**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 10**


# "CONFIGURACIÓN CON VISTAS A AUMENTAR EL RENDIMIENTO DE LOS GESTORES RELACIONALES POSTGRESQL Y ORACLE PARA LAS SOLUCIONES DE INTELIGENCIA DE NEGOCIOS DESARROLLADAS EN DATEC, UCI"

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas**

**Autores**

**Odalmys Castellón Crespo** **Yaima de los A. Elias Alvarez**

**Tutor**

**Msc. Michaell González Jorrín**

**2009 – 2010**

**Declaración de Autoría**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los _ días del mes de junio del año 2010.

Odalmys Castellón Crespo

Yaima de los A. Elias Alvarez

**Firma del Autor(a)**

**Firma del Autor(a)**

Msc. Michaell González Jorrín

**Firma del Tutor**

**Resumen**

En el presente trabajo se propone un **proceso** y guías (“Construyendo almacenes de datos con PostgreSQL” y “Construyendo almacenes de datos con Oracle”) para la configuración del rendimiento, alineados a la Metodología de desarrollo utilizada en la “Línea de soluciones de Almacenes de Datos e Inteligencia de Negocios” de DATEC. La propuesta pretende mediante la **configuración del gestor** guiar el desarrollo hacia la obtención de aplicaciones de **almacenes de datos** más eficientes, con niveles de **rendimiento** aceptables y óptima **utilización de los recursos,** apoyado en **prácticas** agrupadas en **áreas de proceso** pertenecientes a subprocesos (Arquitectura y Diseño, Implementación y Prueba) componentes del “Proceso de configuración del rendimiento”.

**Palabras clave:** proceso, área de proceso, práctica, configuración, almacén de datos, rendimiento, recursos.

## ÍNDICE GENERAL



## ÍNDICE DE TABLAS

## Introducción

La Inteligencia de Negocios (BI)[1] es capaz de facilitar el proceso de toma de decisiones en determinada empresa o institución, mediante el análisis de los datos existentes, la misma está orientada al uso de la tecnología para recolectar y usar efectivamente la información con el fin de mejorar la operación del negocio. *"Un sistema ideal de BI ofrece a los empleados, socios y altos ejecutivos acceso a la información clave que necesitan para realizar sus tareas del día con día, y principalmente para poder tomar decisiones basadas en datos correctos y certeros"* (Inocencio, 2004). BI se compone de una serie de herramientas y técnicas de procesamiento de datos, información relacionada con la empresa o con determinada área de la misma. Según:(Antunez, 2008), (Inocencio, 2004)

Los almacenes de datos (DW)[2] es una de las tecnologías componentes de BI, conjuntamente con las tecnologías de Procesamiento Analítico en Línea[3], minería de datos y herramientas para generar reportes. Un DW es una base de datos con características especiales: es temático, integrado, no volátil (una vez que los datos se cargan en el DW no se eliminan ni modifican en un corto plazo) y almacena gran volumen de datos históricos. Según:(Cabrera, 2007), (Velazco, 2006), (Torres, 2006), (Villanueva, 2006),

La importancia de este tipo de aplicaciones radica principalmente en las funcionalidades dentro de las tecnologías de BI, brindándole a éstas soporte, pues constituyen la plataforma sobre la cual se implantan el resto de tecnologías para procesamiento de datos: OLAP, minería de datos y generadores de reportes. Según:(Vallejos, 2006), (Ruggia, 2008)

Por esta razón resulta vital la calidad de los DW, este tipo de base de datos brinda la posibilidad a las empresas de mejorar la productividad de los responsables de tomar las decisiones y la calidad de las mismas, explotando la facilidad de acceso a una gran variedad de datos provenientes de diversas fuentes externas y obteniendo conocimiento del procesamiento de esta información almacenada. La presente investigación trata específicamente la eficiencia como atributo de calidad.

En Cuba se ha ido introduciendo esta tecnología, pero no ha tenido hasta el momento un desarrollo comparable con el resto del mundo, debido a una serie de factores entre estos se destacan por ejemplo, la inexistencia de una economía desarrollada facilitadora de la informatización de la sociedad, el costo de licencias y patentes en el proceso de asimilación de las tecnologías y el hecho de encontrarse el país

[1] Inteligencia de Negocios en inglés *Bussiness Intelligence,* referida en el documento en lo adelante como BI.
[2] Almacenes de datos en inglés *DataWarehouse,* referidos en el documento en lo adelante como DW.
[3] Procesamiento Analítico en Línea en inglés *On Line Analytical Processing,* referido en lo adelante en el documento como OLAP.

bloqueado económicamente por casi 50 años hecho que limita el acceso a la información científico técnica actualizada, imposibilita el intercambio y especialización de los profesionales. Sin embargo se han obtenido avances teóricos y prácticos dirigidos principalmente a la orientación y capacitación con vistas a alcanzar el desarrollo de los DW.

La sostenibilidad de la Revolución depende de la fortaleza de la economía, por lo que constituye una prioridad nacional. Esto hace necesario profundizar en aquellas tecnologías que permitan tomar decisiones eficientemente y con calidad respecto al manejo de todas las esferas del país; en esta situación los DW pueden considerarse una de las mejores opciones, tomando como base los beneficios brindados por BI ya que estos constituyen la plataforma para este tipo de soluciones y tal como se ha explicado favorece la productividad, competencia y el desarrollo empresarial, siendo un primer aporte a la economía y a la seguridad nacional. Un segundo aporte se observa en la oportunidad de negocio y entrada de divisas al país, pues otros países y organizaciones también necesitan de estas soluciones.

Los problemas de rendimiento además de influir en la velocidad de reacción de las aplicaciones pueden considerarse tema de seguridad basado en; el planteamiento de Raúl Castro *"Considero que es poco, no es cuestión ahora de salir corriendo a repartir sin control, es hacerlo más eficientemente, es hacerlo organizadamente, y que es una tarea de primera prioridad estratégica. Ya uno de los oradores que me antecedió en el uso de la palabra se refería a que es un tema de seguridad nacional producir los productos que se dan en este país y que nos gastamos cientos y miles de millones de dólares —y no exagero— trayéndolos de otros países."* (Castro, 2009)

Donde cataloga la producción de todo lo que se de en Cuba y en general la batalla económica como cuestión de Seguridad Nacional, por lo tanto el aumento de la productividad a través del rendimiento de los productos de software es tarea fundamental, más aquellos que transfieren control sobre cualquier elemento de la economía.

Los resultados de esta investigación proporcionan además la posibilidad de prevenir ataques, tratando aspectos como la recuperación ante fallos, disponibilidad de los datos y bloqueos, estando éstos relacionados con vulnerabilidades que afectan la seguridad informática. Es menos probable que una aplicación tenga un bloqueo que afecte la disponibilidad, si el sistema es capaz de gestionar de manera eficiente las transacciones, conexiones y datos.

En la "Universidad de las Ciencias Informáticas" (UCI)[4] se llevan a cabo actualmente una serie de proyectos pertenecientes al "Centro de Tecnologías de Almacenamiento y Análisis de Datos" (DATEC)[5], donde se desarrollan DW utilizando una serie de herramientas y tecnologías dentro de las cuales se encuentran: "*Oracle Warehouse Builder 10g"*, "*Oracle 10g Database System"*, "*Oracle Enterprise Manager*", "Pentaho Schema Workbench", "PostgreSQL 8.4" y "pgAdmin III".

Independientemente de las ventajas y las oportunidades para el desarrollo de DW brindadas por uno u otro gestor, es necesario tener en cuenta las características del entorno de desarrollo: los recursos variables entre uno y otro tipo de sistemas presentes, por lo tanto no siempre es óptimo el uso de la configuración por defecto del gestor como se hace habitualmente en la "Línea de soluciones de almacenes de datos e Inteligencia de Negocios" (Línea)[6] de DATEC. El resultado del análisis de las consecuencias de este hecho, conduce a un estado desventajoso en términos de rendimiento. En adicción, los problemas de rendimiento originados son difíciles de detectar y resolver, pues tienden a solaparse unos a otros, haciéndose costoso en tiempo y recursos solucionarlos, resultando una baja calidad de las aplicaciones, tardanza en la entrega del producto e insatisfacciones del cliente asociadas.

Hasta aquí se describe una situación problemática, de donde es posible identificar el siguiente problema científico: ¿Cómo configurar el gestor de base de datos sobre el cual se desarrollan los productos de la Línea en DATEC, con vista a aumentar el rendimiento de las aplicaciones desarrolladas?

El **objeto de estudio** de esta investigación es la configuración del gestor de base de datos relacional, así como de los recursos[7] con que el sistema cuenta.

Con vistas a resolver el problema científico definido dentro del objeto de estudio, se definen como objetivos de la investigación los siguientes:

El **objetivo general** es:

Diseñar un proceso que facilite la configuración del rendimiento de las aplicaciones desarrolladas en "Línea de Soluciones de almacenes de datos e Inteligencia de Negocio" en DATEC.

Con los siguientes **objetivos específicos**:

1.Identificar aquellos elementos asociados al rendimiento en DW, de manera que queden relacionados los

[4] "Universidad de las Ciencias Informáticas", en lo adelante UCI.

[5] "Centro de Tecnologías de Almacenamiento y Análisis de Datos", en lo adelante DATEC.

[6] De aquí en adelante referida en el documento como Línea

[7] Se entiende por recursos: Hardware, Red, Sistema Operativo, Gestor de Base de Datos.

problemas de rendimiento, aspectos del desempeño del sistema y los parámetros de configuración de los gestores de base de datos relacionales para gestionar estos problemas.

2. Definir cómo utilizar en los gestores relaciones PostgreSQL y Oracle las principales técnicas de optimización existentes para mejorar la calidad de los productos de la Línea.
3. Diseñar un proceso para la configuración del rendimiento orientado a la configuración de los gestores de base de datos relacionales.
4. Validar el PROCESO diseñado en cuanto a: adecuación a la "Metodología de Desarrollo" (METODOLOGÍA)[8] utilizada, efectividad en cuanto a mejorar el Rendimiento y optimización de la utilización de los recursos.

Contenidos dentro del **campo de acción:** configuración del gestor de base de datos relacional en busca de la eficiencia en un DW.

*"El grado en el que el dato tiene atributos que pueden ser procesados y proporciona los niveles esperados de desempeño, al utilizar las cantidades y los tipos de recursos apropiados en un contexto específico de uso*" (González, 2008), tomando como base esta definición, se sostiene como **hipótesis[9]** que: es posible mediante la optimización del gestor a través de su configuración elevar la eficiencia de los productos desarrollados en la Línea en DATEC.

Para cumplir con estos objetivos han sido propuestas las principales **tareas de la investigación:**

1. Selección y revisión bibliográfica para actualizar los logros y limitaciones existentes sobre la evaluación de productos de software en cuanto al rendimiento.
2. Realización de entrevistas con personas especializadas en el tema de rendimiento en DW.
3. Análisis de modelos de calidad tales como ISO/IEC 9126, McCall, Bohem, etc.
4. Estudio de las herramientas software que soportan el proceso de desarrollo de los productos de DATEC.
5. Análisis de riesgos potenciales asociados con el atributo de calidad estudiado, los métodos y las herramientas.
6. Evaluación de la información obtenida y definir la posición como investigador.
7. Identificación de los involucrados potenciales en el diagnóstico y caracterización su marco de actuación respecto al tema de investigación.

[8] La "Metodología de Desarrollo" se refiere a la metodología utilizada en la Línea. Para diferenciarla, en lo adelante en el documento se referencia como METODOLOGÍA

[9] Esta hipótesis es de tipo proposicional y se prueba en la investigación a través del Método Delphi de validación de expertos.

8. Estudio de las principales técnicas de optimización existentes.
9. Estudio de la aplicación de estas técnicas con los gestores relacionales Oracle y PostgreSQL.
10. Confección de síntesis de la información relacionada con la evaluación de rendimiento.
11. Realización de la descripción del proceso acorde a las condiciones del entorno analizado.
12. Validar propuesta utilizando el Método Delphi.

Hasta el momento en la UCI se han realizado una serie de investigaciones de este tema tanto para PostgreSQL como Oracle. Por ejemplo la "Guía para la optimización de servidores de base de datos de PostgreSQL"[10] del pasado año 2009 de los autores Mariluz Hernández Perdomo y Enrique José González Fernández, trata el rendimiento a través de la configuración. Dado el hecho que en esta tesis tratan rendimiento en aplicaciones utilizando la configuración del gestor indudablemente tiene puntos de encuentro con el presente trabajo, sin embargo no está diseñada para el tratamiento de este factor de calidad en DW.

En la configuración de gestores relacionales, existe un gran cúmulo de información, manuales de configuración y publicaciones orientadas a aspectos específicos. Aún así es un tema muy amplio, la configuración abarca diversos tópicos y la bibliografía es mucha, por lo que orientar esta tarea hacia uno de ellos (Ej. fiabilidad, rendimiento, seguridad, portabilidad, usabilidad, manejabilidad, etc.) presupone primeramente la identificación de especificidades. La presente investigación trata el rendimiento y su repercusión en la configuración del gestor utilizado en este tipo de base de datos, lo cual constituye un elemento de alta importancia y novedad en la investigación.

El aporte de este trabajo radica en la forma de ordenar las actividades de configuración del rendimiento dentro de una única vista: el **"Proceso de configuración del rendimiento para la Línea en DATEC"**[11] y el uso de las guías de configuración del rendimiento para los gestores de base de datos relacionales PostgreSQL y Oracle, denominadas: **"Construyendo almacenes de datos con PostgreSQL"** y **"Construyendo almacenes de datos con Oracle"** respectivamente. El conjunto de PROCESO y guías orientan el proceso de desarrollo a mejorar el rendimiento de los DW atendiendo aspectos como; el comportamiento interno, la utilización de recursos, implementación de técnicas de optimización conocidas,

[10] En este trabajo se estudia el estado del arte de las formas de optimización en la actualidad, realizada por los gestores de bases de datos más conocidos. Esta guía está formada por un grupo de pasos y consultas SQL.
http://bibliodoc.uci.cu/TD/TD_2236_09.pdf

[11] A partir de este momento PROCESO

a través de la configuración, además de las limitaciones de cada gestor para el desarrollo de este tipo de aplicaciones.

El documento de tesis está compuesto por cuatro capítulos; en el primero se expone el estado del arte así como el marco teórico de la investigación. El capítulo dos describe el PROCESO y el capítulo tres las guías de configuración del rendimiento para cada gestor de base de datos: “Construyendo almacenes de datos con PostgreSQL” y “Construyendo almacenes de datos con Oracle”. Finalmente un cuarto capítulo donde se detalla la aplicación del Método Delphi de validación de expertos en dos fases, para el conjunto de solución: PROCESO y guías de configuración descritos en los capítulos anteriores.

## Capítulo 1: Fundamentación Teórica

### *Introducción*

La Inteligencia de Negocios se compone por un conjunto de herramientas y técnicas que combinadas soportan el proceso de la toma de decisiones en una institución determinada. Tienen asociado un DW, el cual es un tipo de base de datos con características especiales: es histórico, no volátil, orientado a temas e integrado. La organización de los datos y el diseño del DW son aspectos muy importantes para la calidad final de los productos, es bien conocido hasta el momento, que la eficiencia es resultado de niveles aceptables de rendimiento aunado a la óptima utilización de recursos, cuestión que los convierte en puntos clave en el tratamiento de las aplicaciones y forma parte de los factores de calidad a tener en cuenta en un cualquier tipo de producto de software. (Coruña, 2007)

Es objetivo de este capítulo evidenciar el beneficio que aporta a las tecnologías componentes de BI, la relación con DW eficientes para el procesamiento de los datos, así como las técnicas de optimización utilizadas hasta el momento por los dos gestores de bases de datos, con características relacionales (PostgreSQL y Oracle) objetos de la investigación, teniendo en cuenta cuánto se ha avanzado y el estado en el que se encuentran. Además una breve descripción de la gestión de procesos, los procesos y estructuras utilizadas para organizar la solución en los siguientes capítulos.

### *1.1.Almacén de Datos*

Un DW es una gran colección de datos que recoge información de fuentes múltiples en sistemas operacionales dispersos, y cuya actividad se centra en la toma de decisiones, es decir, en el análisis de la información en vez de en su captura. Los DW proporcionan al usuario una interfaz consolidada única para los datos, lo que hace más fácil escribir las consultas para la toma de decisiones. (Velasco, 2007). Además de lo descrito es necesario señalar que la estructura y lógica diferenciada de un DW frente a un "Sistema de Procesamiento de Transacciones en Línea" (OLTP)[12] determinan más allá de las funcionalidades, los problemas, el tratamiento requerido a los datos y las necesidades del sistema donde radican.

| Base de Datos Operacional | Almacén de Datos |
|---|---|
| Datos Operacionales | Datos del negocio para Información |
| Orientado a aplicación | Orientado al sujeto |
| Actual | Actual + Histórico |
| Detallada | Detallada + Resumida |
| Cambia continuamente | Estable |

Ilustración 1: Diferencias entre DW y Base de Datos Operacionales. (Cabrera, 2007)

#### 1.1.1.Características

A continuación se describen las características y funcionalidades de los DW tomadas de los autores: (Kafati, 2008), (Cursada, 2009), (Kimball, 2002)

- **Organizado en torno a temas:** La información se clasifica en base a los aspectos que son de interés para la empresa.
- **Integrado:** Es el aspecto más importante. La integración de datos consiste en convenciones de nombres, codificaciones consistentes, medida uniforme de variables, etc.
- **Dependiente del tiempo**: Esta dependencia aparece de tres formas: La información representa los datos sobre un horizonte largo de tiempo, cada estructura clave contiene (implícita o explícitamente) un elemento de tiempo (Día, semana, mes, etc.) o la información, una vez registrada correctamente, no puede ser actualizada.
- **No volátil:** El DW sólo permite cargar nuevos datos y acceder a los ya almacenados, pero no permite ni

[12] Del inglés *On Line Transactional Processing,* de aquí en adelante OLTP.

borrar ni modificar los datos.

Las funcionalidades de este tipo de soluciones son muchas y todas asociadas a procesos de mejoras en la empresa u organización, a competitividad, estrategias de negocios y toma de decisiones. Son infinitas las aplicaciones que se derivan de las ventajas que brindan este tipo de soluciones para el negocio. El DW organiza y orienta la información histórica de una empresa desde la perspectiva del usuario final, provee a las tecnologías de análisis de datos, y esta combinación (almacenamiento y análisis) permite que la información sea consultable a través de la explotación de los siguientes aspectos: Integración de bases de datos heterogéneas (relacionales, documentales, geográficas, archivos, etc.), ejecución de consultas complejas no predefinidas visualizando el resultado en forma gráfica y en diferentes niveles de agrupamiento y totalización de datos, agrupamiento y desagrupamiento de datos en forma interactiva, análisis del problema en términos de dimensiones, control de calidad de datos.

### 1.1.2.Eficiencia del SGBDR

En gestión y administración, el rendimiento de una base de datos se expresa en función de los tiempos de respuestas, estos deben ser razonables. ¿Qué significa que los tiempos de respuesta sean razonables? Se refiere al tiempo que se espera para recibir una respuesta de la base de datos, el mismo debe ser proporcional a la complejidad del procesamiento que el gestor tenga que hacer para dar respuesta a la consulta. (Valladolid, 2009), (Cortes, 2008 )

¿A qué debe prestársele atención cuando se habla de rendimiento en una base de datos? Desde el punto del funcionamiento del sistema, existen tres aspectos fundamentales: La cantidad de tiempo con el máximo o mínimo nivel de funcionamiento, los tiempos de respuesta de procesamiento por lotes o consultas de producción, tiempo de finalización de operaciones de copias de seguridad y la restauración de la base de datos.

El gestor debe permitir la perfecta definición de todos los datos. Es decir, debe permitir incorporar a las estructuras todos aquellos objetos necesarios para completarlas y debe permitir incluir todos los atributos necesarios para definir a los objetos. El principal objetivo de la implantación de una base de datos es poner a disposición de un gran número de usuarios un conjunto integrado de datos y que estos datos puedan ser manipulados por los diferentes usuarios. El SGBD debe garantizar que esos datos seguirán siendo coherentes después de las diversas manipulaciones. (Castillo, 2008), (Serrano, y otros, 2007), (Cortes, 2008 )

### 1.1.3.Calidad de Software

*"La ISO 9126 es un estándar internacional que plantea un modelo normalizado que permite evaluar y comparar productos de software sobre la misma base basándose en seis aspectos: Funcionalidad, Fiabilidad, Usabilidad, Portabilidad, Mantenibilidad y Eficiencia"* (Quiñones, 2009). Éste está revisado por el proyecto SQuaRE, ISO 25000:2005, que sigue los mismos conceptos, esta norma tiene en los modelos McCall y Boehm a dos antecesores que supusieron un gran impacto en la medida de software. El mismo consta de 4 partes fundamentales: Parte 1: Modelo de Calidad, Parte 2: Métricas externas, Parte 3: Métricas internas y Parte 4: Calidad en las métricas de uso.

La norma define un modelo de calidad basado en dos partes bien identificadas: La calidad interna y externa y La calidad de uso. La *calidad interna*, entendida como la totalidad de las características del producto software desde un punto de vista interno, y la *calidad externa* definida como la totalidad de las características de producto software desde un punto de vista externo, influyen en la calidad del proceso, al mismo tiempo que la calidad de uso influye sobre las anteriores. Además establece una serie de categorías y sub-categorías que establecen la calidad del software de la manera siguiente:

| Calidad del sw. (interna y externa) | | | | | |
|---|---|---|---|---|---|
| **Funcionalidad** | **Fiabilidad** | **Facilidad de uso** | **Eficiencia** | **Mantenimiento** | **Movilidad** |
| Idoneidad | Madurez | Fácil comprensión | Comportamiento frente al tiempo | Facilidad de análisis | Adaptabilidad |
| Exactitud | Tolerancia a fallos | Fácil aprendizaje | Uso de recursos | Capacidad para cambios | Facilidad de instalación |
| Interoperatividad | Capacidad de recuperación | Operatividad | Adherencia a normas | Estabilidad | Coexistencia |
| Seguridad | Adherencia a normas | Software atractivo | | Facilidad para pruebas | Facilidad de reemplazo |
| Adherencia a normas | | Adherencia a normas | | Adherencia a normas | Adherencia a normas |

Ilustración 2: Calidad de software. Interna y Externa, (Escorial, 2006) Referente a la Norma ISO/IEC9126.

#### 1.1.3.1.Descomposición jerárquica

Los modelos de calidad asumen en su mayoría el punto de vista del usuario, considerando el software como un producto a evaluar. Partiendo de los denominados *factores de calidad* entendidos como atributos

de calidad (por lo general se trata de atributos externos tales como la facilidad de uso o de mantenimiento, aunque también internos como la eficiencia), éstos se descomponen en otros de más bajo nivel denominados *criterios de calidad* y que suelen ser atributos internos.

Una vez alcanzado este nivel se procede a una segunda descomposición, realizada de forma que los criterios de calidad sean asociados a atributos que pueden ser medidos a través de las denominadas *métricas de calidad*. Factores y criterios han de estar relacionados de forma que la influencia entre ambos se establezca de forma clara.

Se considera métrica como una medida directa de un atributo simple. Las métricas del software se combinarán para obtener la medida de la calidad. (Guerrero, 2006), (Escorial, 2006), (Parra, 2007)

#### 1.1.3.2.Modelo McCall

McCall presenta en su modelo tres puntos de vista de calidad: operación del producto, revisión del producto y transición del producto; incluye 41 métricas, entendidas como medidas directas de los atributos, 21 criterios de calidad y 11 factores de calidad.

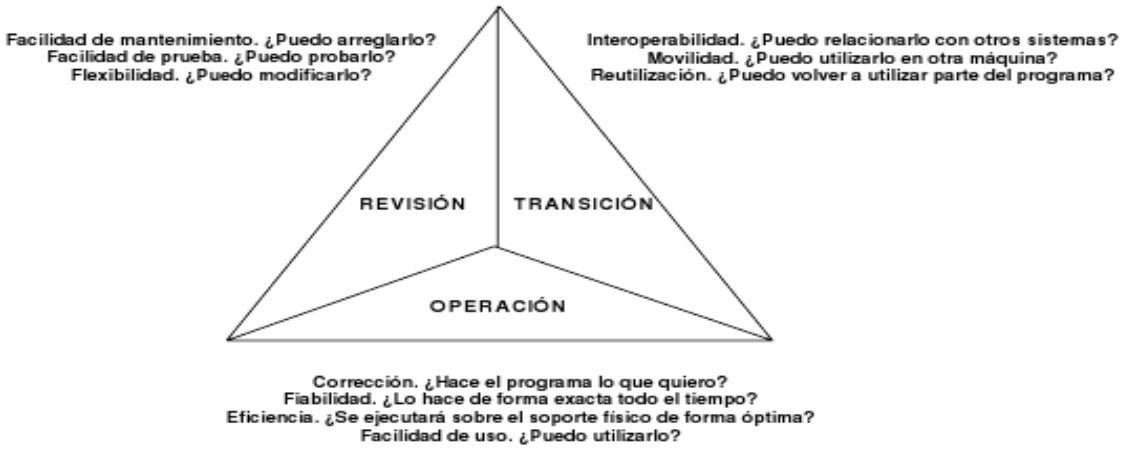


Ilustración 3: Modelo McCall (Escorial, 2006) Referente a la Norma ISO/IEC9126.

#### 1.1.3.3.Modelo Boehm

Este modelo, al igual que el propuesto por McCall, es un modelo fijo sin posibilidad de ser modificado o adaptado por el usuario. Los criterios y factores son determinados y fijos de forma que la medida de la calidad debe ajustarse a estas definiciones y a las relaciones entre criterios y factores de calidad que el modelo propone.

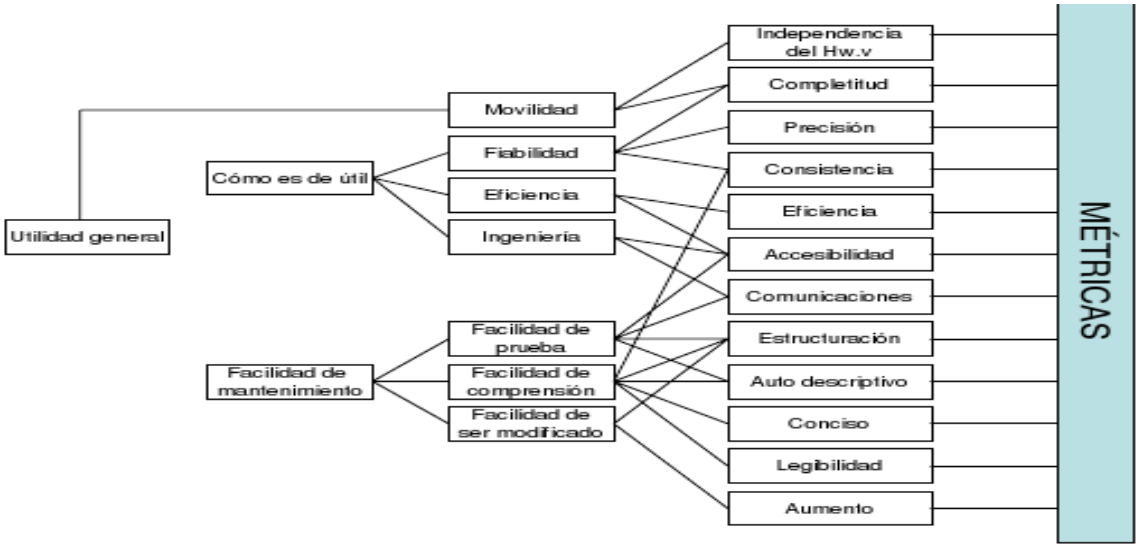


Ilustración 4: Modelo Boehm (Escorial, 2006) Referente a la Norma ISO/IEC9126.

Estos modelos (Boehm y McCall) representaron los primeros intentos por cuantificar la calidad del software como producto a través de la descomposición jerárquica en árbol.

### 1.1.4.Eficiencia en DW

Visto que un DW es un tipo de base de datos que se utiliza en la inteligencia de negocio, por tener la característica de que extrae y depura la información de distintas fuentes y que además, puede considerarse temática, histórica, no volátil, entre otras características y siendo necesario el buen funcionamiento de una gran cantidad de recursos heterogéneos, se llega a la conclusión de que: el rendimiento o la eficiencia en un DW estará dada por la capacidad que tenga de retornar una respuesta que proporcione un conocimiento fiable y un resultado del análisis del contenido de sus datos dentro de límites razonables de tiempo, optimizando la utilización de recursos varios (Ej. Hardware, aplicaciones tanto clientes como de base de datos, la base de datos, red y sistema operativo) (Serrano, y otros, 2007)

## *1.2.Almacén de Datos e Inteligencia de Negocios*

BI es una herramienta que pone a disposición de los usuarios la información correcta en el lugar correcto. Son múltiples los beneficios que ofrece a las empresas, entre ellos: la generación de una ventaja competitiva. Hay una gran variedad de soluciones de BI que en suma, son muy similares y siempre tienen asociado un DW, pero para que se considere completa debe reunir cuatro componentes: multidimensionalidad, minería de datos, agentes y DW. Según:(Bosquet, y otros, 2005), (Larrain, 2005), (López, y otros, 2004), (González, y otros, 2007), (Castro, 2009)

En este trabajo se tratará lo concerniente a los DW, estos repositorios de información representan la plataforma para emitir los análisis de datos y explotación de conocimiento a cargo de los procesos especializados como OLAP y minería de datos. Con respecto a la capa de consulta, esta constituye la herramienta que produce los elementos de información necesarios para la toma de decisiones. Así mismo, al incorporar el nivel de administración de conocimiento, se puede sistematizar la toma de decisiones rutinarias a partir de la información seleccionada del DW. (Vallejos, 2006), (López, y otros, 2004), (Villanueva, 2006)

### *1.3.Proceso y área de proceso*

Con el objetivo de organizar la solución al problema científico identificado en el campo de acción en el que se desenvuelve la presente investigación, se utilizan los siguientes conceptos para definir las estructuras en las que se organiza esta solución: proceso y área de proceso.

#### 1.3.1.Procesos

Para organizar la solución, se utiliza entre todas las estudiadas esta definición por ser la más acorde a la solución propuesta: *"El conjunto de actividades secuenciales que realizan una transformación de una serie de inputs (material, mano de obra, capital, información, etc.) en los outputs deseados (bienes y/o servicios) añadiendo valor"* (Mira, y otros, 2006)

La gestión de procesos plantea que:

Un proceso es un conjunto de actividades que se ejecutan de forma secuencial y organizada y además se desarrollan con un fin común, satisfacer plenamente los requerimientos del cliente al que va dirigido. Consta además con una entrada y produce una salida, utilizando una serie de recursos heterogéneos, se rige por un mecanismo de control. (Mira, y otros, 2006)

Existen distintos tipos de procesos:

- **Proceso clave**: constituye el objeto central de este trabajo, representa básicamente el cumplimiento de un objetivo, y puede estar o no compuesto por procesos de soporte.
- **Proceso de soporte**: Tiene como misión apoyar a un proceso clave.
- **Proceso estratégico**: Orientan y dirigen los procesos en una organización.

Es necesario cuando se describe un proceso comenzar por: Definiendo la misión, identificar el cliente y sus necesidades, definir los responsables de cada una de las actividades del proceso y si es necesario agruparlas en procesos de soporte, establecer el método de evaluación, definir los criterios, indicadores y

estándares de calidad, para evaluar los resultados obtenidos después de concluido el proceso, monitoreo del progreso del proceso, con el fin de aplicar mejoras en el mismo.

#### 1.3.2.Área de proceso y prácticas

CMMI utiliza el concepto de área de proceso para agrupar determinadas prácticas que se desarrollan colectivamente con el fin de alcanzar una meta común. En este caso la solución utiliza las áreas de proceso con el mismo objetivo, en las mismas se agrupan un conjunto de prácticas cuyo objetivo es orientar el trabajo hacia la mejora del rendimiento, aplicando el proceso que representa la solución del problema científico expuesto. (Brualla, 2008), (Palacio, 2006). Las prácticas agrupan un conjunto de actividades que comparten un fin común.

### *1.4.Estado de la técnica en el desarrollo de soluciones para DW.*

#### 1.4.1.Situación Internacional

En la actualidad los DW se han convertido en herramientas indispensables para apoyar los análisis y la toma de decisiones de una institución, sus objetivos incluyen la reducción de los costos de almacenamiento y una mayor velocidad de respuesta frente a las consultas de los usuarios. El mundo evoluciona constantemente, y cada día las empresas que tienen mayor número de aplicaciones automatizadas, almacenan la información diaria en grandes Bases de Datos y pueden conocer al momento datos precisos que deseen, es con este propósito que actualmente es utilizada dicha tecnología.

**Principales productos y empresas que desarrollan DW**

*"Después de años de avance irregular, las empresas líderes han comenzado a basar sus estrategias competitivas en el sofisticado análisis de datos empresariales."*(Davenport, 2009)

Según un artículo de *Gartner Research*, la falta de conocimiento es la mayor amenaza para las empresas modernas. Dentro de las principales empresas que desarrollan este tipo de soluciones a nivel mundial, es posible mencionar entre las más importantes a: "Teradata", "DatAllegro", "Dataupia", "Greenplum", "Aster Data", "Vertica", y encabezando la lista como líder hasta el momento "Netezza". *"Netezza es el líder indiscutible hoy en día en las aplicaciones de almacenes de datos (DWH) y bases de datos analíticas. Pero, una pregunta simple ¿Por qué? Sencillamente porque han desarrollado un único producto que reúne Software + Hardware en un único Rack."* (Valmaseda, 2010). Nuevos productos enfocados al reciente término MPP (Massive Parallel Processing) significan un salto evolutivo en el análisis y el procesamiento de los datos. "*Muchas empresas se han basado en este concepto para desarrollar sus soluciones.*

*Ejemplos son muchos: Netezza, Greenplum, Teradata, Dataupia, Kognitio, Vertica, DATAllegro, Aster Data, Ab Ignitio, HP (con su Neoview) y Oracle (con su producto Exadata)"* (Valmaseda, 2010)

#### 1.4.1.1.Técnicas de optimización

##### *1.4.1.1.1.Procesamiento paralelo.*

Hay tres tipos principales de paralelismos que se pueden implementar en las aplicaciones ETL: de datos que consiste en dividir un único archivo secuencial en pequeños archivos de datos para proporcionar acceso paralelo, de segmentación (pipeline) que permite el funcionamiento simultáneo de varios componentes en el mismo flujo de datos. Un ejemplo de ello sería buscar un valor en el registro número 1 a la vez que se suman dos campos en el registro número 2 y de componente que consiste en el funcionamiento simultáneo de múltiples procesos en diferentes flujos de datos en el mismo puesto de trabajo.

##### *1.4.1.1.2.Particionamiento*

El particionamiento permite dividir tablas e índices en componentes más pequeños y adaptables, y es un requisito clave para cualquier base de datos grande con alto desempeño y alta disponibilidad. (D'Ottone, y otros, 2009)

##### *1.4.1.1.3.Índices*

El **índice** de una base de datos es una estructura de datos que mejora la velocidad de las operaciones, permitiendo un rápido acceso a los registros de una tabla. Al aumentar drásticamente la velocidad de acceso, se suelen usar éstos sobre aquellos campos donde se realizan frecuentes búsquedas. (Cesares, 2009)

##### *1.4.1.1.4.Materialización*

La vista materializada no es más que una vista definida con una sentencia SQL, que además de almacenar su definición, almacena los datos que retorna, realizando una carga inicial y después cada cierto tiempo un refrescamiento de los mismos. (Fernández, 2008)

### 1.4.2.Situación Nacional

En Cuba la aplicación de tecnologías de BI aún se encuentra reducida a un mínimo número de organizaciones, ubicadas fundamentalmente en la capital. A pesar de haberse dedicado esfuerzos orientados a la capacitación en aras de fomentar la utilización de las tecnologías de BI, estos constituyen por el momento apenas primeros pasos hacia un desarrollo futuro de las organizaciones en este campo.

*"Hasta el 2007 solo 35 entidades habían tenido experiencia en el tema, principalmente de la capital, el*

*siguiente grafico muestra como ha sido la aplicación de la BI en Cuba en el periodo comprendido entre los años 1999-2007"* (Antunez, 2008).

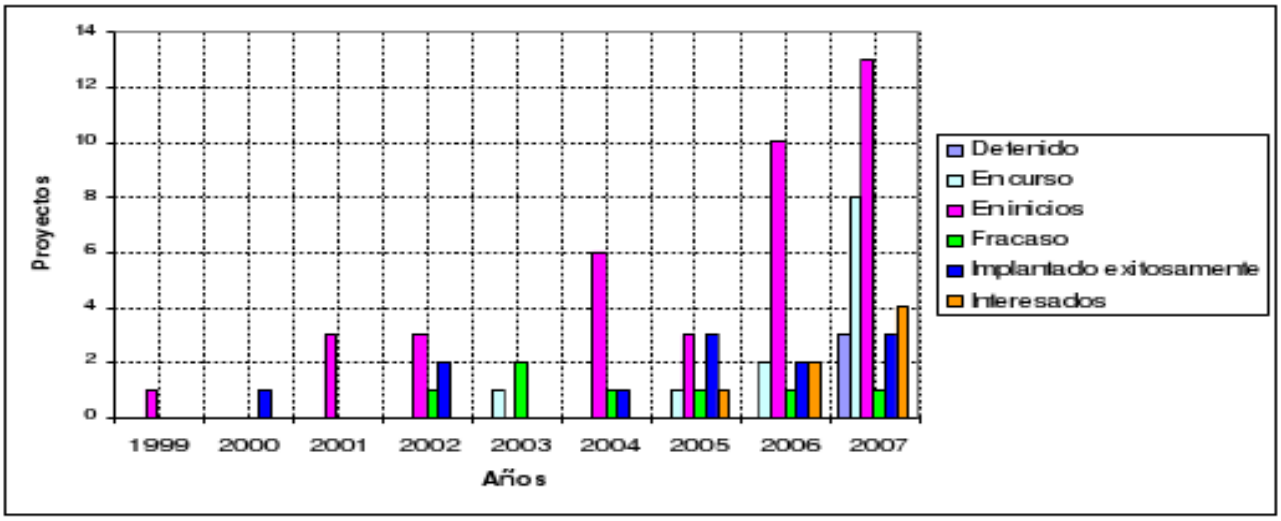


Ilustración 5: Evolución de los proyectos de DW en el periodo (1999 -2007), (Antunez, 2008)

### 1.4.3.Situación en la UCI

En la UCI, se lleva a cabo el desarrollo del estudio y aplicación de los DW, especialmente en DATEC, en este centro se han desarrollado varios proyectos.

#### 1.4.3.1.Proyectos que desarrollan DW en la UCI

El **Proyecto ONE** DW para la "Oficina Nacional de Estadísticas" (ONE). El sistema proporcionará los medios necesarios para analizar la información estadística de recogida en el Modelo 005: Indicadores generales utilizados por la ONE.

**Proyecto Integración** *(propuesta de soluciones para la integración de sistemas informáticos)*, el cual consiste en desarrollar una estrategia para la integración de los sistemas informáticos implantados que incluya el montaje de una solución de inteligencia de negocio y almacenamiento de datos para la toma de decisiones del MPPRIJ (Ministerio del Poder Popular para Relaciones Interiores y Justicia, República Bolivariana de Venezuela) sobre una arquitectura única de sistemas.

**Almacén de Datos para la UCI** es otro de los proyectos que se están realizando siguiendo la metodología. El proyecto surge como necesidad de la UCI en cuanto a la disposición de información para los directivos y poder tomar decisiones precisas ante cada situación particular en las distintas esferas de la UCI.

El **Proyecto DIE** tiene como objetivo desarrollar un DW compuesto por varios Data Mart que recojan toda la información de los procesos del Ministerio, permitiendo a partir de la misma tomar decisiones y tener un control sobre el estado de los procesos del interior y justicia que manejan.

**Sistema de Información para el INE** Desarrollar una solución tecnológica enfocada a optimizar la producción, acceso y divulgación de la información estadística del Instituto, así como, mejorar la comunicación e intercambio entre el INE y los demás entes y organismos del Sistema Estadístico Nacional, además de fortalecer su identidad y su labor.

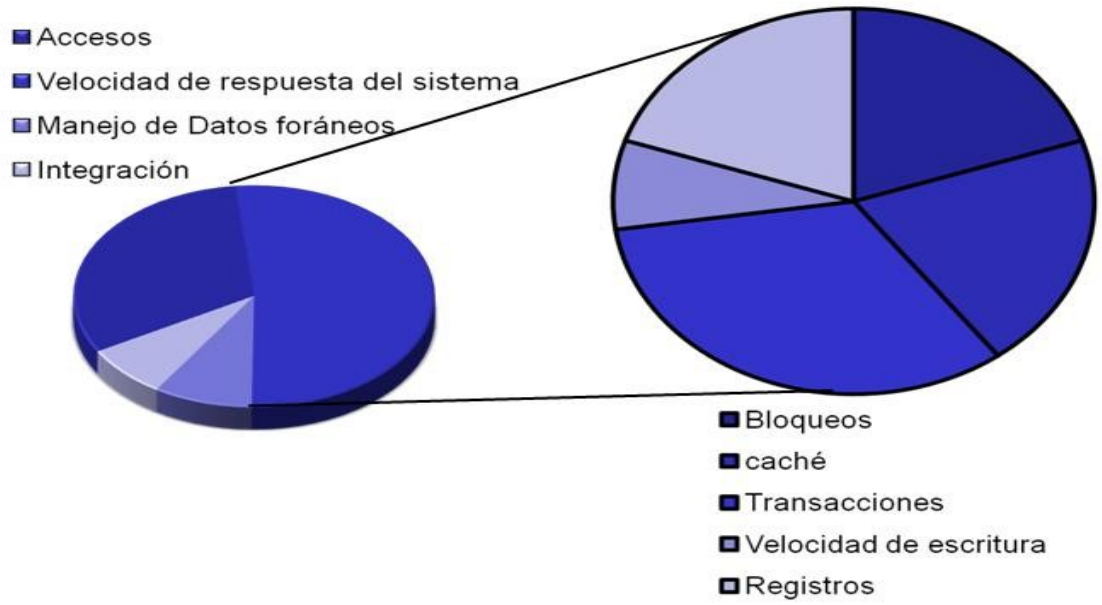


Ilustración 6: Relación problemas de rendimiento - desempeño de DW

Mediante el intercambio con el personal calificado que trabaja en la Línea, y analizando el proceso de desarrollo de los distintos proyectos realizados en la UCI que crean soluciones de DW; se conoce que utilizan la configuración por defecto o mínima del gestor relacional con el cual gestiona la información almacenada, siendo este hecho factor que propicia una serie de problemas. La velocidad de respuesta del sistema es el que más concierne al rendimiento de un DW, es consecuencia de un grupo de aspectos cuyo tratamiento determina la calidad de los productos (Ej. bloqueos, memoria caché, transacciones de los datos, velocidad de escritura etc.).

### *1.5.Construyendo almacenes con PostgreSQL*

PostgreSQL es considerado el gestor de base de datos de código abierto más avanzado en el mundo, no se puede olvidar la importancia que representa para la gestión de un proyecto comercial que PostgreSQL se encuentre bajo la licencia BSD, que posibilita que el software a construir esté libre de costos asociados por licencias, además es desarrollado por una comunidad, lo cual aporta la ventaja de que está en constante proceso de mejora y es posible el acceso al código y *releases* de cada una de estas "nuevas y mejoradas" versiones y está además diseñado según requerimientos de mantenimiento y administración más bajos que gestores propietarios, resulta importante el hecho de que esté disponible para casi la totalidad de UNIX (34 plataformas en total, compatible además con Windows).

A continuación se ilustran cuáles son las características que lo ponen por delante de otros gestores de código abierto. "*PostgreSQL es un gestor de base de datos objeto-relacional capaz de manejar complejas reglas y rutinas, ejemplo de esto es la capacidad que tiene para las transacciones, funcionalidad con consultas SQL y su optimización, entre otros aspectos, es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales*" (PostgreSQL, 2009).

PostgreSQL:

1. Soporta integridad referencial, indispensable para garantizar la validez de los datos de la base de datos
2. Tiene "Control de Concurrencia Multi-Versión" (MVCC)[13], tecnología utilizada para evitar bloqueos innecesarios, mantiene una ruta para todas las transacciones realizadas, siendo entonces capaz de manejar los registros sin la necesidad de que el usuario necesite esperar a que estos estén disponibles.
3. Utiliza una arquitectura proceso-por-usuario cliente servidor, es decir hay un proceso maestro que se ramifica para proporcionar conexiones adicionales a cada usuario que solicite una transacción, multi-proceso en vez de multi-hilo, de esta manera si una transacción se ve afectada las otras podrán seguir ejecutándose normalmente.
4. *Write Ahead Logging* (WAL)[14] es una de las características de PostgreSQL que incrementa la dependencia de la base de datos al registro de cambios, antes de que estos sean escritos en la base

[13] En inglés *Multi-Version Concurrency Control ,* de aquí en adelante MVCC
[14] De aquí en adelante WAL

de datos, lo que garantiza la existencia de un registro de transacciones a partir del cual puede restaurarse en caso de fallo la base de datos.

5. Es una base de datos 100% ACID[15]
6. Tablespaces, constituyen una etiqueta interna para un directorio físico dentro del sistema de archivos que pueden ser creadas o eliminadas en cualquier momento que se requiera y permite almacenar "objetos", tales como tablas e índices en diferentes direcciones, cuestión muy favorable a la escalabilidad y rendimiento del gestor.

El último release, PostgreSQL 8.4 tiene además de las ya mencionadas, cuestiones que hacen de él una opción válida entre los mejores gestores (independientemente de los beneficios que puede aportarle a la UCI y al país que el mismo sea de código abierto):

- Funciones de ventanas: permiten cálculos con rango agregado de rendimiento sobre una partición resultante, mucho más óptimo que un tradicional GROUP BY.
- Expresiones de tabla comunes y consultas recursivas
- Restauración paralela
- Los permisos de columna
- Mejora de índices hash
- Mejoras de rendimiento de JOIN para consultas EXISTS y NOT EXISTS
- Más fácil de usar "Warm Standby"
- Automatización del tamaño del Free Space Map(FSM)16
- Mapa de visibilidad (reduce en gran medida el *vacuum overhead* para determinadas tablas)
- Estadísticas de tiempo de ejecución por funciones.
- Nuevos módulos *contrib*: *pg_stat_statements, auto_explain, citext, btree_gin*
- Llaves sustitutas o *surrogate keys*: identificadores en la base de datos, por lo general una secuencia que combina la llave primaria y la externa, además permite mantener un control en los cambios que se realizan a las dimensiones

Según las fuentes: (Bartollini, 2009), (PostgreSQL, 2010), (PostgreSQL, 2009), (PostgreSQL, y otros, 2005)

[15] Se refiere a pruebas ACID del inglés *Atomicity, Consistency, Isolation, Durability,* de aquí en adelante ACID
[16] En lo adelante FSM

### *1.6. Construyendo almacenes con Oracle*

Cabe destacar que dicho SGBDR contiene muy buenas características y ventajas para el desarrollo de DW, además de ser uno de los gestores propietarios más avanzados del mundo y he aquí una de las desventajas que presenta, aunque es necesario señalar que además existen limitaciones por causa de costo del producto (v*er epígrafe 3.2.5*).

Oracle hace uso de los recursos de los sistemas informáticos en todas las arquitecturas de hardware, lo que permite garantizar su aprovechamiento en ambientes cargados de información, por su capacidad de almacenar y acudir a los datos de forma recurrente y contiene características como: Entorno cliente/Servidor, Gestión de grandes bases de datos, Usuarios concurrentes, Alto rendimiento en transacciones, Autogestión de la integridad de los datos y Compatibilidad. Según las fuentes: (Oracle, 2006), (Oracle, 2005), (Oracle, 2005)

En DATEC, la versión Oracle 10g es la utilizada. Una de las principales ventajas que presenta este gestor de base de datos es su arquitectura multiplataforma en el desarrollo de sus aplicaciones. Las cargas de trabajo del DW a menudo son complejas, para cubrir estas demandas dicho gestor incluye un grupo de técnicas de optimización, como son:

Técnicas para la optimización del desempeño para todo tipo de consulta y carga de trabajo:

- Cubos OLAP
- Optimizaciones de consultas
- Indices de mapa bits
- Vistas materializadas
- Particionamiento
- Paralelismo

Para concluir: crear un DW utilizando Oracle es una buena manera de consolidar y organizar todos los datos para que se puedan administrar, ver y analizar fácilmente. Con una sola interfaz de administración, características de autodiagnóstico y autoajuste.

***Conclusiones***

1.Los DW son un tipo de base de datos con características especiales cuyos requerimientos son satisfechos de forma efectiva, con la utilización de gestores relacionales.

2.La Línea de DATEC que desarrolla esta tecnología, utiliza los gestores relacionales PostgreSQL y Oracle con su configuración por defecto, lo cual resulta poco eficiente para el desarrollo de DW, por ello es necesario la creación de un proceso que se adecue a las características de la Línea y que satisfaga sus necesidades de optimización.

3.Oracle es un tipo de gestor que brinda muchas facilidades para DW, desarrolla esta tecnología desde sus inicios, por lo que en la actualidad posee las principales técnicas y herramientas para el desarrollo de este tipo de soluciones.

4.PostgreSQL aporta el beneficio de ser un potente gestor relacional de código abierto, cuestión requerida debido al proceso de migración al software libre (SWL)[17] en la Línea, la UCI y el país, escogido además por las oportunidades para el trabajo con este tipo de soluciones que es posible explotar con vistas a facilitar el desarrollo, convirtiéndola sino en la mejor, en una de las mejores opciones para DATEC.

5.La solución propuesta a la problemática que atañe a esta investigación se basa y organiza sobre conceptos de Gestión de procesos: proceso clave, subproceso de soporte y área de proceso y prácticas.

[17] Software Libre de aquí en adelante SWL

## Capítulo 2: Proceso y áreas de proceso

### *Introducción*

En este capítulo se describe el PROCESO dentro de la propuesta de solución compuesta además por las guías de configuración del rendimiento, ("Construyendo almacenes con PostgreSQL" y "Construyendo almacenes con Oracle") detalladas en el "Capítulo 3: Guías de configuración del rendimiento" de este trabajo. La razón de implementarlo está dada por la necesidad de mejorar el rendimiento de las aplicaciones desarrolladas sobre los gestores relacionales utilizados; pues el uso de la configuración por defecto implica un bajo rendimiento y calidad de los productos.

La cota para medir el resultado del PROCESO, es decir, la calidad de estas aplicaciones la brinda el ISO/IEC 9126 mediante la definición de la eficiencia y las subcaracterísticas asociadas, a las que se pretende llegar. El conjunto solución aporta las actividades y parámetros de configuración indicados para el desarrollo de DW, además resulta determinante que este se alinee a la METODOLOGÍA, por lo tanto se establece su relación partiendo de las fases de la misma en las que interviene el uso y configuración de los gestores de base de datos.

El PROCESO cuenta con **subprocesos** (Análisis y Diseño, Implementación y Prueba) lo cuales están vinculados con el ciclo de vida de la METODOLOGÍA. Las **áreas de procesos** contenidas en los subprocesos de soporte, se componen por un conjunto de **prácticas** que agrupan a su vez, las **actividades** que conforman el PROCESO y están diseñadas con el fin de resolver los problemas de rendimiento de los DW construidos en la Línea, mediante la configuración de los gestores de base de datos relacionales (PostgreSQL y Oracle).

### *2.1."Proceso de configuración del rendimiento para la Línea en DATEC".*

El PROCESO describe un camino dirigido a la optimización de las aplicaciones de DW, a través de la configuración del gestor relacional. Antes de detallar la solución es necesario, familiarizarse con los conceptos de: **área de proceso** como un conjunto de prácticas que se desarrollan colectivamente con el fin de alcanzar una meta común y ayudar a alinear el PROCESO con la METODOLOGÍA y **prácticas** compuestas por un conjunto de actividades que igualmente tienen un objetivo común. Como muestra la siguiente ilustración.

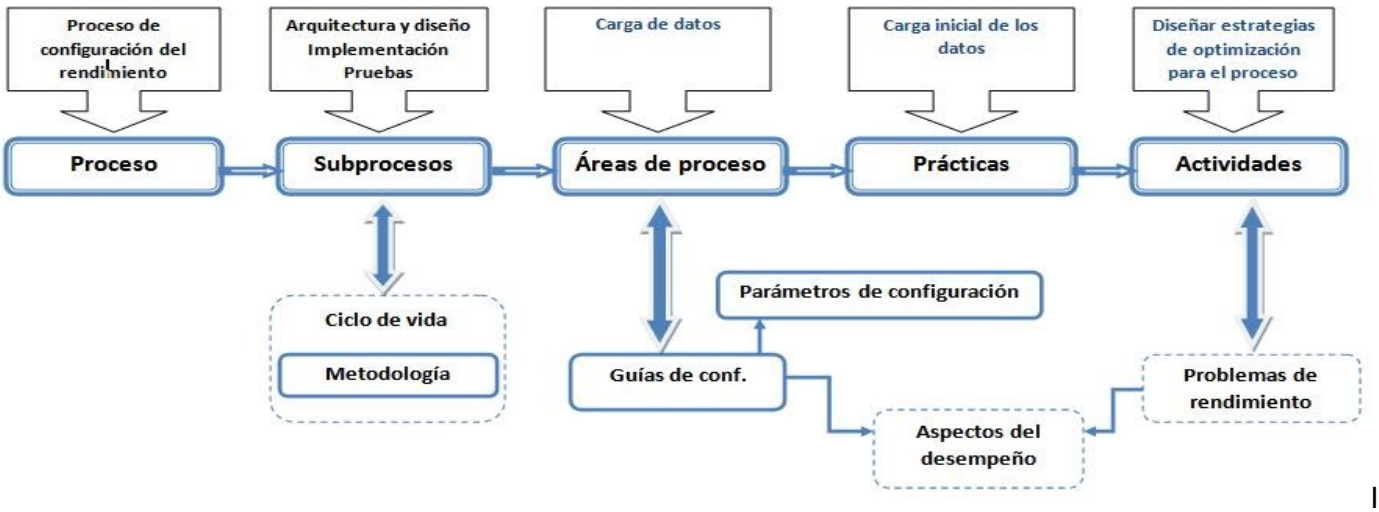


Ilustración 7: Componentes del modelo del PROCESO

Está compuesto por tres **subprocesos** de soporte que se relacionan con el ciclo de vida del DW de la manera en que se explica, donde se agrupan una serie de **Áreas de proceso** específicas, dentro de estas las pertenecientes al subproceso de Implementación se asocian a las **guías de configuración** de PostgreSQL y Oracle, estas guías contienen las herramientas y los **parámetros de configuración** que influyen en el rendimiento de las aplicaciones, necesarios para el completamiento de las actividades del subproceso Implementación. Estas áreas de proceso contienen un conjunto de **prácticas** conformadas por las **actividades** que propone la guía, dirigidas a optimizar el rendimiento de los DW y que se ejecutan de manera independiente.

De forma general las actividades tratan los principales **problemas** relacionados con el rendimiento diagnosticado en este tipo de aplicaciones, relacionados con: escaneos secuenciales, consultas

inefectivas o actualizaciones masivas, que afectan determinados **aspectos del desempeño** sobre los cuales influyen los parámetros de configuración descritos en las guías.

Se enunció en el capítulo anterior como proceso, al conjunto de actividades relacionadas que se desarrollan con un fin común, mediante las cuales se transforman entradas en una salida y necesita una serie de recursos y mecanismos de control.

Como entrada se tiene la **configuración básica** o por defecto del gestor relacional (PostgreSQL u Oracle) utilizado por la Línea y los **requerimientos no funcionales** dentro de los cuales se establece el rendimiento esperado de la aplicación y como salida, una **configuración adecuada** con vistas a proporcionar eficiencia a las aplicaciones desarrolladas. Los recursos utilizados son: **la tecnología y el capital humano de la Línea** y como mecanismo de control: las **guías de configuración** del rendimiento que se detallan en el siguiente capítulo las cuales describen la influencia y recomendaciones de configuración de los parámetros y herramientas con que cuentan los gestores para gestionar el rendimiento y los recursos, brindando el soporte necesario para la realización de las actividades del subproceso de Implementación. El ciclo de vida definido en la **METODOLOGÍA** determina según la etapa, subprocesos, trabajadores y las actividades a realizar en el PROCESO y las **normativas del ISO 9126** brindan las métricas para el rendimiento y utilización de los recursos (*véase Ilustración 9*).

El PROCESO**;** ¿Quién realiza las actividades del PROCESO? ¿Qué objetivos se persiguen con estas? ¿Quién es el beneficiario? Es decir, *trabajadores* que intervienen en el PROCESO, *objetivos* del mismo y *cliente*.

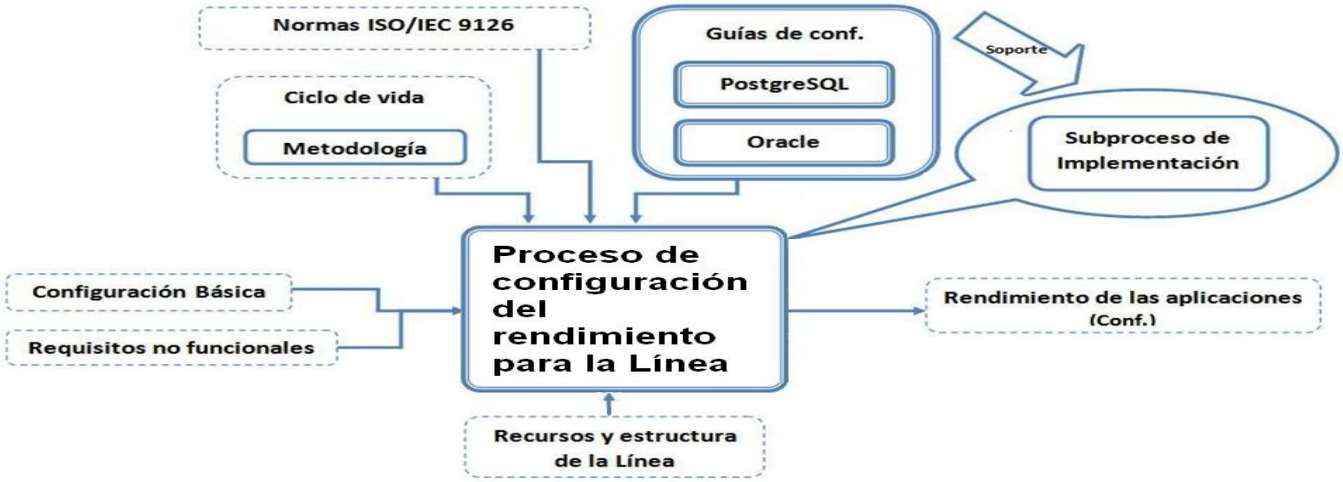


Ilustración 8: Entorno de la solución

Los **trabajadores** que intervienen son los miembros de la Línea de soluciones de almacenes de DATEC, los cuales a su vez se convertirían en los **clientes** del PROCESO, debido a que es la propia Línea el beneficiario inmediato. Los **objetivos** que se persiguen con este PROCESO son:

1.Guiar el trabajo con los gestores PostgreSQL y Oracle en la Línea con vistas a aumentar el rendimiento de las aplicaciones desarrolladas.

2.Mostrar aquellos aspectos, herramientas y elementos configurables que cada gestor brinda para gestionar el rendimiento y utilización de recursos.

3.Identificar y evaluar los aspectos que determinan el rendimiento de las aplicaciones de DW y cómo gestionarlos a través del gestor.

### *2.2.Metodología de desarrollo*

También se han venido desarrollando las metodologías para la implantación de este tipo de soluciones. En este caso, para introducir la METODOLOGÍA a utilizar en la Línea de DATEC es necesario aclarar que aún no se encuentra registrada, se tomó como base para su implantación, la "Metodología de Kimball" y como complemento a la misma y fortaleciendo la etapa del levantamiento de requisitos; se tomó lo planteado por Leopoldo Zenaido Zepeda Sánchez en su Tesis de Doctorado, orientando así el trabajo a los casos de uso, logrando estar mejor alineada con las tendencias y normas de la UCI. (Villa, y otros, 2009)

La METODOLOGÍA define los flujos de trabajo: **Estudio Preliminar y planeación, Requerimientos, Arquitectura y Diseño, Implementación, Pruebas, Despliegue, Soporte y Mantenimiento y Gestión y Administración del proyecto.**

Los grupos de trabajo se organizan por áreas de conocimiento, determinadas por la fase en la que intervienen, las tecnologías que utilizan y los componentes del DW, estos se encargan de tareas: **"Grupo de Análisis" (G-Análisis)[18], "Grupo Almacenes" (G-DW)[19], "Grupo de Extracción, Transformación y Carga de los datos" (G-ETL)[20] y "Grupo de Inteligencia de Negocios" (G-BI)[21]**

En el proceso de desarrollo intervienen según los flujos de trabajo, los grupos de la siguiente manera:

[18] En lo adelante para identificarlos como trabajadores del PROCESO, G-Análisis.
[19] En lo adelante para identificarlos como trabajadores del PROCESO, G-DW
[20] En lo adelante para identificarlos como trabajadores del PROCESO, G-ETL
[21] En lo adelante para identificarlos como trabajadores del PROCESO, G-BI

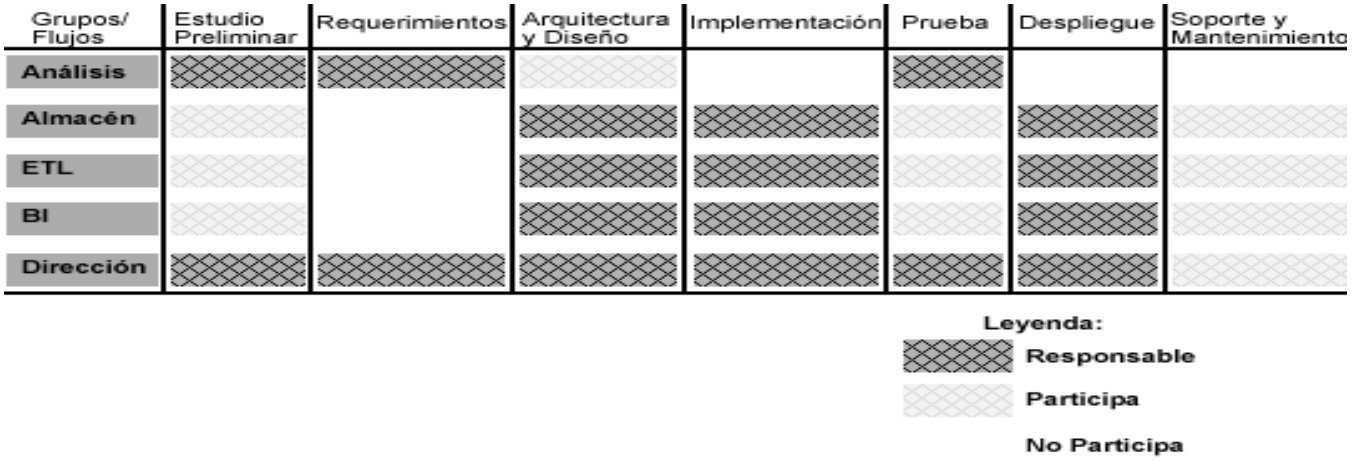


Ilustración 9: Relación entre flujos de trabajo y grupos de trabajo(Villa, y otros, 2009)

### 2.2.1. Relación entre la METODOLOGÍA y los componentes del PROCESO

Resulta entonces una necesidad adecuar el PROCESO a diseñar a esta METODOLOGÍA, como se persigue mejorar el rendimiento de las aplicaciones, se deben identificar las etapas o fases de esta METODOLOGÍA en las que interviene el gestor de base de datos (Arquitectura y Diseño, Implementación y Pruebas) y orientar las prácticas a la optimización del mismo.

Los subprocesos se llevan a cabo junto a las fases de manera secuencial: el subproceso Arquitectura y Diseño tiene como entradas la entrada del PROCESO, los requisitos no funcionales que establecen el nivel de rendimiento esperado y la configuración básica del gestor y tiene como salida los documentos de Arquitectura (“Especificaciones del diseño físico”, “Especificaciones de tablas de Hechos”, “Mapa lógico” según define la METODOLOGÍA) donde se describe el diseño del DW.

La salida de este primer subproceso conjuntamente con la configuración básica del gestor, funcionan como entradas del subproceso de Implementación y tiene como salida, los registros de las nuevas configuraciones (los ficheros de configuración específicos de cada gestor de base de datos). Finalmente estos ficheros y la configuración básica constituyen las entradas del subproceso de Pruebas, siendo el rendimiento de las aplicaciones producto de la adecuada configuración la salida de este último

La METODOLOGÍA funciona como mecanismo de control aplicado en todo momento durante el PROCESO, definiendo el momento de actuación y lo documentos donde hacer los registros. Las guías de configuración del rendimiento se aplican solamente al subproceso de Implementación, pues contienen los elementos configurables de apoyo para las actividades definidas en este, y las normas del ISO/IEC 9126

aplicadas únicamente al subproceso de Pruebas, que brindan las métricas para evaluar el rendimiento de las aplicaciones.

En un subproceso actúan los grupos de trabajo de la Línea (G-ETL, G-DW, G-BI) de acuerdo a las tareas definidas en la METODOLOGÍA para cada rol, según las actividades especificadas para cada área de proceso, descrito por el documento de roles y responsabilidades de la Línea. La siguiente ilustración muestra como se utilizan los recursos y estructura de la Línea (REL) en los subprocesos.

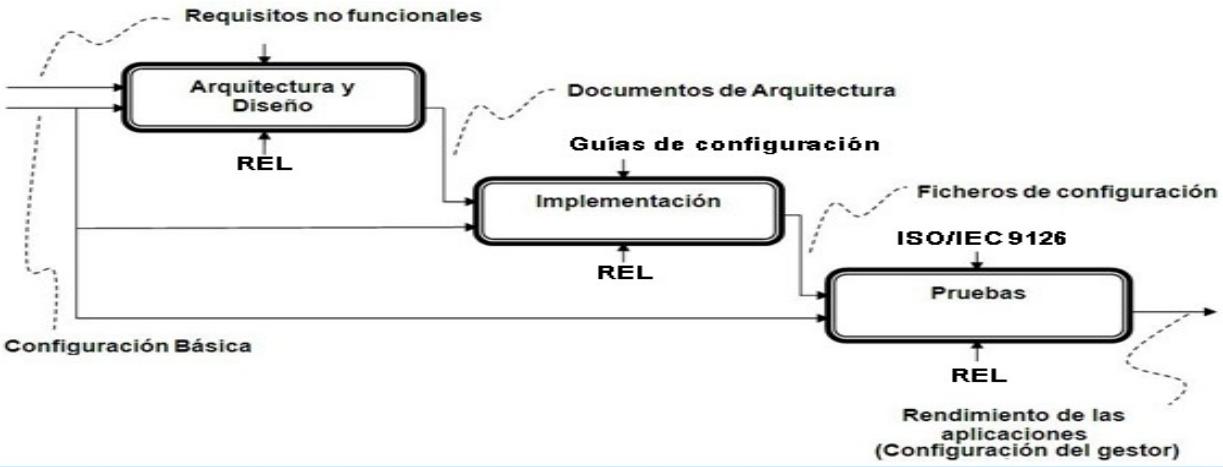


Ilustración 10: Relación entre los subprocesos del PROCESO

Los subprocesos agrupan áreas de procesos, prácticas y actividades diseñadas para cada una de estas fases respectivamente. De igual manera, por razones de capacitación se utilizan los miembros de los grupos de trabajo de la METODOLOGÍA, como trabajadores de los subprocesos de la siguiente manera:

**Tabla 1: Trabajadores de los subprocesos**

| Subprocesos de soporte | Arquitectura y Diseño | Implementación | Prueba |
|---|---|---|---|
| Grupos de trabajo que intervienen | *G-ETL, G-DW, G-BI* | *G-ETL, G-DW, G-BI* | *G-Análisis* |

### *2.3Subprocesos, Áreas de procesos, prácticas y actividades*

#### 2.3.1.Subproceso Arquitectura y Diseño

Durante la definición de la Arquitectura y el diseño, se estructura el sistema en función de estructuras de almacenamiento, reglas de extracción, transformación y carga de los datos y la arquitectura de la

información que guiará el proceso de desarrollo. Aquí se concentra el mayor volumen de actividades en las aplicaciones de BI, el diseño, modelación dimensional y perfilado de los datos.

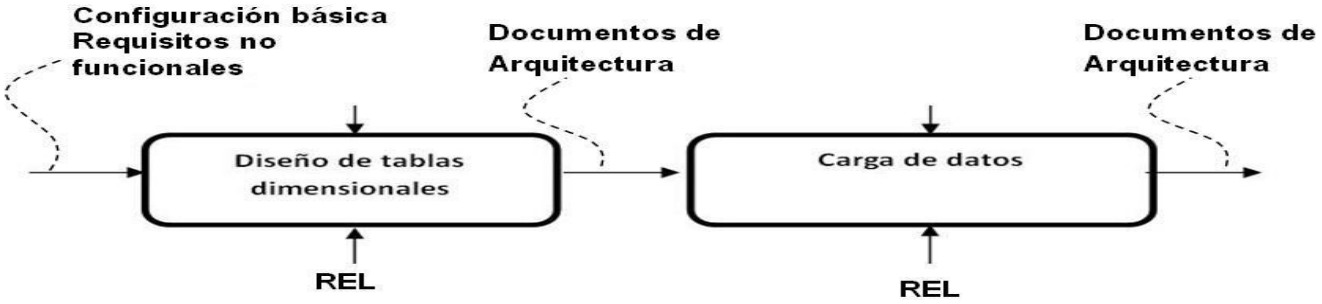


Ilustración 11: Relación entre las áreas de proceso del subproceso Arquitectura y Diseño

Las actividades de áreas de proceso del subproceso de Arquitectura y Diseño, se ejecutan en el orden que se señala y utilizan como mecanismo de control la METODOLOGÍA, aplicable a todo el PROCESO, y como recursos, los recursos y estructura de la Línea (REL), primero las incluidas dentro de la práctica "Carga inicial de los datos" de "Diseño de tablas dimensionales" que tienen como entrada la configuración básica y requisitos no funcionales para convertirlos en especificaciones en los documentos de diseño de la arquitectura física del DW y luego Diseño apropiado de las tablas de "Carga de Datos" el cual utiliza esta salida y realiza las actualizaciones pertinentes en el documento de arquitectura, de acuerdo a las estrategias de carga de datos.

### 2.3.2.Subproceso Implementación

Durante este subproceso se lleva a cabo el diseño físico del repositorio de los datos, se crean las estructuras del almacenamiento y el área temporal de almacenamiento, se ejecutan las reglas de extracción, transformación y carga y se configuran e implementan las herramientas de BI para obtener reportes.

Debido a que las prácticas definidas para el subproceso de implementación requieren de parámetros que difieren de uno a otro gestor (PostgreSQL y Oracle), en esta sección se describen las Áreas de procesos. Los parámetros de configuración y las herramientas contenidas en cada uno de los gestores de base de datos se presentan en el siguiente capítulo, como parte de las Guías "Construyendo almacenes de datos con PostgreSQL" y "Construyendo almacenes de datos con Oracle".

#### *2.3.2.1.Gestor de base de datos*

Las prácticas definidas en esta área, exponen los elementos configurables, cuáles y cuál es el valor recomendable en aplicaciones de DW a tener en cuenta que permitan al gestor comportarse y gestionar las consultas de manera que optimice el rendimiento influyendo en las estadísticas y manejo de las transacciones. Es importante mencionar que no se especifican valores, debido a que estos varían de acuerdo a las características específicas del sistema y estas a su vez tienden a cambiar.

#### *2.3.2.2. Utilización de los recursos*

Las prácticas definidas en esta área, exponen los elementos configurables, cuáles son estos y cuál es el valor recomendable en aplicaciones de DW, que permitan al gestor optimizar la utilización de los recursos, ya sea Disco I/O, Red, Memoria o CPU que necesite. Es importante mencionar que no se especifican valores, debido a que estos varían de acuerdo a las características específicas del sistema y estas a su vez tienden a cambiar.

#### *2.3.2.3. Técnicas de Optimización*

En esta área de proceso, las prácticas agrupadas están orientadas a la implementación de las técnicas de optimización existentes para DW para uno u otro gestor respectivamente.

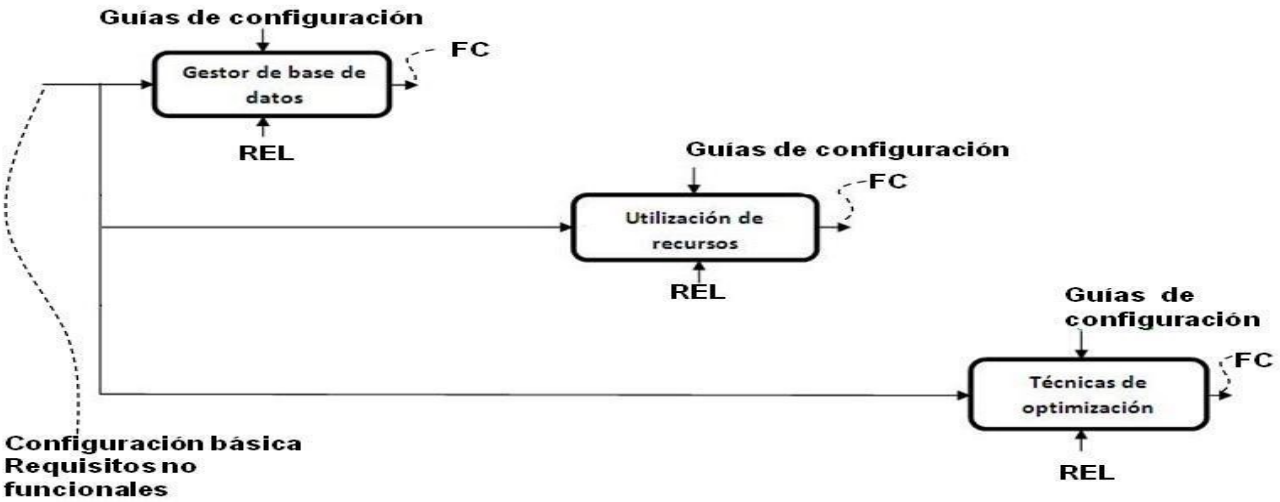


Ilustración 12: Relación de las áreas de proceso del subproceso Implementación

Las actividades dentro de las áreas de proceso del subproceso Implementación se realizan de la manera señalada; es posible llevarlas a cabo de forma independiente, debido a que las mismas relacionan los parámetros de configuración de las guías de configuración del rendimiento (“Construyendo almacenes con PostgreSQL” y “Construyendo almacenes con Oracle”), es decir, es posible realizar independientemente

del orden las prácticas “Tratamiento de consultas” y “Configuración interna” de “Gestor de base de datos”, “Memoria”, “CPU”, “Red” y “Disco” de “Utilización de recursos” o finalmente, “Particionamiento”, “Vistas materializadas”, “Índices” y “Procesamiento paralelo” de “Técnicas de Optimización”. Cada una de estas áreas de proceso utiliza como mecanismo de control además de la METODOLOGÍA, las guías de configuración que aportan los elementos para cada gestor según se necesite, utilizan como entrada la configuración básica para crear las actualizaciones en los ficheros de configuración, producto de las actividades realizadas, mediante el uso de los recursos y estructura de la Línea (REL).

### 2.3.3.Subproceso Pruebas

El flujo de trabajo Pruebas es donde se realizan las pruebas del sistema, desde las unitarias hasta las de aceptación del cliente final. Debido a la naturaleza de los problemas de rendimiento, pues generalmente cuando se realizan pruebas al sistema, un problema grande encontrado es muy probable que esté solapando otro, por ello, y ya que tienen su origen en las capas inferiores relacionadas con el hardware, por esta razón es necesario comenzar la pesquisa de forma ascendente desde esta capa escalando progresivamente hasta encontrar el problema. De esta manera el subproceso itera entre las áreas definiendo las siguientes prácticas:

#### *2.3.3.1. Recolección de Información (Estadísticas)*

Tratándose de un DW, es crucial esclarecer ciertos aspectos como: qué intentan hacer las aplicaciones, cómo utiliza la base de datos (consultando las estadísticas de procesamiento de las transacciones por parte del gestor de base de datos), qué tipo de problemas o riesgos relacionados con el rendimiento se prevén (apoyándose en datos históricos de otras aplicaciones elaboradas), resumiendo, cómo se realiza el trabajo en el sistema e intentar identificar áreas problemáticas en el mismo, analizando los cambios realizados a la configuración básica recogidos en los ficheros de configuración (FC), para obtener como salida el Plan y los Casos de prueba (PCP) y los registros de las estadísticas (*logs*).

#### *2.3.3.2. Comprobación del sistema*

Utilizando los registros de las estadísticas de procesamiento del gestor (*logs*), el Plan y Casos de prueba, se comprueba la configuración del uso de los recursos (la configuración del hardware, del sistema operativo, del gestor y de las aplicaciones que esté utilizando el sistema) y se obtiene como salida la actualización de los registros y ficheros de configuración.

#### *2.3.3.3. Identificación de problemas de rendimiento*

Luego del análisis de las áreas anteriores, entonces será posible la identificación de problemas de los componentes que indiquen si existe alguna dificultad relacionada con el rendimiento, y asociarla a alguno de los indicadores, sea de rendimiento o utilización de recursos de acuerdo a las métricas propuestas. Utilizando Plan y Casos de prueba conjuntamente con los ficheros de configuración, para obtener las posibles soluciones a los problemas de rendimiento descritas en el Plan y Casos de prueba y configuraciones del gestor de base de datos.

#### *2.3.3.4. Solución*

Aplicar estrategias definidas para la configuración en las prácticas con el objetivo de solucionar las dificultades de rendimiento detectadas y obtener finalmente una configuración adecuada a DW del gestor de base de datos.

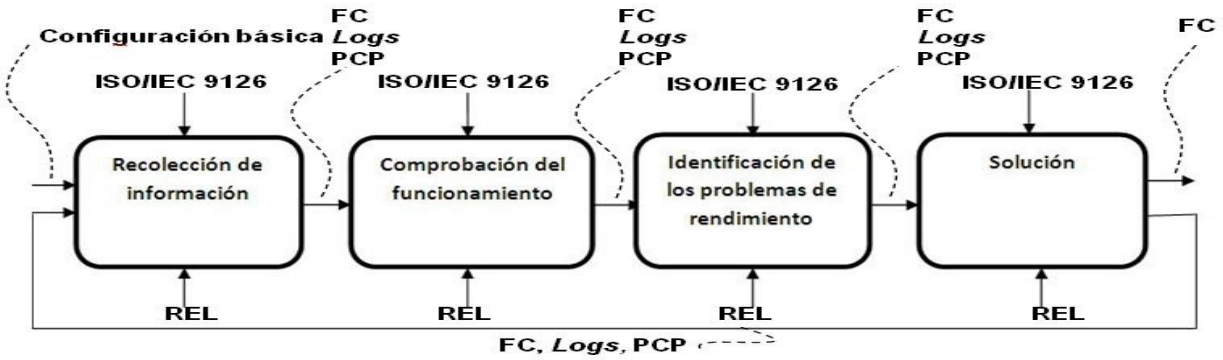


Ilustración 13: Relación entre las áreas de proceso del subproceso Pruebas

Por lo anteriormente explicado, las actividades dentro de las prácticas de las áreas de proceso se ejecutan en el orden: “Estadísticas” de “Recolección de información”, “Funcionamiento de componentes” de “Comprobación del funcionamiento”, “Relación entre problemas, aspectos del desempeño y parámetro de configuración” de “Identificación de los problemas de rendimiento”, “Solución de las dificultades detectadas” de “Solución” y dependiendo de si se alcanza el nivel de rendimiento esperado, entonces se ejecutan o no nuevamente todas las actividades definidas en el subproceso Pruebas en el mismo orden.

**Tabla 2: Relación entre subproceso, área de proceso, práctica y actividades**

| Subproceso | Áreas de proceso | Prácticas | Actividades |
|---|---|---|---|
| **Arquitectura y Diseño** | Diseño de tablas dimensio-nales | Diseño apropiado de las tablas | Utilizar llaves sustitutas para todas las dimensiones, excepto para la temporal |
| | | | Mantener el menor número de enteros como llave |
| | | | Crear una tabla con una columna por cada atributo de cada nivel que se ha modelado |
| | | | Evitar relaciones de dependencia que obliguen a una relación de llave foránea entre las tablas de hechos y las dimensionales |
| | | | Diseñar las tablas con N llaves y M columnas de tipo numérico que contienen medidas de un proceso de negocio |
| | Carga de datos | Carga inicial de los datos | Diseño de estrategias que optimicen el Proceso |
| | | | Diseñar estrategias para eliminar viejos datos eficientemente |
| | | | Renombrar las nuevas tablas que reemplazarán las antiguas |
| | | | Utilizar eficientemente particionamiento e índices |
| **Implemen-tación** | Gestor de base de datos | Tratamiento de consultas | Utilizar los parámetros de configuración establecidos para el tratamiento de las consultas en los epígrafes 3.1.1 y 3.2.1 de las guías de configuración |
| | | Configuración interna | Utilizar los parámetros de configuración establecidos para inicialización del gestor de las guías de configuración en los epígrafes 3.1.1.1 al 3.1.1.4 y 3.2.1.1 |
| | Utilización de recursos | Disco I/O | Utilizar los parámetros de configuración establecidos para manejo de peticiones I/O de las guías de configuración en los epígrafes 3.1.2.2 y 3.2.2.1 |
| | | Memoria | Utilizar los parámetros de configuración establecidos para memoria de las guías de configuración en los epígrafes 3.1.2.1 y 3.2.2.2 |
| | | CPU | Utilizar los parámetros de configuración establecidos para uso de CPU del gestor de las guías de configuración en los epígrafes 3.1.2.3 y 3.2.2.3 |

| | | | |
|---|---|---|---|
| | | Red | Utilizar los parámetros de configuración establecidos para manejo de conexiones del gestor de las guías de configuración en los epígrafes 3.1.2.4 y 3.2.2.4 |
| | Técnicas de optimización | Índices | Seguir las indicaciones de las guías para la implementación de la técnica de índices en el epígrafe 3.1.3.1 |
| | | Particiona-miento | Seguir las indicaciones de las guías para la implementación de la técnica de particionamiento en el epígrafe 3.1.3.2 |
| | | Procesamien-to paralelo | Seguir las indicaciones de las guías para la implementación de la técnica de procesamiento paralelo en el epígrafe 3.1.3.3 |
| | | Vistas materializa-das | Seguir las indicaciones de las guías para la implementación de la técnica de vistas materializadas en el epígrafe 3.1.3.4 |
| **Pruebas** | Recolección de información | Estadísticas | Determinar la función e interacción de los componentes del DW |
| | Comproba-ción del funciona-miento | Funcionamien-to de componentes | Comprobar el correcto funcionamiento de los recursos utilizados y la funcionalidades del sistema |
| | | | Recolección de los datos del diagnóstico por elemento probado |
| | Identificación de los problemas de rendimiento | Relación entre problemas, aspectos del desempeño y parámetro de configuración | Determinar a partir de los problemas detectados, a qué aspecto del desempeño tributan y los parámetros de configuración adecuados para su tratamiento |
| | Solución | Solución de las dificultades detectadas | Revisión de la configuración del gestor según problema y con la ayuda de las guías de configuración |

### *Conclusiones*

Llegados a este punto se puede concluir que el “Proceso de configuración del rendimiento”, se diseña de acuerdo a las características, flujos de trabajo y grupos que intervienen en la METODOLOGÍA de DW de la Línea, definiendo por razones, mayormente de capacitación, tres subprocesos: Arquitectura y Diseño, Implementación y Prueba, cada uno de estos contiene a su vez un conjunto de Áreas de procesos donde se agrupan prácticas a seguir que definen el trabajo con el gestor, dirigidas a aumentar el rendimiento, cuya meta se alcanza.

Dentro de este PROCESO se describen además aquellas Áreas de procesos que resultan comunes para ambos gestores, PostgreSQL y Oracle, los parámetros que complementan las prácticas propuestas pertenecientes al subproceso de Implementación, se describen en el siguiente capítulo, como parte de las Guías de configuración del rendimiento para los gestores relacionales PostgreSQL y Oracle.

## Capítulo 3: Guías para la configuración del rendimiento

### *Introducción*

En este capítulo se describen las guías para la configuración del rendimiento “Construyendo almacenes de datos con PostgreSQL” y “Construyendo almacenes de datos con Oracle” que completan la solución. Las mismas están constituidas por la descripción de los parámetros y las herramientas que completan las actividades de contenidas en las prácticas pertenecientes a las áreas de procesos: “Gestor de base de datos”, “Utilización de recursos” y “Técnicas de optimización”, pertenecientes al subproceso Implementación del proceso de configuración del rendimiento descrito en el anterior capítulo.

### *3.1. Construyendo almacenes de datos con PostgreSQL*

#### 3.1.1. Área de proceso: Gestor de base de datos

Con respecto a PostgreSQL durante la etapa de implementación, donde se configuran las herramientas y cargan los datos es necesario:

El tratamiento para las consultas más lentas realizarlo mediante los parámetros *log_min_duration_statement* en postgresql.conf (fichero donde se guardan las configuraciones), estos valores hacen que la duración de cada declaración completa se registre, si la declaración corrió por lo menos durante el número especificado de milisegundos. Al establecer esta a cero imprime todas las duraciones de las declaraciones, menos uno (el valor predeterminado) deshabilita el registro de la duración de la declaración. Por ejemplo, si lo establece en 250 ms, todas las sentencias SQL que se ejecutan 250 ms o más se registran. La habilitación de este parámetro puede ser útil en la búsqueda de las consultas no optimizadas en sus aplicaciones, por lo que deben ser primeramente altos, ir bajando poco a poco y evitar el exceso de logs, además es posible con la ayuda de la herramienta *pg_fouine*, leer los logs en busca de información de la actividad de las consultas, las más lentas y las más usadas, en dependencia de los parámetros mencionados. En la nueva versión 8.4 podría ejecutarse el *auto_explain*, instalando el módulo contrib.

Hacer dormir al *vacuum* periódicamente no es aconsejable, sin embargo habilitando el *autovacuum* (ON) en postgresql.conf, se evita tener que manipular esta opción periódicamente, en otro caso dejar que el gestor se encargue de esta tarea, evitará que consuma todo el I/O disponible y posibilita que otros procesos puedan continuar trabajando sin interrupciones.

El *bgwriter* o *background writer* es un proceso independiente que se encarga de escribir periódicamente en disco, entre checkpoints, posibilitando un mejor manejo de las peticiones I/O favoreciendo el rendimiento, tiene la ventaja de posibilitar que los procesos puedan traer páginas a *shared_buffers* sin la necesidad de escribir en disco, y como desventaja que cada escritura signifique I/O desperdiciado, por lo que hay que tener mucho cuidado a la hora de manejar este parámetro.

Deshabilitar el *autocommit* cuando se realizan múltiples INSERT, lo que significa comenzar la sentencia con BEGIN y realizar un único *commit* al finalizar, de esta manera no se interrumpirá el proceso con datos a medio cargar, muy recomendable para instruir al gestor en hacer uso eficiente de la transacción.

*BEGIN*

*booktom=# CBEmE TABLE teat (id W,n6 m terd);*

*CRFATE*

*booktm=# C a m T* ***mm;***

*COMMIT*

Utilizar el comando COPY para todas las filas en una sola orden, en lugar de realizar una serie de INSERT, este comando está diseñado para cargar grandes volúmenes de datos y aunque es menos flexible que INSERT incurre indudablemente en menos *overhead* y no es necesario deshabilitar *autocommit* para realizar la operación. El comando COPY en la última versión 8.4 de PostgreSQL fue mejorado considerablemente con las aportaciones de la empresa "Aster Data".

***COPY*** *[* ***BINARY*** *I table [ WITH* ***OIDS*** *I*

*FROM {* ***'filename' I stdin 1***

*[[* ***USDE ]*** ***DELIMITERS 'delimiter'*** *I*

*[* ***WITH*** *NULL* ***AS 'nullstring'*** *I*

***COPY*** *[* ***BINARY*** *I table [ WITH O I X* ***I***

*TO {* ***Ifilename' I stdout 1***

*[[* ***USDG*** *I* ***DELIMITERS 'delimiter' I***

*[* ***WITH WLT., AS 'null-string'*** *I*

EXPLAIN permite examinar el plan de una consulta, permite ver el plan de ejecución.

A continuación se explican algunos de los parámetros que también contiene el postgresql.conf, que como aquellos que se encargan de especificar la utilización de los recursos de PostgreSQL, estos establecen un comportamiento interno, es necesario tenerlos en cuenta al perfilar el rendimiento.

#### *3.1.1.1. Checkpoints*

***Checkpoint_segment*****:** representan un punto en el tiempo donde se garantiza que todos los datos "sucios" (actualizados o añadidos) se hayan escrito en disco. especifican la máxima distancia entre los *checkpoints* automáticos del WAL en el segmento de archivos de logs, es decir, define el máximo número de los segmentos de las transacciones, es recomendable para las aplicaciones de DW tenerlo alto para mejor manejo y recuperación. Por tanto el espacio en disco requerido se calcula (*checkpoint_segment* * 2 + 1) * 16 Mb, establecer este parámetro en 128 siempre y cuando sea posible, es aconsejable para soportar los pesados procesos ETL.

***Checkpoint_timeout*****:** tiempo máximo entre los checkpoints automáticos del WAL.

***Checkpoint_warning*:** envía un mensaje a los logs del servidor, si los *checkpoint* causados por las entradas a los archivos de segmento suceden con mayor frecuencia que el número establecido en segundos.

***Commit_delay*:** tiempo en milisegundos que demora la escritura y envío de un pedido en el buffer de escritura de WAL al disco.

***Commit_siblings*:** número mínimo de transacciones concurrentes abiertas que se requieren antes de ejecutar el c*ommit_delay.*

#### *3.1.1.2. Optimización de consultas*

Estos parámetros son métodos del planeador en el plan de consultas, el valor por defecto de los mismos es ON. Es necesario mencionar sin embargo, que es importantísimo tener mucho cuidado al trabajar con ellos, por ejemplo, con el comando EXPLAIN, ya que es posible que una consulta que esté utilizando una búsqueda secuencial funcione mejor si se establece la búsqueda por índices. En esta última versión de PostgreSQL es posible cambiar estos parámetros en tiempo de ejecución con la cláusula SET: *Enable_hashagg, Enable_hashjoin, Enable_indexscan, Enable_mergejoin, Enable_nestloop, Enable_seqscan, Enable_sor y Enable_tidsca.*

Cada uno de los siguientes parámetros es miembro de las constantes de costo del planeador, que establecen la estimación del costo de optimización de las consultas para el procesamiento: *Random_page_cost, Cpu_tuple-cost, Cpu_index_tuple_cost y Cpu_operator_cost.*

Los siguientes parámetros pertenecen al optimizador de consultas por estimación genética.

*Geqo*: habilita o deshabilita la optimización de las consultas, realizada a través de un algoritmo que intenta hacer una planificación de una consulta sin realizar trabajo excesivo, recomendable cuando se usan más de 12 JOIN.

*Geqo_threshold*: es el número mínimo de sentencias FROM que debe tener una consulta para que requiera la utilización de algoritmos de optimización, las sentencias JOIN también están incluidas y es necesario aclarar que el valor que se usa usualmente es 11. Este parámetro como el optimizador tratará de mezclar las subconsultas FROM.

Los últimos 5 parámetros se utilizan para el rendimiento del algoritmo genético de optimización de las consultas. El tamaño del pool, es el número de individuos en una población, el *effort*, es utilizado para calcular el número por defecto para cada generación, y estas últimas establecen el número de iteraciones

del algoritmo genético: *Geqo_selection_bias, Geqo_pool_size, Geqo_effort, Geqo_generations y .Geqo_random_seed*

#### *3.1.1.3. Estadísticas de consultas e índices*

Estas banderas establecen qué información se envía al proceso de recolección de estadísticas y también se encuentran en postgresql.conf: comandos actuales y estadísticas de bloqueos de actividades. El valor por defecto de todas es OFF debido a que requieren un costo adicional por consulta: *Stats_start_collector, Stats_reset_on_server-start, Stats_command_string, Stats_row_level, Stats_block_level y Default_statistics_target:* establece el objeto por defecto de las estadísticas para las columnas de las tablas que no han tenido una vía establecida a una columna específica. Valores elevados de este parámetro puede ralentizar el proceso ANALYZE.

#### *3.1.1.4. Otros modificadores de consultas*

*Explain_pretty_print*: determina que formato utilizará EXPLAIN para mostrar los detalles de las consultas.

*From_collapse_limit*: el planeador mezclará las subconsultas con las consultas si el resultado de la lista de FROM no tiene más elementos que este parámetro.

*Join_collapse_limit*: establece el número con el que se compara la cantidad de elementos resultantes de la lista de FROM para decidir si se incluyen los INNER JOIN, usualmente se establece en 1, para permitir que se ejecute el JOIN.

*Max_expr_depth*: la máxima profundidad de las expresiones anidadas en el parser.

### 3.1.2. Área de proceso: Utilización de recursos

Mayormente, los problemas de rendimiento en DW son consecuencias de una pobre utilización de los recursos, los principales están dados por los procesos de I/O en disco, poca RAM disponible, consultas mal programadas, búsquedas secuenciales y cargas masivas de datos, resaltando la cantidad de estos necesaria para desarrollar y administrar el proceso. Es necesario recordar que la eficiencia de un producto depende linealmente de la de cada uno de sus componentes, en DW específicamente: Memoria, CPU, Red y Disco I/O.

A los que se hace referencia a continuación, con el objetivo de brindar al lector una panorámica de cómo tratar este delicado aspecto que en ocasiones representa la mayor cantidad y complejidad y resulta tan difícil detectar o diagnosticar los problemas que produce un mala gestión de los mismos.

#### *3.1.2.1. Memoria*

La memoria RAM como un recurso limitado divido en segmentos. Los segmentos, de un tamaño fijo forman páginas de memoria y en esta se guarda todo lo que el CPU necesita para hacer su trabajo, esto incluye programas, datos requeridos por los programas, el kernel y por supuesto, las zonas de trabajo de PostgreSQL. El reto está en asignarle las dimensiones adecuadas para que sea posible aprovechar la mayor cantidad de memoria posible (de esta manera resulta más ventajoso para el rendimiento pues es el recurso de almacenamiento más rápido con que se cuenta) y realizar este proceso sin que ocasione afectaciones a las demás operaciones que requieren su uso. PostgreSQL brinda una serie de parámetros para gestionar su uso de la memoria:

***Shared_buffers*:** es el que más afecta el rendimiento de PostgreSQL, indica el número de bloques de memoria que reservará como zona de trabajo, se configura en el fichero postgresql.conf por defecto con una valor 1000, claramente insuficiente para las demandas de un DW, escoger esta cantidad depende tanto de la cantidad de memoria disponible como del tamaño de la base de datos y la cantidad de consultas concurrentes que suele manejar. Miembros de la Comunidad de PostgreSQL han encontrado valores aceptables en el rango entre 1000 y 6000, más cada sistema tiene sus propias exigencias. Usualmente para DW se recomienda darle a este parámetro el 25% de la memoria RAM.

***Work_mem*:** es utilizado para especificar la cantidad de memoria que PostgreSQL utilizará en cada una de sus operaciones, la configuración por defecto establece 1024, es decir 1MB, pero se debe tener en cuenta que en las consultas complejas se ejecutan distintas operaciones en paralelo, se recomienda por tanto establecer el parámetro proporcional al número de conexiones y aumentarlo en consultas complejas, como por ejemplo volcado de datos, o dejarlo en 128 Mb.

***Vacuum_mem*:** Este parámetro especifica la cantidad de memoria que utiliza el *vacuum* para mantener el control sobre las tuplas a mostrar. El valor por defecto se establece en 1KB, pero para consultas complejas en grandes cantidades de datos es recomendable un valor mayor para aumentar el rendimiento.

***Effective_cache_size*:** Este parámetro permite a PostgreSQL hacer el mejor uso posible de la memoria de la que dispone, utilizando como dato el tamaño de la cache del sistema operativo es capaz de realizar diferentes planes de ejecución que optimicen el procesamiento. Suponiendo que la cantidad de shared_buffers es mayor que la de este parámetro y este a su vez, mayor que la cantidad requerida para la operación, este puede trazar un plan que optimice el uso de los índices y realizar la operación de manera eficiente, de otra manera, cuando la cantidad requerida es mayor puede entonces trazar una

estrategia que se centre en lograr un escaneo secuencial eficiente para los procesos I/O, lo recomendable es dejarlo en 2/3 de la RAM, en caso de que el servidor se dedique completamente a PostgreSQL.

#### *3.1.2.2. Disco*

**WAL:** *Write Ahead Log*, establece la escritura en el disco. En el fichero de postgresql.conf, existen además una serie de parámetros que establecen distintos comportamientos:

***Fsync*:** asegura que la base de datos se recupere a un estado consistente luego de algún fallo del sistema operativo o el hardware, toma valores de True o False. Es importante recalcar que nunca se debe deshabilitar este parámetro ya que garantiza la consistencia de los datos en caso de fallos.

***WAL_sync_method*:** define el tipo de método mediante el cual se actualizará la información almacenada en el disco, que pueden ser *fsync, fdatasync, open_sync u open_datasync* cada una para diferentes plataformas.

***WAL_buffers*:** establece el número de páginas del disco usadas como *buffers* en memoria compartida para logs de WAL. Elevar este valor puede aumentar la velocidad de escrituras para las transacciones, por defecto se tiene 1Mb, lo recomendable es subirlo a 8Mb.

Existen además una serie de prácticas útiles para WAL: Evitar la escritura de páginas completas a WAL, antes de una modificación, Chequear después de cada escritura si existen bloqueos en sistema de archivos durante la recuperación, Escribir páginas completas solo durante la escritura en el sistema de archivos y no cuando la página es modificada en el *buffer* del cache y Reducir el tráfico de WAL de manera tal que se escriban ciertos valores y no filas completas.

**RAID:** la configuración óptima de RAID es 0 cuando las actualizaciones se realizan a través de múltiples discos, lo que multiplica su velocidad, 1 cuando las actualizaciones de replican en múltiples discos, que es la opción menos costosa en caso de separar el directorio xlog que es donde se almacenan los archivos del WAL, pero recientemente en caso del directorio $PG_DATA se recomienda utilizar el valor 10 cuando se tienen varios discos, aunque represente una opción más costosa provee extra confiabilidad y rendimiento. Otra de las opciones que proporciona gran rendimiento, es una mejora de hardware, el controlador de RAID *battery-backed cache*, las compañías que lo comercializan son "LSI", "*Logic", "Compaq*" e "Intel", la forma usual del producto es un controlador PSI SCSI o SATA que soporta 16 o más discos y un cache de 128MB o más con una batería.

***Max_files_per_process*:** establece el número de archivos abiertos simultáneamente en cada subproceso del servidor, para evitar bloqueos.

***Preload_libraries*:** establece las librerías que se utilizarán una vez iniciado el servidor.

***Max_fsm_pages*:** establece el número máximo de páginas del disco para las cuales se controla espacio libre en el mapa compartido de espacios libres. Sin embargo es necesario especificar que se encuentra aún solo en versiones inferiores a la 8.4, ya que en esta última existe una nueva implementación de FSM, para usar el mapa compartido directamente en el disco y no en memoria como se hacía en versiones anteriores

***Max_fsm_relations*:** establece el número máximo de relaciones (tablas) para las cuales se controla espacio libre en el mapa compartido de espacios libres, sólo en versiones anteriores por las razones anteriormente explicadas.

#### *3.1.2.3.CPU*

La capacidad del CPU es consumida por la manipulación de consultas en memoria y tiende a ser costoso cuando se trabaja con consultas que procesan gran cantidad de datos, por consiguiente gran cantidad de memoria, como resultado es usual que se le preste más atención al uso de la memoria que al uso del CPU, pero en general cuando las consultas son complejas mientras más CPU mejor. Si el sistema maneja consultas sencillas, lo cual no es el caso más común en el trabajo con DW, entonces puede prestársele más atención a la RAM o subsistema de discos para lograr mejoras de rendimiento.

#### *3.1.2.4. Red*

Cada conexión necesaria tiene un costo en términos de latencia, de esta manera puede resultar más costoso establecer un gran número de conexiones para pequeñas transacciones, que realizar pocas consultas que transporten gran cantidad de información. PostgreSQL establece algunos parámetros para la configuración de este recurso:

***Tcpip_socket:*** guarda un valor True si el servidor acepta conexiones de tipo TCP/IP, y False en caso contrario.

***Max_connections*:** establece el número de conexiones concurrentes que será la base de datos capaz de manejar, no es aconsejable permitir un número muy alto (más de algunos cientos) de conexiones concurrentes, en su lugar utilizar un pooling de conexiones (*PgPool-II, PgBouncer*), los cuales hacen un uso eficiente de las conexiones mediante la reutilización siempre que sea posible, es decir, en caso de que sea el mismo usuario a la misma base de datos y desde el mismo host cliente. Es necesario tener en cuenta que de la configuración de este parámetro dependen otros como *Work_mem* tratándose de configuraciones con vistas al rendimiento

***Port*:** el puerto que el servidor atiende.

***Unix_socket-directory*:** especifica el directorio del socket del dominio Unix desde donde el servidor recibirá las solicitudes de las aplicaciones cliente.

***Unix_socket_group*:** especifica el grupo administrador del socket de Unix.

***Unix_socket_permissions*:** especifica los permisos

### 3.1.3. Área de proceso: Técnicas de optimización de DW

Mediante la utilización de estas técnicas de optimización es posible mejorar aún más la eficiencia de los DW, constituyen estrategias precisamente diseñadas con ese objetivo, a continuación se explica cómo utilizarlas con el gestor objeto-relacional PostgreSQL.

#### *3.1.3.1.Índices*

Los índices se utilizan para mejorar el rendimiento, es decir, optimizar las consultas mejorando el tiempo de respuesta de las base de datos, PostgreSQL implementa los índices igualmente con el objetivo de evitar complejas búsquedas y hacer las consultas más efectivas. Una vez que el índice ha sido creado, ninguna intervención es necesaria. Para realizar las búsquedas el sistema se encargará de decidir cuándo se utiliza y lo actualizará automáticamente cuando se actualice la tabla, más será necesario ejecutar ANALYZE para el planificador de consultas del sistema.

PostgreSQL permite que los lectores (SELECT) se ejecuten en paralelo con la creación del índice, debido a que en una tabla muy grande tiende a ser un proceso largo, mientras que los procesos que escriben (UPDATE, DELETE e INSERT) son bloqueados hasta que la creación del índice concluya.

En este tipo de aplicaciones (DW) la utilización de índices como estrategia para la optimización es muy útil, sin embargo se debe tener cuidado a la hora de crearlos, ya que si existen muchos es posible que el planificador entonces en vez de mejorar el plan de consultas lo ralentice.

Existen varios tipos de índices (B-tree, Hash, GiST y GIN) y cada uno tiene diferentes algoritmos, aquellos que se ajusten más a cada uno de los tipos, aunque este gestor de base de datos crea por defecto cuando no se especifica B-tree, que se ajusta a las situaciones más comunes.

B-tree: maneja consultas de igualdad y rango dentro de datos parcialmente ordenados aunque en general el planificador utiliza este tipo de índices cuando alguno de los operadores <; <=; =; >=; > se involucra. Es también utilizado cuando se usan sentencias equivalentes a estos.

Hash: se considerará su uso cuando involucra únicamente comparaciones de igualdad con el operador =.

GiST: no son un tipo de índice específico, sino que representan diferentes estructuras mediante las cuales

es posible implementar diferentes estrategias de indexación, las que además determinan qué operadores se emplearán, usualmente son utilizados con clases de operadores (<<; &>; &<; >>: <<|; &<|; |&>; |>>; @>; <@; &&) para tipos de datos en dos dimensiones.

GIN: es un tipo de índice invertido que tiene la capacidad de manejar valores con más de una llave y al igual que los índices GiST soportan diferentes tipos de estrategias de indexado definidas por el usuario. El tipo de operadores con que trabaja dependen linealmente de esta estrategia. La distribución estándar de PostgreSQL incluye clases de operadores (<@; @>; =; &&) para arreglos de una dimensión.

A continuación se explican las distintas implementaciones posibles en PostgreSQL utilizando los tipos de índices anteriormente explicados:

- **Índices Multicolumnas**

Los índices pueden ser definidos en más de una columna y en las últimas versiones de PostgreSQL los tipos: B-tree, GiST y GIN, soportan este tipo de índices y pueden especificar hasta 32 columnas, valor que puede alterarse en el fichero postgresql.conf. Un índice B-tree multicolumna puede utilizarse para condiciones de consultas que involucren otras especificaciones de los índices, pero se obtiene mejor rendimiento cuando estos están ubicados en la parte izquierda de la tabla, de esta manera se reduce la porción de datos adicional que es necesario escanear.

Por otro lado, los tipos de índices GiST multicolumna, se usan en las mismas circunstancias con la salvedad de que la efectividad de la consulta no se ve afectada por la localización del índice. Este tipo de índices es utilizado en raras ocasiones, la mayor parte de las veces es posible optimizar utilizando un solo índice en una columna, ahorra tiempo y espacio.

- **Los índices y el GROUP BY**

Mediante esta solución es posible encontrar las filas que debe retornar una consulta y devolverlas en orden, según los requerimientos de la consulta sin necesidad de un paso intermedio de ordenamiento, de los diferentes tipos de índices soportados por PostgreSQL únicamente los B-tree produce una salida ordenada, por defecto este tipo de índices retorna de manera ascendente con NULL al final, pero es posible ajustarles el orden incluyendo ASC, DESC, NULL FIRST y/o NULL LAST cuando se crea el índice.

- **Combinando múltiples índices**

El escaneo único de un índice puede solamente utilizar cláusulas que usen la columna del índice con operadores de su clase, que estén unidos por AND. PostgreSQL sin embargo tiene la capacidad de combinar múltiples índices y además los usos de los mismos, con vista a manejar requerimientos que no

pueden ser completados con la utilización de un índice único. Un ejemplo de esto sería una consulta WHERE x = 23 OR x = 44 OR x=33; combinando 4 búsquedas separadas con un índice en x y OR para producir un único resultado.

- **Índices únicos**

Pueden ser utilizados para reforzar la singularidad del valor de una columna o la combinación de valores de múltiples columnas. Hasta el momento solo los índices B-tree pueden ser declarados únicos. Cuando un índice es declarado único no permite índices de distintas columnas en diferentes tablas con el mismo valor, este se crea en PostgreSQL automáticamente cuando se declara una llave primaria o restricción única constituyendo el mecanismo que garantiza la singularidad de estos elementos.

- **Índices en Expresiones**

Es necesario que una columna índice no sea simplemente una columna en la tabla, sino una expresión escalar o de función computada por una o más columnas de la tabla. Esta propiedad es muy útil para obtener más rápido acceso a las tablas basado en los resultados, cuando la velocidad de respuesta es más importante que la velocidad de inserción o actualización de los datos.

- **Índices parciales**

Son definitivamente los más usados en DW, se crean en un segmento o fragmento de la tabla que es definido por una expresión condicional. Este tipo de índice solo contiene entradas para aquellas filas que satisfacen el predicado. Son una propiedad especializada pero muy útil, por ejemplo, para evitar valores comunes de indexado, otro caso sería para agilizar las búsquedas evitando aquellas que incluyen columnas que no contienen valor en absoluto para la consulta, esto reduce el tamaño del índice y acelera la transacción.

#### *3.1.3.2. Particionamiento*

En PostgreSQL no es posible particionar por el momento, se espera conseguir el mismo efecto utilizando tablas heredadas, otra manera es realizar el proceso mediante RAID de manera que el sistema operativo guarde los datos de las tablas a través de varios discos. Mediante la herencia en PostgreSQL una tabla puede heredar de otras, en tal caso la unión de las tablas es definida por las tablas padre y cualquier columna declarada en la definición de la tabla hija, es añadida a esta. Si alguna tiene el mismo nombre en más de una tabla padre, o en tablas padre e hija, estas se mezclan de manera que aparezca solo en la tabla hija.

La herencia es útil para definir tablas que conceptualmente mantienen elementos en común, pero que también requieren de datos que las hacen diferentes. En la programación orientada a objetos es un mecanismo que permite derivar una clase de otra de manera que extienda su funcionalidad, ahorra tiempo de codificación, maximiza el reutilización de código y es una característica propia de las bases de datos orientadas a objetos, que permite grandes posibilidades de diseño con vistas a aumentar el rendimiento bajando el costo de las transacciones, lo cual la hace una de las características más aprovechables para "Construyendo almacenes de datos con PostgreSQL".

```
CREATE TABLE capitals (
    state          char(2)
) INHERITS (cities);
```

#### *3.1.3.3.Procesamiento paralelo*

En PostgreSQL, el procesamiento paralelo debe simularse a través de consultas paralelas, ya que PostgreSQL utiliza un modelo multiproceso y no multi-hilo, es decir, cada conexión tiene su propio proceso desde el punto de vista del servidor, obtenido desde el postmaster mediante un *fork.* Desde otro punto de vista, si uno de los hilos abre una conexión a PostgreSQL, entonces esa conexión debe usarse por solamente un hilo en un momento dado, si se envía una consulta en un hilo y después otra en otro, entonces el protocolo se desincroniza y la consulta queda en un estado bloqueado. Para hacer conexiones en paralelo es necesario abrir otra conexión, es decir, otro proceso en el servidor (lanzar dos consultas independientes, si se envía una sola sentencia PostgreSQL hará una sola).

Otra de las técnicas que se desarrollan por el momento para proporcionar rendimiento a través de la posibilidad de procesamiento en paralelo, es la utilización de un clúster, que provea de mayor capacidad de procesamiento y disponibilidad de los datos, excepcional para trabajar con los grandes volúmenes de información propios de los DW. Dentro de este estudio destaca: "PgCluster", Sistema de replicación asíncrona multimaster con el objetivo de lograr alta disponibilidad en los datos y buenos límites de rendimiento para transacciones mediante el uso de un clúster, involucrando tres tipos de servidores: de base de datos, balance de carga y réplica. Con esta opción se disminuyen mucho el costo de las transacciones, aunque la herramienta aún no implementa el almacenamiento compartido, se espera en próximas versiones de "PgCluster".

Bucardo, con el mismo uso pero de tipo sincrónica. "PgCluster" es uno de los sistemas insignias de la replicación multimaster para PostgreSQL, desarrollado por "Atsushi Mitani". El proyecto "Chronos" utiliza

como base “PgCluster” para el desarrollo y la empresa “Cybertech” también usa este software para sus negocios.

Es necesario aclarar que el procesamiento paralelo posible por la característica de PostgreSQL de ser multi proceso en vez de multi hilo, es una de las mayores fortalezas que presenta. Quien haya trabajado en un entorno multi-usuario de base de datos puede referirse a la conocida expresión de "se bloquea", como respuesta a la espera de si el sistema de base de datos está utilizando la tabla de nivel: el nivel de página, columna nivel o el bloqueo de fila. El mismo molesto problema persiste: Lectores (SELECT) deben de esperar a los escritores (UPDATE) hasta el final, y los escritores (UPDATE) esperar a los lectores (SELECT) hasta el final. PostgreSQL evita este problema mediante el uso de MVCC.

#### *3.1.3.4. Vistas Materializadas*

Aún existen personas que oponen a incluir en PostgreSQL la implementación de vistas materializadas, debido a que a pesar de que la técnica indudablemente aumenta el rendimiento de la base de datos, usualmente es utilizada para minimizar las consecuencias de un mal diseño del DW, antes de aplicar cualquiera de las técnicas de optimización propias del gestor de base de datos, cualquiera que sea el que se vaya a utilizar, es necesario que exista un buen diseño, de otra manera, lo único que se estaría haciendo sería poner parches sobre un sistema que con mala base y mal funcionamiento en sí mismo.

En qué consisten las vistas materializadas, es un “caching” de aquellos elementos mayormente requeridos por el usuario del sistema, una tabla en caché, fuera de la base de datos alimentada con los datos desde dentro que se comporta como una vista. Existen igualmente distintos tipos de implementaciones utilizadas para distintos tipos de necesidades:

***Snapshot***: es una de las de implementación más sencilla, y se actualizan únicamente de forma manual, esto puede resultar irrelevante si la actualización es grande, pero bastante trabajoso sin son sólo unos campos a actualizar, y debe realizarse regularmente ya que quedaría out-of-sync, es decir, que no tenga relación con la base de datos en cuanto los datos comenzaran a cambiar.

***Eager***: estas son siempre actualizadas inclusive dentro de una transacción, cuestión que implica un alto costo en cambios, que puede resultar beneficioso si los datos si los datos que la alimentan no sufren modificaciones continuamente, pero completamente desventajoso de otra manera. Existe igualmente la posibilidad de que quede out-of-sync si existen dependencias con funciones mutantes, una función para actualización especializada puede resolver el problema, pero encontrar el algoritmo e igualmente muy complejo.

***Lazy***: estas ofrecen un balance entre las snapshots y las eager, realizando todas las actualizaciones de una sola vez, esto tiene la desventaja de que entre un cambio y otro puede que existan inconsistencia entre los datos de la base de datos y los almacenados en la vista, y al igual que las eager, puede que quede out-of-sync en caso de dependencias a funciones mutables. La implementación adecuada de este tipo de vista suele ser muy compleja.

***Very lazy***: es como la snapshots pero con actualizaciones menos pesadas, y al igual que estas quedan out-of-sync en cuanto es actualizada la base de datos. En pocas palabras, tiene la ventaja de la facilidad de implementación de las snapshots pero es más rápida y utiliza menos recursos.

Independientemente del hecho de no ser una implementación propia, se simula ya sea mediante tablas temporales, no utilizada para DW debido a que las mismas son solo visibles en la sesión en que se crean y se pierden cuando se reinicia la conexión y además mediante tablas almacenadas directamente en caché y gestionadas por medio de triggers y funciones de PostgreSQL como muestra el ejemplo:

```
create table inspecciones_maximas(
idinspeccion numeric(18) not null,
idvehiculo  numeric(18) not null,
CONSTRAINT pk_inspecciones_maximas PRIMARY KEY (idvehiculo)
);
create unique index ix_inspecciones_maximas on
inspecciones_maximas(idinspeccion)

CREATE TRIGGER tr_inspecciones_maximas BEFORE INSERT OR UPDATE OR DELETE
   ON inspecciones FOR EACH ROW
   EXECUTE PROCEDURE max_inpeccion();
-- manejar la anulación correctamente.
CREATE OR REPLACE FUNCTION max_inpeccion() RETURNS TRIGGER AS
$tr_inspecciones_maximas$
  DECLARE
    maxima_inspeccion numeric(18);
  BEGIN
   -- ESTO VA SIEMPRE
```

```
   DELETE FROM INSPECCIONES_MAXIMAS where idvehiculo = NEW.idvehiculo;
   SELECT MAX(idinspeccion) INTO maxima_inspeccion FROM INSPECCIONES  where
idvehiculo = NEW.idvehiculo and anulada is null;
   IF (TG_OP = 'INSERT' OR TG_OP = 'UPDATE' ) THEN
      IF maxima_inspeccion is not null THEN
         INSERT INTO INSPECCIONES_MAXIMAS VALUES (maxima_inspeccion,
NEW.idvehiculo);
      END IF;
      RETURN NEW;
   END IF;
 END;
```

### *3.2. Construyendo almacenes de datos con Oracle*

De acuerdo con lo antes expuesto es recomendable crear un modelo estrella independiente de la base de datos transaccional y crear el proceso ETL para obtener y transformar los datos de manera adecuada.

#### 3.2.1. Área de proceso: Gestor de base de datos

Con respecto a Oracle, durante la etapa de implementación donde se configuran las herramientas y se cargan los datos, es necesario conocer que Oracle cuenta con un amplio grupo de capacidades para la extracción, carga y transformación de los datos. Estas características son aprovechadas por "*Oracle Warehouse Builder*" las siguientes son:

- *Database Gateways* (base de datos de puertas de enlace) para acceder a sistemas que no sean de Oracle.
- Servicio de Carga para realizar cargas rápidas de datos de archivos planos.
- Extensiones SQL para transformaciones de datos: sentencia MERGE (hace posible ejecutar operaciones de actualización o inserción definiendo una sola sentencia SQL, en la cual se especifican determinadas condicionales).
- Funciones de Tabla: transformaciones eficientes y paralelas definidas por el usuario.
- Cambio de la captura de datos por la captura de baja latencia basada en registros desde bases de datos Oracle.

Oracle cuenta con muchas ventajas y una de ellas son las optimizaciones de desempeño para cada tipo de entorno de DW. Esta herramienta cubre en su totalidad las demandas del desempeño del DW por brindar un amplio grupo de técnicas de optimización.

También hay que tener en cuenta la configuración de ciertos parámetros de inicialización, los cuales se encuentran ubicados en el fichero de parámetros denominado **init.ora.**

#### *3.2.1.1. Parámetros de inicialización de la Base de Datos*

Una instancia de Base de Datos en Oracle 10g es configurada usando parámetros de inicialización, especificados en el archivo init.ora. Estos parámetros influyen en el funcionamiento de la instancia de base de datos, incluyendo su rendimiento. A continuación se muestran los parámetros de inicialización más relevantes y su descripción:

***DB_BLOCK_SIZE*:** Especifica el tamaño de los bloques de la base de datos almacenados en los *data files* y cacheados en el Área Global del Sistema (SGA) [22]. El rango de valores depende del Sistema Operativo, pero es normalmente 8192 bytes para OLTP y mayor para sistemas de bases de datos tipo DW.

***SGA_TARGET*:** Especifica el tamaño total de todos los componentes del SGA. Si el parámetro está especificado, entonces a todos los componentes del SGA se les asigna un tamaño automáticamente.

***PROCESSES*:** Especifica el máximo número de procesos que pueden ser inicializados por la instancia. Este es el parámetro más importante a inicializar, porque el valor de muchos parámetros es deducido de él mismo.

***UNDO_MANAGEMENT*:** Especifica cuál modo de administración de espacio de UNDO el sistema debe usar. El modo AUTO es el más recomendado.

***UNDO_TABLESPACE*:** Especifica el *tablespace* de *UNDO* que el sistema debe usar al iniciarse la instancia de base de datos.

***Db_block_buffer*:** Especifica el número de bloques de la base de datos en el SGA.

También cuando se definen los bloques, que es la unidad de acceso a disco de una base de datos Oracle, o sea la unidad mínima de transferencia de información, existen parámetros de configuración de los cuales se debe estar pendientes; como son:

***PCTFREE**:* Es el mínimo porcentaje de un bloque que se reserva para actualizaciones de filas. Por defecto 10% del tamaño útil.

[22] Del inglés *System Global Area*, a partir de este momento SGA

***PCTUSED***: Es el mínimo porcentaje de un bloque que debiera estar ocupado para no admitir más inserciones. Por defecto 40%.

Una de las mejores opciones en cuanto a la optimización del rendimiento, es la configuración y optimización del DW, la cual está basada en un diseño y configuración cuidadosa de la base de datos, para evitar que se agoten los recursos del sistema y causar pérdidas de escalabilidad. Para esto se debe de realizar un estudio de los elementos que afectan al rendimiento de la base de datos que da soporte al sistema del propio DW, a continuación se expondrán los pasos a tener en cuenta para la recomendación de un mejor rendimiento: Revisión de la configuración del subsistema de almacenamiento, Revisión del diseño físico de la base de datos, Revisión de los parámetros de la base de datos asociados al rendimiento del sistema DW y Revisión del modelo de datos asociado.

Hay que destacar que Oracle cuenta con determinadas opciones que al ser ejecutadas hacen un estudio efectivo del sistema y orienta para que se hagan los cambios pertinentes, ejemplo de esto se tiene al ADDM (*Automatic Database Diagnostics Monitor*), este es un consejero, que automáticamente realiza análisis del sistema, identifica los posibles problemas y sus causas potenciales, y por último plantea recomendaciones para solucionarlos.

### 3.2.2. Área de proceso: Utilización de los recursos

El rendimiento en sistema está más ligado al diseño y construcción del mismo sistema como tal, o sea, los problemas de rendimiento son usualmente resultado de la contención o agotamiento de algún recurso del sistema. Cuando un recurso del sistema se agota, el sistema es incapaz de escalar a mayores niveles de rendimiento. Entonces eliminando conflictos de recursos, los sistemas pueden ser escalados hasta lograr el máximo rendimiento de los recursos disponibles y los niveles de respuesta deseados para el DW. Para esto se deben de tener en cuenta los principales componentes de hardware que son un factor importante para mejorar el rendimiento del DW. Los principales componentes de hardware son: Disco I/O, Memoria, CPU y Red.

#### *3.2.2.1. Disco I/O*

El subsistema de I/O es un componente vital de una base de datos Oracle, el rendimiento de muchos sistemas de software está limitado por el I/O a disco. Oracle está diseñado de manera tal que si una aplicación está bien escrita, su rendimiento no debería ser limitado por I/O. Los siguientes requerimientos de I/O deben ser considerados al diseñar el subsistema de I/O de una base de datos Oracle:

Almacenamiento: la capacidad mínima de discos, Accesibilidad: la continuidad u horarios laborales solamente y Rendimiento: tiempo de respuesta de la aplicación.

**Configuración del subsistema I/O:**

Para diseñar una eficiente I/O del subsistema, necesita la siguiente información: Usar de dispositivos sin formato o software de terceros que permiten escribir directamente en el disco, evitando la lectura y escritura en la memoria caché, Usar I/O del disco que tiene la caché de memoria suficientemente grande y Evitar la configuración RAID 5 para aplicaciones intensivas en escritura.

1.I/O de concurrencia.

I/O de concurrencia mide el número de procesos distintos que a la vez hacen peticiones al subsistema de I/O. Desde el punto de vista de Oracle, éste se considera el número de procesos al mismo tiempo que la expedición de I/O de peticiones. Por lo que un alto grado de I/O de concurrencia implica que hay muchos procesos distintos al mismo tiempo que las peticiones I/O. Y un bajo grado de concurrencia implica que algunos procesos son a la vez peticiones de I/O.

2.I/O *size*.

I/O *size*, es el tamaño de la solicitud de I/O desde la perspectiva de Oracle. El mínimo de I/O *size* es el tamaño de bloques del sistema, mientras que el máximo es típicamente un factor del tamaño de bloques multiplicados por el número de bloques a leer. Aunque el tamaño de I/O puede variar dependiendo del tipo de operación, hay algunas estimaciones razonables en general que se puede hacer, dependiendo de la naturaleza de la solicitud.

Con un sistema de DSS la mayoría de los subsistemas de I/O típicamente van a ser grandes, aproximadamente n * *DB_BLOCK_SIZE.*

3.Disponibilidad.

Aquí está presente la tecnología RAID para satisfacer las necesidades de recuperación y de cualquier medida de seguridad específica de Oracle, como la creación de reflejo de *redo logs*, para el cual debe de archivar los registros y archivos de control. Esto asegura que los archivos de datos y los archivos de registro no puedan ser perdidos en un fallo de disco único.

4.Tamaño de almacenamiento.

En estos casos es necesario considerar los siguientes métodos de maximizar el espacio de almacenamiento: Usar diferentes tipos de configuraciones RAID, Usar más discos de almacenamiento y Usar discos con una capacidad más grande.

A continuación se listan parámetros de Oracle y del Sistema Operativo que se pueden usar para especificar el tamaño de I/O:

- ***DB_BLOCK_SIZE*:** determina el tamaño de un solo bloque de peticiones I/O. Este parámetro también es se usa en combinación con los parámetros para determinar tamaño de la petición de multibloques de I/O.
- ***OSblocksize*:** Determina I/O de tamaño de registro de rehacer y las operaciones de archivo de registro.
- ***Maximum* OS I/O *size*:** Coloca un límite superior para el tamaño de una sola solicitud de I/O.
- ***DB_FILE_MULTIBLOCK_READ_COUNT*:** calcula el máximo tamaño de I/O para escaneos completos de tabla el cual es multiplicando este parámetro con *DB_BLOCK_SIZE.*
- ***SORT_AREA_SIZE*:** Determina los tamaños I/O y la concurrencia para operaciones en orden.
- ***HASH_AREA_SIZE*:** Determina el tamaño I/O para las operaciones de *hash*.

#### *3.2.2.2. Memoria*

Oracle almacena la información en memoria caché del disco. Ya que el acceso a la memoria es mucho más rápido que el acceso a disco. Este trabaja escaneando el disco físico de I/O donde se toma una cantidad significativa de tiempo en comparación con el acceso de la memoria, típicamente del orden de 10 milisegundos. El disco físico también aumenta los recursos del CPU necesario, debido a la longitud del camino en el dispositivo, los conductores y planificadores de eventos del Sistema Operativo. Por esta razón es más eficiente para las solicitudes de datos de acceso frecuente, ser satisfecha únicamente por la memoria en lugar de exigir también el acceso a disco.

Un objetivo del rendimiento es reducir la I/O física tanto como sea posible, ya sea por los datos requeridos por la memoria o el proceso de recuperar los datos necesarios de un modo más eficiente.

**Memoria caché:**

Los principales depósitos de memoria cache que afectan a Oracle son: *Shared pool, Large pool, Java pool, Buffer Caché, Log buffer* y Proceso privado de la memoria.

El tamaño de estos depósitos de memoria, se pueden configurar mediante la configuración de parámetros de inicialización. Ya que los valores de estos parámetros son configurables de forma dinámica mediantes la instrucción ALERT SYSTEM (la cual es una excepción del *log buffer* y *java pool* que son estáticos después del arranque de la base de datos Oracle).

**Dinámica de cambio de tamaño de caché:**

Como se ha dictado anteriormente, se puede configurar dinámicamente el tamaño de *sharedpool*, *largepool*, *buffer caché* y procesos privados de la memoria.

Entonces todas estas memorias caché, son asignadas en unidades de gránulos. En términos generales, en la mayoría de las plataformas, ya sea Windows o Linux el tamaño de un gránulo es de 4 MB, por lo tanto si el tamaño total de SGA está entre 16 MB y 128 MB puede ser entonces que exista cierta dependencia de la plataforma.

A continuación se resaltan algunas de las vistas que nos facilitaran información referente a la disponibilidad del SGA: **V $ SGA_CURRENT_RE*SIZE*_OPS:** información acerca de las operaciones para cambiar el tamaño del SGA, **V $ SGA_RE*SIZE*_OPS:** información los últimos cambios de operaciones en el SGA, **V $ SGA_DYNAMIC_COMPONENTS:** Información sobre los componentes dinámicos en el SGA y **V $ SGA_DYNAMIC_FREE_MEMORY:** Información sobre la cantidad de memoria SGA disponible para las operaciones del futuro dinámico tamaño SGA.

#### *3.2.2.3. CPU*

El CPU en vista a las aplicaciones depende de la complejidad y volumen de trabajo de las mismas. La mayoría de los Sistemas Operativos los estados del CPU incluyen usuarios, sistema, tiempo de espera, y los componentes. En este caso Oracle no tiene acceso directo a las estadísticas de la utilización del CPU, en todo caso, cuenta con las métricas secundarias del CPU. Teniendo en cuenta además que Oracle utiliza un pequeño porcentaje de carga del CPU ya que es totalmente ajena a la instancia, porque cuenta con sus propios destinos de almacenamiento, esto sería el PGA (*Program Global Area*) y el SGA que ya ha sido expuesto anteriormente.

En Oracle existen diferentes actividades que ayudaran a administrar la base de datos en vista a las tareas diarias que esta ejecute, en caso del manejo del CPU, se podrá administrar las secciones del usuario, la cual cumple con diferentes características como: *Top Session Finder*, *Session Browser* y SGA

Las cuales definen qué unidad de consumo desea observar como (CPU, I/O, memoria, entre otros) y luego puede clasificar a los usuarios de acuerdo al recurso. También puede ser ejecutada la revisión de oportunidades de optimización de aplicaciones la cual permite a la base de datos revisar rápidamente y clasificar las secciones que mayor consumo de recursos tienen.

Otra ventaja de Oracle es que tiene su propio kernel, el cual es el corazón del SGBDR Oracle y que es cargado en la memoria al inicio de las operaciones y usado por cada base de datos existente en el equipo.

En conclusión, durante las horas máximas de carga de trabajo con Oracle, el 90% de utilización del CPU es aceptable. Los recursos de hardware deben de ser adecuados para las necesidades de las aplicaciones específicas y para evitar cuellos de botella de rendimiento relacionado con hardware, cada componente de hardware debería funcionar a no menos de 80% de la capacidad.

#### *3.2.2.4.Red*

En cuanto a al tratamiento de la Red, Oracle utiliza tres archivos (listener.ora, tnsnames.*ora* y *SQLNET.ora*) para configuración de red. Hay que decir además que tiene una arquitectura cliente-servidor; que es la más extendida en la actualidad y constituye la base para la mayoría de los sistemas de información modernos. Además existen dos modos de conexión diferentes, pero la indicada a utilizar la base de datos para DW es el Modo Servidor Dedicado.

Esta herramienta cuenta con diferentes parámetros configurables de los servicios de red de Oracle en vista al rendimiento, estos parámetros son:

- ***cluster_database*** en TRUE para activar la opción *RealApplicationClusters*.
  Rango de Valores: TRUE | FALSE Valor por Defecto: FALSE
- ***processes***: Especifica el número máximo de procesos de usuario del Sistema Operativo que se pueden conectar simultáneamente a Oracle Server. Este valor debe tener en cuenta todos los procesos en segundo plano, como por ejemplo, procesos de la cola de trabajos y de ejecución en paralelo.
  Rango de Valores: 6 a un valor que depende del Sistema Operativo.
  Valor por Defecto: Depende de *PARALLEL_MAX_SERVERS*
- ***pga_aggregate_target***: Especifica las memorias PGA agregadas de destino de todos los procesos del servidor adjuntos a la instancia. Es mejor definir este parámetro en un valor positivo antes de activar la definición automática de áreas de trabajo. Esta memoria no reside en SGA. La base de datos utiliza este parámetro como cantidad de memoria PGA de destino que utiliza. Al definir este parámetro, reste la SGA de la memoria total del sistema disponible para la instancia Oracle. La memoria restante se puede asignar a *pga_aggregate_target*.
  Rango de Valores: Valores enteros más la letra K, M o G para especificar este límite en kilobytes, megabytes o gigabytes. El valor mínimo es 10 M y el máximo es 4000 G
  Valor por Defecto: "No Especificado", que significa que el ajuste automático de las áreas de trabajo está completamente desactivado.

- ***shared_servers***: Especifica el número de procesos del servidor para crear un entorno de servidor compartido cuando se inicia una instancia.
  Rango de Valores: Depende del Sistema Operativo.
  Valor por Defecto: 1
- ***local_listener***: lista de direcciones de red de Oracle que identifica las instancias de base de datos en la misma máquina que los *listeners* de red de Oracle. Cada instancia y distribuidor se registra con el *listener* para activar las conexiones del cliente. Este parámetro sobrescribe los parámetros *MTS_LISTENER_ADDRESS* y *MTS_MULTIPLE_LISTENERS* que se quedan obsoletos a partir de 8.1.
  Rango de Valores:
  Lista de direcciones de red de Oracle.
  Valor por Defecto: (ADDRESS_LIST= (Dirección= (Protocolo=TCP) (Host=hostlocal) (Puerto="No.")) (Dirección= (Protocolo=IPC) (Clave=nombreBD))).

### 3.2.3. Área de proceso: Técnicas de optimización

*Véase Anexo 2*

***Conclusiones***

1.Los elementos configurables que determinan el comportamiento interno del gestor, así como la utilización de los recursos, influyen positivamente gestionado, de la manera en que se plantean en las prácticas descritas en las Áreas de procesos: “Gestor de base de datos” y “Utilización de recursos”.

2.A pesar de que las técnicas de optimización se conciben originalmente para Oracle, las prácticas contenidas dentro del área de proceso “Técnicas de optimización”, describen cómo implementarla para PostgreSQL, las mismas están descritas en los Anexos para Oracle, pues como ya se explicó fueron concebidas originalmente para él, por lo que se considera que el verdadero aporte radica en el área de proceso de la guía “Construyendo almacenes de datos con PostgreSQL”.

## Capitulo 4: Validación. Método Delphi

### *Introducción*

En este capítulo se exponen los detalles de la validación de la adecuación del PROCESO a la METODOLOGÍA, así como las guías de configuración del rendimiento diseñadas para aumentar la eficiencia de las aplicaciones de DW, desarrolladas en la Línea de DATEC.

Las guías de configuración constituyen una aproximación teórica a aquellos aspectos configurables que brindan los gestores PostgreSQL y Oracle, con los que se trabaja en la Línea, para manejar el comportamiento del gestor y la utilización de los recursos del sistema que necesita para su funcionamiento. Se pretende demostrar que mediante la configuración adecuada de estos aspectos es posible mejorar la eficiencia de las aplicaciones, manejando indistintamente aquellos elementos que determinan el comportamiento del gestor y como consecuencia inciden en el rendimiento de las aplicaciones, dígase, tiempos de espera, respuesta, reacción y productividad e igualmente optimizar la utilización de los recursos.

Esta validación está dirigida a tres aspectos fundamentales; la adecuación del PROCESO a la METODOLOGÍA utilizada en la Línea, la influencia positiva que tenga en el rendimiento, la “Utilización de los recursos” y la efectividad de las guías. Siguiendo estas directrices, se expone la evolución del método escogido para validar la solución: el Método Delphi de validación mediante expertos, escogido dada la naturaleza investigativa del presente trabajo, el mismo se describe en dos etapas, la validación independiente del PROCESO dividido en “Construyendo almacenes con PostgreSQL” y “Construyendo almacenes con Oracle”, con el objetivo de eliminar la dependencia del éxito del mismo para uno u otro gestor.

### *4.Descripción del Método Delphi de validación mediante expertos*

El método procede mediante la interrogación a expertos, personas con cierta experiencia en el trabajo enmarcado en el campo de acción definido, con la ayuda de cuestionarios sucesivos con el objetivo de poner de manifiesto convergencias de opiniones y deducir un consenso con respecto a la efectividad del PROCESO a validar referente al rendimiento de las aplicaciones de DW.

Dicho método se ejecutará en dos fases individuales, cada una de ellas con un panel de expertos, modelador (miembro del equipo de investigación que se encarga de distribuir las encuestas, garantizar el anonimato entre los miembros del panel de expertos y tabular los datos resultantes) y cuestionarios diferentes. Con el objetivo de validar la efectividad del PROCESO y las guías diseñadas para cada uno de los gestores (PostgreSQL y Oracle) por separado, garantizando así la independencia del éxito del PROCESO para uno u otro gestor, de la siguiente manera:

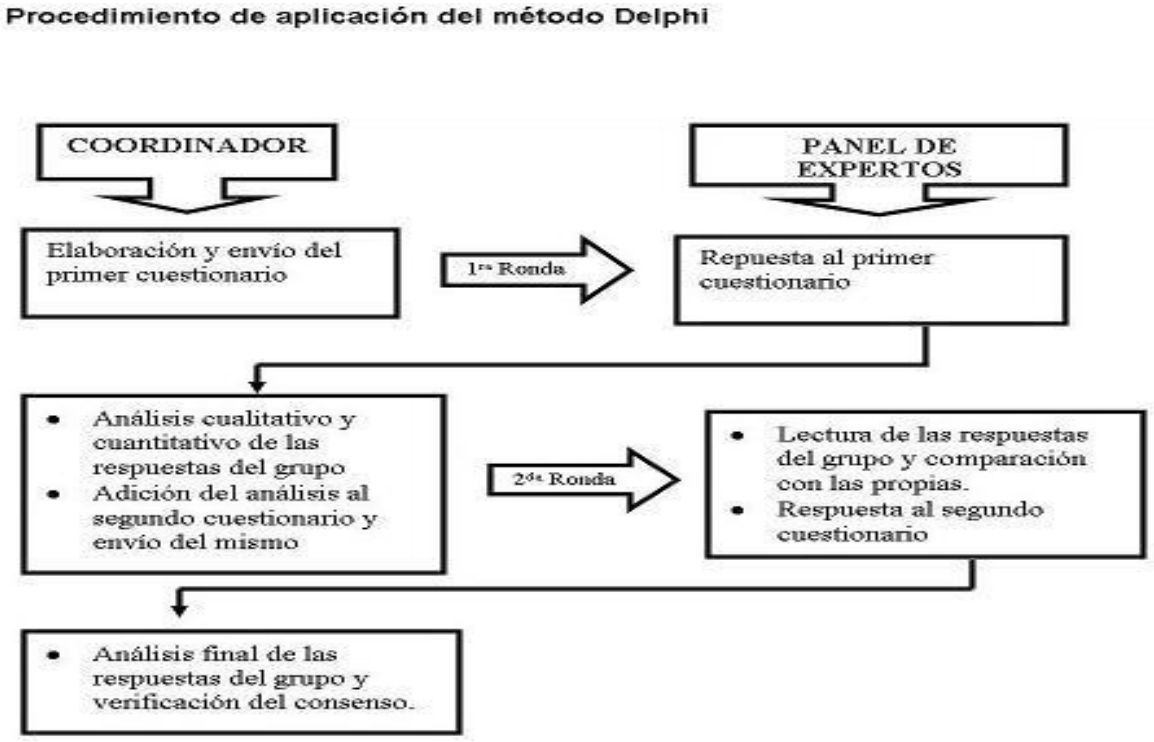


Ilustración 14: Iteraciones del Método Delphi de validación de expertos

### *4.1.Objetivos que se persiguen con la aplicación del Método*

**Objetivo General**

Validar el PROCESO diseñado para optimizar el rendimiento de las aplicaciones de DW, desarrolladas en la Línea de DATEC.

**Objetivos Específicos**

1. Comprobar que sean adecuadas las prácticas incluidas en las actividades definidas por la METODOLOGÍA utilizada en la Línea.
2. Comprobar que los elementos conjuntamente con las técnicas de optimización descritas, tratados en las prácticas inciden de forma positiva en el rendimiento de las aplicaciones desarrolladas.
3. Comprobar que el tratamiento que se le da a los recursos que utiliza el sistema define un uso más eficaz de los mismos haciendo más eficientes las aplicaciones desarrolladas.

### *4.2.Consideraciones generales con respecto al tratamiento de los resultados de la encuesta*

Siendo la notación para cada una de las tablas (x para expresar la acuerdo acerca del tema en cuestión y – para indicar desacuerdo), el sistema escogido para recolectar la opinión de los expertos.

Según las especificaciones del Método Delphi, se considera que los expertos han llegado a un consenso cuando el porciento de coincidencias en un aspecto es mayor que 60, y de esta manera se tratan los resultados en la evaluación de cada una de las iteraciones del método.

### *4.3.Construyendo almacenes de datos con PostgreSQL*

A continuación se describe el progreso de la aplicación del Método Delphi de validación de expertos, para el PROCESO de optimización diseñado para PostgreSQL. Se presenta el panel de expertos, así como los temas, las encuestas y la tabulación de los datos para cada una de las iteraciones del mismo, dando finalmente una conclusión según los resultados obtenidos.

#### 4.3.1.Panel de Expertos

Nombre y Apellidos: **Daymel Bonne Solís**

e-mail: dbonne@uci.cu

Título: Ingeniero en Ciencias Informáticas

Ubicación Laboral: Centro de Tecnologías de Gestión de Datos

Categoría Docente: Instructor Recién Graduado

Resumen Experiencia en el trabajo con PostgreSQL: Participación por 3 años como Administrador de Base de Datos en el proyecto del Centro de Tratamiento y Análisis de Información de Seguridad Ciudadana. Estuvo a cargo de la migración de la base de datos del sistema informático de Oracle hacia PostgreSQL. Miembro del Grupo de Soporte de la Subdirección de Gestión de Negocios del Centro de

Tecnologías de Almacenamiento y Análisis de Datos. Cubre el rol de desarrollador en el proyecto Clúster de altas prestaciones de *PostgreSQL* está encaminado al diseño y montaje de un clúster de alto rendimiento y alta disponibilidad usando el gestor *PostgreSQL*. Participó en el curso internacional de PostgreSQL impartido por Álvaro Herrera

Nombre y Apellidos: **Marcos Luis Ortiz Valmaseda**
e-mail: mlortiz@uci.cu
Título: Ingeniero en Ciencias Informáticas
Ubicación Laboral: Centro de Tecnologías de Gestión de Datos
Categoría Docente: Instructor Recién Graduado
Resumen Experiencia en el trabajo con PostgreSQL: Miembro del Grupo de Soporte de la Subdirección de Gestión de Negocios del Centro de Tecnologías de Almacenamiento y Análisis de Datos, se desempeña como arquitecto del proyecto Clúster de altas prestaciones de *PostgreSQL* está encaminado al diseño y montaje de un clúster de alto rendimiento y alta disponibilidad usando el gestor *PostgreSQL*. Participó en el curso internacional de PostgreSQL impartido por Álvaro Herrera

Nombre y Apellidos: **Héctor Miguel Beltrán Lugo**
e-mail: hmbeltran@uci.cu
Título: Ingeniero en Ciencias Informáticas
Ubicación Laboral: Centro de Tecnologías de Gestión de Datos
Categoría Docente: Instructor Recién Graduado
Resumen Experiencia en el trabajo con PostgreSQL: Miembro del Grupo de Soporte de la Subdirección de Gestión de Negocios del Centro de Tecnologías de Almacenamiento y Análisis de Datos. Brinda servicios de soporte técnico en el proyecto Clúster de altas prestaciones de *PostgreSQL* está encaminado al diseño y montaje de un clúster de alto rendimiento y alta disponibilidad usando el gestor *PostgreSQL*. Participó en el curso internacional de PostgreSQL impartido por Álvaro Herrera

#### 4.3.2.1ra Iteración

##### 4.3.2.1.Preguntas

1. Son las prácticas definidas para cada una de las fases (Diseño, Implementación y Prueba), adecuadas con respecto a:

- La fase en la que están enmarcadas, o sea, la parte de la METODOLOGÍA en que son incluidas.
- Objetivos que se persiguen con dichas prácticas en cada una de las fases.

  --------- Si--------- No (Argumente)

2. ¿Influyen positivamente en el rendimiento del sistema?

   --------- Si---------- No (Argumente)

3. ¿Es posible afirmar que la "Utilización de los recursos" que se describe en el PROCESO influyen positivamente la eficiencia de las aplicaciones desarrolladas?

   ---------- Si---------- No (Argumente)

4. Marque aquellos aspectos que usted considera es posible mejorar con la aplicación del PROCESO descrito durante la explotación del DW.

   --------- tiempo de respuesta del sistema

   --------- cantidad de usuarios atendidos simultáneamente

   --------- tiempo de espera entre peticiones

   --------- tiempo de reacción del sistema ante una petición

   --------- tiempos de máxima y mínima función del DW

   --------- productividad (Índice de la velocidad de ejecución)

5. El tratamiento que se propone de los recursos del sistema influyen en:

- El tiempo de espera del sistema por la culminación de una petición I/O.
- La razón comprendida entre la cantidad de fallos del sistema en determinado intervalo de tiempo.

  ---------- Si--------- No (Argumente)

#### 4.3.2.2.Conclusiones de la 1ra iteración

Luego de procesar los datos de la primera encuesta se arriba a las siguientes conclusiones:

- Las prácticas propuestas en el PROCESO en cada una de las fases son adecuadas en cuanto a:

✓ La fase de la METODOLOGÍA en que son incluidas

✓ Efectividad en lograr la eficiencia de las aplicaciones desarrolladas con la METODOLOGÍA.

- Siguiendo el PROCESO descrito es posible influir positivamente en el "Rendimiento de las aplicaciones" en cuanto a: Tiempo de espera, Tiempo de reacción, Tiempo de respuesta y Productividad

- Siguiendo el PROCESO descrito es posible optimizar la “Utilización de los recursos” a través de la configuración de PostgreSQL, con vistas a aumentar la eficiencia de las aplicaciones desarrolladas en la Línea.
- Las prácticas propuestas en el PROCESO permiten mejorar el tiempo de espera por peticiones I/O, y la relación entre los fallos en el tiempo del sistema.

#### 4.3.3.2da iteración

##### 4.3.3.1.Preguntas

1. En las prácticas propuestas en el PROCESO para el flujo de trabajo Arquitectura y Diseño, se consideran una serie de aspectos, marque aquellos que usted considera correctos y mencione los que Ud. considere resultan importantes si no están incluidos:

------ Diseño apropiado de las tablas

------ Utilización de llaves sustitutas para las dimensiones

------ Evitar dependencias entre las tablas de hechos y las dimensiones

2. Para el comportamiento de PostgreSQL con respecto al manejo de los datos, se propone:

- Centrar el PROCESO de optimización en las tablas con más accesos.
- Utilizar Vistas Materializadas en las tablas con más accesos.
- Realizar el tratamiento a las consultas más lentas utilizando: *log_min_duration_statement*, *pg_fouine* y *auto_explain,* con vistas a identificarlas y optimizarlas, posteriormente.
- Utilizar herramienta *pg_bulk_loader* y el comando COPY, para operaciones en lotes.
- Habilitar el *autovacuum* (ON).
- Utilización del *bgwriter* o *background writer* para posibilitar que los procesos puedan traer páginas a *shared_buffers* sin la necesidad de escribir en disco.
- Deshabilitar el *autocommit* cuando se realizan múltiples INSERT.
- Utilizar COPY para todas las filas en una sola orden en lugar de realizar una serie de INSERT

Responda si considera estas consideraciones adecuadas:

------- Si------- No (Argumente)

3. Para la configuración que especifica el comportamiento de PostgreSQL en el tratamiento y análisis de sus transacciones, se definieron una serie de parámetros a tener en cuenta, agrupados en las siguientes categorías:

- Checkpoints

- Parámetros para la optimización de consultas
- Parámetros para manejar la estadística de las consultas
- Modificadores de consultas

Responda si estos parámetros seleccionados por categorías son adecuados.

------- Si-------- No (Argumente)

4. Para la configuración que especifica el manejo de los recursos por parte de PostgreSQL, se definieron una serie de parámetros a tener en cuenta, agrupados por recursos de la manera siguiente: Memoria, CPU, Red y Disco I/O.

Responda si estos parámetros seleccionados por recursos son adecuados.

------- Si-------- No (Argumente)

6. Diga si el proceso que se define en el PROCESO para la fase de Pruebas del DW es adecuado para detectar y resolver problemas de rendimiento.

------- Si------- No (Argumente)

7. Responda si es posible o no aplicar las técnicas de optimización para DW según lo referido en el PROCESO.

------ Si------- No (Argumente)

#### 4.3.3.2.Conclusiones de la 2da Iteración

Como resultado del análisis de la segunda ronda de encuestas se cumple:

- Es apropiado durante la definición de la Arquitectura y Diseño: Diseño apropiado de las tablas, Evitar dependencias entre los Hechos y Dimensiones y Uso de llaves sustitutas en las Dimensiones. Durante este proceso es además necesario incluir: Particionamiento eficiente de los datos y Uso de índices.
- Las prácticas que dictan cómo debe ser el comportamiento del gestor son correctas.
- Es necesario para lograr mejor rendimiento en las aplicaciones, seguir las recomendaciones dadas en el PROCESO en cuanto a: Tratamiento de consultas, Utilización de los recursos, Pruebas y Utilización de técnicas de optimización.

### 4.4.4. Conclusiones de "Construyendo almacenes con PostgreSQL"

Luego de la aplicación del Método Delphi de validación mediante expertos, se arriba a la siguiente conclusión con respecto al PROCESO asociado a la guía **"**Construyendo almacenes con PostgreSQL**":**

- El PROCESO descrito es perfectamente adecuado a la METODOLOGÍA utilizada en la Línea, en los flujos de trabajo en los que se propone incluir las prácticas que lo conforman.

- Mediante la aplicación de este PROCESO es posible aumentar el Rendimiento de las aplicaciones desarrolladas.
- La Utilización de Recursos que se proponen devienen en eficiencia para las aplicaciones.
- Es posible implementar en PostgreSQL las Técnicas de Optimización para DW que se describen con el objetivo de mejorar la calidad de las aplicaciones.

## *4.4.Construyendo almacenes de datos con Oracle*

A continuación se describe el progreso de la aplicación del Método Delphi de validación de expertos para el PROCESO diseñado para Oracle. Se presenta el panel de expertos, así como los temas, las encuentras y la tabulación de los datos, para cada una de las iteraciones del mismo, dando finalmente una conclusión según los resultados obtenidos.

### 4.4.1.Panel de Expertos

Nombre y Apellidos: **Yanet Peña Vázquez**

e-mail: ypenav@uci.cu

Título: Ingeniero en Ciencias Informáticas y Máster en Informática Aplicada.

Categoría Docente: Profesor Instructor

Ubicación Laboral: Centro de Tecnologías de Almacenamiento y Análisis de Datos (DATEC)

Resumen Experiencia en el trabajo con Oracle:

Trabajo con Oracle desde el 2006, hace 4 años aproximadamente. Su primera experiencia fue con Oracle versión 9i, luego profundizó en Oracle 10g. Se especializó en el área de Inteligencia de Negocio, con "*Oracle BI Estándar Edition*", "*Oracle BI Enterpise Edition*"y "*Oracle BI Enterpise Edition Plus*".

Nombre y Apellidos: **Asniobys Hernández López**

e-mail: ahernandezlo@uci.cu

Título: Ingeniero en Ciencias Informáticas, Máster en Informática Aplicada.

Categoría Docente: Instructor

Ubicación Laboral: Centro de Tecnologías de Almacenamiento y Análisis de Datos (DATEC)

Resumen Experiencia en el trabajo con Oracle:

Experiencia trabajando con base de datos Oracle. Y un 1 año trabajando como diseñador de base de datos.

Nombre y Apellidos: **Doris Medina Mustelier**
e-mail: dmedina@uci.cu
Título: Ingeniero en Ciencias Informáticas.
Categoría Docente: Instructor Recién Graduado.
Ubicación Laboral: Centro de Tecnologías de Almacenamiento y Análisis de Datos (DATEC)
Resumen Experiencia en el trabajo con Oracle:
Experiencia trabajando con base de datos Oracle por un 1 año.

#### 4.4.2.1ra Iteración

En esta primera iteración de validación del PROCESO diseñado para Oracle se tratarán los mismos temas, con igual selección de preguntas que la 1ra iteración de la validación del PROCESO diseñado para PostgreSQL.

#### 4.4.2.1.Conclusiones de la 1ra iteración

Luego de procesar los datos de la primera encuesta se arriba a las siguientes conclusiones:

- Las prácticas propuestas en el PROCESO en cada una de las fases son adecuadas en cuanto a:
  - La fase de la METODOLOGÍA en que son incluidas.
  - Efectividad en lograr la eficiencia de las aplicaciones desarrolladas con la METODOLOGÍA.
- Siguiendo el PROCESO descrito es posible influir positivamente en el Rendimiento de las aplicaciones en cuanto a:
  - Tiempo de espera
  - Tiempo de reacción
  - Tiempo de respuesta
  - Productividad
- Siguiendo el PROCESO descrito es posible optimizar la “Utilización de los recursos” a través de la configuración de PostgreSQL, con vistas a aumentar la eficiencia de las aplicaciones desarrolladas en la Línea.
- Las prácticas propuestas en el PROCESO permiten mejorar el tiempo de espera por peticiones I/O, y la relación entre los fallos en el tiempo del sistema.

#### 4.4.3.2da Iteración

##### 4.4.3.1.Preguntas

1. Las prácticas propuestas en el PROCESO para el flujo de trabajo Arquitectura y Diseño, están considerados una serie de aspectos, marque aquellos que usted crea correctos y mencione los que Ud. considere resultan importantes si no están incluidos:

   ------- Utilización de *tablespaces*

   ------ Utilización de tablas de hechos como cubos OLAP

2. Para el comportamiento de Oracle con respecto al manejo de los datos, se propone:

- Utilización de la herramienta "*Oracle Warehouse Builder"*.
- Utilizar Vistas Materializadas en las tablas con más accesos.
- Utilizar *Tablespaces* para mejorar la organización de los datos.
- Realizar el tratamiento de optimización del rendimiento en la base de datos, configurando los parámetros de inicialización.
- Utilizar opciones que incluye Oracle para evitar que se agoten los recursos.

Responda si considera estas consideraciones adecuadas:

------- Si------- No (Argumente)

3. Para la configuración que especifica el comportamiento de Oracle en el tratamiento y acceso a las tablas, se definieron 2 tipos de accesos: Secuencialmente (FULL SCAN) y Usando índices.

Responda si de estos, el acceso por índice es el más adecuado.

------- Si-------- No (Argumente)

4. Para mejorar la eficiencia y tiempo de respuesta de los informes, es de buenas prácticas la utilización de tablas agregadas y un recurso de optimización física que puede aportar grandes mejoras es la utilización de vistas materializadas. En cuanto al rendimiento del DW es de buenas prácticas utilizar vistas materializadas para:

-------Ir actualizando tablas intermedias que alimenten los esquemas de DW

------ Ir directamente para implementar tablas agregadas que se refrescarán a partir de las tablas base.

5. Para la configuración que especifica el manejo de los recursos por parte de Oracle, se definieron una serie de parámetros a tener en cuenta, agrupados por recursos de la manera siguiente: Memoria, CPU, Red y Disco I/O.

Responda si estos parámetros seleccionados por recursos son adecuados.

------- Si-------- No (Argumente)

6. Diga si la secuencia de actividades que se define en el PROCESO en el subproceso de Pruebas del DW es adecuado para detectar y resolver problemas de rendimiento utilizando las herramientas incluidas en Oracle.

------- Si------- No (Argumente)

7. Responda si es posible o no aplicar las técnicas de optimización para DW según lo referido en el PROCESO.

------ Si------- No (Argumente)

#### 4.4.3.2.Conclusiones de la 2da Iteración

Como resultado del análisis de la segunda ronda de encuestas se cumple:

1.Es apropiado durante la definición de la Arquitectura y Diseño: Tablas de hechos como cubos OLAP, *Tablespaces y* Índice y particionamiento.

2.Los parámetros de inicialización para especificar el comportamiento del gestor en el tratamiento de los datos según las prácticas son los correctos.

3.La configuración de la base de datos Oracle para evitar que se agoten los recursos del sistema es la adecuada, así como el tratamiento de los recursos y la implementación de las técnicas de optimización propuestas en el PROCESO.

4.La practica definida para el proceso de pruebas, permite llevar a niveles de rendimiento aceptables las aplicaciones de DW construidas sobre el gestor.

#### 4.4.4.Conclusiones de "Construyendo almacenes con Oracle"

Luego de la aplicación del Método Delphi de validación mediante expertos, se arriba a la siguiente conclusión con respecto al Proceso asociado a "Construyendo almacenes con Oracle"**:**

5. El PROCESO descrito es perfectamente adecuado a la METODOLOGÍA utilizada en la Línea, en los flujos de trabajo en los que se propone incluir las prácticas que lo conforman.
6. Mediante la aplicación de este PROCESO es posible aumentar el Rendimiento de las aplicaciones desarrolladas.
7. La Utilización de Recursos que se proponen devienen en eficiencia para las aplicaciones.
8. Las Técnicas de Optimización para DW descritas en el PROCESO son capaces de mejorar la calidad de las aplicaciones.

***Conclusiones***

Durante este capítulo ha sido posible comprobar el éxito del PROCESO diseñado en sus dos variantes, “Construyendo almacenes de datos con PostgreSQL” y “Construyendo almacenes de datos con Oracle”, en cuanto a:

- Adecuación con la METODOLOGÍA utilizada en la Línea.
- Elevar el rendimiento de las aplicaciones desarrolladas a niveles aceptables, que cumplan con los requisitos de calidad del ISO 9126.
- Optimización de la utilización de los recursos por parte del gestor, cuestión que repercute en el rendimiento final de las aplicaciones.
- Posibilidad de implementar las técnicas de optimización de DW con el gestor, con vistas a mejorar el rendimiento.

### Conclusiones generales

1. Es necesario para mejorar el rendimiento de DW configurar el comportamiento interno del gestor (estadísticas, monitoreo y modificadores de consultas), los parámetros que brindan los gestores para gestionar memoria, disco, CPU y red para incidir directamente en los problemas de rendimiento que se ponen de manifiesto en el desempeño de las aplicaciones.
2. El PROCESO diseñado define cómo utilizar en los gestores relaciones PostgreSQL y Oracle las principales técnicas de optimización existentes para mejorar la calidad de los productos de la Línea.
3. El PROCESO cumple los objetivos propuestos mediante la configuración de los parámetros que establecen la utilización de los recursos y el gestor de bases de datos utilizados para el desarrollo de los productos en la Línea de DATEC.
4. El PROCESO diseñado tuvo éxito en cuanto a: adecuación a la METODOLOGÍA utilizada, efectividad en vista a mejorar el Rendimiento, optimización de la “Utilización de los Recursos” y la implementación de las técnicas de optimización para DW.

### Recomendaciones

Luego de realizada la investigación y cumplidos los objetivos del trabajo, los autores recomiendan:

- Realizar según las características específicas de los recursos y el tipo de aplicaciones, una actualización del PROCESO para el desarrollo en la Línea.
- Aplicar el PROCESO diseñado para el desarrollo de los almacenes en la Línea de DATEC.
- Actualizar el PROCESO descrito con las modificaciones y nuevas versiones del gestor, ya sea PostgreSQL u Oracle.
- Realizar según los requerimientos expuestos para los DW, un estudio de las herramientas de prueba adecuadas, para medir el rendimiento de las aplicaciones desarrolladas.

### *Bibliografía Consultada*

**(s.f.)Oracle, Oracle. 2010.** Oracle Corporation. [En línea] Mayo de 2010. http://www.oracle.com/global/es/products/solutions/business intelligence/dw_home.html.

**2003, ISO/IEC. 2003.***Anexo B: Tablas contenedoras de las métricas.* 2003.

**2005, Norma Cubana. 2005.***ISO/IEC 9126:2005 Ingeniería de Software. Calidad del producto. Parte 1: Modelo de calidad. .* 2005.

**Bartollini, GAbrielle. 2009.***Data warehousing con PostgreSQL. PgDay 2009.* 2009.

**Cabrera, María .E Casales. 2007.** Data Warehouse (Almacenes de Datos). [En línea] 2007. http://hp.fciencias.unam.mx/~alg/bd/dwh.pdf.

**Castillo, Carlos. 2008.***Tema 2. Sistemas gestores de bases de.* 2008.

**Castro, Dr. Rogelio S. Silverio. 2009.***Papel de la Minería de Datos dentro de los Sistemas de Información Empresariales .* 2009.

**Corporation, Oracle. 2010.** Oracle. [En línea] 17 de Abril de 2010. http://www.oracle.com/corporate/princing/technologi-princ-list.pdf.

**D'Ottone, Lic. Rosana y Boccaloni, Lic. Gustavo. 2009.** Considerando el particionamiento. [En línea] 2009. http://www.ixora.com.au/tips/design/partitioning.htm.

**Escorial, Jorge Salamanca. 2006.***La Norma ISO/IEC 9126 y los modelos de calidad Boehm y Mc Call. 2006.* 2006.

**Fogel, S. 2002.***Oracle Database. Administrator´s guide.* 2002.

**Goodwin, Candince. 2003.***Bussiness Intelligence - Assault on the data mountain.* 2003.

**Inocencio, Beatriz Canales. 2004.** Nueva Economía, Internet y tecnología. [En línea] Julio de 2004. http://www.gestiopolis.com/canales2/gerencia/1/busint.htm.

**ISO/IEC. 2003.***Software Engineering. Product quality. Part 3: Internal metrics.* 2003.

**John Worsley, Joshua Drake. 2009.** Practical PostgreSQL. [En línea] 2009. www.postgresql.org/docs/awbook.html.

**Mendez, A., y otros. 2004.** Fundamentos de Data Warehouse. [En línea] 2004.

**Momjian, Bruce. 2008.** PostgreSQL: Introduction and Concepts. [En línea] 2008. http://developer.postgresql.org/todo.php http://archives.postgresql.org/pgsql-es-.

**Parra, Jaime G. Orjuela. 2007.***Características de calidad para la arquitectura de software.* 2007.

**PostgreSQL, Comunidad Internacional de Desarrollo de. 2009.***PostgreSQL 8.4.1 Documentation.* 2009.

**PostgreSQL, Comunidad Internacional de Desarrollo de y Pg_Foundry. 2005.***PostgreSQL. Hardware performance tunning.* 2005.

**PostgreSQL, Comunidad Internacional de Desarrollo. 2010.***PostgreSQL 9.0 beta Documentation.* 2010.

**Velazco, Roberto Hernando. 2006.** Almacenes de datos (Datawarehouse). [En línea] 2006. http://www.rhernando.net/modules/tutorials/doc/bd/dw.html.

**Wiles, Frank. 2009.***Performance Tunning PostgreSQL.* 2009.

### *Bibliografía Referenciada*

**Antunez, Ivette Marrero. 2008.** *La Inteligencia de Negocio desde la perspectiva cubana: retos y tendencias.* s.l. : Instituto de Información científica y tecnológica, 2008.

**Bartollini, GAbrielle. 2009.** *Data warehousing con PostgreSQL. PgDay 2009.* 2009.

**Bosquet, Isabel Dapena, Roque, Antonio Muñoz San y Miralles, Álvaro Sánchez. 2005.** Sistemas de Información Orientados a la Toma de Decisiones: el enfoque multidimensional. [En línea] mayo-junio de 2005.

**Brualla, Cecilia Rigoni. 2008.** *Mejora del proceso en fábricas de software.* 2008.

**Cabrera, María .E Casales. 2007.** Data Warehouse (Almacenes de Datos). [En línea] 2007. http://hp.fciencias.unam.mx/~alg/bd/dwh.pdf.

**Castillo, Carlos. 2008.** *Tema 2. Sistemas gestores de bases de.* 2008.

**Castro, Dr. Rogelio S. Silverio. 2009.** *Papel de la Minería de Datos dentro de los Sistemas de Información Empresariales .* 2009.

**Castro, Raúl. 2009.** Discurso pronunciado el 26 de julio de 2009 . [En línea] 2009. http://www.bohemia.cu/2009/07/26/noticias/raul-castro-discurso-26.html.

**Cecchet, Emmanuel. 2009.** Building PetaByte Warehouses with PostgreSQL. [En línea] Mayo de 2009. http://www.pgcon.org/2009/schedule/attachments/135_PGCon%202009%20-%20Aster%20v6.pdf.

**Cesares, Claudio. 2009.** *Datawarehousing.* 2009.

**Corporation, Oracle. 2010.** Oracle. [En línea] 17 de Abril de 2010. http://www.oracle.com/corporate/princing/technologi-princ-list.pdf.

**Cortes, Angel R. 2008 .** *Parallel Structure and Performance.* 2008 .

**Coruña, A. 2007.** ¿Qué es Business Intelligence? [En línea] 2007. http://www.sinnexus.com/business_intelligence/index.aspx.

**Cursada. 2009.** Introducción al Data Warehouse,Olap y Minería de datos. [En línea] 2009.

**D'Ottone, Lic. Rosana y Boccaloni, Lic. Gustavo. 2009.** Considerando el particionamiento. [En línea] 2009. http://www.ixora.com.au/tips/design/partitioning.htm.

**Davenport, Thomas. 2009.** Ejecutivos consideran a LA INTELIGENCIA DE NEGOCIO (BI) y a la inteligencia analítica como una ventaja competitiva crucial. [En línea] 2009.

**Escorial, Jorge Salamanca. 2006.** *La Norma ISO/IEC 9126 y los modelos de calidad Boehm y Mc Call. 2006.* 2006.

**González, Erika Vilches y Broitman, Iván A. Escobar. 2007.** *Minería de Datos.* 2007.

**Guerrero, Eduardo Leyton. 2006.** *Calidad de componentes de software. ISO 9126.* 2006.

**Inocencio, Beatriz Canales. 2004.** Nueva Economía, Internet y tecnología. [En línea] Julio de 2004. http://www.gestiopolis.com/canales2/gerencia/1/busint.htm.

**John Worsley, Joshua Drake. 2009.** Practical PostgreSQL. [En línea] 2009. www.postgresql.org/docs/awbook.html.

**Kafati, Elizabeth Gutierrez. 2008.** Datawarehouse y sus principales características. [En línea] 2008.

**Kimball, Ralph. 2002.** *The Data Warehouse Toolkit (2nd edition).* 2002.

**Larrain, Carlos Hurtado. 2005.** *Repositorios (data warehouses) OLAP.* 2005.

**López, José Manuel Molina y Herrero, Jesús García. 2004.** *TÉCNICAS DE ANÁLISIS DE DATOS.* 2004.

**Martínez, Rafael. 2009-2010.** Sobre PostgreSQL. [En línea] 2009-2010. http://www.postgresql-es.org/sobre_postgresql.

**Mendez, A., y otros. 2004.** Fundamentos de Data Warehouse. [En línea] 2004.

**Mira, José Joaquín, y otros. 2006.** *La Gestión por procesos.* s.l. : Universidad Miguel Hernández, 2006.

**Momjian, Bruce. 2008.** PostgreSQL: Introduction and Concepts. [En línea] 2008. http://developer.postgresql.org/todo.php http://archives.postgresql.org/pgsql-es-.

**Oracle. 2005.** *Administrator Guide Oracle 10g release 2.* 2005.

**—. 2006.** *Database Performance Tunning Guide 10g release 2.* 2006.

**—. 2005.** *Release Notes, 10g release 2.* 2005.

**Palacio, Juan. 2006.** Sinopsis de los modelos SW-CMM y CMMI. [En línea] 2006. http://www.navegapolis.net/files/articulos/sinopsis_cmm.pdf.

**Parra, Jaime G. Orjuela. 2007.** *Características de calidad para la arquitectura de software.* 2007.

**PostgreSQL, Comunidad Internacional de Desarrollo de. 2009.** *PostgreSQL 8.4.1 Documentation.* 2009.

**PostgreSQL, Comunidad Internacional de Desarrollo de y Pg_Foundry. 2005.** *PostgreSQL. Hardware performance tunning.* 2005.

**PostgreSQL, Comunidad Internacional de Desarrollo. 2010.** *PostgreSQL 9.0 beta Documentation.* 2010.

*Razones por las que invertir en Bussiness Intelligence.* **Research, gartner. 2008.** 2008.

**Rivera, Ricardo Mendoza. 2009.** BI Inteligencia de Negocios. [En línea] 2009. http://www.google.com.cu/imgres?imgurl=http://bp2.blogger.com/_TGCPAPvSmlc/SFYU-8CbxwI/AAAAAAAAATE/CuQDHwYyTwI/s400/RoadMap.png&imgrefurl=http://rimenri.blogspot.com/2008/06/kimball-cognos-rimenri-roadmap-bidw.html&usg=__JREqKZz7lrXX_J-P_G7Mj9xse8Y=&h=400.

**Ruggia, Dr. Ing. Raul. 2008.** Tecnologías de Business Intelligence y panorama general de TI en BPS . [En línea] 2008. http://webcache.googleusercontent.com/search?q=cache:15qWk0MnoFAJ:www.eurosocialfiscal.org/uploads/documentos/20080529_170556_Eurosocial-BI%26TI-BPS(RR).ppt+BI+tecnolog%C3%ADas&cd=3&hl=es&ct=clnk&gl=cu.

**Serrano, Manuel, y otros. 2007.** *Una propuesta de un modelo conceptual de calidad de almacenes.* 2007.

**Torres, Liudmila Padrón. 2006.** Almacenes de datos: Importancia en el estándar. [En línea] Enero de 2006. http://www.mailxmail.com/curso-almacenes-datos-importancia-estandar/almacenes-datos-definicion.

**Valladolid, Dpto. Informática-ETSII-U. 2009.** *Evaluación y explotación de sistemas informáticos. Introducción a la evaluación del rendimiento.* 2009.

**Vallejos, Sofía J. 2006.** Minería de Datos. [En línea] 2006. http://exa.exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Mineria_Datos_Vallejos.pdf.

**Valmaseda, Ing. Marcos Luis Ortiz. 2010.** Netezza Performance Server: Lider en desarrollo de DWH y Analytics DBs. [En línea] 19 de Marzo de 2010. http://personas.grm.uci.cu/+marcos/?p=45#more-45.

**Velazco, Roberto Hernando. 2006.** Almacenes de datos (Datawarehouse). [En línea] 2006. http://www.rhernando.net/modules/tutorials/doc/bd/dw.html.

**Villa, Ing. Madelys Cuesta y López, Msc. Asnioby Hernández. 2009.** *METODOLOGÍA PARA EL DESARROLLO DE SOLUCIONES DE ALMACENES DE DATOS E INTELIGENCIA DE NEGOCIO EN CENTALAD.* 2009.

**Villanueva, Wladimiro Díaz. 2006.** Alamcenes de datos. [En línea] 2006. http://informatica.uv.es/iiguia/DBD/Teoria/data-warehouses.pdf.