demora de los procesos batch de la noche y se tenían que procesar durante el día lo que afectaba los procesos en línea.

6.5.2 Métricas de Medición: Eficiencia del Administrador de Base de Datos.

Caso A: Software de alerta en la ejecución de procedimientos almacenados.

En este caso mediremos la eficiencia por la cantidad de veces que se presenta el problema de cancelación de procedimientos almacenados y por el tiempo empleado por el Administrador de la Base de Datos en un periodo de un mes, esto debido a:

- Como indicamos anteriormente, cuando un procedimiento almacenado se detiene se deja de brindar un servicio o no se ejecuta una transacción por lo que se afecta la atención a los clientes lo que genera malestar en los clientes lo que afecta a la institución financiera. La medición de la cantidad de veces que se presenta el problema de la cancelación de procedimientos almacenados y el tiempo empleado por el Administrador de la Base de Datos para resolver el problema, nos permite cuantificar el tiempo empleado para identificar, analizar, reportar y resolver permanentemente o temporalmente los errores que se presentan en la cancelación de un procedimiento almacenado y que lo deje operativo, para que no se vean afectados las siguientes ejecuciones del procedimiento almacenado y no se vean afectados los servicios que brinda.
- El periodo de un mes se eligió porque generalmente las cancelaciones de los procedimientos almacenados se daban con mayor frecuencia alrededor de la quincena y alrededor de fin de mes porque eran los días de pagos de planilla, pero eventualmente se presentaba cancelaciones de otros procedimientos almacenados durante los otros días de un mes. Desde el punto de vista del Administrador de Base de Datos en un periodo de un mes se cubría todas las posibilidades de cancelaciones de procedimientos almacenados.

Caso B: Software de alerta en la ejecución de transacciones de fraude.

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 ABD en un periodo de un mes, esto debido a:

- Cuando se produce un encolamiento de transacciones de depósito que van a ser enviadas a analizar por el sistema de fraudes, en la mayoría de los casos generaba que el Administrador de la Base de Datos invierta una buena cantidad de su tiempo en el proceso para resolver el problema, dejando de dedicar su tiempo en otras tareas propias de la ABD.
- Como se indicó anteriormente, en el caso de encolamiento de transacciones de fraude, se eligió el periodo de un mes porque las transacciones de depósito se encolaban principalmente en la quincena o para fin de mes o en días en que había demora de los procesos batch de la noche y se tenían que procesar durante el día lo que afectaba los procesos en línea. Durante estos días el Administrador de Base de Datos debía dedicar su tiempo a estar monitoreando que no se produzca encolamiento y si sucedía encolamiento tenía que dedicar su tiempo a resolverlo

6.6 Resultados.

La ejecución de los casos de estudio se realizó en el ambiente de Producción de la institución financiera.

Caso de estudio A: Eficiencia del Software de alertas en la ejecución de procedimientos almacenados (Sin Self-Management)

Tabla 1

Registro de Incidentes 1

DIA	INCIDENTES	TIEMPO DEL ABD	TIEMPO DE RESOLUCION
1	2	38	5
2	0	32	0
3	3	40	6
4	1	34	3
5	0	33	0
6	0	39	0
7	4	33	8
8	1	32	3
9	0	37	0
10	1	35	3
11	1	33	3
12	0	40	0
13	1	38	3
14	2	35	5
15	4	45	8

16	2	39	5
17	0	37	0
18	3	40	4
19	0	37	0
20	4	38	10
21	0	33	0
22	2	35	4
23	0	34	0
24	1	36	3
25	1	37	2
26	0	40	0
27	3	34	5
28	1	32	2
29	2	38	3
30	6	42	15

Caso de estudio A: Eficiencia del Software de alertas en la ejecución de procedimientos almacenados (Con Self-Management)

Tabla 2

Registro de Incidentes 2

DIA	INCIDENTE	TIEMPO DEL ABD	TIEMPO DE RESOLUCION
1	2	0	0.5
2	0	0	0
3	1	0	0.5
4	3	0	0.8
5	4	0	0.8
6	2	0	0.5
7	0	0	0
8	3	0	0.5
9	0	0	0
10	4	0	0.6
11	2	0	0.5
12	0	0	0
13	3	0	0.8
14	1	0	0.5
15	2	0	0.5
16	0	0	0
17	4	0	0.8
18	1	0	0.5
19	0	0	0
20	3	0	0.5
21	7	0	0.8
22	3	0	0.5
23	0	0	0
24	1	0	0.5
25	6	0	0.8
26	5	0	0.5
27	0	0	0
28	3	0	0.5
29	6	0	0.8
30	10	0	0.8

Caso de estudio B: Eficiencia del Software de alerta en la ejecución de transacciones de fraude (Sin Self-Management)

Tabla 3

Registro de Incidentes 3

DIA	INCIDENTE	TIEMPO DEL ABD	TIEMPO DE RESOLUCION
1	10	120	90
2	8	80	60
3	6	50	40
4	8	90	70
5	7	80	60
6	8	90	70
7	11	130	100
8	4	20	15
9	6	25	20
10	5	20	15
11	3	15	10
12	7	80	60
13	8	80	60
14	9	80	60
15	15	140	100
16	9	30	20
17	7	30	20
18	6	25	15
19	7	30	20
20	11	130	100
21	9	30	20
22	6	25	15
23	8	30	20
24	7	25	15
25	11	130	100
26	10	130	100
27	10	130	100
28	12	140	100
29	14	160	120
30	17	170	140

Caso de estudio B: Eficiencia del Software de alerta en la ejecución de transacciones de fraude (Con Self-Management)

Tabla 4
Registro de Incidentes 4

DIA	INCIDENTE	TIEMPO DEL ABD	TIEMPO DE RESOLUCION
1	8	20	12
2	8	20	12
3	7	20	12
4	8	20	12
5	7	20	12
6	6	20	12
7	12	20	14
8	5	20	12
9	6	20	12
10	5	20	12
11	4	25	12
12	8	25	12
13	12	25	14
14	11	30	14
15	18	30	15
16	11	25	14
17	8	25	12
18	7	20	12
19	6	20	12
20	7	20	12
21	6	20	12
22	6	20	12
23	7	20	12
24	9	20	12
25	8	20	12
26	11	20	14
27	12	20	14
28	14	25	10
29	17	25	10
30	20	30	10

Tabla 5

Resumen de los Resultados

Caso de estudio A: Eficiencia del Software de alertas en la ejecución de procedimientos almacenados		
	Eficiencia del Sistema de Base de Datos	Eficiencia del Administrador de Base de Datos
Sin Self-Management	42	12.5
Con Self-Management	0	0

Caso de estudio B: Software de alerta en la ejecución de transacciones de fraude		
	Eficiencia del Sistema de Base de Datos	Eficiencia del Administrador de Base de Datos
Sin Self-Management	26	25
Con Self-Management	0	0

6.7 Resumen del Capítulo.

En este capítulo se ha descrito las pruebas que se realizaron para validar el software desarrollado aplicando el enfoque de Self-Management en base de datos para confirmar el cumplimiento de los objetivos del trabajo de tesis. Se describió el proceso de diseño de la validación del software desarrollado. Luego, se describieron los dos casos de estudio que se emplearon para la validación y la forma como se implementó en la empresa financiera. Finalmente se describieron las métricas utilizadas para la validación del modelo propuesto y se presentaron los resultados obtenidos en la validación de los dos casos de estudio.

CAPÍTULO 7:

CONCLUSIONES Y TRABAJOS FUTUROS

7.1 Conclusiones.

Al final del presente trabajo de tesis podemos indicar las siguientes conclusiones:

7.1.1 Conclusión General.

En la presente tesis se ha desarrollado un modelo de Self-Management en base de datos que, a diferencia de los modelos existentes, permite resolver problemas críticos específicos de una organización relacionados a la ABD. El modelo de Self-Management desarrollado permite integrar los componentes de un sistema de base de datos y tiene como base el enfoque de la Administración General. Para la implementación del modelo de Self-Management se utilizó un sistema experto, haciendo que la ABD sea más efectivo y que se reduzcan los costos que se generan, lo que brinda una mejora en la eficiencia de la productividad en la ABD.

7.1.2 Conclusiones Específicas.

De acuerdo con los objetivos específicos planteados inicialmente, se presentan las conclusiones específicas de la tesis.

OE1: Revisar la literatura especializada, buscando modelos y enfoques que permitan aplicar el enfoque de Self-Management en la ABD.

Se realizó una revisión de la literatura sobre Self-Management en base de datos y se identificaron tres productos que tienen enfoque fuera de línea y tres productos en línea. Además, se encontraron cuatro modelos de Self-Management en base de datos.

OE2: Diseñar un modelo de Self-Management que considere requerimientos específicos de ABD que permita solucionar problemas críticos en las organizaciones.

Se diseñó un modelo de Self-Management que automatiza la ABD en base a cuatro de las funciones básicas de Administración General (monitoreo, análisis, evaluación y transformación), de forma que se pueden resolver problemas críticos específicos de ABD que tienen las organizaciones.

OE3: Desarrollar un software que permita aplicar los conceptos desarrollados en el modelo de Self-Management propuesto.

Se desarrolló un software que permita validar el modelo de Self-Management de base de datos propuesto. En el caso 2 de la validación se recolectó información para posteriormente graficarla y monitorear los eventos de activación del software de Self-Management. Se desarrolló un sistema experto con lo que permite validar el componente que se le da énfasis a los requerimientos específicos de una organización con respecto a la ABD, generándose soluciones basadas en procedimientos que pueden aplicarse o adaptarse a las organizaciones independientemente de los DBMS que tengan implementados.

OE4: Realizar dos casos de estudio para comprobar que el modelo de Self-Management propuesto brinda mejoras en la eficiencia de la productividad de la ABD.

Para validar la aplicación del enfoque de Self-Management, se desarrollaron dos casos de problemas críticos especiales que tenía una institución financiera de nuestro país. La solución que se aplicó permitió solucionar ambos casos en el corto tiempo y evitando que se produzcan pérdidas económicas que afecten a la organización.

- Con el objetivo de determinar si se debe realizar una reconfiguración en el SBD, en el presente trabajo, se definió un conjunto de tablas para recolectar datos sobre la ejecución de las transacciones en línea y de los procesos batch para medir el desempeño de las tareas ejecutadas y analizar y comparar con las metas fijadas y al identificar desvíos se tomen acciones en forma automática para corregirlo o se generen alertas para el ABD elabore una estrategia de solución, de forma que la función de ABD sea más eficiente.
- Uno de los objetivos del presente trabajo es no generar una sobrecarga en la tarea de monitoreo para identificar posibles problemas, adelantándose a que se conviertan en problemas. Para alcanzar este objetivo se generó un conjunto de tablas que almacenan datos sobre diferentes procesos y eventos, de forma que los procesos no consuman muchos recursos y se puedan hacer más eficientes los procesos de ABD.
- En el caso de aplicación del modelo propuesto de Self-Management, los resultados obtenidos cumplieron con el objetivo del presente trabajo de tesis.

7.2 Limitaciones.

- El modelo de Self-Management desarrollado, se ha limitado a cuatro funciones (monitoreo, análisis, evaluación y transformación), debido a que estas funciones se pueden automatizar mediante software.
- El modelo desarrollado, fue validado a través de un sistema experto orientado a un DBMS específico.

7.3 Trabajos Futuros.

- Con el fin de ampliar los casos específicos de problemas de ABD que ocurren en las organizaciones, se propone a futuro desarrollar un repositorio en donde se pueda registrar y utilizar estos casos específicos.

- Generalizar el sistema experto desarrollado, con el fin de que se pueda aplicar a diversos DBMS.
- En el presente trabajo se ha aplicado utilizando un DBMS en particular, sería recomendable aplicar dichos conceptos en otros DBMS 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 con el DBMS como por ejemplo en el caso de MS SQL Server, Integration Services o Reporting Services.
- También debemos considerar en el futuro la generación de una interface gráfica o la integración con software que permita graficar indicadores.

REFERENCIAS BIBLIOGRÁFICAS

- Barreto, A. & Wongsaroj, B. & King, T. & Cameron, T. & Diaz, S. & Cilli, J. & Muqueet, A. & Bullard, S. & Adjouadi, M. & Wolfson, O. & Graham, S. & Rishe N. (2007) Evaluation Criteria for Self-Management in DBMSs Published in Enterprise Information Systems and Web Technologies
- Belknap, P. & Beresniewicz, P. & Dageville, P. & Dias, K. & Shaft, U. & Yagoub, K. (2011), A Decade of Oracle Database Manageability, Published in IEEE Data Eng. Bull.
- Bertucci, P., Silverstein, A., Gallelli, C. & Rankins, R. (2015) Microsoft SQL Server 2014 Unleashed. USA: Sams Publisher.
- Bruno, N. & Chaudhuri S. & Konig, A. & Narasayya, V. & Ramamurthy, R. & Syamala, M. (2011), AutoAdminProjectatMicrosoftResearch: LessonsLearned, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering
- Castillo, E. (1998) Sistemas Expertos y Modelos de Redes Probabilísticas. España.
- Chaudhuri, S. & Narasayya, V. (2007), Self-Tuning Database Systems: A Decade of Progress, Microsoft Research
- Chen, W., Comeau, B., Ichikawa, T., Kumar, S. & Miskimen, M. (2008). DB2 Workload Manager for Linux, UNIX, and Windows. USA: International Business Machine Corporation.
- Club BMP (2009) <http://www.club-bpm.com/ApuntosBPM/ApuntosBPM01.pdf>
- del Mar, J. (1988). Criterios para el Diseño del Modelo Conceptual de una Base de Datos, revista MICROBYTE, Santiago de Chile, CHILE.
- del Mar, J. (1988). Criterios para la Evaluacion y Seleccion de un Sistema de Gestion de Base de Datos, revista MICROBYTE, Santiago de Chile, Chile.

- del Mar, J. (1988). Implementación Relacional para el Desarrollo de un Diccionario de Datos usando SQL/DS, revista MICROBYTE, Santiago de Chile, Chile.
- del Mar, J. (1992). Selecting a DBMS: DB2 a Very Good Alternative, 4th International DB2 User Group Conference, New York, USA.
- del Mar, J. (2018). 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 de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, Lima, Perú.
- del Mar, J. (2018). La Investigación Operativa en el Self-Management para mejorar la productividad en la Administración de una Base de Datos, Congreso Latino Iberoamericano de Investigación de Operaciones CLAIO XIX, Lima, Perú.
- Association for Computing Machinery (ACM) Edgar F. Codd. Bibliografía de Codd: ganador del premio AM Turing. (s. f.). Recuperado 6 de octubre de 2019, de http://amturing.acm.org/bib/codd_1000892.cfm
- Elmasri, R. & Navathe, S. (2010). Fundamentals of Database System. USA: Editorial Addison-Wesley, 6ta Edición.
- Elnaffar, S. & Powley, W. & Benoit D. & Martin P. (2003) Today's DBMSs: How autonomic are they? Canada, Technical Report 2003-469, School of Computing, Queen's University, Kingston, Ontario, Canada K7L 3N6.
- Feinberg, D., Adrian, M. & Cook, H. (2019) Gartner Magic Quadrant for Operational Database Management Systems. USA: Noviembre G00376881, Stamford, Connecticut.
- Foote, K. (2017) A Brief History of Database Management. Recuperado de <http://www.dataversity.net/brief-history-database-management/> [Consulta: 03 de enero de 2019].
- Giatarano, J. & Ryley, G. (2004) Expert Systems Principles and Programming. USA: Fourth Edition, Course Technology.

- Giatarano, J. & RYLEY, G. (1998) Expert Systems Principles and Programming.
- Gil, V., Olmeda, M. & Rosas, X. (2012). Fundamentos de Tecnologías de Información: Viviendo en una Sociedad Tecnológica (Volumen I). México: Editorial Digital. Instituto Tecnológico y de Estudios Superiores de Monterrey.
- Gomez, A. & Regalado, J. & Gutierrez J. (2017). Fundamentos sobre la Gestión de Base de Datos. Mexico
- Hammer, M. & Chan, A. (1976), Index Selection in a Self-Adaptive Data Base Management System, Proc. SIGMOD Int. Conf. on Management of Data (ACM)
- Harder, T & Reuter, A. (1983). Concepts for Implementing a Centralized Database Management System. Alemania, Proceedings International Computing Symposium on Application Systems Development.
- Holze, M. (2012) Self-Management Concepts for Relational Database Systems. Tesis de Doctorado en Matemática, Informática y Ciencias Naturales. Alemania: Universidad de Hamburg.
- Horn, P. (2001) Autonomic computing: IBM perspective on the state of information technology. USA: IBM.
- Hoyos, J. & Valencia, A. (2012) El papel de las TIC en el entorno organizacional de las PYMES. Colombia: Revista Trilogía número.
- IBM (2005). An Architectural blueprint for automatic computing. USA: IBM.
- Ideas: O'reilly media. (s. f.). Recuperado 6 de octubre de 2019, de <https://www.oreilly.com/ideas>.
- Kälkäjä, M. (2015) Self-Management and its pat in knowledge workers'experience of high performance. Finlandia: Tesis de maestría Department of Management and International Business. OULU Business School, University of OULU.

- Kolaczowski Piotr & Rybinski Henryk (2011) Online Index Selection in RDBMS by Evolutionary Approach. Conference: Database and Expert Systems Applications - 22nd International Conference, DEXA 2011, Toulouse, France, August 29 - September 2, 2011, Proceedings, Part II
- Kuhnash, J. & Salyer, S. (2013). Automated Management of Tier-1 Applications on VMware. USA: VMworld.
- Lorig, K. & Holman H. (2003) Self-Management Education: History, Definition, Outcomes, and Mechanisms. USA: Stanford University School of Medicine.
- Mullins, C. (2015) Database Administration: The Complete Guide to Practices and Procedures. USA: Addison Wesley Professional.
- Mullins, C. (2018). Why DBAs Must Be Jacks-of-All-Trades. USA: Pearson Education, Informit.
- Narasayya, V. & Park, H. & Syamala, M. (2011), Automatic Workload Driven Index Defragmentation, the 37th International Conference on Very Large Data Bases. Proceedings of the VLDB Endowment, Vol. 4, No. 12.
- Niu, B. & Martin, P. & Powley, W. & Horman, R. & Bird, P. (2006), Workload adaptation in autonomic DBMSs, Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative Research, Indianapolis, IN, USA.
- Oracle (2003). Oracle Database 10g: The SelfManaging Database. USA: An Oracle White Paper.
- ORACLE (2018). Viewing Automatic SQL Tuning Results. Oracle Database Online Documentation 12c Release 1 (12.1). USA: Database Administration.
- ORACLE (2019). Statistics Collection Enhancements in Oracle Database 12c Release 1 (12.1). Recuperado 21 de junio del 2020, de <https://oracle-base.com/articles/12c/statistics-collection-enhancements-12cr1>

- Padfield, B., Alapati, S. & Kuhn, D. (2011). Expert Indexing in Oracle Database 11g: Maximum Performance for Your Database. USA: Apress Media.
- Pinal, D. (2018) SQL SERVER – Introduction to Database Engine Tuning Advisor. Recuperado de <https://blog.sqlauthority.com/2015/05/11/sql-server-introduction-to-database-engine-tuning-advisor-a-k-a-dta/> [Consulta: 03 de enero de 2019].
- Popovic, J., Ljepava, D., Raveler, C. (2018). Automatic tuning in Azure SQL Database. USA: Microsoft Azure.
- PowerData (2002) Data Warehouse: todo lo que necesitas saber sobre almacenamiento de datos. Recuperado de <https://www.powerdata.es/data-warehouse>. [Consulta: 03 de enero de 2019].
- Pratt, P. & Adamski, J. (2011) Concepts of Database Management. USA: Cengage Learning.
- Quest Software Inc. (2018). Index Planning. Community Driven Innovation Toad World. Recuperado de <https://community.toadworld.com/platforms/ibmdb2/w/wiki/7310.index-planning> [Consulta: 03 de enero de 2019].
- Rani, P. & Kumar, Singh, P. & Sharma, H., (2011) Self Tuning of Oracle Database Using SQL Scripts, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, July 2011
- Robbins, S. & Coulter, M. (2014). Administración. México: 12ava edición, Pearson Educación de México.
- Rockspace & London School Economic (2017). The Cost of Cloud Expertise Report. China: Vanson Bourne.
- Sánchez, A. (2013) 16 hechos de Big Data que hablan por sí solos. Data IQ Blog, Big Data & Business Intelligence, Mercado BI, Slider, Tendencias BI. Recuperado de <https://dataiq.com.ar/blog/16-hechos-de-big-data/> [Consulta: 03 de enero de 2019].

- Shah, A. (2013) Scholar South Asian University New Delhi – India. Recuperado de <https://www.slideshare.net/assultani/dbms-and-its-models> [Consulta: 03 de enero de 2019].
- Skelley, A. (2000). DB2 Advisor: An Optimizer Smart Enough to Recommend its own Indexes. USA: Proceeding ICDE '00 Proceedings of the 16th International Conference on Data Engineering.
- Stonebraker, M. (2015) New DBMS Architectures, Reading in Database Systems, 5th Edition
- Stonerbraker, M (1989 The case for partial indexes SIGMOD Record
- Storm, A. & Garcia-Arellano, C. & Lightstone, S. & Diao, Y. & Surendra, M. (2006) Adaptive Self-Tuning Memory in DB2 Proceeding VLDB '06 Proceedings of the 32nd international conference on Very large data bases
- Valchev, M. (n.d.). Self-management Skills: List, Definition, Tips & Techniques. USA: Businessphrases.net.
- Weikum, G. & MonKukberg, A. & Hasse, C. & Zabback, P. (2002). Self-Tuning Database Technology and Information Services. Hong Kong: VLDB '02 Proceedings of the 28th international conference on Very Large Data Bases.
- Whitehead, A. (1949). Introducción a las Matemáticas. Buenos Aires: Editorial: Emecé, Biblioteca Emecé de Obras Universales, Sección II, Ciencias.
- Wilensky, R. (1983) Planning and Understanding: A Computational Approach to Human Reasoning, Addison- Wesley
- Zilio, D. & Lightstone, S. & Guy, K. (2001) SELF-MANAGING TECHNOLOGY IN IBM DB2 UNIVERSAL DATABASE Proceedings of the 2001 ACM CIKM: Tenth International Conference on Information and Knowledge Management, USA

ANEXOS

ANEXO A: Base de Conocimiento del Sistema Experto.

a) Base de Hechos.

- Tabla de Transacciones
 - Numero de secuencia
 - Indicador de envio
 - Datos de la transacción
 - Numero de secuencia de envio
 - Numero de secuencia de confirmación

- Tabla de Estado Actual de Procedimientos Almacenados Invocados
 - Esquema
 - Dueño
 - Nombre del Procedimiento Almacenado
 - Estado
 - Total de veces que el procedimiento almacenado ha estado activo
 - Total de instancias actuales que el procedimiento almacenado está en espera
 - Total de veces que el procedimiento almacenado que han estado en espera
 - Total de veces que el procedimiento almacenado que han cancelado (abend)
 - Total de veces que el procedimiento almacenado que han cancelado por tiempo de espera
 - Total de veces que el procedimiento almacenado ha cancelado desde el ultimo inicio

- Tabla de Procedimientos Almacenados
 - Esquema

- Dueño
- Nombre del Procedimiento Almacenado
- Creado por
- Identificador interno
- Lenguaje
- Identificador de la colección
- Residente
- Nombre del WLM
- Fecha de creación
- Fecha de modificación

- Tabla de Ejecución de Jobs
 - Fecha de ejecución
 - Nombre del job
 - Número del job
 - Fecha de ejecución
 - Hora de inicio
 - Hora de termino
 - Código de retorno
 - Identificador de usuario ejecutor
 - Tiempo de duracion
 - Tiempo de consumo de CPU
 - Identificación del sistema
 - Operaciones de entrada/salida

- Tabla de Ejecución de Pasos de Job
 - Fecha de ejecución
 - Nombre del job
 - Número del job
 - Nombre del paso
 - Fecha de ejecución
 - Hora de inicio
 - Hora de termino
 - Código de retorno

- Identificador de usuario ejecutor
 - Tiempo de duración
 - Tiempo de consumo de CPU
 - Identificación del sistema
 - Operaciones de entrada/salida
 - Nombre del programa que ejecuto el paso
- Tabla de Estadísticas de Transacciones
 - Fecha
 - Identificación del sistema
 - Numero de secuencia
 - Total de requerimientos de GET
 - Total de requerimientos de GET UPDATE
 - Total de requerimientos de BROWSE
 - Total de requerimientos de UPDATE
 - Total de requerimientos de ADD
 - Total de requerimientos de DELETE
- Tabla de Tablas
 - Nombre
 - Creador
 - Tipo
 - Nombre de la base de datos
 - Nombre del espacio de tabla
 - Identificador interno
 - Fecha de creación
 - Estado
 - Fecha de actualización de estadísticas
- Tabla de Índices
 - Nombre
 - Creador
 - Nombre de la tabla
 - Unicidad

- Clustering
- Base de datos
- Pool asignado
- Indicador de cierre
- Fecha de actualización de estadísticas
- Tabla de Respaldos
 - Nombre de la base de datos
 - Nombre del espacio de tablas
 - Fecha
 - Nombre del job que lo genero
 - Tipo de respaldo

b) Reglas.

- IF (EstadoActualprocedimiento almacenado = 'Activo')
 THEN puede procesarse en cualquier momento, alerta verde
 continuar
 ELSE
 IF (EstadoActual del procedimiento almacenado = 'Detenido')
 THEN detenido por una cancelacion previa del procedimiento
 almacenado, alerta roja
 ir a Transformacion
- IF (cantidad de registros no procesados) < 10000
 THEN no hay problema, alerta verde
 Continuar
 ELSE
 IF (cantidad de registros no procesados) < 15000
 THEN puede haber problema, alerta amarilla
 Continuar
 ELSE
 IF (cantidad de registros no procesados) >= 15000
 THEN hay problema, alerta roja

ir a Transformacion

- IF (un procedimiento almacenado no tiene un WLM)
THEN hay problema, alerta roja
ir a Transformacion
- IF (esta el procedimiento almacenado SPPPAUA1)
THEN hay problema, alerta roja
ir a Transformacion
- IF (esta el procedimiento almacenado SPPAGRTF)
THEN hay problema, alerta roja
ir a Transformacion
- IF (una de las principales tablas no tiene indices)
THEN hay problema, alerta roja
ir a Transformacion
- IF (un espacio de tabla tiene mas de una tabla)
THEN hay problema, alerta roja
ir a Transformacion
- IF (una de las principales tablas no tiene registros)
THEN hay problema, alerta roja
ir a Transformacion
- IF (WLM no activo)
THEN hay problema, alerta roja
ir a Transformacion
- IF (falta levantar un servicio del DBMS)
THEN hay problema, alerta roja
ir a Transformacion

- IF (no hay respaldo de una tabla)
 THEN hay problema, alerta roja
 ir a Transformacion

- IF (no estan actualizadas las estadisticas de una tabla)
 THEN hay problema, alerta roja
 ir a Transformacion

- IF (codigo de retorno de un job = 00)
 THEN no hay problema, alerta verde
 continuar

- IF (codigo de retorno de un job > 04)
 THEN no hay problema, alerta verde
 Continuar

- IF (hora de inicio de un job >= 9:00 AM y < 12:00 PM)
 THEN hay problema, alerta roja
 ir a Transformacion

- IF (hora de termino de un job >= 9:00 AM y < 12:00 PM)
 THEN hay problema, alerta roja
 ir a Transformacion

- IF (tiempo de consume de CPU > 30 minutos)
 THEN hay problema, alerta roja
 ir a Transformacion

- IF (incidente se resolvio)
 THEN validar si se resolvio
 enviar alerta verde
 continuar

