

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**

**POSTGRADO EN INFORMÁTICA**

**MAESTRÍA EN INGENIERÍA DE SOFTWARE**



**TESIS DE MAESTRIA**

**METODOLOGÍA PARA LA MIGRACIÓN DE FORMA SEGURA DE  
SISTEMAS DE GESTIÓN DE BASES DE DATOS RELACIONALES A  
SOFTWARE LIBRE Y ESTÁNDARES ABIERTOS**

Caso de estudio: Instituciones y Entidades Públicas del Estado Plurinacional de  
Bolivia

**POSTULANTE:** Lic. José Arturo Celis Rioja

**TUTOR:** Ing. M.Sc. Sergio Antonio Toro Tejada

**La Paz, Bolivia**

**2015**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

# ***INDICE GENERAL***

Dedicatoria.....	VI
Agradecimientos .....	VII
Resumen.....	VIII
Abstract.....	IX
Introducción .....	X
Capítulo I: Marco del problema .....	1
1.1 Estado del arte .....	1
1.2 Planteamiento del problema.....	5
1.3 Formulación del problema de investigación .....	5
1.4 Planteamiento de objetivos.....	6
1.5 Planteamiento de hipótesis .....	6
1.6 Diseño metodológico.....	8
1.7 Delimitación.....	9
Capítulo II: Marco teórico .....	11
2.1 Referencia teórica.....	11
2.2 Desarrollo conceptual.....	17
2.3 Marco legal o institucional.....	19
Capítulo III: Situación y factibilidad.....	21
3.1 Uso actual de bases de datos .....	21
3.2 Capacidades.....	27
3.3 Licencias.....	31
3.4 Interpretación de resultados .....	34
3.5 Propuesta de factibilidad .....	36
3.6 Manejo de migraciones no factibles.....	41
Capítulo IV: El proceso de migración .....	43
4.1 Ambientes de trabajo.....	43
4.2 La Migración .....	48
4.3 Pruebas de las aplicaciones .....	52
4.4 Roles y responsabilidades .....	54
4.5 Herramientas a desarrollar .....	57
Capítulo V: Marco de resultados .....	61
5.1 Estado de los objetivos.....	61
5.2 Estado de la hipótesis .....	62
5.3 Conclusiones y recomendaciones.....	63
Bibliografía .....	67
Anexos .....	69
Apéndices.....	87
Glosario.....	100

## **INDICE DE CUADROS**

Cuadro 1, Tipos de SGBD utilizados.....	22
Cuadro 2, SGBD propietarios utilizados. ....	22
Cuadro 3, Soporte técnico a SGBD. ....	22
Cuadro 4, SGBD sin soporte técnico. ....	23
Cuadro 5, SGBD propietarios sin soporte técnico. ....	23
Cuadro 6, Tipos de sistemas de información. ....	24
Cuadro 7, Lenguajes de programación utilizados.....	24
Cuadro 8, Lenguajes de programación propietarios utilizados.....	24
Cuadro 9, Lenguajes de programación sin soporte técnico. ....	25
Cuadro 10, Lenguajes de programación propietarios sin soporte técnico. ....	25
Cuadro 11, Tipo de servidor de aplicaciones utilizados. ....	25
Cuadro 12, Servidores de aplicaciones utilizados.....	26
Cuadro 13, Soporte técnico en servidor de aplicaciones ....	26
Cuadro 14, Servidor de aplicaciones sin soporte técnico ....	26
Cuadro 15, Frecuencia de información en Gb .....	27
Cuadro 16, Capacitación en los dos últimos años.....	27
Cuadro 17, Capacitación de los funcionarios de sistemas en TIC .....	28
Cuadro 18, Tipo de financiamiento de la capacitación.....	28
Cuadro 19, Nivel del área de sistemas en el organigrama. ....	28
Cuadro 20, Tipo de servidores utilizados. ....	29
Cuadro 21, Tipos de sistemas operativos utilizados. ....	29
Cuadro 22, Tipos de sistemas operativos propietarios.....	29
Cuadro 23, Soporte técnico a sistemas operativos.....	30
Cuadro 24, Tipos de sistemas operativos sin soporte técnico.....	30
Cuadro 25, Sistemas operativos propietarios sin soporte técnico.....	30
Cuadro 26, Licenciamiento de uso SGBD .....	31
Cuadro 27, SGBD sin licencia.....	31
Cuadro 28, SGBD propietarios sin licencias .....	32
Cuadro 29, Licenciamiento de sistemas operativos .....	32
Cuadro 30, Sistemas operativos propietarios sin licencia.....	33
Cuadro 31, Detalle de sistemas operativos propietarios sin licencias .....	33
Cuadro 32, Licenciamiento de lenguajes de programación .....	34
Cuadro 33, Propiedad o acceso al código fuente.....	37
Cuadro 34, Documentación de los sistemas de información .....	38
Cuadro 35, Documentación actualizada de sistemas de información.....	38
Cuadro 36, Plan de migración a SGBD de software libre y estándares abiertos .....	63

\*\* La Fuente de todos los cuadros es de las encuestas de diagnóstico e-gobierno DGGE, BID, REDES de Diciembre 2014

## ***INDICE DE FIGURAS***

Figura 1, Tipo de diseño. Fuente: Tamayo, El proceso de investigación científica .....	8
Figura 2, Desarrollo conceptual del marco teórico .....	18
Figura 3, Ambientes necesarios para la migración. ....	44
Figura 4, Ambiente de desarrollo para la migración.....	45
Figura 5, Ambiente de certificación para la migración.....	46
Figura 6, Pruebas unitarias realizadas por el desarrollador .....	53
Figura 7, Prueba de sistemas.....	54
Figura 8. Ejemplo de script de una tabla.....	59
Figura 9, Estructura de carpetas para la extracción de objetos .....	60
Figura 10, Estructura del servidor Oracle .....	87
Figura 11, Estructuras de memoria y procesos para acceder a la base de datos Oracle .....	88
Figura 12, Principales componentes del PostgreSQL.....	93
Figura 13, Conexión básica de PostgreSQL .....	94

## ***INDICE DE TABLAS***

Tabla 1, Variables de la hipótesis .....	7
Tabla 2, Definición conceptual de variables de la hipótesis .....	7
<i>Tabla 3, Definición operacional de las variables de la hipótesis.....</i>	<i>8</i>
Tabla 4, Resumen de interpretación de resultados.....	35
Tabla 5, Descripción de elementos o componentes de evaluación operativa .....	36
<i>Tabla 6, Descripción de elementos o componentes de evaluación técnica .....</i>	<i>39</i>
Tabla 7, Tiempo en horas por etapa del proceso de migración .....	40
Tabla 8, Ejemplo de base de datos para cálculo de tiempo .....	40
Tabla 9, Resultado de tiempos aplicando la Tabla 6 y 7 .....	40
Tabla 10, Costo de la migración por tipo consultor o funcionario, en dólares americanos.....	41
Tabla 11, Información general del Microsoft SQL Server .....	91
Tabla 12, Historial de versiones Microsoft SQL Server.....	92

## **INDICE DE ANEXO Y APENDICES**

Anexos .....	69
Anexo A: Cuestionario Gobierno, 001 - SISTEMAS .....	69
Anexo B: Lista de instituciones encuestadas .....	75
Anexo C: Factibilidad .....	77
Anexo D: Factibilidad Operativa .....	78
Anexo E: Factibilidad Técnica .....	79
Anexo F: Inventario del Servidor .....	80
Anexo G: Inventario de la Base de Datos .....	81
Anexo H: Modificación de objetos .....	82
Anexo I: Comparación de Objetos .....	83
Anexo J: Comparación de Objetos (Continuación) .....	84
Anexo K: Secuencia de Pasos Migración .....	85
Anexo L: Pruebas de Componentes .....	86
Apéndices.....	87
Apéndice A: Gestor ORACLE.....	87
Apéndice B: Gestor Microsoft SQL Server .....	91
Apéndice C: Gestor PostgreSQL.....	93
Apéndice D: Gestor MySQL.....	96

## ***INDICE DE ACRONIMOS***

ADO.NET, 10, 72, 100  
ADSIB, 20, 68  
ASP.NET, 24, 33, 73  
BCB, 64  
COBOL, X, 13, 63, 100  
COPLUTIC, 20, 68  
DBA, 13, 47, 52, 55, 65  
DBMS, IX, 11, 13  
DDL, 19, 55, 56, 91, 100  
DGE, 20, 67  
DML, 50, 55, 56, 90, 91, 100  
G2G, 2  
HTML, 58  
JAVA, 27, 57, 63, 73  
JODBC, 10, 72, 100  
JSP, 27, 73  
MOPSV, 20  
MPD, 20  
ODBC, 10, 72, 96, 100  
OLE DB, 10, 92, 100  
PHP, 27, 63, 73, 96  
RUA, 64  
SAFCO, X, 100  
SGBD, VIII, X, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 27, 28, 31, 32, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 65, 77, 78, 79, 82, 83, 85, 93, 100  
SICOES, X, 100  
SIDUNEA, 27  
SIGMA, X, 100  
SIIF, X, 63  
SIM, 5  
SIN, 64  
SIRAT, X, 100  
SP, 14, 80  
SQL, IV, 4, 5, 14, 19, 22, 23, 32, 48, 49, 58, 59, 68, 71, 89, 90, 91, 92, 93, 97, 100  
SSMA, 4  
TIC, 10, 21, 27, 28, 100  
T-SQL, 18, 37, 91, 100

\*\* Para encontrar la descripción de las abreviaturas, vea el Glosario de términos, al final del presente documento.

## *Dedicatoria*

Quiero dedicar el presente trabajo a tres personas, que me enseñaron a ser papá y por consiguiente me han convertido en una mejor persona, resultado de lo anterior es que hoy, soy más conocido como el papá de Kevin, Michelle o Danny:

Kevin Celis, futuro Psicólogo, por siempre mostrarme el camino de la sabiduría y temor a Dios, por sus sabios consejos, a través de sus distintas vivencias como el ajedrez y la universidad.

Michelle Celis, futura Diseñadora, por su cariño y comprensión a pesar de mi carácter. Y por su dedicación a su carrera que tanto le encanta. Diseño Gráfico y Comunicación Visual.

Danny Celis, futura bachiller, por alegrarme cada uno de mis días con sus travesuras e inquietudes.

## *Agradecimientos*

Sobre todas las cosas, quiero agradecer a Dios, por brindarme la salud, energía y fortaleza de cada día y no abandonarme nunca.

Agradezco a Yolanda Rioja Sandy, mi mamá, por todo su apoyo, cariño y comprensión en todas las etapas que me tocó vivir hasta ahora, a pesar de mis convicciones y carácter.

También quiero dar las gracias al M.Sc. Aldo Valdez, Coordinador del Postgrado en Informática, por darme la oportunidad de terminar mis estudios en el Postgrado de Informática de la UMSA.

No quiero olvidarme del Ing. M.Sc. Sergio Toro, mi tutor, quien prácticamente me alentó, empujo y aconsejo a realizar y terminar el presente trabajo. También van mis agradecimientos para el tribunal revisor conformado por el MIB Iván Mercado Lorberg y el Ph.D. Eduardo Delgado Morales, por sus sabios consejos y observaciones que ayudaron a enriquecer el presente trabajo de investigación.

Quiero agradecer a todo el personal administrativo del Postgrado en Informática, por su paciencia, consejos y guías para la elaboración de los trámites administrativos y otros similares que uno debe cumplir con la UMSA y el Postgrado de Informática.

Finalmente mis agradecimientos para Milton Berbetti, sin cuyo apoyo, confianza y paciencia no hubiera sido posible que adquiriera el conocimiento y practica de migración de SGDB necesario para la realización del presente trabajo, en importantes empresas del rubro bancario y telecomunicaciones tanto en Chile como en Bolivia. Gracias MilteX!!!!

## *Resumen*

En el presente trabajo de investigación se realiza un levantamiento de información de las principales instituciones y entidades del estado para determinar la situación actual del uso de bases de datos y como determinar la factibilidad de migrar de Sistemas de Gestión de Bases de Datos (SGBD) propietarios, licenciados o privativos a SGBD de software libre y estándares abiertos. También contiene métodos, técnicas y procedimientos que se pueden utilizar y facilitaran la migración a SGBD de software libre y estándares abiertos en las instituciones y entidades del Estado.

En el Capítulo I, se hace un estudio de situación (estado del arte) de las migraciones de bases de datos en la región, se formula y plantea el problema, se plantean los objetivos de la investigación, así como se plantea la hipótesis, se define el diseño metodológico que se seguirá el presente trabajo de investigación y se delimita los límites y alcances.

En el Capítulo II, se hace referencia al marco teórico que se utilizará. La referencia teórica basada en la seguridad de la información, bases de datos migraciones de bases de datos y pruebas de software. El desarrollo conceptual de la referencia teórica, así como el marco legal o institucional que se aplicará.

En el Capítulo III, se realiza el análisis de la situación actual de uso de Bases de Datos en las instituciones y entidades del Estado, así como se plantea la factibilidad operativa, técnica y económica que deberían cumplir las aplicaciones que utilizan Bases de Datos a ser migradas a software libre y estándares abiertos.

En el Capítulo IV, se plantea la metodología de migración a SGBD de software libre y estándares abiertos, que ambientes de trabajo son necesarios, los pasos que se deben seguir, así como inventario y categorización, el desarrollo de la migración, como migrar los datos de origen a destino, comparar los servidores origen y destino, y finalmente como realizar el control de cambios.

En el Capítulo V, se expone el estado de los objetivos y de la hipótesis, luego del desarrollo del presente trabajo, también se exponen las conclusiones y recomendaciones.

Palabras Claves: Migración, Migración de bases de datos, Migración a Software libre.

## *Abstract*

In the present research work gathering information from key institutions and state entities is performed to determine the current use of databases and how to determine the feasibility of migrating from Database Management Systems (DBMS) licensed or proprietary to FOSS DBMS and open standards. It also contains methods, techniques and procedures that can be used and facilitate DBMS migration to free software and open standards in institutions and entities.

In Chapter I, a study of situation (state of the art) of migration of databases in the region is made, is formulated and posed the problem, the objectives of the research are raised, and the hypothesis is proposed, defines the methodological design that this research will continue and the limits and scope is delimited.

In Chapter II, it is referring to the theoretical framework to be used. The theoretical reference based on information security, database migration database and software testing. The conceptual development of the theoretical framework and the legal and institutional framework that will apply.

In Chapter III, the analysis of the current situation of use of databases in institutions and state entities is performed as well as the operational, technical and economic feasibility should meet the applications that use databases to be arises migrated to free software and open standards.

In Chapter IV, the methodology DBMS migration to free software and open standards is proposed that work environments are necessary to perform the migration, the steps you must follow the migration as inventory and categorization, development of migration, such as migrating data, compare the source and destination servers, and finally as perform change control.

In Chapter V, the state of the aims and the hypothesis, after the development of this paper, we discuss the findings and recommendations are also presented.

Keywords: Migration, Migrating databases, migration to free software.

## *Introducción*

Es común en nuestra sociedad, la acelerada digitalización de la información que las instituciones públicas del Estado plurinacional de Bolivia generan o tratan. Todos nosotros estamos siendo parte de este proceso por ejemplo, cuando debemos escribir una carta lo hacemos previamente en un procesador de texto para posteriormente imprimirlo. Por lo que cada vez los datos o información que creamos son almacenados en formato digital, y buena parte de ellos jamás conocerán los soportes convencionales (por ejemplo papel) que durante toda la historia de la escritura han sido el depósito natural de la información.

Una consecuencia inmediata de lo anterior, es que los Sistemas de Gestión de Bases de Datos (SGBD) se han convertido velozmente en el almacén primordial en donde buscar y hallar la información que instituciones públicas y nosotros mismos precisamos, Por lo tanto en los SGBD existe información muy sensible que debe cumplir con las premisas de la seguridad de información: Confidencialidad, Integridad y Disponibilidad.

La información que almacenan las instituciones públicas en SGBD debe ser resguardada de forma segura para evitar que personas no autorizadas, por ejemplo, tengan acceso a esta información. La información es un recurso que, como el resto de los importantes activos que tiene una institución pública del Estado, tiene valor y por consiguiente debe ser debidamente protegida. Al mismo tiempo debe cumplir con las regulaciones y normativas (leyes, decretos y reglamentos) que definen el uso de los SGBD en las instituciones públicas del Estado.

Con la promulgación de la ley 1178 SAFCO, el país se ve en la necesidad de administrar de mejor manera la información proveniente de las instituciones, pues a partir de esta ley se transfiere la responsabilidad por la función pública a las instituciones, es así que se implementan sistemas de manejo de Archivos (antecesor de los SGBD), la primera experiencia que se tiene al respecto es en sistemas WANG administrada por COBOL, la cual luego se migro en menos de dos años a Unix-Infomix, implementándose el SIIF, Sistema Integrado de Información Financiera.

La buena experiencia lograda en la Contaduría General del Estado al migrar la administración de archivos que la realizaba en COBOL-WANG a un SGBD Unix-Infomix, es imitada por la Administración Tributaria la cual en primera instancia migra sus sistemas de gestión de archivos de COBOL-WANG a SGBD en Unix-Infomix para luego de tres años realizar una nueva migración de Unix-Infomix a Oracle, con lo que se demuestra que en el Estado Boliviano tiene la experiencia previa de realizar migraciones en periodos relativamente cortos, teniendo como únicos requisitos, el orden, el tener un norte claro y el tener planes adecuados de migración y recuperación de datos e información.

La implementación de SGBD en las instituciones públicas del Estado ha sido necesaria estos últimos 20 años. De pasar a procesar la información en archivos y administrarlos con COBOL, se pasó a la implementación de complejos sistemas de información como el SIGMA, SIRAT o el SICOES que almacena su información en SGBD como el Oracle.

# ***Capítulo I: Marco del problema***

## ***1.1 Estado del arte***

En los últimos años la tendencia del software libre y estándares abiertos se ha propagado por la región, a continuación se realiza un desglose de los países donde su gobierno ha implementado o ha intentado implementar a través de leyes y decretos el Software libre, finalmente y a continuación se presenta los trabajos realizados que se encontraron acerca de la migración de bases de datos y herramientas utilizadas.

### ***1.1.1 Cuba***

En 2005, Cuba inició su migración a software libre y plataformas de código abierto en general, después de que el Consejo de Ministros decidiera comenzar paulatinamente el traslado de las instancias de la Administración Central del Estado a la nueva tecnología.

El sistema operativo, llamado Nova, fue creado por estudiantes y profesores de la Universidad de las Ciencias Informáticas de La Habana (UCI) y su principal objetivo es facilitar el proceso de migración de Microsoft Windows a software libre<sup>1</sup>.

### ***1.1.2 Perú***

En el año 2001 en el Perú una iniciativa de los parlamentarios fue desestimada, por el entonces presidente Alejandro Toledo, quien era considerado el mejor amigo del principal fabricante de software. Posteriormente en el año 2005 no fue necesaria realizar una ley ya que la mayoría de las instituciones utilizaban software libre de propia iniciativa.

### ***1.1.3 Venezuela***

En diciembre del año 2004, el entonces Presidente Hugo Chávez, decide sacar adelante un decreto<sup>2</sup> que instaba a la administración pública a utilizar Software Libre en sus plataformas tecnológicas.

### ***1.1.4 Ecuador***

En el Ecuador su gobierno ha promulgado el uso del software libre, a través un Decreto<sup>3</sup> por la Presidencia de la República.

### ***1.1.5 Colombia***

Según la Asociación Colombiana de Ingenieros de Sistemas (ACIS)<sup>4</sup> realizo una encuesta a las empresas que usan software libre, el 63% utiliza Linux como sistema operativo, el 54% lo utiliza como servidor y el 27% lo usa en los computadores de escritorio.

---

<sup>1</sup> [https://es.wikipedia.org/wiki/Nova\\_\(sistema\\_operativo\)](https://es.wikipedia.org/wiki/Nova_(sistema_operativo))

<sup>2</sup> Decreto 3390 del año 2004.

<sup>3</sup> Decreto 1014, emitido el 10 de Abril del 2008.

<sup>4</sup> <http://www.elmundo.com/portal/resultados/detalles/?idx=176377#.Vi1E3djfPIU>

En cuanto al uso del software libre en el Estado colombiano falta mucho camino por recorrer, principalmente por el temor que existe al cambiar las plataformas tradicionales. Vale la pena destacar la decisión de la Superintendencia de Servicios Públicos de migrar a software libre. Este proyecto se llevó a cabo con todas las formalidades de licitaciones, con resultados de contexto social como es la liberación bajo la licencia GNU/GPL del software de Gestión Documental y de Procesos llamado Orfeo.

### ***1.1.6 Chile***

En Chile no existe por el momento una ley o decreto que norme o reglamente el uso de software libre en las instituciones del estado. El diputado Mirosevic, del Partido Liberal de Chile, pensaba presentar un proyecto de ley referente al software libre y promover su utilización por parte del Estado. El proyecto presentado por el diputado Mirosevic fue suscrito también por otros nueve legisladores que buscaban que el Estado tome en cuenta la alternativa del software libre frente al software de licencia, por el cual se pagan más de 60 millones de dólares al año. Conscientes de la amenaza sobre sus intereses, los representantes de Microsoft acudieron a entrevistarse con Mirosevic a fin de persuadirlo de bajar esta moción.

### ***1.1.7 Argentina***

Desde el año 2000 en Argentina existieron varias iniciativas para promocionar o priorizar el software libre en el estado Argentino, pero hasta la fecha solo quedo en ante proyecto de ley, esto debido a las presiones que los legisladores recibieron de todo tipo, de los gigantes de la industria de software propietaria.

### ***1.1.8 Brasil***

Software Libre en la Administración pública de Brasil se inició durante la primera presidencia de Luis Ignacio Lula da Silva (2003-2006). El decreto<sup>5</sup>, en el cual el presidente amparado en las facultades que le confería la constitución, decretaba la creación de comités técnicos establecidos por el Comité Ejecutivo de Gobierno Electrónico<sup>6</sup>, con el propósito de coordinar y promover la planificación y ejecución de proyectos y acciones en sus respectivos ámbitos de competencia, con los siguientes nombres:

- Aplicación del Software Libre
- Inclusión Digital
- Integración de Sistemas
- Sistemas heredados y licencias de software
- Gestión de Sitios y Servicios On-line
- Infraestructura de red
- Gobierno para Gobierno – G2G
- Gestión del Conocimiento y de Información Estratégica.

La adopción de estas acciones cubre todo el ámbito nacional, sus 26 estados federados, más el Distrito Federal, divididos en 5564 municipios. El gobierno del presidente Lula, comenzó en sus primeros meses, realizando modificaciones importantes en el programa de gobierno electrónico

---

<sup>5</sup> Decreto del 29 de octubre del año 2003.

<sup>6</sup> Creado por el Decreto del 18 de octubre del año 2000.

brasileño, coordinadas por el ministro jefe de la Casa Civil, José Dirceu. Se crearon dos cámaras técnicas: la Cámara Técnica de Implementación de Software Libre y la de Inclusión Digital.

### ***1.1.9 Metodología para la migración de datos***

El año 2012 se elaboró en la Universidad Técnica de Ambato - Ecuador, el trabajo de investigación titulado “Metodología para la migración de datos que permita asegurar y conservar la integridad y consistencia de la información administrada por la empresa vsystems”<sup>7</sup>. El proceso de migración, tiene como origen el SGBD Oracle10g y como destino al SGBD PostgreSQL. Se utiliza la herramienta de migración de software libre Ora2Pg que facilita la migración de Oracle a PostgreSQL interviniendo en el proceso de extracción y carga de información. También se realiza el diseño y ejecución de una metodología específica para migración de datos que permite solucionar algunos de los inconvenientes como:

- Incoherencia en la información que atraviesa un proceso de migración de datos.
- Inconsistencia en la base de datos.
- Reducción de tiempo de depuración y activación de la base de datos.

Tiene como objetivo general. Diseñar una metodología de migración para la empresa VSYSTEMS que permita la conservación de la integridad y consistencia de la información. Y como objetivos específicos:

- Desarrollar una guía técnica para ejecutar el proceso de la migración.
- Asegurar la integridad de la información.
- Garantizar la consistencia de la información luego de pasar por un proceso de migración.

Se realiza un análisis de factibilidad operativo, técnico y económico. También se realiza un análisis y descripción de las diferentes herramientas que se pueden utilizar para migrar un SGBD, lo que no queda muy claro en este trabajo, es como se realizaron las pruebas de software del aplicativo o sistema afectado por la migración.

### ***1.1.10 Servicios de migración estrategica***

STRATEGY 7 Migration Services<sup>8</sup>, por su nombre en inglés utiliza la palabra migración para describir procedimientos y métodos empleados en mover datos y programas a un nuevo SGBD (nuevo servidor). Los procedimientos y métodos se utilizan para resolver con éxito las diferencias operacionales y posibles incompatibilidades entre la plataforma origen y la plataforma destino. Cada método o proceso apunta a una incompatibilidad o problema conocido.

El primer paso en resolver un problema es detectando su existencia en el código fuente de las aplicaciones o en los datos, este descubrimiento de problemas se realiza generalmente a través de herramientas especializadas de migración, que buscan en el código fuente de los programas o en

---

<sup>7</sup> Susana Beatriz Caraguay Martínez, Metodología para la migración de datos que permita asegurar y conservar la integridad y consistencia de la información administrada por la empresa vsystems, Universidad Técnica de Ambato Ecuador, Septiembre del 2012.

<sup>8</sup> STRATEGY 7 Migration Services, <http://www.s7.com/S7-Migration-Services.php>

los datos por un problema determinado. Una vez detectado, otras herramientas se encargan de convertir el código fuente o los datos a formas que son aceptables por el SGBD destino.

Cuenta con una metodología de migración compuesta por 6 fases que se detallan a continuación:

- Mapeo, Tiene la siguientes actividades:
  - Análisis de mapeo de datos
  - Análisis de integridad
  - Diseño detallado, descripción vías de extracción
  - Determinar medios de transformación
  - Determinar reglas de validación
  - Documentos de requerimientos
- Diseño. Consta de las siguientes actividades:
  - Estrategia para ejecutar la migración
  - Determinar configuración de software
  - Estrategias para pruebas
  - Estrategias de aseguramiento de calidad
  - Estrategias para certificación
- Diseño detallado. Consiste en las siguientes actividades:
  - Crear proyecto de migración detallado
  - Determinar estándares a usarse
  - Desarrollar especificación de programas
  - Desarrollar planes y benchmark
  - Diseñar funciones, excepciones, manejo, logeos, etc
- Construcción. Está compuesta por las siguientes actividades:
  - Comunicar estándares de programación
  - Ejecutar motor de migración (herramienta de migración). Extracción, transformación, validación y carga
  - Pruebas de unidad
- Test y deploy. Tienen las siguientes actividades:
  - Ejecutar plan de test
  - Realizar migraciones simuladas
  - Mejorar reglas de negocio

### ***1.1.11 Asistente de migración para servidores***

SQL Server Migration Assistant (SSMA)<sup>9</sup>, es la metodología utilizada por Microsoft para migrar bases de datos Oracle a Microsoft SQL Server. Es una herramienta gratuita que acelera el proceso de migración de bases de datos, migra las tablas, vistas, índices, procedimientos almacenados, funciones y triggers a Microsoft SQL Server desde bases de datos fuentes como: Oracle, Sybase, MySql y MS-Access. SSMA consiste en tres tareas:

- Crear el Reporte. Genera reportes para análisis, estadísticas de objetos de cada base de datos y genera una lista de problemas como características no soportadas que requieren intervención manual.

---

<sup>9</sup> <http://www.mssqltips.com/sqlservertip/2230/sql-server-migration-assistant-2008-ssma/>

- Convertir el esquema. Configuración de maqueo de esquemas y convierte objetos de base de datos a Microsoft SQL Server
- Migrar Datos. Realiza la migración de los datos a Microsoft SQL Server.

### ***1.1.12 Otros trabajos y artículos***

Otro trabajo de investigación similar se realizó en la Universidad Cooperativa de Colombia, Seccional Medellín, que titula “Metodología para la construcción de un migrador universal de bases de datos”<sup>10</sup>. Esta metodología solo considera la migración de los datos entre un SGBD origen y destino, no así otros objetos como vistas, índices, trigger, procedimientos almacenados, etc.

También en Colombia se elabora el “Plan de migración del Sistema de Información (SIM) de la Procuraduría General de la Nación”<sup>11</sup>, realizado por la empresa Synapsi en el año 2007, utilizando la herramienta Oracle Warehouse Builder, para migrar los datos de un origen basado en SGBD en Oracle y como servidor destino, también un SGBD Oracle, no se menciona si se migraron otros objetos (índices, trigger, vistas, procedimientos almacenados, etc.)

En México se realizó “Factores críticos de éxito en el proceso de migración de bases de datos relacionales”<sup>12</sup>, por Susana Laura Corona Correa, Universidad Nacional Autónoma de México, Dirección de Sistemas. Tienen una definición de migración de SGBD en un esquema de servidor origen y destino, que factores críticos de éxito se deben considerar para la migración basados en etapas: Planeación, Aprobación del plan, Análisis del esquema actual y nuevo, Selección y/o desarrollo de las aplicaciones para migrar, Pruebas, Migración, Resultados y Cierre de Proyecto.

## ***1.2 Planteamiento del problema***

Las instituciones y entidades públicas del estado tienen funcionando en sus servidores de SGBD para almacenar y utilizar información, por lo que se han visto enfrentados al problema de migrar sus SGBD licenciados a SGBD de software libre y estándares abiertos, sin la existencia de una metodología que guíe la migración.

## ***1.3 Formulación del problema de investigación***

Por lo planteado en la sección anterior el problema central es:

Existe una metodología, que determine si se realiza o no la migración de los SGBD licenciados a SGBD de software libre y estándares abiertos de forma segura?

Como consecuencia del problema central surgen los problemas específicos:

---

<sup>10</sup> Javier Fernandez y Juan Camilo Giraldo, Metodología para la construcción de un migrador universal de bases de datos, Universidad Cooperativa de Colombia, Seccional Medellín, Mayo 2010.

<sup>11</sup> Synapsi, Plan de Migración del Sistema de Información de la Procuraduría General de la Nación, Colombia, 2007.

<sup>12</sup> Corona Correa Susana Laura, Factores críticos de éxito en el proceso de migración de bases de datos relacionales, Universidad Nacional Autónoma de México, Dirección de Sistemas.

- Es altamente dificultoso determinar la factibilidad de la migración de los SGBD licenciados a SGBD de software libre y estándares abiertos, en las instituciones y entidades públicas del estado.
- Los métodos, procedimientos y técnicas existentes son limitados para realizar la migración de forma segura de los SGBD licenciados a SGBD software libre y estándares abiertos en las instituciones y entidades públicas del estado.
- Los métodos, técnicas y procedimientos existentes son limitados para probar y validar las aplicaciones que utilizan los SGBD que se van a migrar a software libre y estándares abiertos.
- Las definiciones y descripción de roles y responsabilidades de los participantes en el proceso de migración de los SGBD licenciados a SGBD de software libre y estándares abiertos, son insuficientes.
- Las herramientas actuales son insuficientes para soportar el proceso de migración de los SGBD licenciados a SGBD de software libre y de estándares abiertos.

## ***1.4 Planteamiento de objetivos***

### ***1.4.1 Objetivo general***

Diseñar una metodología, que determine la factibilidad para realizar la migración de forma segura de los SGBD licenciados a SGBD de software libre y estándares abiertos en las instituciones y entidades públicas del Estado.

### ***1.4.2 Objetivos específicos***

- Determinar la factibilidad de la migración de los SGBD licenciados a SGBD de software libre y estándares abiertos, en las instituciones y entidades públicas del estado.
- Diseñar y describir métodos, procedimientos y técnicas que permitan realizar la migración de forma segura de los SGBD licenciados a SGBD software libre y estándares abiertos en las instituciones y entidades públicas del estado y otras que así lo consideren.
- Diseñar y describir métodos, técnicas y procedimientos que permitan probar y validar las aplicaciones que utilizan los SGBD que se van a migrar a software libre y estándares abiertos.
- Definir y describir roles y responsabilidades de los participantes en el proceso de migración de los SGBD licenciados a SGBD de software libre y estándares abiertos.
- Establecer un marco de trabajo para la elaboración de herramientas que soporten el proceso de migración de los SGBD licenciados a SGBD de software libre y de estándares abiertos.

## ***1.5 Planteamiento de hipótesis***

De acuerdo al planteamiento general y específicos descritos se plantea la hipótesis del presente estudio.

“El Diseño de una metodología, para realizar la migración de los SGBD licenciados a SGBD de software libre y estándares abiertos en las instituciones públicas del Estado, aumentara la factibilidad con aseguramiento de los datos.”

Tabla 1, Variables de la hipótesis

Variables	Descripción
Independiente	El Diseño de una metodología
Dependiente	La migración
Interviniente	La factibilidad con aseguramiento de los datos

La definición de conceptos relacionados con las variables de la hipótesis se describe en la siguiente tabla.

Tabla 2, Definición conceptual de variables de la hipótesis

	Variables	Definición de Conceptos
<b>Independiente</b>	El Diseño de una metodología	<b>Metodología.</b> Es la aplicación de métodos, procedimientos y técnicas para lograr algo. En este caso la migración.
<b>Interviniente</b>	La factibilidad con aseguramiento de los datos	<b>Factibilidad.</b> Es medir la eficacia y eficiencia de realizar algo, generalmente se la mide en base a la factibilidad operativa, técnica y económica. La seguridad de los datos están definidos por el concepto de: Confidencialidad, integridad y disponibilidad de los datos en el proceso de migración.

La definición operacional de las variables independientes e interviniente se puede observar en la Tabla 3, a continuación.

Tabla 3, Definición operacional de las variables de la hipótesis

	Variables	Definición Operacional
<b>Independiente</b>	El Diseño de una metodología	<p>La metodología se medirá por la existencia de:</p> <ul style="list-style-type: none"> <li>• Métodos. Es un conjunto de operaciones ordenadas con que se pretende obtener un resultado.</li> <li>• Procedimientos. Es la forma de realizar una actividad. Forma de decir o hacer algo con orden.</li> <li>• Técnicas. Habilidad y conocimientos especiales de realizar una actividad con determinado recurso.</li> </ul>
<b>Interviniente</b>	La factibilidad con aseguramiento de los datos	<p>La factibilidad se medirá de la siguiente forma:</p> <ul style="list-style-type: none"> <li>• Operativa. Si existe el código fuente de los programas que utilizan los SGBD, si tienen la documentación de los sistemas y programas, el nivel decisional del área de sistemas y la capacidad del personal para la migración a SGBD de software libre y estándares abiertos.</li> <li>• Técnica. Se realizará inventario a dos niveles: de servidor y de base de datos.</li> <li>• Económica. No debería ser un factor que influya en la decisión de migrar a SGBD de software libre y estándares abiertos, debido a la Ley No 164.</li> </ul>

## 1.6 Diseño metodológico

El diseño y estructura a seguir en el presente trabajo, será el sugerido por Tamayo<sup>13</sup>, que propone realizar el diseño de la investigación basado en el diseño bibliográfico y diseño de campo.

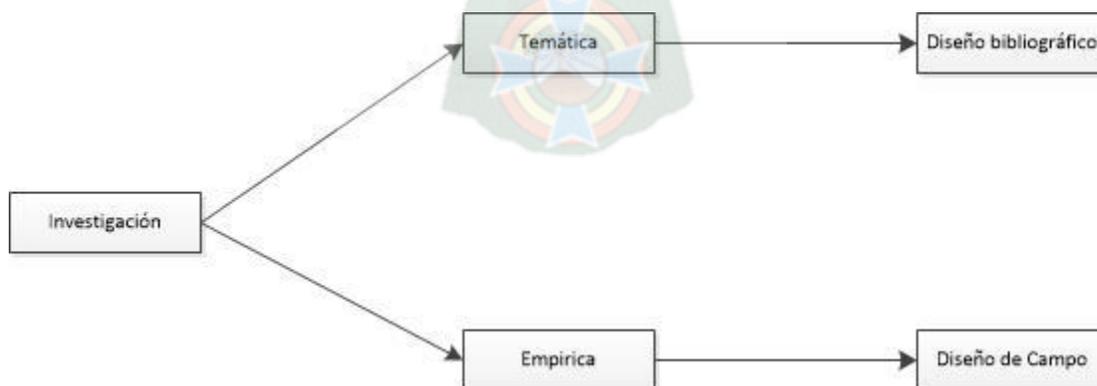


Figura 1, Tipo de diseño. Fuente: Tamayo, El proceso de investigación científica

<sup>13</sup> TAMAYO Mario, El proceso de la investigación científica, Quinta Edición, pp.113

### **1.6.1 Diseño bibliográfico**

El uso de datos secundarios, es decir aquellos trabajos y estudios obtenidos por otras personas, se realizará al buscar y revisar la bibliografía a través de internet y bibliotecas especializadas que traten el tema de migración de SGBD. En ese sentido se realizará un estudio de los trabajos definidos en la sección 1.1 Estado del arte, específicamente:

- Metodología para la migración de datos, definido en la sección 1.1.6.
- Servicios de migración estratégica, en la sección 1.1.7.
- Asistente de migración para servidores, de la sección 1.1.8.
- Otros trabajos y artículos, definidos en la sección 1.1.9.

### **1.6.2 Diseño de campo**

Para recoger los datos directamente de la realidad, se realizarán encuestas dirigidas a funcionarios públicos de instituciones y entidades del Estado, muestra representativa, así como entrevistas en profundidad que permitan obtener mayor información. En resumen:

- Se desarrollara el análisis de información
- Se recolectará información a través de cuestionarios y entrevistas.
- Se aplicaran herramientas cualitativas como entrevistas en profundidad a funcionarios y autoridades relacionadas con el tema.
- Se evaluaran diversos documentos referenciales:
  - Planes de desarrollo Informático.
  - Presupuestos para las áreas de modernización.
  - Planes de migración y otros referidos a la tecnología.

Trabajo a ser desarrollado con autoridades; profesionales y técnicos, fundamentalmente de las áreas unidades de planeación y sistemas.

## **1.7 Delimitación**

En esta sección de acuerdo al consejo del Dr. Escalera<sup>14</sup>, se realizará una delimitación del presente estudio, donde se definirá los límites y alcances del presente estudio.

### **1.7.1 Límites**

No incluye la decisión de migrar a SBGD de software libre y estándares abiertos. El presente trabajo es una guía que ayudará a tomar la decisión de migrar o no a SGBD de software libre y estándares abiertos.

Tampoco incluye la migración, si bien se diseñarán métodos, procedimientos y técnicas para realizar la migración, el presente documento no es un manual paso a paso de cómo realizar la migración, ya que en este caso se deben considerar los aspectos particulares de cada institución y

---

<sup>14</sup> ESCALERA S., Técnicas de Investigación y Desarrollo en Ciencias y Tecnologías, Segunda Edición, pp.278

entidad del Estado, como por ejemplo que SGBD utilizan y la forma en como administran, manejan y explotan su SGBD.

No incluye el plan de migración a SGBD de software libre y estándares abiertos, ya que cada institución y entidad del Estado tiene sus propias (como se mencionó anteriormente) características y particularidades. La elaboración del plan de migración a SGBD de software libre y estándares abiertos debe ser elaborada por cada institución y entidad del Estado.

No incluye los pasos y recomendaciones de instalaciones y configuraciones de los servidores que serán utilizados en el proceso de migración en los diferentes ambientes que serán necesarios.

### ***1.7.2 Alcances***

El presente trabajo se realizará considerando una muestra representativa de las instituciones y entidades públicas del Estado ubicada en la sede de gobierno, a través de cuestionarios y entrevistas donde se evaluara fundamentalmente:

- El uso de SGBD si es comercial, propietario o de software libre y estándares abiertos. De que fabricante/proveedor fue adquirido la licencia.
- El volumen de información que se maneja (Anual, mensual, semanal y diario).
- Si se tiene acceso al código fuente de las aplicaciones que utilizan los SGBD a migrar.
- Existe documentación de las aplicaciones que se conectan a los SGBD a migrar.
- El tipo de conectividad utilizan (ODBC, JODBC, OLE DB, ADO.NET, etc.).
- Existencia y el nivel en el organigrama del área responsable de los Sistemas o TIC.
- Análisis de capacidades actuales del personal que trabaja en las áreas de Sistemas o TIC.
- Análisis y tabulación de información recolectada.

Juntamente con el análisis de la información recolectada por cuestionarios y entrevistas, se realizará el diseño de una metodología que en una primera etapa determine la factibilidad de realizar la migración a SGBD de software libre y estándares abiertos. Y si la migración es factible se utilizarán métodos, técnicas y procedimientos para realizar la migración a SGBD de software libre y estándares abiertos.

## ***Capítulo II: Marco teórico***

### ***2.1 Referencia teórica***

#### ***2.1.1 Seguridad de la información***

Las instituciones públicas y privadas consideran a la información como un recurso o activo intangible de igual o mayor importancia que sus demás activos comerciales, por esta razón cada vez más instituciones u organizaciones se preocupan por proteger debidamente su información.

La seguridad de la información se define como la preservación de las siguientes características<sup>15</sup>:

- **Confidencialidad:** la información debe ser accesible solo por personas debidamente autorizadas.
- **Integridad:** garantizar la exactitud y totalidad de la información.
- **Disponibilidad:** que las personas debidamente autorizadas accedan a la información y a sus recursos relacionados toda vez que se necesite.

Proteger o asegurar la información se puede lograr aplicando un conjunto adecuado de controles, que van desde políticas, normas, procedimientos, etc. La aplicación de estos controles, garantizan que se logren los objetivos de la seguridad de la información.

#### ***2.1.2 Bases de datos***

Las Bases de Datos son básicamente un conjunto de información almacenada en algún medio físico (hardware) que puede ser utilizado por usuarios o aplicaciones computarizadas a través de una interfaz (software) llamada Sistema de Gestión de Base de Datos (SGBD) o por sus siglas en inglés DBMS.

Los usuarios de las bases de datos pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo:

- Agregar nuevos archivos vacíos a la base de datos.
- Insertar datos en los archivos existentes.
- Recuperar datos de los archivos existentes.
- Modificar datos en archivos existentes.
- Eliminar datos de los archivos existentes.
- Eliminar archivos existentes de las bases de datos.

---

<sup>15</sup> Norma ISO IEC 17799

Otra definición de base de datos se puede encontrar en Date<sup>16</sup>, donde se considera a la base de datos como “un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones o consultas”.

Una base de datos tiene cuatro componentes principales que son:

- Datos,
- Hardware,
- Software y
- Usuarios

A continuación pasaremos a definir el o los componentes que serán objeto de estudio e investigación en el presente trabajo.

### ***Datos***

En la actualidad existe mucha confusión entre el termino datos e información, algunos autores como por ejemplo Date, consideran a los datos e información como un sinónimo, mientras otros consideran a los datos como aquello que está almacenado en la base de datos e información para hacer referencia al significado de esos datos como lo entendería algún usuario.

En el presente trabajo haremos referencia a los datos como la unidad básica de información que luego de ser transformada (proceso) se convierte en información de utilidad para algún usuario. Cabe hacer notar que al mismo tiempo la información se puede convertir en dato, si la información es utilizada como insumo o entrada de otro proceso de transformación o es utilizada como retroalimentación para el mismo proceso.

### ***Hardware***

El hardware de una base de datos es todo el medio físico que es utilizado para el almacenamiento de los datos, está formado por:

- Los volúmenes de almacenamiento secundarios, que son los discos duros que se utilizan para contener a los datos, juntamente con los dispositivos asociados de Entrada y Salida (unidades de discos), los controladores de dispositivos, los canales de E/S entre otros.
- Los procesadores de Hardware y la memoria principal asociada utilizados para soportar el SGBD.

### ***Software - Sistema de Gestión de Base de Datos***

Entre la base de datos física y los usuarios o aplicaciones, existe una capa de software, conocida como el SGBD. Todas las solicitudes de acceso a la base de datos son manejadas por el SGBD, así como las distintas operaciones que se quieren realizar. El SGBD es un software especializado,

---

<sup>16</sup> DATE C. J., Introducción a los sistemas de Bases de Datos.

que permite a los usuarios y aplicaciones agregar, recuperar, modificar y eliminar información de la base de datos. El SGBD sirve de interfaz entre el componente hardware, los usuarios y aplicaciones que necesitan utilizar la información contenida en la base de datos. Se compone de un lenguaje de:

- Definición de datos,
- Manipulación de datos, y
- Un lenguaje de consulta.

Los términos SGBD y DBMS (por sus siglas en inglés), son equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Base de Datos y DataBase Management System.

El propósito general de los SGBD es el de administrar y manejar de manera clara, sencilla y ordenada un conjunto de información almacenada en una base de datos.

### ***Usuarios***

Se consideran tres clases de usuarios:

- Programadores de aplicaciones, que son responsables de escribir los programas o aplicaciones de base de datos en algún lenguaje de programación como COBOL, PL/I, C++, Java o algún lenguaje de alto nivel de cuarta generación.
- Usuarios finales, quienes interactúan con la base de datos desde estaciones de trabajo o terminales. Un usuario final accede a la información de la base de datos, a través de un programa o aplicación mencionado en el párrafo anterior.
- Administrador de base de datos, o DBA (por sus siglas en inglés) tiene como función principal decidir qué datos se deberán almacenar en la base de datos y establecer políticas para mantener y manejar esos datos una vez almacenados. Un ejemplo de las políticas podría ser una que indicara quien puede realizar que operaciones sobre ciertos datos y bajo qué circunstancias. En otras palabras, una política de seguridad de los datos.

#### ***2.1.3 Actualización/Migración de base de datos***

La migración, según el diccionario de la lengua española<sup>17</sup>, está definido como la “Acción y efecto de pasar de un país a otro para establecerse en él. Se usa hablando de las migraciones históricas que hicieron las razas o los pueblos enteros. Desplazamiento geográfico de individuos o grupos, generalmente por causas económicas o sociales”, por lo que podemos concluir que la migración de base de datos es el proceso de llevar los datos y objetos asociados (Tablas, Vistas, Procedimientos Almacenados, Índices, Triggers, etc.) de un servidor origen o antiguo a un servidor destino o nuevo.

La migración de bases de datos es generalmente una tarea compleja que no sólo supone transferir datos entre tipos de almacenaje y formatos de un servidor origen de base de datos a otro servidor

---

<sup>17</sup> Diccionario de la lengua española (DRAE), Edición 22, 2001. [www.rae.es](http://www.rae.es)

destino, sino que también supone reescribir sentencias SQL o incluso procedimientos almacenados (SP) de lógica de negocio, con la finalidad de adaptarlos al nuevo servidor de destino.

Existen muchas estrategias de actualización y migración de SGBD. Algo común en todas las estrategias es la necesidad de instalar/ejecutar el software del SGBD en el que será el servidor destino o nuevo. Es una buena idea ejecutar más de una prueba del proceso de actualización y migración, antes de realizarlo en el ambiente de producción, por lo que será necesario contar con un ambiente de pruebas.

A continuación se detallan y describen las estrategias comunes y más utilizadas por los principales fabricantes de SGBD<sup>18</sup>.

### ***Actualización In-Place***

Probablemente la estrategia de actualización y migración más fácil de realizar. En este caso el servidor origen y el servidor destino, es el mismo servidor. Mientras el servidor cumpla con los requerimientos mínimos del nuevo SGBD, el instalador permitirá la migración o como se trata del mismo servidor se le suele llamar solamente actualización del SGBD.

La ventaja que tiene esta estrategia es:

- No se requiere nuevo hardware para el servidor destino. Los costos de Hardware de nuevo servidor son reducidos.
- No es necesario cambiar las cadenas de conexiones de las aplicaciones que apuntan a este SGBD.

Las desventajas que presenta son:

- Si bien se puede probar en un ambiente de prueba, el momento de realizarlo en el ambiente de producción no existe posibilidad de vuelta atrás. La única posibilidad de vuelta atrás incluye restaurar backups previos a la actualización del SGBD, lo cual dejaría al SGBD fuera de servicio, mientras se realiza la restauración.
- Es necesario una ventana de tiempo para realizar este proceso, que generalmente necesita bajar los servicios del SGBD.
- Se utiliza esta estrategia para actualizar y migrar a una edición/versión del SGBD superior a la que se está utilizando. Dependiendo de los requerimientos de cada proveedor/fabricante de SGBD.
- No se puede cambiar la plataforma o sistema operativo del servidor, como parte del proceso de actualización.

### ***Actualización lado a lado***

La idea de esta estrategia es sencilla y fácil, se debe instalar una nueva instancia del software de SGBD en el mismo servidor Lado a Lado, donde se está ejecutando la instancia antigua (origen)

---

<sup>18</sup> Los principales proveedores y fabricantes de SGBD son ORACLE y MICROSOFT.

que se quiere actualizar. Una vez instalada el software de SGBD de la nueva instancia y que este con sus servicios funcionando, se debe simplemente cambiar las bases de datos de usuarios desde la antigua instancia (origen) y cargarlas en la nueva instancia (destino) y conectar su aplicación a la nueva instancia.

En caso de encontrarse con problemas, se debe restaurar los backups de las bases de datos de usuarios a la instancia antigua y reconectar las aplicaciones. Una vez que se está seguro que la actualización Lado a Lado está bien, se puede desinstalar la instancia antigua.

Las ventajas que tiene esta estrategia de actualización es:

- No se requiere nuevo hardware para el servidor destino. Por lo que los costos de Hardware de nuevo servidor son reducidos.
- Se puede contar con un plan de vuelta atrás, utilizando la instancia antigua u origen.

Las desventajas de esta estrategia, son:

- Se debe cambiar la cadena de conexión de las aplicaciones que apuntan a la instancia antigua (origen), para que apunten a la nueva instancia (destino).
- Es necesario una ventana de tiempo para realizar este proceso, que generalmente necesita bajar los servicios del SGBD.
- No se puede cambiar la plataforma o sistema operativo del servidor, como parte del proceso de actualización.

### ***Actualización por migración***

Es la forma ideal para realizar una actualización, básicamente se debe proporcionar un nuevo servidor (destino) y realizar la instalación del software del SGBD, restaurar los backups de las bases de datos, reconectar las cadenas de conexión de las aplicaciones al nuevo servidor, si se encuentran problemas, se puede volver atrás utilizando el servidor antiguo (origen), hasta que se resuelvan los problemas.

Esta estrategia es más limpia, ya que asegura que solo estén instalados los bits del nuevo software de SGBD, de esta forma los administradores de SGBD no deben preocuparse por mantener los bits antiguos del software de SGBD. Existe toda clase de trucos que se pueden utilizar para permitir que la migración al nuevo servidor (destino) sea lo más rápido posible y minimice al mismo tiempo el tiempo de parada del servicio del SGBD mientras se realiza la migración.

La ventaja de esta estrategia es:

- Se puede contar con un plan de vuelta atrás, utilizando el servidor antiguo u origen.
- Se puede minimizar el tiempo que deja de proporcionar servicios el SGBD antiguo u origen.
- Se puede cambiar la plataforma o sistema operativo del servidor destino. Aprovechando que el Hardware es nuevo.

- Permite la consolidación de bases de datos en un solo servidor de SGBD, esto por contar un nuevo Hardware que seguramente será de mayor capacidad.

Las desventajas que tiene son:

- Necesita de inversión en hardware, ya que se debe proporcionar un nuevo servidor (destino) donde se migrará el SGBD.
- Se debe cambiar la cadena de conexión de las aplicaciones que apuntan al antiguo servidor (origen)
- Requiere de un ambiente de pruebas similar al de producción, donde se pruebe la migración de los datos y objetos asociados (Tablas, Vistas, Procedimientos almacenados, Índices, etc.), así como se debe probar el funcionamiento de las aplicaciones que utilizan las bases de datos.

#### ***2.1.4 Pruebas de software***

Las actividades fundamentales de las pruebas de software son: La prueba de componentes, se prueba una parte del sistema y la prueba de sistemas, donde se prueba el sistema como un todo<sup>19</sup>.

El propósito de la prueba de componentes es descubrir errores probando componentes de programas, que pueden ser funciones, objetos o componentes reutilizable. Durante las pruebas del sistema, estos componentes se integran para formar subsistemas o el sistema completo. Inevitablemente, los defectos en los componentes que no se han detectado durante la etapa de pruebas de componentes, se descubrirán en las pruebas de sistemas<sup>20</sup>.

El proceso de software de pruebas de software tiene dos objetivos que son distintos:

- Para demostrar que el software satisface los requerimientos. Esto significa que debería haber pruebas por cada requerimiento en el documento de requerimientos, En caso de tratarse de productos de software genérico, se debería realizar pruebas para todas las características del sistema que se incorporan en la entrega del producto.
- Para descubrir errores en el software. La prueba de errores está relacionada con la eliminación de todos los tipos de comportamiento del sistema no deseable, como caídas del sistema, interacciones no permitidas con otros sistemas, cálculos incorrectos y corrupción de datos.

#### ***Pruebas de componentes***

Las pruebas de componentes, también llamadas pruebas unitarias, es el proceso de realizar pruebas a los componentes individuales del sistema. Es un proceso de pruebas de defectos, por lo que su objetivo es encontrar errores en los componentes. Para la mayoría de los sistemas, los responsables de las pruebas de componentes, son los desarrolladores de componentes.

---

<sup>19</sup> SOMMERVILLE Ian, Ingeniería del Software

<sup>20</sup> SOMMERVILLE Ian, Cit. Op.

Existen tres tipos de componentes que pueden probarse en esta etapa:

- Funciones individuales o métodos dentro de un objeto.
- Clases de objetos que tienen varios atributos y métodos.
- Interfaces, Las prueba de interfaces es la prueba de componentes compuestos, formados por diferentes funciones y métodos. Estos componentes compuestos tienen una interfaz definida que se utiliza para acceder a su funcionalidad.

Las funciones o métodos individuales son el tipo más simple de componente y sus pruebas son un conjunto de llamadas a estas rutinas con diferentes valores de parámetros de entrada. Pueden utilizarse las aproximaciones para diseñar los casos de prueba y para diseñar las pruebas de las funciones o métodos.

### ***Pruebas de sistemas***

Las pruebas de sistemas, es la prueba de dos o más componentes que implementen funciones del sistema o características y seguidamente se prueba el sistema integrado. En un proceso de desarrollo iterativo, las pruebas del sistema se ocupan de probar un incremento que va ser entregado al cliente, en un proceso en cascada, las pruebas del sistema se ocupan de probar el sistema completo.

Para la mayoría de los sistemas complejos, existen dos fases distintas de pruebas del sistema<sup>21</sup>:

- Pruebas de integración. En estas pruebas el equipo que realizará las pruebas tiene acceso al código fuente del sistema. Cuando se descubre un problema, el equipo de pruebas intenta encontrar la fuente del problema e identificar los componentes que tienen que ser depurados. Las pruebas de integración se ocupan principalmente de encontrar errores en el sistema.
- Pruebas de entregas. En las que se realiza la prueba a una versión del sistema que podría ser entregada a los usuarios. El equipo de pruebas se ocupa de validar que el sistema satisface los requerimientos y asegurar que el sistema es confiable. Las pruebas de entregas son generalmente pruebas de caja negra en las que el equipo de pruebas se ocupa de demostrar que el sistema funciona o no correctamente. Los problemas son comunicados al equipo de desarrollo cuyo trabajo es depurar el sistema. Cuando los clientes se involucran en las pruebas de entregas, a menudo se denominan pruebas de aceptación. Si la entrega es lo suficientemente buena, el cliente puede entonces aceptarla para su uso.

## ***2.2 Desarrollo conceptual***

De la referencia teórica definida en la sección anterior, se procederá a definir en qué etapas se utilizarán o se desarrollarán los conceptos definidos para la migración a SGBD de software libre y estándares abiertos, vea la figura 2.

La idea principal es proporcionar un nuevo servidor, que será llamado destino, donde se realizará la instalación y configuración del software del SGBD de software libre y estándares abiertos.

---

<sup>21</sup> SOMMERVILLE Ian, Ingeniería del Software

Realizar la migración del o de las bases de datos, así como de todos sus objetos como tablas vistas, procedimientos almacenados, funciones, índices, triggers, etc. Reconectar las cadenas de conexión de las aplicaciones al nuevo servidor de SGBD de software libre y estándares abiertos, en caso de encontrarse problemas, se puede volver atrás utilizando el servidor antiguo o de origen, hasta que se resuelvan los problemas.



Figura 2, Desarrollo conceptual del marco teórico

Antes de realizar la migración en el ambiente de producción, se deben tener desarrollados como mínimos dos ambientes, estos ambientes deben ser lo más parecido posible al ambiente de producción en su servidor de origen:

- Ambiente de Desarrollo. En este ambiente se desarrollará la migración, se detectará incompatibilidad de sentencias T-SQL y se realizarán pruebas de migración por parte del equipo de desarrollo de migración.
- Ambiente de Certificación. Será utilizado por los usuarios que manejan los aplicativos o programas que acceden a los SGBD que se van a migrar, pero apuntando al SGBD migrado (destino), donde realizarán pruebas de componentes y de sistemas con el propósito de certificar que los aplicativos funcionan correctamente con el nuevo SGBD de software libre y estándares abiertos.

### 2.2.1 Desarrollo conceptual de seguridad de la información

La referencia teórica de la seguridad de la información, será utilizada para proteger debidamente la información que se está migrando de un SGBD propietario, comercial o privativo a un SGBD

de software libre y estándares abiertos, considerando que la seguridad de la información se define<sup>22</sup> como la preservación de la confidencialidad, integridad y disponibilidad de la información.

### **2.2.2 Desarrollo conceptual de bases de datos**

La base de datos son repositorios de información que son utilizados por los usuarios para almacenar y consultar información<sup>23</sup>, poseen el lenguaje de definición de datos, DDL por sus siglas en inglés, que ayudan a definir las tablas y objetos relacionados, así como a manipular los datos a través de un lenguaje de manipulación de datos y utilizan el lenguaje de consultas para recuperar o consultar información.

### **2.2.3 Desarrollo conceptual de actualización por migración**

La migración se realizará de un SGBD origen que será de tipo propietario, comercial o privativo a un SGBD de software libre y estándares abiertos, por lo que será necesario realizar la migración por actualización<sup>24</sup> utilizando un nuevo servidor que será denominado destino, en este servidor destino se realizará la instalación del software de gestión de base de datos (SGBD) y las configuraciones necesarias de acuerdo a las mejores prácticas del SGBD.

Posteriormente se desarrolla la migración, es decir se realizan pruebas de llevar los objetos a nivel servidor y base de datos al servidor destino, con la idea de detectar incompatibilidades de los motores de SGBD, anotarlos y modificarlos.

### **2.2.4 Desarrollo conceptual de pruebas de software**

Luego se debe desarrollar las pruebas de componentes de aquellos que tuvieron que ser modificados producto de la migración. Con el propósito que no altere su resultado si se lo compara contra el resultado que arroja el mismo componente en producción.

Finalmente se deben probar todos los componentes del sistema para asegurar que el cambio producido en un componente no altere o modifique el resultado o comportamiento del sistema en general.

## **2.3 Marco legal o institucional**

En fecha 8 de agosto de 2011, el Estado promulga la Ley<sup>25</sup> con el propósito de asumir como política de Estado la implementación de estrategias para incorporar el conocimiento y aplicación de nuevas tecnologías de información y comunicación. Esta Ley declara en su título IV “*prioridad nacional la promoción del uso de las tecnologías de información y comunicación para procurar el vivir bien de todas las bolivianas y bolivianos*” y dispone que se promueva la incorporación del Gobierno Electrónico y se priorice la utilización del Software Libre y Estándares Abiertos, para lo cual se deberán elaborar planes y lineamientos.

---

<sup>22</sup> Norma ISO IEC 17799.

<sup>23</sup> DATE C. J., Introducción a los sistemas de Bases de Datos.

<sup>24</sup> SQL Server 2012 Upgrade and Application Compatibility, Choose an Upgrade Strategy

<sup>25</sup> Ley 164, General de Telecomunicaciones y Tecnologías de Información y Comunicación, 8 Agosto 2011.

A través de un Decreto Supremo<sup>26</sup>, se aprueba el Reglamento a la Ley número 164, de 8 de agosto de 2012, para el Desarrollo de Tecnologías de Información y Comunicación, también se establece el uso de Software Libre y Estándares Abiertos. Se define el plazo de dieciocho (18) meses a partir de la publicación del Decreto Supremo, para la elaboración del Plan de Implementación del Gobierno Electrónico y el Plan de Implementación de Software Libre y Estándares Abiertos que deberán ser aprobados mediante Decreto Supremo.

En la disposición transitoria quinta se establece que el plazo máximo para la migración de los sistemas de las entidades públicas a Software Libre y Estándares Abiertos, será de siete (7) años desde el inicio de la migración.

Esta norma establece que el Ministerio Planificación del Desarrollo (MPD) es la entidad encargada, en coordinación con el Ministerio de Obras Públicas, Servicios y Vivienda (MOPSV), a través del Viceministerio de Telecomunicaciones, y la Agencia para el Desarrollo de la Sociedad de la Información en Bolivia (ADSIB), de elaborar ambos planes.

En concordancia al plan de implementación de Gobierno Electrónico el Estado Boliviano aprueba el Decreto Supremo<sup>27</sup>, donde se dispone la creación de la “*Dirección de Gobierno Electrónico (DGE)*” dependiente del ministerio de Planificación del Estado Plurinacional de Bolivia. Actualmente esta dirección se encuentra trabajando en el plan de implementación del Gobierno Electrónico, mismo que tiene como plazo mínimo de entrega en mayo del 2015.

A la fecha de la culminación del presente trabajo, el Estado Plurinacional de Bolivia ha creado el Comité Plurinacional de Tecnologías de la Información y Comunicación - COPLUTIC<sup>28</sup> basado en el Artículo 73 de la Ley 164 de telecomunicaciones del 8 de agosto de 2011, formado por las siguientes instituciones: Ministerio de Obras Públicas, Servicios y Vivienda, Ministerio de Comunicación, Ministerio de Educación, Ministerio de Planificación del Desarrollo y la Agencia de Desarrollo para la Sociedad de la Información en Bolivia – ADSIB.

El COPLUTIC, en el marco de la Ley 164 de Telecomunicaciones y del Decreto Supremo 1793 y con el fin de recabar iniciativas en la elaboración del Plan de Implementación del Gobierno Electrónico y del Plan de Implementación de Software Libre y Estándares Abiertos ha presentado un borrador (16 de junio del 2015) preliminar a las entidades públicas y a la sociedad en su conjunto.

A iniciativa de la Senadora Nelida Sifuentes, Primera Vicepresidenta de la Cámara de Senadores, se realizó el Seminario de Socialización de los Planes de Software Libre, Estándares Abiertos y Gobierno electrónico, con el objetivo de avanzar en la elaboración de ambos planes y para recoger iniciativas de la sociedad civil. Este seminario se llevó a cabo en el ADSIB<sup>29</sup> el día martes 14 de abril del año 2015 en el Hall de la Vicepresidencia.

---

<sup>26</sup> Decreto Supremo 1793 de fecha 13 de noviembre del año 2013.

<sup>27</sup> Decreto Supremo 1874 de fecha 23 de enero de 2014.

<sup>28</sup> <http://coplutic.gob.bo/>

<sup>29</sup> [http://www.adsib.gob.bo/adsibnueva/noticiacompleta.php?codigo\\_noticia=Vm0wd2VFMUdiRmhTV0d4V1YwZG9XVmx0ZUV0WFJteHIWbTVrVm1KR2NlbfFdWm1JMVIVaV1ZVMUVhejA9](http://www.adsib.gob.bo/adsibnueva/noticiacompleta.php?codigo_noticia=Vm0wd2VFMUdiRmhTV0d4V1YwZG9XVmx0ZUV0WFJteHIWbTVrVm1KR2NlbfFdWm1JMVIVaV1ZVMUVhejA9)

## ***Capítulo III: Situación y factibilidad***

Para determinar la factibilidad de la migración de los SGBD licenciados a SGBD de software libre y estándares abiertos en las instituciones y entidades públicas del estado. Es necesario previamente realizar el análisis de la situación de los SGBD para conocer el impacto y las perspectivas de adopción de SGBD de software libre y estándares abiertos en las instituciones y entidades del Estado. Una mejor comprensión del problema contempla aspectos importantes como: el uso actual de SGBD, las capacidades en las instituciones y entidades del Estado y las características de las licencias que poseen las instituciones y entidades del estado, entre otras.

Para conocer la situación de los SGBD en las instituciones y entidades del Estado se ha utilizado la información generada por las siguientes encuestas:

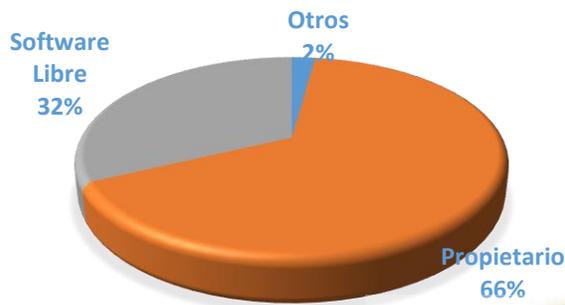
- Encuesta realizada a través del “Formulario de Gobierno Electrónico” a instituciones y entidades estatales en junio de 2014 que reúne 76 entidades.
- Encuesta realizada a través del “Formulario de Software Libre” a instituciones y entidades estatales en junio de 2014 que reúne 84 entidades.
- Encuesta realizada para determinar la situación de servicios entregados a la ciudadanía a través del uso de Tecnología Información y Comunicación (TIC) con el Cuestionario 001 - Área de Sistemas (Información de Base, ver Anexo A). Segmentado las instituciones públicas para que sea representativa. (Total de entidades identificadas 47, ver Anexo B), segmentadas en:
  - 18 Ministerios
  - 18 Servicios y Direcciones
  - 5 Empresas Estatales
  - 5 Autoridades
  - 1 Entidad de Control (Contraloría General del Estado)

La información producida por estas encuestas, fue utilizada por la Dirección General de Gobierno Electrónico – DGGE, dependiente del Ministerio de Planificación del Desarrollo, para realizar el análisis de situación del Software Libre en el Estado, categorizando el uso de Software Libre en: software de servidor, software de desarrollo, software de escritorio. Del cual para el presente trabajo nos interesa el software de servidor.

### ***3.1 Uso actual de bases de datos***

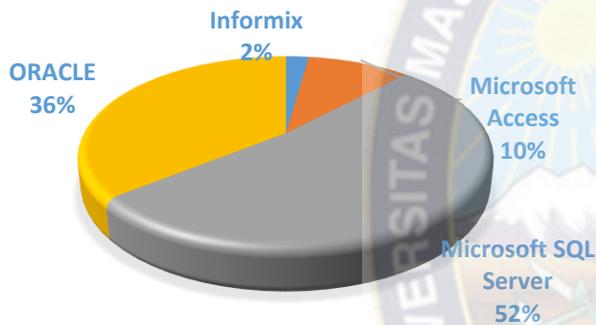
En la actualidad, de acuerdo a datos recogidos a través de las encuestas mencionadas anteriormente, las instituciones y entidades del Estado usan SGBD de sistemas libres y estándares abiertos así como SGBD de tipo comercial y privativo. Se hace notar que aquellas registros que mencionen el uso de SGBD comercial y privativo, no necesariamente significa que es SGBD con licencia.

### 3.1.1 Bases de datos utilizados



Del Cuadro 1, se puede observar que los SGBD más utilizados en las instituciones y entidades más representativas del Estado son de tipo propietario, privativo o comercial con el 66%, 32% utilizan software libre y estándares abiertos.

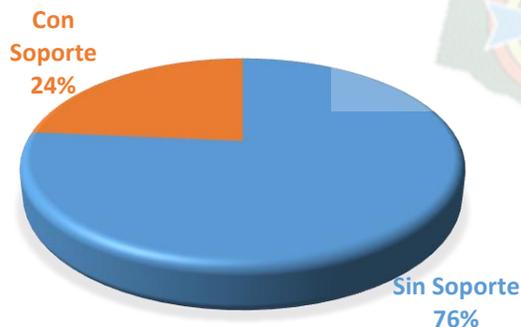
Cuadro 1, Tipos de SGBD utilizados. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



Del Cuadro 2, se muestra que las instituciones y entidades más representativas del Estado, prefieren utilizar Microsoft SQL Server con 52%, Oracle con 36%, Microsoft Access 10% e Informix con 2%.

Cuadro 2, SGBD propietarios utilizados. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.

De los cuadros 1 y 2, se puede concluir que las instituciones y entidades más representativas del Estado prefieren utilizar SBGB propietario y como consecuencia prefieren utilizar Microsoft SQL Server en cualquiera de sus versiones o ediciones.



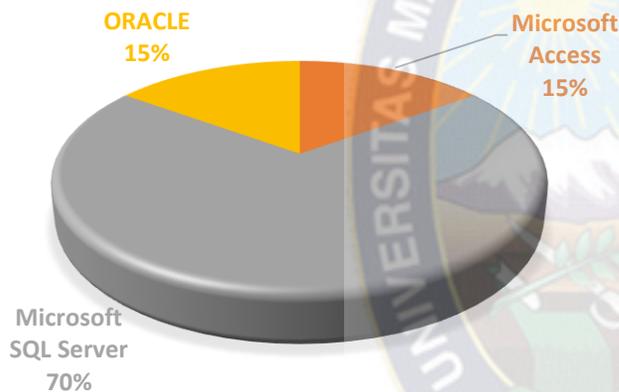
Del Cuadro 3, se puede apreciar que el soporte técnico contratado por las instituciones y entidades más representativas del Estado para el manejo y administración de sus SGBD es de 76% Sin soporte técnico contra 24% que tiene soporte técnico.

Cuadro 3, Soporte técnico a SGBD, Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



En el Cuadro 4, también se puede apreciar que aquellos SGBD que no tienen soporte técnico, están conformados por 56% de tipo propietario, privativo o comercial su SGBD que no tiene soporte técnico, 41% de tipo software libre y estándares abiertos su SGBD que no tiene soporte técnico y 3% otros tipos de SGBD que no tienen soporte técnico.

Cuadro 4, SGBD sin soporte técnico. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



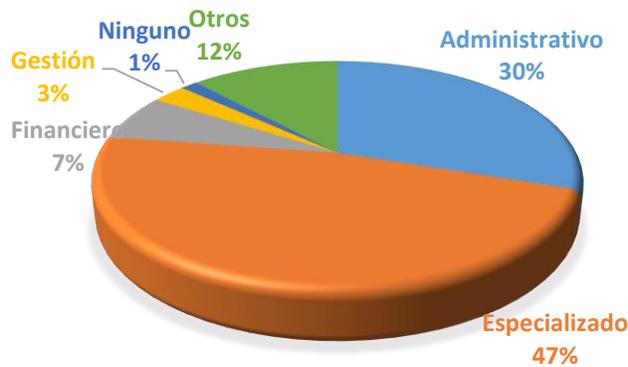
Finalmente en el Cuadro número 5, podemos ver el detalle de SGBD propietario, privativo o comercial sin soporte técnico que está conformado por: 70% Microsoft SQL Server en cualquier versión o edición, 15% Oracle y 15% Microsoft Access.

Cuadro 5, SGBD propietarios sin soporte técnico. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.

De los Cuadros 3, 4 y 5 se puede concluir que las instituciones y entidades más representativas del Estado no cuentan con soporte técnico para los SGBD que utilizan, de los SGBD que no tienen soporte técnico son en general de tipo propietario y que el SGBD que no cuenta con soporte técnico es del fabricante Microsoft.

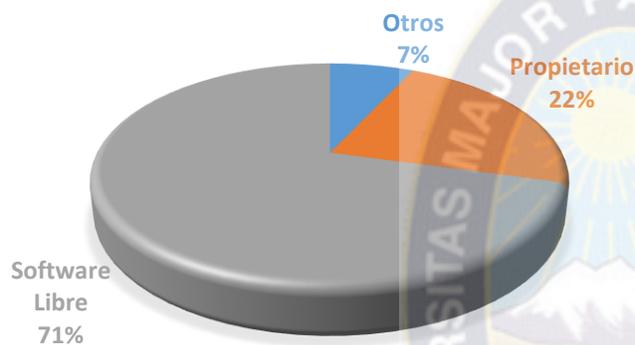
### 3.1.2 Sistemas de información que utilizan bases de datos

A continuación se presentan los resultados obtenidos en relación al uso de sistemas de información o aplicaciones y el uso que hacen del tipo de sistemas de información y que SGBD o tipo de SGBD utilizan.



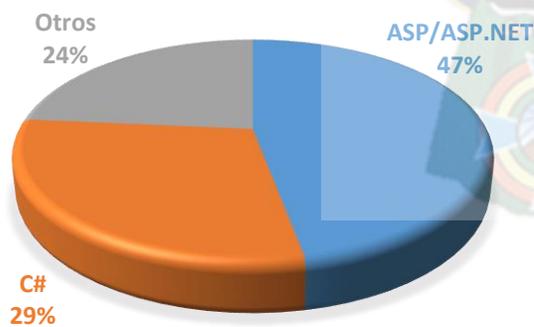
En el Cuadro 6, tipo de sistemas de información en las instituciones y entidades más representativas del Estado, se puede visualizar que el 47% son sistemas de información especializados, el 30% corresponden a sistemas de información administrativos, 12% otros, 7% financieros, 3% de gestión y 1% no tiene ningún tipo de sistemas de información.

Cuadro 6, Tipos de sistemas de información. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



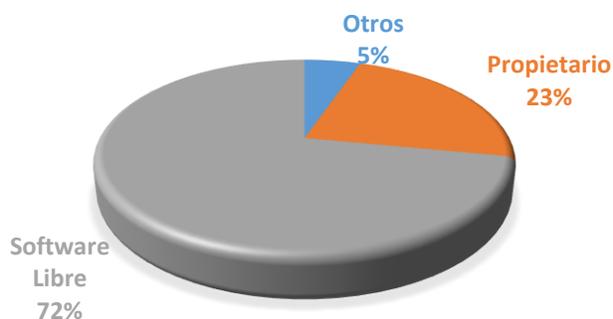
En el Cuadro 7, lenguaje de programación utilizado en sistemas de información, se puede observar que el 71% fueron desarrollados utilizando lenguaje de programación de software libre y estándares abiertos, 22% fue desarrollado en lenguajes de programación propietario, comercial y privativo y 7% por Otros lenguajes de programación.

Cuadro 7, Lenguajes de programación utilizados. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



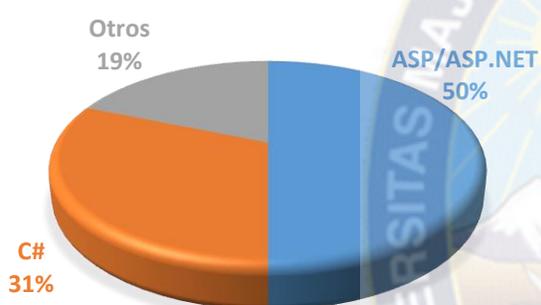
En el Cuadro 8, lenguajes de programación propietarios utilizados en los sistemas de información, haciendo un acercamiento a la información del cuadro 7, se puede observar que el 47% fueron desarrollados utilizando Microsoft ASP/ASP.NET, 29% utiliza C# y 24% otros tipos de lenguajes de programación propietarios.

Cuadro 8, Lenguajes de programación Propietarios utilizados, Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



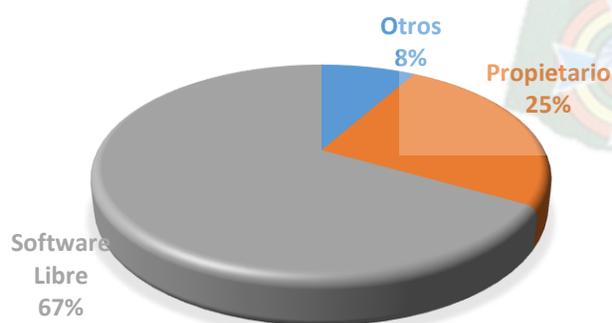
En el Cuadro 9 se muestra la relación de la ausencia de soporte técnico a lenguajes de programación, donde se puede ver que el 72% pertenece a Software Libre y estándares abiertos, 23% a propietarios, comercial o privativo y 5% a otros.

Cuadro 9, Lenguajes de programación sin soporte técnico, Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



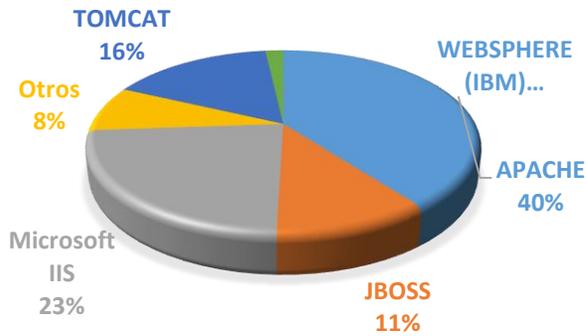
Del Cuadro 10, se realiza un acercamiento del cuadro 9, para determinar que lenguajes de programación propietarios no cuentan con soporte técnico. 50% corresponde a Microsoft ASP/APS.NET, 31% C# y 19% Otros

Cuadro 10, Lenguajes de programación propietarios sin soporte técnico. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



En relación a los servidores de aplicaciones utilizados, se puede observar en el Cuadro 11, que el 67% utiliza servidores de aplicaciones de software libre y estándares abiertos, 25% propietarios, privativos o comercial y 8% otros.

Cuadro 11, Tipo de servidor de aplicaciones utilizadas, Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



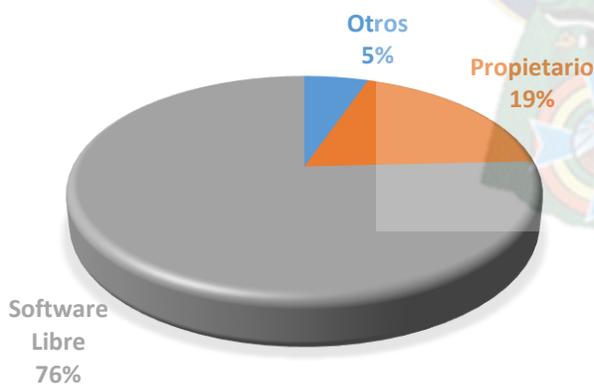
Del Cuadro 12, podemos visualizar que el servidor de aplicaciones preferido es con 40% Apache, 23% Microsoft IIS, 16% Tomcat, 11% Jboss, 8% Otros y 2% Websphere de IBM.

Cuadro 12, Servidores de aplicaciones utilizados. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



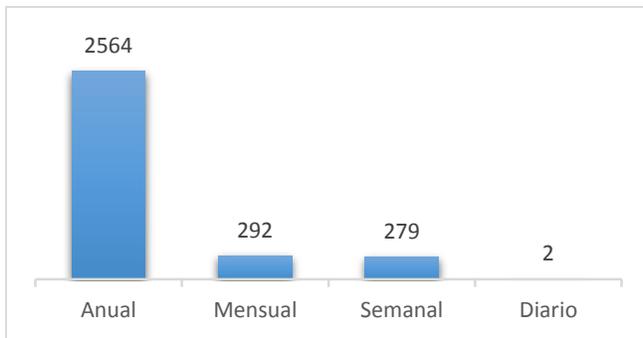
En lo relacionado a soporte técnico para los servidores de aplicaciones, podemos ver en el Cuadro 13, que el 86% no cuenta con soporte técnico para sus servidores de aplicaciones y que el 14% si tiene soporte técnico para sus servidores de aplicaciones.

Cuadro 13, Soporte técnico en servidor de aplicaciones, Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



Del Cuadro 14, donde se realiza el detalle de servidores de aplicaciones sin soporte técnico, se puede ver que el 76% es de Software libre y estándares abiertos, 19% propietario, privativo o comercial y 5% otros.

Cuadro 14, Servidor de aplicaciones sin soporte técnico, Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



La frecuencia de información que se maneja utilizando bases de datos está conformado por, según el cuadro 15: 2.5 TB anual, 292 GB Mensual, 279 GB Semanal y 2 GB Diario, se hace notar que no todas las instituciones manejan la frecuencia de información al mismo tiempo es decir, no todos manejan información anual, mensual semanal y diaria.

Cuadro 15, Frecuencia de información en GB. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.

En esta sección de sistemas de información utilizados, se puede concluir que los tipos de sistemas de información más utilizados (adquirido o desarrollado) por la instituciones y entidades más representativas del Estados son los Especializados, por ejemplo la Aduana Nacional que tiene el SIDUNEA que es utilizado para administrar las importaciones y exportaciones.

Para el desarrollo de sus sistemas de información prefieren utilizar lenguajes de programación de software libre y estándares abiertos como JAVA, JSP y PHP. Los servidores de aplicaciones preferidos son de software libre y estándares abiertos como el APACHE y TOMCAT.

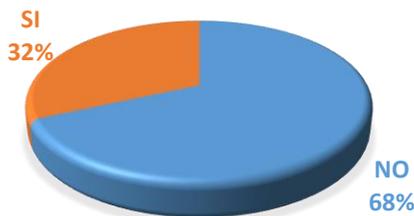
Y en general los lenguajes de programación como los servidores de aplicaciones no cuentan con soporte técnico por parte del fabricante del software.

### 3.2 Capacidades

Para un mejor análisis, esta sección se dividirá en: Capacitación y su financiamiento, el nivel del área de sistemas en el organigrama y la infraestructura tecnológica.

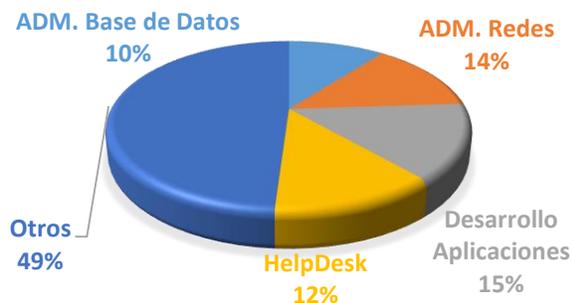
#### 3.2.1 Capacitación y su financiamiento

La capacitación y habilidades actuales del personal del área de sistemas de las instituciones y entidades del Estado, están directamente relacionado a los SGBD que administran y manejan, lo que significa, que las habilidades se encuentran enfocados en gran porcentaje al SGBD comercial y privativo.



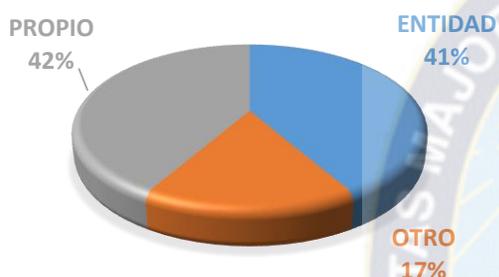
Del Cuadro 16, se puede observar que los funcionarios de las instituciones y entidades del Estado han recibido capacitación en TIC en los dos últimos años solo en un 32%, mientras que la mayoría de los funcionarios no han recibido capacitación en los dos últimos años, que representan un 68%.

Cuadro 16, Capacitación en los dos últimos años. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.



Del Cuadro 17, se puede determinar que la capacitación que reciben los responsables de las áreas de sistemas de las instituciones y entidades del Estado en SGBD representa el 10% del total.

Cuadro 17, Capacitación de los funcionarios de sistemas en TIC. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.

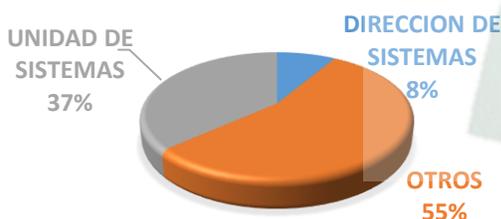


En el Cuadro 18 se puede ver que las instituciones y entidades del Estado solo el 41% invierten en capacitación de sus funcionarios ya que existe un 42% de los funcionarios que se financia por cuenta propia la capacitación en TIC. Significa que el personal del área de sistema no tiene la capacitación suficiente en SGBD de software libre y estándares abiertos para encarar un proceso de migración.

Cuadro 18, Tipo de financiamiento de la capacitación. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.

La conclusión de esta sección de capacitación y su financiamiento, es que los funcionarios de las instituciones y entidades públicas más representativas del Estado, cuentan con poca capacitación en administración de SGBD y que en general la capacitación es financiada con fuentes propios, por no existir presupuesto.

### 3.2.2 Nivel en el organigrama



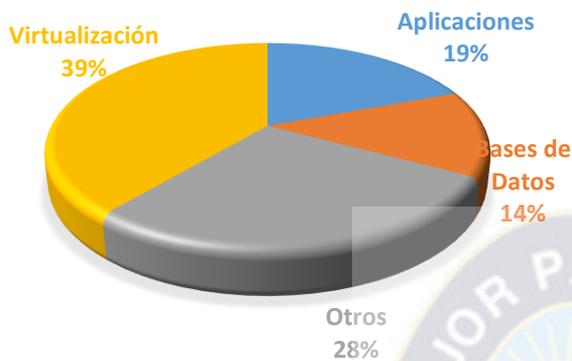
En lo relacionado al nivel del área de sistemas, se puede observar en el siguiente cuadro que el 8% de las instituciones y entidades del Estado más representativas tienen una Dirección de Sistemas, 37% tienen una unidad de sistemas y 55% tienen otro tipo de denominación.

Cuadro 19, Nivel del área de sistemas en el organigrama.

Existen muy pocas instituciones y entidades del Estado que están preparadas a nivel institucional para tomar decisiones relacionadas con la migración de sus SGBD a software libre y estándares abiertos.

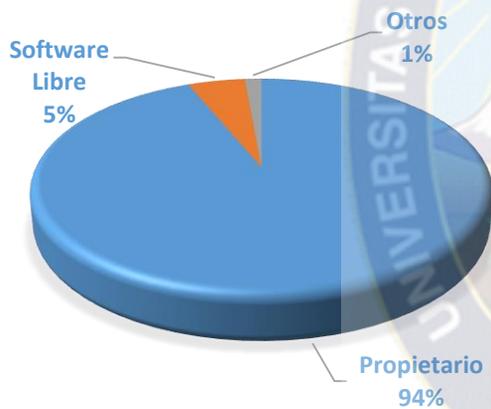
### 3.2.3 Infraestructura tecnológica

Lo relacionado a capacidades de infraestructura, se realizará el análisis de los tipos de servidores con los que cuentan las instituciones y entidades del Estado, que sistema operativo (servidor, escritorio) tienen y si cuentan con el soporte técnico respectivo.



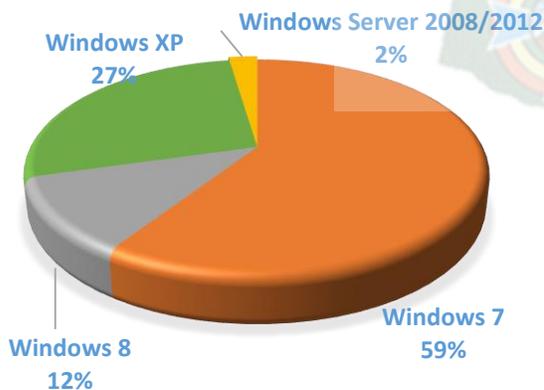
Cuadro 20, Tipo de servidores utilizados.

En el Cuadro 20, se puede apreciar que las instituciones y entidades más representativas del estado utilizan 39% Servidores de virtualización, 28% otros que están confirmados por servidores de archivo/impresión, servidores de correo electrónico, de páginas web, etc., 19% servidores de aplicaciones y 14% Servidores de bases de datos.



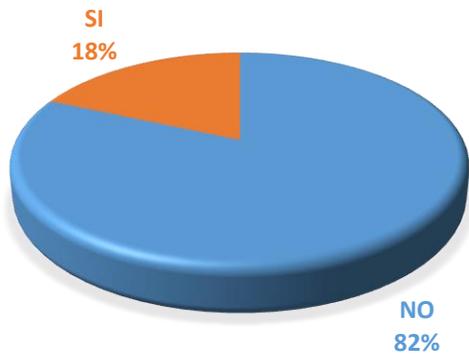
Cuadro 21, Tipos de sistemas operativos utilizados.

Del Cuadro 21, tipos de sistemas operativos utilizados, que incluyen a servidores y escritorio. Se tiene que el 94% utiliza sistema operativo propietario, 5% software libre y 1% otro tipo de sistema operativo.



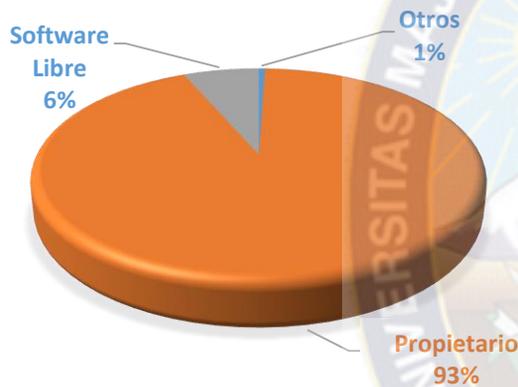
Cuadro 22, Tipos de sistemas operativos propietarios

En el Cuadro 22, se tiene el detalle de sistemas operativos propietarios utilizados, donde el 59% corresponde a Microsoft Windows 7, 27% a Microsoft Windows XP, 12% a Microsoft Windows 8 y 2% a Microsoft Servers.



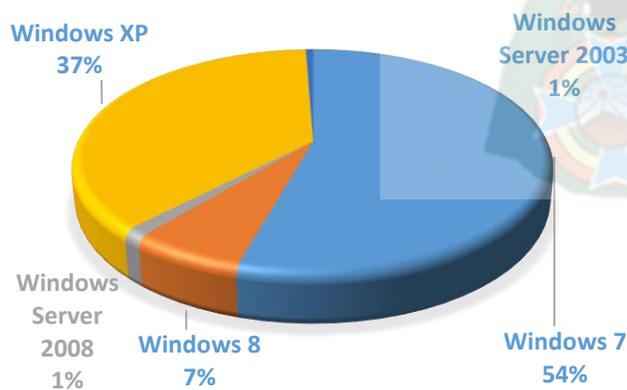
En lo relacionado al soporte técnico de los sistemas operativos utilizados, se tienen que el 82% de los sistemas operativos no tiene soporte técnico, mientras el 18% si cuenta con soporte técnico.

Cuadro 23, Soporte técnico a sistemas operativos



El detalle de los sistemas operativos sin soporte técnico en el Cuadro 24, nos muestra con 93% a sistemas operativos propietarios que no tienen soporte técnico, 6% sistemas operativos de software libre sin soporte técnico y 1% a otros sistemas operativos sin soporte técnico.

Cuadro 24, Tipos de sistemas operativos sin soporte técnico



En el Cuadro 25, se puede ver el detalle de sistemas operativos propietarios sin soporte técnico. Encabezados por: 54% Windows 7, 37% Windows XP, 7% Windows 8 y los servidores Windows con 1%. Se hace notar que los Windows XP ya no tienen soporte técnico.

Cuadro 25, Sistemas operativos propietarios sin soporte técnico

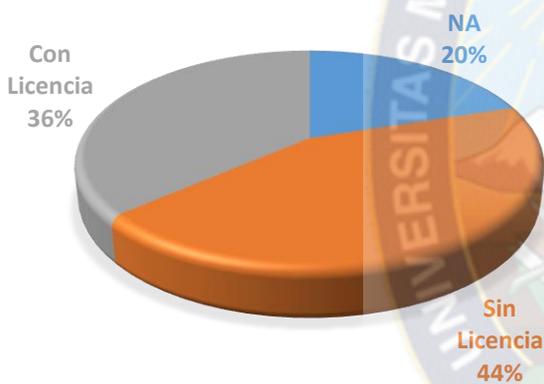
La conclusión de esta sección es que las instituciones y entidades más representativas del estado utilizan servidores para virtualizar, que los sistemas operativos más utilizados son de tipo propietario y que por lo general no tienen soporte técnico contratado.

### 3.3 Licencias

En lo relacionado con las licencias en las instituciones y entidades del Estado, se presentan dos problemas:

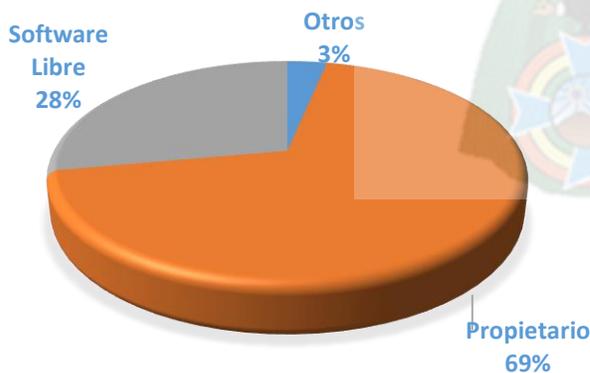
- El pago por concepto de licencia de SGBD.
- El uso de SGBD que no cuenta con licencia.

#### 3.3.1 Licencias de bases de datos



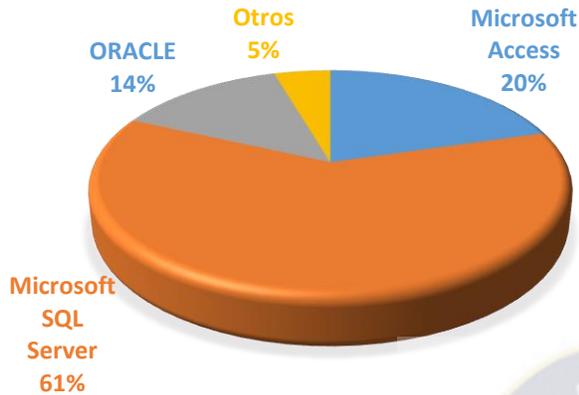
En el Cuadro 26, se muestra el licenciamiento del uso de SGBD en instituciones y entidades del Estado, donde se muestra que el 44% no tiene licencia, 36% cuentan con licencia de uso de SGBD y 20% no aplica la compra de licencia.

Cuadro 26, Licenciamiento de uso SGBD



En el Cuadro 27 se puede observar el detalle de SGBD sin licencia, donde el 69% son SGBD de tipo propietario, 28% a software libre o que no aplica y 3% a otros

Cuadro 27, SGBD sin licencia



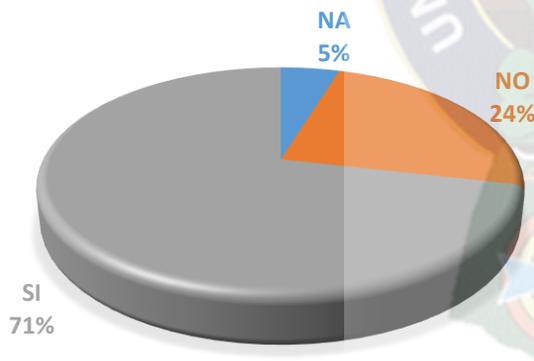
El detalle SGBD sin licencias de tipo propietario, muestra que el 61% es Microsoft SQL Server, 20% Microsoft Access, 14% Oracle y 5% Otros tipos de SGBD propietarias.

Cuadro 28, SGBD propietarios sin licencias

Como conclusión del licenciamiento de uso de SGBD, se aprecia que las instituciones y entidades del estado no tienen en su mayoría licencia de uso de SGBD, de las cuales en general son de tipo propietario y de esta último concluimos que las licencias de uso no adquiridas corresponden en su mayoría al fabricante Microsoft.

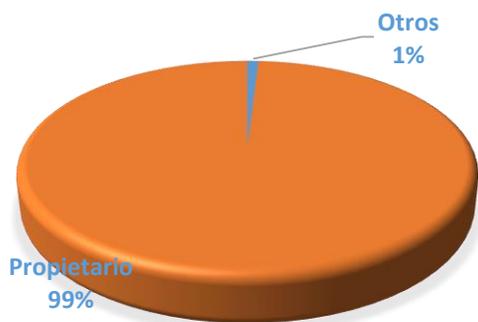
### 3.3.2 Infraestructura

En lo relacionado a infraestructura se mostrará principalmente el licenciamiento de uso de sistemas operativos en instituciones y entidades del Estado, tanto a nivel servidores como de escritorio. Esto debido a que están directamente relacionado al uso de SGBD.



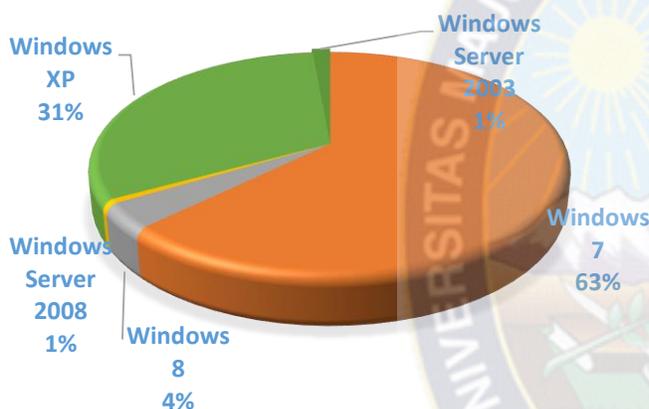
En el Cuadro 29, se puede apreciar que el 71% cuenta con licencia de uso, 24% no tiene su licencia de uso de sistemas operativos y 5% no aplica por ser sistemas operativos de software libre y estándares abiertos.

Cuadro 29, Licenciamiento de sistemas operativos



En el Cuadro 30, se muestra los sistemas operativos sin licencia, donde el 99% corresponde a sistemas operativos de tipo propietario y el resto a otros.

Cuadro 30, Sistemas operativos propietarios sin licencia



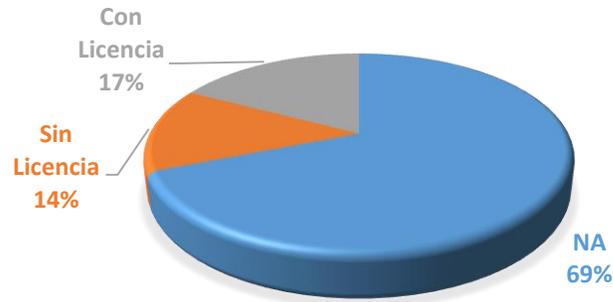
En el Cuadro 31, se muestra el detalle de sistemas operativos propietarios sin licencia, donde 63% corresponde a Windows 7, 31% a Windows XP, 4% Windows 8 y el 1% para los servidores Windows 2008 y 2012.

Cuadro 31, Detalle de sistemas operativos propietarios sin licencias

En general podemos concluir que las instituciones y entidades del estado cuentan con su licencia de uso de los sistemas operativos de servidores y escritorios, los que no cuentan con licencia de uso corresponden a Microsoft Windows 7 y Microsoft Windows XP, este último ya sin soporte por parte del fabricante.

### 3.3.3 Lenguajes de programación

El uso de licencias de lenguajes de programación actuales en las instituciones y entidades del Estado es en su mayoría corresponden en 69% a No Aplica (NA) o lenguajes de programación basados en software libre y estándares abiertos, 17% pertenece a la categoría de lenguajes de programación con su licencia de uso y 14% corresponde a lenguajes de programación sin licencia, que por otra parte son lenguajes de programación del tipo propietario, privativo o comercial, como el ASP.NET o C#. De acuerdo al cuadro 32.



Cuadro 32, Licenciamiento de lenguajes de Programación

Del Cuadro 32, se puede concluir que las instituciones y entidades más representativas del Estado utilizan para el desarrollo de aplicaciones, páginas web y otros, los lenguajes de programación de software libre y estándares abiertos.

### 3.4 Interpretación de resultados

Como interpretación de resultados de los cuadros de la sección 3.1 a la 3.3 en las instituciones y entidades más representativas del Estado, se puede decir que:

- El uso de bases de datos se prefieren utilizar SGBD propietario y como consecuencia utilizan Microsoft SQL Server en cualquiera de sus versiones o ediciones y que además no cuentan con soporte técnico para los SGBD que utilizan, de los SGBD que no tienen soporte técnico son en general de tipo propietario y que el SGBD que no cuenta con soporte técnico es del fabricante Microsoft.
- En lo relacionado a sistemas de información utilizados, los tipos de sistemas de información más utilizados (adquirido o desarrollado) son los Especializados. Y para el desarrollo de sus sistemas de información prefieren utilizar lenguajes de programación de software libre y estándares abiertos como JAVA, JSP y PHP.
- Los servidores de aplicaciones preferidos son de software libre y estándares abiertos como el APACHE y TOMCAT. Y en general los lenguajes de programación de los servidores de aplicaciones no cuentan con soporte técnico por parte del fabricante del software.
- En relación a capacitación y su financiamiento, los funcionarios cuentan con poca capacitación en administración de SGBD y que en general la capacitación es financiada con fuentes propios, por no existir presupuesto asignado para tal efecto.
- En cuanto al nivel de los unidades/direcciones de sistemas en el organigrama, existen muy pocas instituciones y entidades del Estado que están preparadas a nivel institucional para tomar decisiones relacionadas con la migración de sus SGBD a software libre y estándares abiertos.

- Respecto a la infraestructura tecnológica, se utilizan servidores para virtualizar, que los sistemas operativos más utilizados son de tipo propietario y que por lo general no tienen soporte técnico contratado.
- En el licenciamiento de uso de SGBD, no tienen en su mayoría licencia de uso de SGBD, de las cuales en general son de tipo propietario y de esta último concluimos que las licencias de uso no adquiridas corresponden en su mayoría al fabricante Microsoft.
- El licenciamiento de infraestructura en general podemos decir que se cuentan con su licencia de uso de los sistemas operativos de servidores y escritorios, los que no cuentan con licencia de uso corresponden a Microsoft Windows 7 y Microsoft Windows XP, este último ya sin soporte por parte del fabricante.
- Finalmente en lo relacionado a lenguajes de programación que utilizan para el desarrollo de aplicaciones, páginas web y otros, son los lenguajes de programación de software libre y estándares abiertos.

A continuación se muestra la Tabla 4, donde se hace un resumen de los hallazgos encontrados, a través de las encuestas ya mencionadas.

*Tabla 4. Resumen de interpretación de resultados*

Descripción	Propietario	Libre	Observaciones
Uso Bases de datos	68% (Microsoft)	32%	Sin soporte
Tipos de Sistemas Información		(JAVA, PHP)	47% Especializado 30% Administrativo Sin soporte
Servidores Aplicaciones	29%	71% (APACHE, TOMCAT)	Sin soporte
Capacitación			Sin presupuesto, financiamiento propio
Nivel Organigrama			No están preparados para tomar decisiones
SO Servidores	95% Microsoft	5%	Sin soporte
Licencias SGBD	72% Microsoft	28%	Sin licencias de uso
Licencias Escritorio	99% Microsoft	1%	Sin licencias de uso
Licencias Lenguajes Programación	14%	86% JAVA, PHP	Sin soporte

### 3.5 Propuesta de factibilidad

De las secciones 3.1 a 3.3 del presente capítulo se evidenció que las instituciones y entidades del Estado cuentan, utilizan y administran en su mayoría SGBD propietario, comercial y privativo, además no se encuentran lo suficientemente capacitados y no tienen el nivel organizacional dentro de su institución para enfrentar un proceso de migración a SGBD de software libre y estándares abiertos.

Para poder determinar si la migración en las instituciones y entidades del Estado es factible se recurre a las pruebas de factibilidad propuesta por James Senn<sup>30</sup>, donde se considera tres tipos de factibilidad que se deben evaluar, las cuales se describen a continuación:

#### 3.5.1 Factibilidad operativa

La factibilidad operativa determinará, si se puede realizar la migración a SGBD de software libre y estándares abiertos. Por otra parte si los aplicativos y programas que utilizan los SGBD seguirán funcionando y no presentaran problemas el momento de su puesta en marcha.

Para determinar si la migración a SGBD de Software libre y estándares abiertos en las instituciones y entidades del Estado, es factible operativamente se debe recurrir al Anexo D. En la Tabla 5, a continuación podrá encontrar una descripción de cada elemento o componente que se evalúa para determinar la factibilidad operativa de la migración.

Tabla 5, Descripción de elementos o componentes de evaluación operativa

Elementos	Descripción
Código fuente de los aplicativos	Existe acceso al código fuente de los aplicativos y programas que acceden o utilizan los SGBD o bases de datos a migrar. Es el componente de factibilidad operativa más importante, ya que de no existir al código fuente de los aplicativos, no se podrá realizar la migración.
Documentación	Si existe la documentación de los aplicativos y programas que acceden o utilizan a los SGBD.
Documentación actualizada	Se mantiene una relación de control de versiones de la documentación. Esta actualizada la documentación.
Nivel Decisional	El área de sistemas, dentro del organigrama institucional, es de dirección de sistemas o superior.

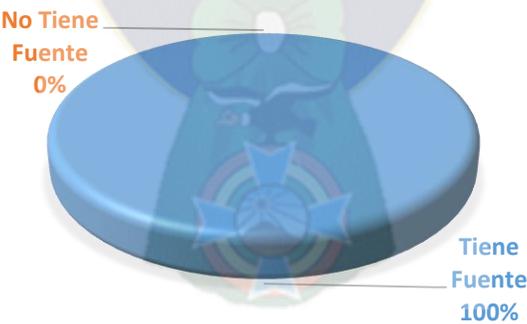
<sup>30</sup> SENN James, Análisis y Diseño de Sistemas de Información, pp.55.

Elementos	Descripción
Capacidad del Personal	El personal del área de sistemas, se encuentra capacitado en SGBD de software libre y estándares abiertos.
	El personal del área de sistemas, cuenta con alguna clase de certificación internacional en SGBD de software libre y estándares abiertos.

El elemento o componente que prácticamente decide si la migración es factible operativamente a SGBD de software libre y estándares abiertos, es la existencia de acceso a los códigos fuentes de los aplicativos o programas que acceden y utilizan los SGBD a migrar, ya que es en el código fuente donde se define generalmente la cadena de conexión<sup>31</sup> al SGBD y por otra parte es en el código fuente donde se realizarán las modificaciones de incompatibilidad de sentencias T-SQL entre SGBD (origen y destino).

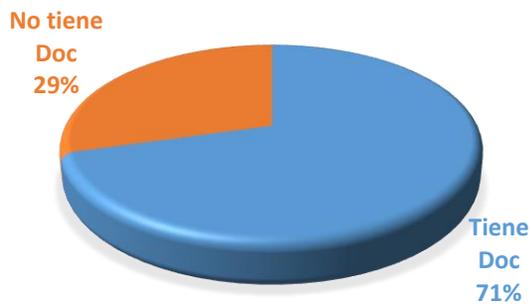
La salvedad al párrafo anterior, sería que la cadena de conexión se defina por fuera del código fuente de los aplicativos, es decir como parámetro, pero de todas formas quedaría las sentencias T-SQL que utiliza el aplicativo a la cual no se tendría acceso.

De acuerdo a las encuestas realizadas y ya mencionadas, se puede evidenciar que las instituciones y entidades más representativas del Estado, tienen acceso al código fuente de los aplicativos y programas que acceden o utilizan los SGBD o bases de datos a migrar, como se puede observar en el Cuadro número 33.



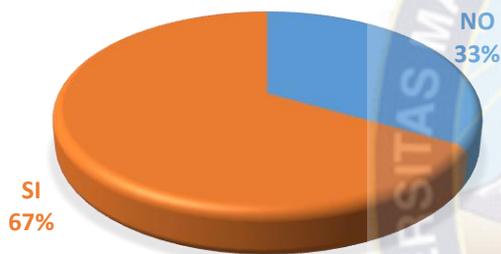
Cuadro 33, Propiedad o acceso al código fuente

<sup>31</sup> Cadena de conexión, contiene información de inicialización que se transfiere como un parámetro desde un proveedor de datos a un origen de datos. La sintaxis depende del proveedor de datos.



Del Cuadro 34, que representa la existencia de la documentación de los aplicativos y programas que acceden o utilizan a los SGBD. Se puede apreciar que el 71% de las instituciones y entidades más representativas del Estado tienen la documentación de sus aplicativos y programas que utilizan los SGBD.

Cuadro 34, Documentación de los sistemas de información



En el Cuadro 35, relacionado con, si la documentación de los aplicativos que acceden a SGBD está actualizada o no. Se puede ver que el 67% de las instituciones y entidades más representativas del Estado, cuentan con documentación actualizada, mientras que el 33% no está actualizada.

Cuadro 35, Documentación actualizada de sistemas de información

### 3.5.2 Factibilidad técnica

La factibilidad técnica determina si las instituciones y entidades del Estado cuentan con los suficientes recursos técnicos para llevar adelante el proceso de migración a SGBD de software libre y estándares abiertos, en otras palabras, se verifica si tienen la tecnología necesaria y adecuada, como servidores adicionales a los de producción y redes aisladas de producción donde se puedan realizar inicialmente el desarrollo de la migración, para posteriormente realizar las pruebas que certifiquen que la migración a SGBD de software libre y estándares abiertos funcionará.

Otro elemento o componente importante que ayudará a determinar si la migración a SGBD de software libre y estándares abiertos, es si se realizó o si se tiene un inventario de objetos del servidor que se quiere migrar, que contemple los objetos a nivel servidor y a nivel base de datos.

Para determinar si la migración a SGBD de software libre y estándares abiertos en las instituciones y entidades del Estado, es factible técnicamente se puede recurrir al Anexo E. En la Tabla 6 se podrá encontrar una descripción de cada elemento o componente que se evalúa para determinar la factibilidad técnica de la migración.

Tabla 6, Descripción de elementos o componentes de evaluación técnica

Elementos	Descripción
Inventario Servidor	Existe inventario de objetos a nivel servidor, por ejemplo una base de datos es un objeto a nivel servidor.
Inventario Base de Datos	Existe inventario de objetos a nivel bases de datos, por ejemplo una base de datos está compuesto por objetos como: tablas, vistas, procedimientos almacenados, funciones, índices, triggers, etc.
Infraestructura	<p>Se requiere contar con la infraestructura adicional que permita realizar la migración de SGBD, garantizando la continuidad de los aplicativos y programas.</p> <ul style="list-style-type: none"> <li>• Existe servidores adicionales para desarrollar ambientes de pruebas.</li> <li>• Existe los recursos para armar una red adicional</li> </ul>

En caso de que la institución o entidad del Estado no cuente con un inventario de objetos a nivel servidor y base de datos, puede recurrir a los anexos F y G respectivamente para poder obtener un inventario de lo solicitado.

### 3.5.3 Factibilidad económica

En relación a la factibilidad económica, se proponen una serie de tablas que dan una idea de números aproximados de cuánto costaría migrar a SGBD de software libre y estándares abiertos en todas las instituciones y entidades del Estado esto basado a la promulgación de la Ley No 164<sup>32</sup> y sus reglamentos contenidos en el Decreto Supremo No 1793, donde se prioriza el uso de software libre y estándares abiertos como factor de descolonización del conocimiento y como tal se deberían brindar los recursos económicos y tecnológicos necesarios para llevar adelante los procesos de migración a SGBD de software libre y estándares abiertos en todas las instituciones y entidades del Estado.

En la Tabla 7, se muestra el tiempo aproximado en cada etapa del proceso de migración, considerando una base de datos que contenga una tabla, una vista y un procedimiento almacenado, se puede apreciar que en este caso migrar a SGBD de software libre y estándares abiertos en este tomaría cuatro (4) horas y media, en todas las etapas del proceso de migración.

<sup>32</sup> Ley general de telecomunicaciones, tecnologías de información y comunicación, 8 de agosto del 2011.

Tabla 7, Tiempo en horas por etapa del proceso de migración

Descripción	Horas			Total Horas
	Tabla	Vista	Procedimiento Almacenado	
Desarrollo	0:15	0:15	1:00	1:30
Pruebas	0:15	0:15	1:00	1:30
Migración	0:15	0:15	1:00	1:30
Total	0:45	0:45	3:00	4:30

Si aplicamos los tiempos, expresados en horas, de la Tabla 7 a una base de datos que contenga 40 tablas, 30 vistas y 200 procedimientos almacenados, como se muestra en la tabla siguiente, la Tabla 8 se puede determinar de realizar el inventario y categorización del servidor origen que se quiere migrar. Vea la sección 4.2.1 inventario y categorización.

Tabla 8, Ejemplo de base de datos para cálculo de tiempo

Descripción	Horas			Total Horas
	Tabla	Vista	Procedimiento Almacenado	
Base Datos	40	30	200	

A continuación, se muestra la Tabla 9, que muestra el resultado de aplicar la Tabla 7 y 8, se puede observar que el total de horas se obtiene el 20% que es la cantidad de horas a considerar para migración, este porcentaje se obtiene como resultado de la experiencia de realizar múltiples procesos de migraciones de SGBD, donde este porcentaje representa la cantidad de tablas, vistas y procedimientos almacenados que requerirán de atención y modificación en el proceso de desarrollo de migración.

Tabla 9, Resultado de tiempos aplicando la Tabla 6 y 7

Descripción	Horas			Total Horas
	Tabla	Vista	Procedimiento Almacenado	
Desarrollo	10.00	7.50	200.00	217.50
Pruebas	10.00	7.50	200.00	217.50
Migración	10.00	7.50	200.00	217.50
Total	30.00	22.50	600.00	652.50
20% Modificados				130.50

La Tabla muestra que el proceso completo de migración para una base de datos definida con el número de tablas, vistas y procedimientos almacenados en la Tablas 8, se podría realizar en aproximadamente tres semanas, considerando que se deben realizar la instalación de los SGBD en el ambiente de desarrollo y pruebas.

Finalmente en la Tabla 10, se muestra el costo del proceso de migración, tomando en cuenta el tipo de consultor que llevaría a cabo el proceso de migración o si lo realizaría un funcionario de la institución, los costos que se exponen son promedios de costos de mercado por hora, expresado en dólares americanos.

Tabla 10, Costo de la migración por tipo consultor o funcionario, en dólares americanos

Consultor	Costo	Total
Internacional	120	15,660.00
Local	45	5,872.50
Línea	15	1,957.50
Funcionario	8	1,044.00

Se debe tomar en cuenta que los costos más altos, se deben a que los consultores poseen la experiencia y conocimiento de realizar migraciones de SGBD, contra un funcionario público que probablemente no cuente con la experiencia y conocimiento del proceso de migración.

### 3.6 Manejo de migraciones no factibles

No todos los SGBD comerciales, propietarios y privativos que tienen las instituciones y entidades del Estado se podrán migrar a SGBD de Software libre y estándares abiertos inicialmente, ya sea debido a temas operativos o técnicos.

A partir de este punto el Estado tiene dos alternativas: Desarrollo de un nuevo aplicativo o adquisición de un nuevo aplicativo. En ambos casos se debe considerar que deben cumplir la norma de software libre y estándares abiertos en la Ley No 164 y sus respectivos reglamentos.

Para tal efecto se recomienda que el Estado forme una empresa de desarrollo de software que sea descentralizada del Estado, pero que el Estado sea el mayor accionista, que se encargue de desarrollar software libre con estándares abiertos, de acuerdo a los requerimientos de las instituciones y entidades de Estado, y que realice la migración a SGBD de software libre y estándares abiertos

La ventaja para el Estado de contar con esta empresa de desarrollo de software libre y estándares abiertos que realice la migración a SGBD de software libre y estándares abiertos, son:

- El conocimiento o “know how” y la experiencia adquirida en el desarrollo de los aplicativos o migraciones a software libre y estándares abiertos para las diferentes instituciones y entidades del Estado, no se perdería, quedaría centralizada, almacenada y registrada en esta empresa.
- Tendría metodologías de desarrollo de software libre y estándares abiertos, que se aplicarían en todas las instituciones y entidades del Estado por igual.
- Podría centralizar y administrar los distintos servidores que se utilizan para el desarrollo y certificación de la migración en entornos virtuales, para todas aquellas entidades o instituciones del Estado que vayan a realizar migraciones a SGBD de software libre y estándares abiertos.

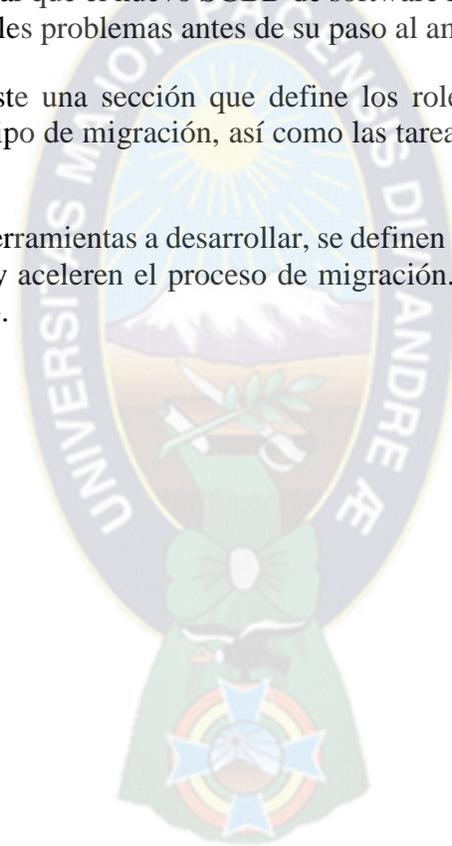
- Estaría a cargo del manejo de las adquisiciones de nuevo software libre y estándares abiertos para todas las instituciones y entidades del Estado.

En el siguiente Capítulo, se planteara la propuesta de la metodología del proceso de migración a SGBD de software libre y estándares abiertos, los ambientes de trabajo que serán necesarios para realizar el desarrollo de la migración, los pasos que se deben seguir, la documentación que se producirá en cada etapa o paso, así como el inventario y categorización de objetos a nivel servidor y a nivel de bases de datos, el desarrollo de la migración, como migrar los datos de origen a destino, como comparar los servidores origen y destino, y finalmente como realizar el control de cambios antes de realizar el proceso de migración en el entorno de producción.

También se verá las pruebas que se realizarán, como las pruebas unitarias y las pruebas de sistemas, esto con la finalidad de asegurar que el nuevo SGBD de software libre y estándares abiertos quede operativo y se anticipen posibles problemas antes de su paso al ambiente de producción.

En el capítulo siguiente, existe una sección que define los roles y responsabilidades de cada miembro o integrante del equipo de migración, así como las tareas y responsabilidades que están a su cargo.

Finalmente en la sección de herramientas a desarrollar, se definen los algoritmos para el desarrollo de herramientas que ayuden y aceleren el proceso de migración. Estas herramientas pueden ser desarrolladas en JAVA o C++.



## ***Capítulo IV: El proceso de migración***

En la sección 2.1.3 del presente trabajo se definió la migración como el proceso de llevar los objetos del SGBD origen a un SGBD destino, en la sección 2.2.3 se definió como realizar este proceso, es decir sacando los objetos del origen y probándolos en el SGBD destino para poder determinar si las sentencias de cada objetos son compatibles con en el SGBD destino.

Para el desarrollo de la migración se deben proporcionar ambientes de trabajo donde se pueda realizar las pruebas de migración y su posterior certificación, poder categorizar las bases de datos ya sea por su importancia, criticidad o volumen de información, probar la migración de los datos, comparar el destino contra el origen y realizar las pruebas de las aplicaciones.

### ***4.1 Ambientes de trabajo***

En este sentido deben existir tres ambientes de trabajo considerando que uno de ellos es el entorno actual que se viene ejecutando en producción y que pasará por un proceso de migración a un SGBD de software libre y estándares abiertos. Los ambientes a los que se hace referencias son: Desarrollo, Certificación y Producción.

Los ambientes necesarios para la migración a un SGBD de software libre y estándares abiertos, se puede apreciar en la figura 3, donde se puede notar que cada ambiente se inicia a partir de la obtención de un backup o respaldo del ambiente de producción para generar el respectivo ambiente, ya sea de desarrollo o de certificación.

Para la formación de los ambientes de desarrollo y certificación se puede utilizar herramientas de virtualización como el VMWare, Hyper V u otro similar, para la formación de los servidores de origen y destino.

A pesar del uso de herramientas de virtualización para los ambientes de desarrollo y certificación, se puede notar que para la migración a SGBD de software libre y estándares abiertos, serán necesarios dos ambientes para poder migrar un SGBD, lo cual puede resultar en un número de ambientes difícil de manejar y administrar, motivo por el cual se recomienda estandarizar los nombres de los servidores de cada ambiente, relacionado con el SGBD que se está migrando.

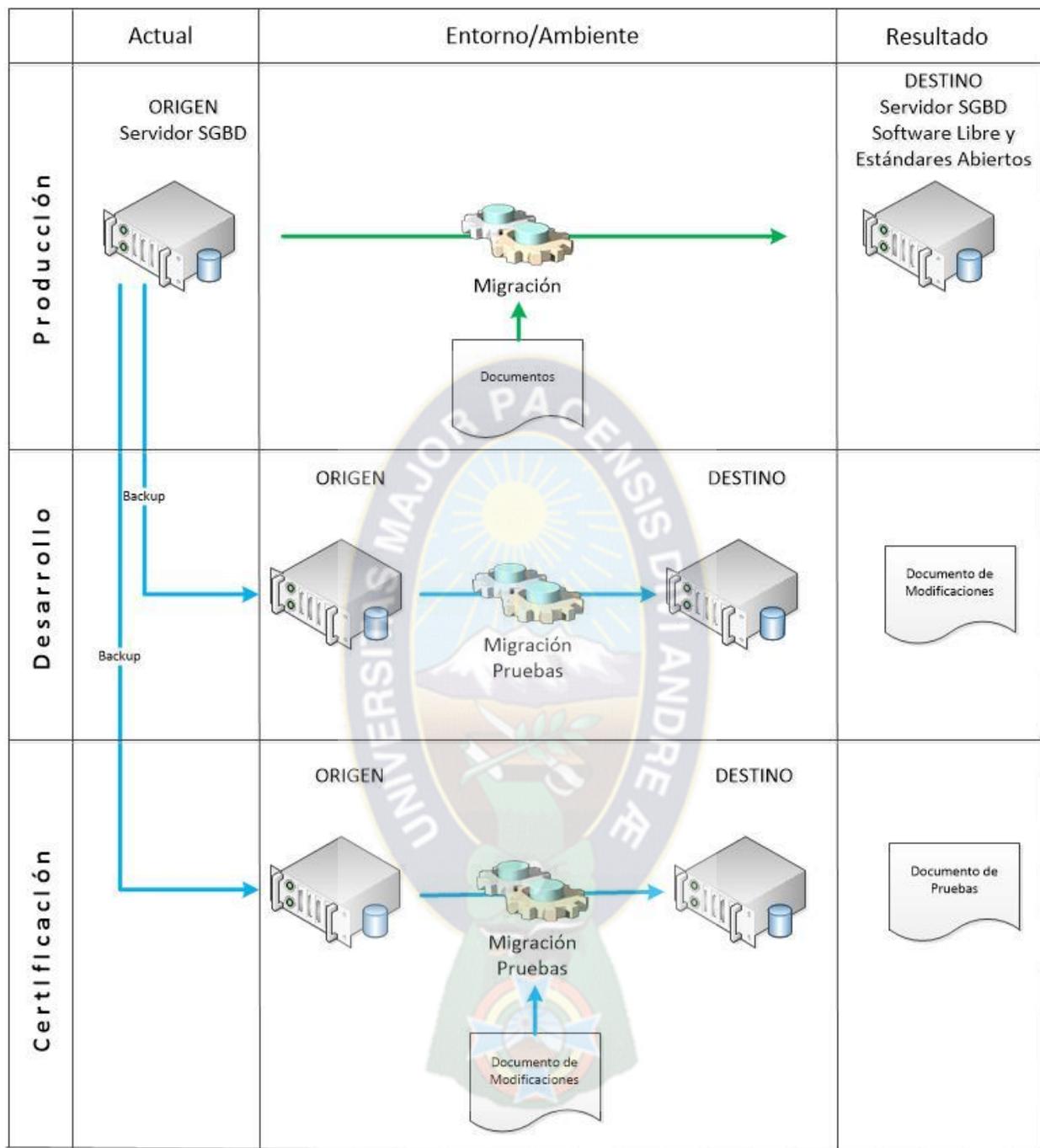


Figura 3, Ambientes necesarios para la migración a SGBD de software libre y estándares abiertos.

#### 4.1.1 Desarrollo

El entorno de desarrollo está compuesto por los siguientes servidores y procesos, vea la figura 4:

- a) Un servidor origen, este servidor es un SGBD lo más parecido posible al servidor de producción, mientras más se parezca al servidor de producción las pruebas de extracción de objetos, copia de datos y pruebas de compatibilidad en el servidor destino, serán más reales y podrá reproducir posteriormente cuando se lo realice en el entorno de producción,

lo que dará mayor seguridad, disponibilidad y confiabilidad al proceso de migración de SGBD de software libre y estándares abiertos. El responsable de la instalación, configuración y restauración del backup de este servidor es el Administrador de base de datos, vea la sección 4.4.2 de roles.

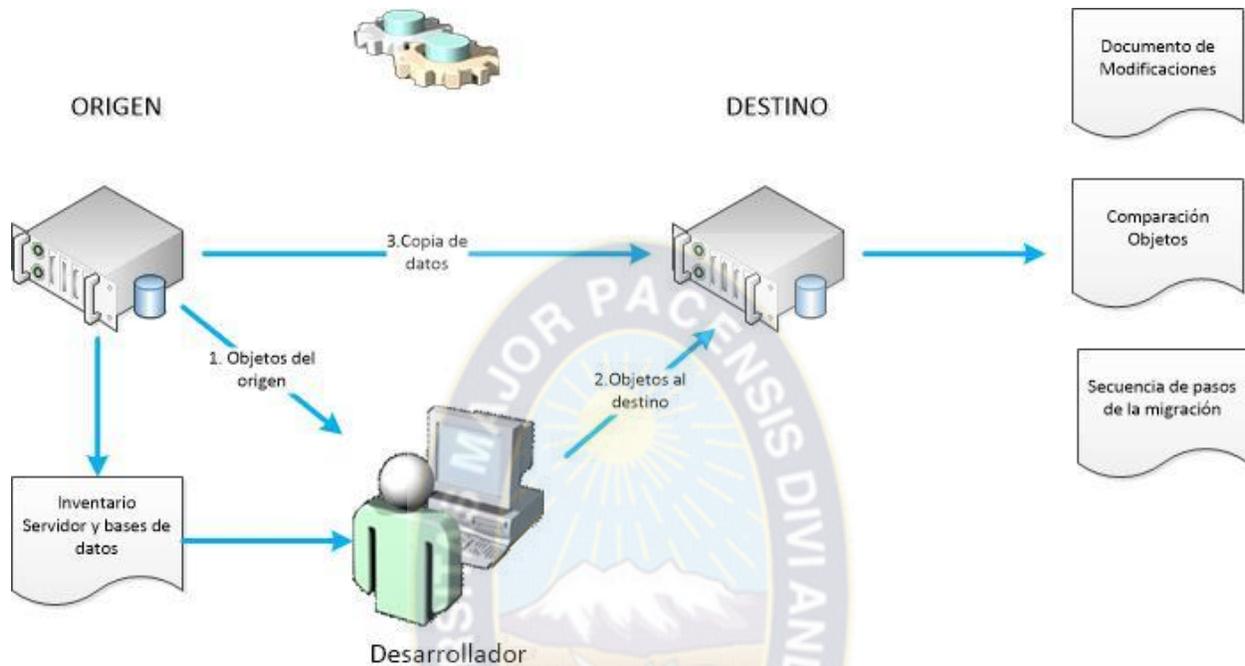


Figura 4, Ambiente de desarrollo para la migración a SGBD de software libre y estándares abiertos.

- b) El desarrollador, es la persona responsable de desarrollar el proceso de migración, probar la copia de datos desde el servidor origen al servidor destino y realizar la comparación de objetos entre el servidor origen y el servidor destino, las tareas y roles del desarrollador están descritas a mayor detalle en la sección 4.4.3.
- c) El servidor destino, que es el servidor que recibirá los objetos del servidor origen, además de ser el servidor que tendrá el SGBD de software libre y estándares abiertos que será utilizado posteriormente por los usuarios y sus respectivas aplicaciones. La instalación y configuración es responsabilidad del administrador de bases de datos, vea la sección de roles en la sección 4.4.2.
- d) Inventario del servidor y bases datos, es la línea base con la que trabajará el desarrollador, en base a este inventario realiza la extracción de objetos del servidor origen y verificará posteriormente que no falten objetos en el servidor destino.
- e) El proceso de extracción de objetos, es el proceso de sacar o extraer los objetos del servidor origen, para llevarlos y probarlos en el servidor destino. Este proceso se lo puede realizar de forma manual, uno por uno cada objeto. O realizarlo de forma automática, para lo cual el desarrollador deberá tener la capacidad de desarrollar herramientas que le permitan realizar este trabajo de forma rápida y eficiente.

- f) El proceso de llevar los objetos al servidor destino, después de ser extraído del servidor origen el objeto se lleva al servidor destino, donde debe ser probado su compatibilidad, en caso de error se debe registrar y documentar la modificación.
- g) El proceso de copia de datos, que se realiza una vez extraídos los objetos de tipo tablas en las bases de datos y llevados al destino, el número de filas de cada tabla debe ser la misma que el servidor origen.
- h) Documentación, donde el desarrollador genera tres tipos de documentos: Documento de modificaciones, de comparación de objetos y de secuencia de pasos de la migración, el contenido de cada documento se puede ver en los Anexos H, I y J.

### 4.1.2 Certificación

El entorno de certificación está compuesto por los siguientes servidores y procesos, vea la figura 5:

- a) Servidor origen, debe ser un servidor SGBD lo más parecido posible al servidor de producción, el backup o respaldo debe ser el mismo que se utilizó para el servidor origen del entorno de desarrollo.

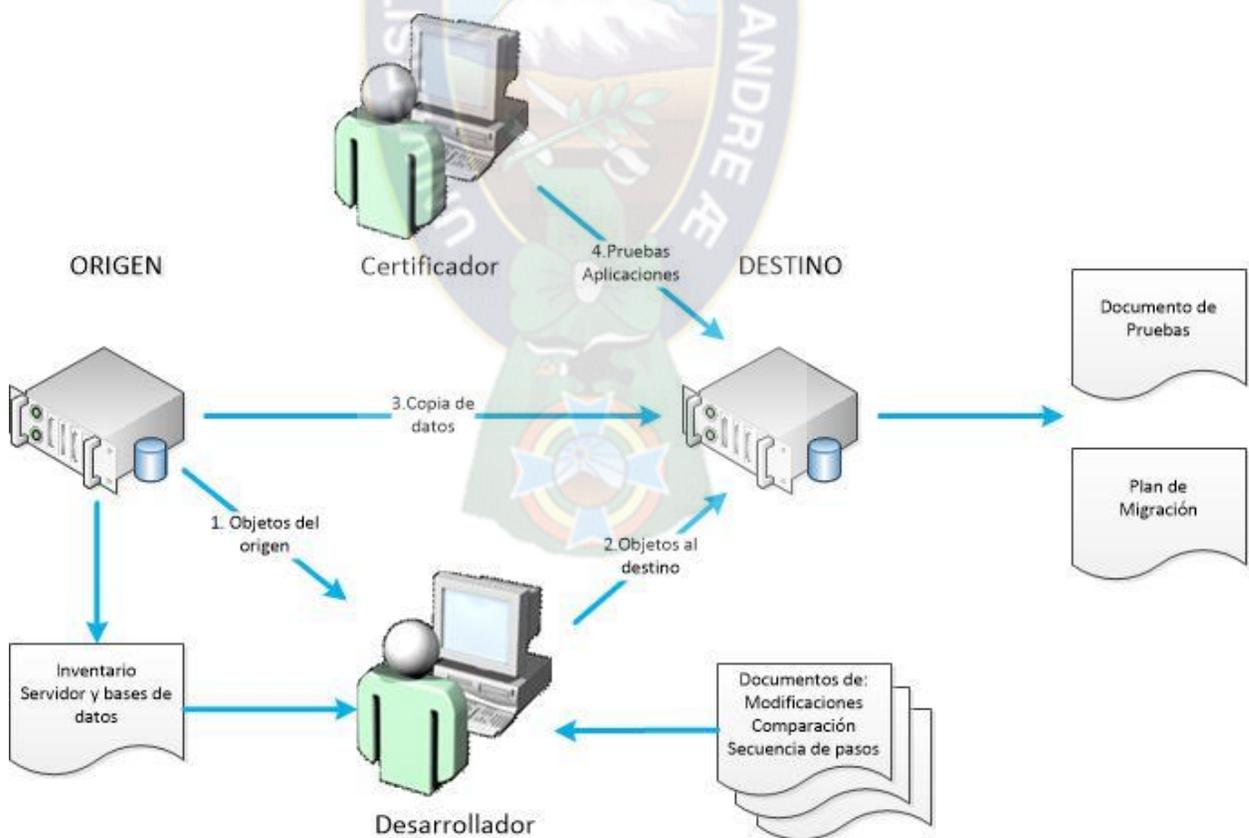


Figura 5. Ambiente de certificación para la migración a SGBD de software libre y estándares abiertos.

- b) Servidor destino, es el servidor SGBD de software libre y estándares abiertos donde se realizará el proceso de migración desarrollado en el entorno de desarrollo.
- c) El proceso de migración, es la aplicación del proceso desarrollado en el entorno de desarrollo, considerando los documentos de modificaciones, de comparación de objetos y de secuencia de pasos de la migración que fueron elaborados en el proceso de desarrollo, como entradas.
- d) Certificador, es la persona responsable por realizar las pruebas unitarias y de sistemas para aquellos componentes que hayan sido modificados, producto del desarrollo de la migración.
- e) Documentos de pruebas realizadas y plan de migración, es responsabilidad del certificador elaborar el documento de las pruebas realizadas. El plan de migración debe ser elaborado conjuntamente el administrador de la base de datos y el administrador del proyecto.

### **4.1.3 Producción**

Si bien el ambiente de producción ya tiene su servidor origen de SGBD funcionando, la idea es que en este ambiente se realice la migración a SGBD de software y estándares abiertos, está conformado por:

- a) Servidor origen, que es el servidor que está funcionando actualmente brindando servicios en el entorno de producción.
- b) Servidor destino, que será el nuevo servidor con el SGBD de software libre y estándares abiertos, estará configurado con las recomendaciones, pruebas realizadas y contenidas en el plan de migración, producto del desarrollo de la migración así como su certificación, la persona responsable de la instalación y configuración de este servidor es el administrador de bases de datos (DBA). Vea Sección 4.4 roles y responsabilidades.
- c) El equipo de migración, es el conjunto de personas lideradas por el Project Manager y conformado por el administrador de base de datos, desarrollador y certificadores, quienes realizan la migración en el entorno de producción utilizando los documentos producidos en el desarrollo de la migración y la certificación.
- d) Documentos, este ambiente utiliza los documentos producidos en el ambiente de desarrollo (Documento de modificaciones, de comparación de objetos y secuencia de pasos de la migración) y el ambiente de certificación (Documentos de pruebas y plan de migración)

Antes de encarar la migración en este entorno, se debe tener presente que este servidor sufre cambios en los distintos objetos del SGBD por las operaciones del día a día que se realizan, por lo que se debe considerar una fecha donde se congelan los cambios a objetos del SGBD y se obtiene un nuevo backup que se restaura en el entorno de desarrollo, esto con la finalidad de descubrir nuevos cambios que se hayan producidos a los objetos del SGBD, los cuales deben pasar por el mismo proceso de desarrollo y pruebas. Para mayor detalle puede ver la sección 4.2.5 Control de cambios.

## ***4.2 La Migración***

El proceso de migración, está dividido en varias etapas, donde la primera etapa es la realización del inventario del SGBD que se va migrar y su categorización, posteriormente se desarrolla la migración y se la prueba, se migran o copian los datos al servidor destino y finalmente se realiza una comparación de objetos para verificar que se haya llevado todos los objetos del servidor origen al servidor destino.

El proceso de migración es realizado por un equipo de personas que está conformado por el Project manager, el administrador de base de datos, el desarrollador de la migración y los certificadores quienes son los que finalmente dan el visto bueno de la migración. Vea sección 4.4 Roles y Responsabilidades.

A continuación se describen y detallan los pasos de cada etapa del proceso de migración.

### ***4.2.1 Inventario y categorización***

El inventario es una de las primeras actividades que se deben realizar, ya sea que se esté realizando la factibilidad de la migración o se esté encarando el proceso de migración, para la ejecución del inventario se pueden utilizar los formulario descritos en los Anexos F y G.

Se debe considerar, que en general todo SGBD tiene dos tipos de objetos: objetos a nivel servidor y objetos a nivel bases de datos, ya que el Anexo F servirá para los objetos a nivel servidor, mientras el Anexo E se podrá utilizar para los objetos a nivel base de datos.

El inventario generalmente se la realiza en el entorno de desarrollo contra el backup proporcionado del servidor origen de producción, esto para evitar incongruencias entre el inventario y el backup del entorno de desarrollo.

Una vez obtenido el inventario, se debe proceder a categorizar los objetos, sobre todo aquellos objetos que son del tipo base de datos, esta categorización podría ser considerada por su importancia (Alta, media y baja), por su criticidad (Alta, media y baja) o por su tamaño (base de datos con más de 40 Gb, etc.) o cada entidad o institución podría brindar su propio criterio de categorización.

La idea de la categorización, es poder determinar que objetos de bases de datos son más importantes, críticos o de mayor tamaño para considerarlos en el proceso de desarrollo de la migración y como se van a trabajar estos objetos de bases de datos, por lo general los objetos de bases de datos que reciben la categorización son las tablas y vistas ya que contienen la información de la institución.

### ***4.2.2 Desarrollo de la migración y pruebas***

Una vez restaurado el backup en el servidor origen del entorno de desarrollo, comienza el trabajo del desarrollador extrayendo cada uno de los objetos del SGBD origen, un objeto puede ser referido a una base de datos que contiene otros objetos como tablas, vistas, índices, procedimientos almacenados, triggers, etc. También se deben considerar otros tipos de objetos al margen de las base de datos, como usuarios (logins) y sus respectivos permisos sobre objetos, también se debe considerar que cada SGBD tiene sus propios objetos, por ejemplo el Microsoft SQL Server utiliza

objetos llamados job y dtsx<sup>33</sup> que se deben considerar el momento de extraer los objetos del SGBD origen. Como se nota en la Figura 4, la extracción de objetos del SGBD debe estar basado en el inventario del servidor origen, así como de cada objeto que contiene cada base de datos. El backup realizado y restaurado en el SGBD origen deben tener concordancia con el inventario tomado del servidor y sus bases de datos.

Los pasos que debe seguir el desarrollador, así como la descripción de cada paso se detalla a continuación:

- a) Revisa que el inventario del servidor origen y sus bases de datos, sea congruente con el backup restaurado en el servidor origen del entorno de desarrollo.
- b) Debe considerar cada objeto a nivel SGBD, el principal objeto que interesa son las bases de datos y se deben considerar otros objetos de acuerdo al SGBD y su versión respectiva. Por ejemplo Logins y permisos.
- c) Extraer cada objeto a nivel servidor y probarlo en el servidor SGBD destino, en caso que existan errores o incompatibilidad en la creación de objetos deben ser corregidos, probados y documentar la modificación que se debe realizar en el SGBD destino. Este paso puede ser automatizado con el desarrollo de herramientas, vea la sección 4.5 Herramientas a desarrollar.
- d) Documentar aquellos objetos que tuvieron que ser modificados para soportar la compatibilidad del SGBD de software libre y estándares abiertos. Vea el Anexo H.
- e) Realiza la comparación de objetos entre el servidor origen y el servidor destino, con el propósito que se haya desarrollado y probado todos los objetos del servidor origen. Para este propósito utiliza los Anexos H e I.
- f) Realiza y documenta la secuencia de pasos que serán necesarios llevar adelante para el proceso de migración. Vea el Anexo K.

### **4.2.3 Migrar los datos**

Migrar los datos, no es otra cosa que la copia de datos de las tablas del servidor o SGBD origen al servidor o SGBD destino, para migrar los datos se debe tomar en cuenta lo siguiente:

- a) Crear los índices después de realizar la copia de datos, ya que el proceso de indexado producirá demora en el proceso de copia o migración de los datos.
- b) Considerar el tipo de dato de cada tabla en cada SGBD, por ejemplo el tipo de datos integer que se utiliza en Microsoft SQL Server no tiene el mismo número de bytes que el SGBD de software libre y estándares abiertos.

---

<sup>33</sup> DTSX, Data Transformation Services XML en inglés, o Servicios de Transformación de datos en XML

- c) También se debe considerar aquellas tablas de gran tamaño o volumen, ya que estas se podrían copiar por partes utilizando algún criterio, por ejemplo gestiones, años, departamentos, etc.

Los pasos que debe seguir el desarrollador, así como la descripción de cada paso se detalla a continuación:

- a) Copia las tablas, del servidor o SGBD origen al servidor o SGBD destino, esta tarea puede ser realizada utilizando sentencias DML o en su defecto desarrollando herramientas que permitan copiar tablas con determinado criterio.
- b) Asegurar que todas las tablas del servidor o SGBD origen, estén en el servidor o SGBD destino, para este fin utiliza el inventario de bases de datos y debe obtener un inventario de las bases de datos del servidor o SGBD Destino, vea los Anexos F y G, Inventario de Servidor y Bases de Datos.
- c) Verificar para cada tabla a ser migrada, que el número de filas de la tabla origen sea igual al número de filas de la tabla destino, caso contrario debe indagar e investigar las causas de las diferencias.
- d) Documentar el proceso de migración de datos, para ello utilizara el Anexo I, Comparación de Objetos (tablas)

#### ***4.2.4 Comparar origen y destino***

La comparación entre el servidor o SGBD de origen y el servidor o SGBD destino, se la realiza una vez que se ha finalizado la extracción de los objetos del servidor origen y han sido llevados al servidor destino y migrado sus datos.

El objetivo de comparar lo objetos a nivel servidor y a nivel base de datos, es asegurar que todos los objetos (a nivel servidor y bases de datos) estén en el servidor o SGBD destino. Vea el Anexo I para las tablas y el anexo J para el resto de objetos.

Los pasos que debe seguir el desarrollador para realizar la comparación de objetos entre el servidor origen y el servidor destino, así como la descripción de cada paso se detalla a continuación:

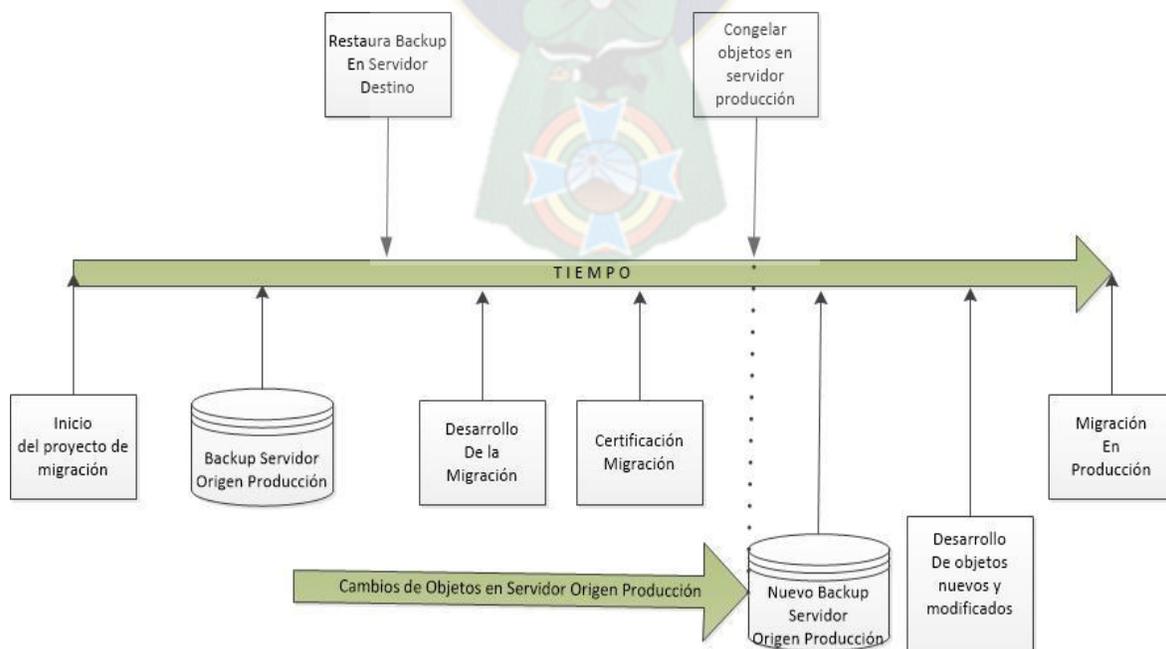
- a) El inventario base es el inventario obtenido, del servidor de origen, en el entorno de desarrollo.
- b) Se obtiene un nuevo inventario de objetos a nivel servidor y bases de datos, del servidor o SGBD destino.
- c) Se comparan los inventarios, del servidor origen contra el inventario del servidor destino, para determinar si falta algún objeto que no fue migrado. También se debe comparar los números de filas de cada tabla del servidor origen, con su respectiva tabla y número de filas del servidor destino.

- d) Documentar la comparación del servidor origen y el servidor destino. Vea los Anexos I para las tablas y el Anexo J para los demás objetos.
- e) En caso de encontrar diferencias entre el servidor origen y el servidor destino, se debe indagar e investigar las mismas, con la finalidad de arreglarlas o en caso contrario documentarlas.

#### 4.2.5 Control de cambios

El control de cambios se produce debido a cambios que se realizan a los objetos de los servidores o SGBD origen del entorno de producción, mientras se realiza el desarrollo, pruebas y certificación de la migración, por ejemplo cambios a tablas, vistas, índices, procedimientos almacenados, etc. El control de cambios también se produce por la información que se almacena en los objetos de tipo tablas, que constantemente están recibiendo, actualizando y eliminando información, por esta razón se debe considerar realizar dos tipos de control de cambios del servidor o SGBD origen de producción.

- a) El primero relacionado con no permitir actualizaciones de objetos del servidor o SGBD origen de producción, o como se lo llama comúnmente, congelar los cambios de objetos, generalmente este congelamiento se la realiza dos semanas antes de la migración del servidor de producción, esto para dar oportunidad de desarrollar los nuevos cambios, modificaciones y eliminación de los objetos del servidor o SGBD origen.
- b) El segundo control de cambio está más relacionado con la información que contienen los objetos de tipo tablas, en este caso el congelamiento de la información se la debe realizar horas antes de la ejecución del proceso de migración en el entorno de producción, caso contrario los resultados de las comparaciones de los objetos tablas en su números de filas no serán congruentes.



El encargado de realizar y comunicar de los congelamientos en el ambiente de producción es el administrador de bases de datos. Los pasos que debe seguir el desarrollador para realizar el control de cambios, así como la descripción de cada paso se detalla a continuación:

- a) Solicita un nuevo backup del entorno de producción, al administrador de bases de datos (DBA), una vez definida y realizada el congelamiento de objetos del servidor de producción.
- b) Restaura el nuevo backup en el entorno de desarrollo, en el servidor origen.
- c) Obtiene un nuevo inventario a nivel servidor y a nivel bases de datos.
- d) Compara el inventario nuevo con el primer obtenido para determinar que objetos fueron creados, eliminados o modificados.
- e) Extrae los objetos nuevos o modificados del servidor origen del entorno de desarrollo y lo lleva al servidor destino del entorno de desarrollo con el objetivo de probar la compatibilidad de los objetos nuevos o modificados. Aquellos objetos que fueron eliminados, del servidor origen, deben ser eliminados del servidor destino.
- f) Actualizar el documento de modificación de objetos, con los objetos nuevos, modificados o eliminados. También se debe actualizar el documento de comparación de objetos.
- g) Probar el proceso de migración con la migración de datos, para verificar que no falten objetos y que la cantidad de filas de las tablas del servidor origen sean las mismas del servidor destino.
- h) Comunicar de los cambios realizados a los certificadores, para que estos consideren si es necesario realizar las pruebas de las aplicaciones.

### ***4.3 Pruebas de las aplicaciones***

Existen dos tipos de pruebas que se deben realizar a todos aquellos objetos que hayan sido modificado producto de la migración al servidor destino como consecuencia de adaptaciones por incompatibilidad con el servidor destino, estas pruebas son unitarias de los componentes u objetos modificados y las pruebas de sistemas, este último con el propósito de comprobar el impacto de la modificación en el componente en todo el sistema.

Cuando las pruebas son realizadas por el desarrollador se llevan a cabo en el ambiente de desarrollo, mientras que si son realizadas por el certificador se deben realizar en el ambiente de certificación.

El propósito del certificador es probar el impacto de los cambios en la base de datos en las aplicaciones, por esta razón se debe realizar previamente la migración de la base de datos con todos sus objetos e información y cambiar las cadenas de conexión de las aplicaciones para que apunten al SGBD o servidor destino.

### 4.3.1 Pruebas unitarias

Las pruebas unitarias son realizadas por el desarrollador en el proceso de migrar cada objeto o componente al servidor destino, en este proceso el desarrollador debe verificar o probar que los cambios aplicados, que son generalmente por compatibilidad entre el servidor origen y el servidor destino, no provoquen cambios en el funcionamiento del objeto o componente tratado.

En este sentido el desarrollador realiza o ejecuta el objeto o componente contra el servidor origen y registra el resultado, como por ejemplo el número de filas que arroja como resultado el componente. Posteriormente procede a probar el objeto o componente modificado contra el servidor destino, el resultado debe ser el mismo que se registró del objeto o componente contra el servidor origen (vea la figura 6), en caso que no se obtengan los mismos resultados, se debe indagar por las causas de las diferencias, finalmente si las diferencia persisten, el objeto o componente debe ser registrado con observaciones, vea el Anexo L.

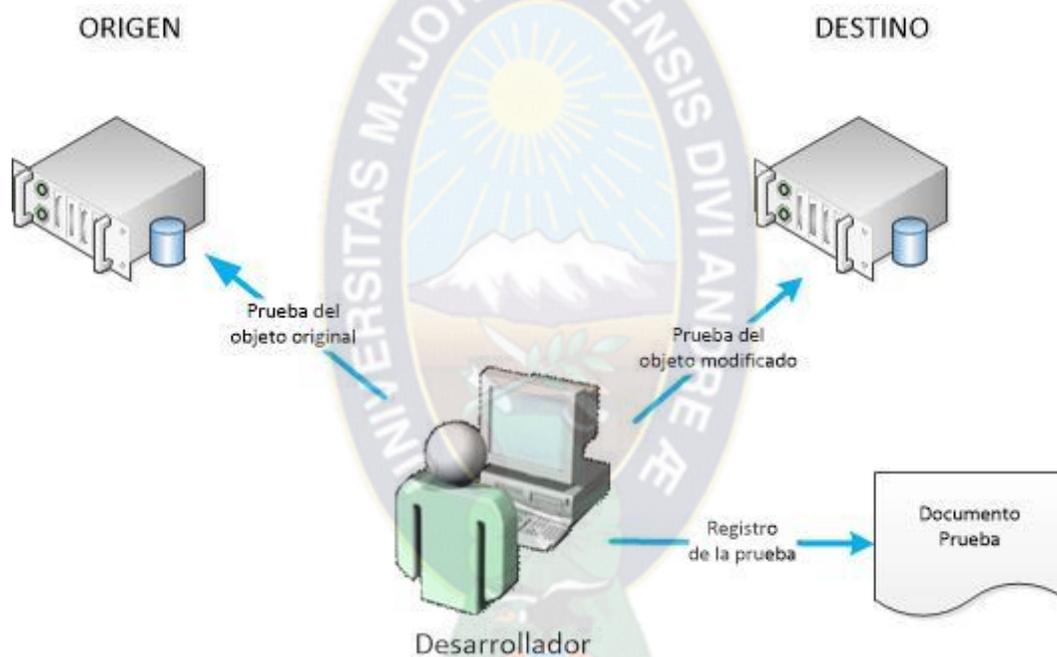


Figura 6, Pruebas unitarias realizadas por el desarrollador

Las pruebas contra el servidor destino, se deben realizar una vez realizada la copia de datos del servidor origen al servidor destino y de verificar que la copia o migración de datos fue satisfactoria y sin diferencias entre el número de filas de las tablas del servidor origen contra el número de filas de las tablas del servidor destino.

Es una buena práctica que las pruebas unitarias sean realizadas, además, por otro desarrollador, esto para asegurar la calidad de la prueba unitaria. El desarrollador generalmente realiza las pruebas mientras unitarias y no así las pruebas de sistemas, las cuales están reservadas para los certificadores, quienes deben probar el total de las aplicaciones que acceden al SGBD que se está migrando.

### 4.3.2 Pruebas de sistemas

Los certificadores son los responsables por llevar adelante las pruebas de sistemas, lo cual realizan en el entorno de certificación definido en la sección 4.1.2. Los certificadores no son personal de sistemas, al contrario son usuario que conoce muy bien el sistema de información que utilizan los SGBD que se están migrando.

Los certificadores reciben los documentos de modificaciones, así como el documento de comparación, esto con el propósito de enfocar las pruebas en aquellas áreas que hayan sido modificadas.

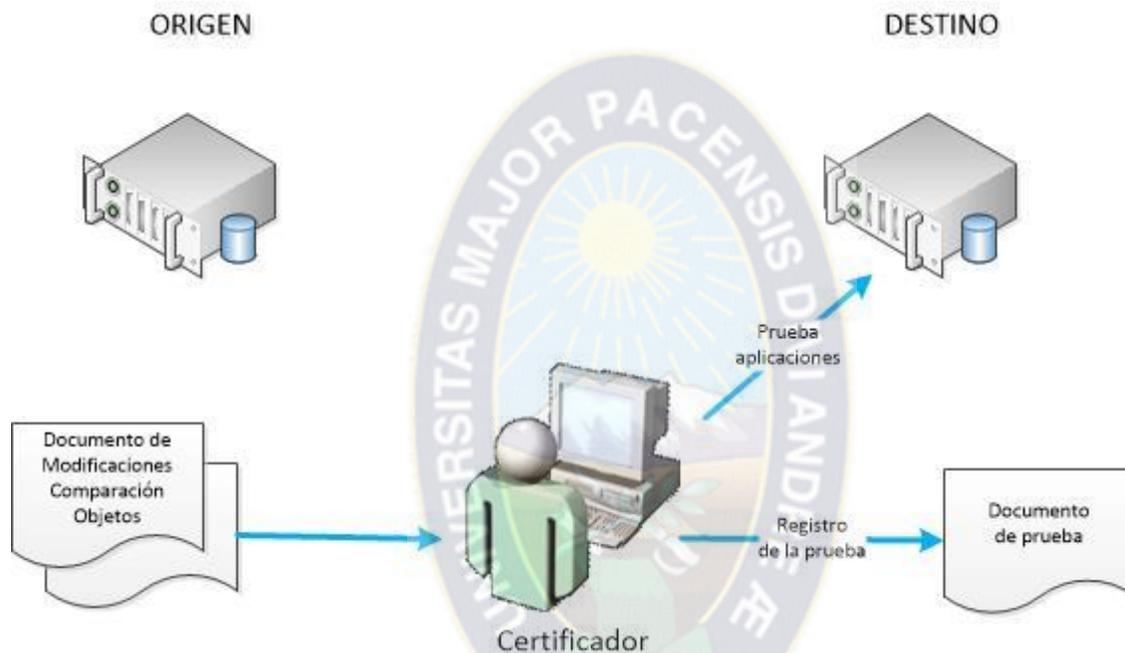


Figura 7, Prueba de sistemas

El certificador, es justamente quien da el visto bueno o no, que las aplicaciones o sistemas de información, funcionarán con acceso al SGBD de software libre y estándares abiertos. Razón por la cual es importante seguir de cerca las pruebas que se vayan a realizar. Luego del visto bueno del certificador se procede a realizar la migración en producción. En caso que el certificador no proporcione su visto bueno se debe indagar las razones y motivos.

El certificador realiza una prueba de aceptación del aplicativo o sistema de información con el objetivo de certificar que el aplicativo será válido con SGBD de software libre y estándares abiertos, si las pruebas documentadas a satisfacción del certificador, el aplicativo o sistema de información serán considerados correcto y por lo tanto adecuado para su puesta en producción.

## 4.4 Roles y responsabilidades

A continuación se definen los roles y responsabilidades del personal que estará involucrado en el proceso de migración a SGBD de software libre y estándares abiertos, en la medida de lo posible cada rol debe ser asignado a una persona diferente, sobre todo en el caso de los desarrolladores y

certificadores, esto para evitar que sean juez y parte en el desarrollo de la migración y la prueba del desarrollo.

#### **4.4.1 Administrador del proyecto**

El Project Manager o administrador de proyectos es la persona que administra o gestiona todo el proceso de migración a SGBD de software libre y estándares abiertos como un proyecto, es un profesional especialista y experto en administración de proyectos de migración de SGBD, realiza las siguientes tareas y actividades:

- a) Realiza la planificación del proyecto, estimando tiempos de cada tarea y actividad.
- b) Define las tareas, actividades y responsables de cada tarea y actividad.
- c) Establece el cronograma de desarrollo de la migración y las pruebas de las aplicaciones.
- d) Proporciona la logística necesaria para el proyecto.
- e) Coordina las actividades del equipo.
- f) Informa sobre el estado de avance del proyecto de migración a SGBD de software libre y estándares abiertos.
- g) Con la colaboración del administrador de la base de datos y los desarrolladores, genera el plan de migración para producción.
- h) Supervisa la migración a SGBD de software libre y estándares abiertos en el entorno de producción.

#### **4.4.2 Administrador de bases de datos**

Administrador de base de datos, o DBA (por sus siglas en inglés) tiene como responsabilidad principal<sup>34</sup>:

- Decidir qué datos se deberán almacenar en la base de datos.
- Decidir por la estructura de almacenamiento y la estrategia de acceso.
- Vincularse con los usuarios.
- Definir los controles de autorización y los procedimientos de validación.
- Definir una estrategia de respaldo y recuperación.
- Controlar el desempeño y responder a los cambios de requerimientos.

El administrador de bases de datos es un profesional especialista en DDL (Data Definition Language, por sus siglas en inglés) lenguaje de especificación de datos y en DML (Data Manipulation Language) lenguaje de manipulación de datos, tanto en el SGBD origen, como en el SGBD de software libre y estándares abiertos.

El administrador de bases de datos, en el proceso de migración a SGBD de software libre y estándares abiertos, se encarga de proporcionar el SGBD de origen y destino, a través de realizar las siguientes tareas:

- a) Realiza el inventario a nivel servidores y bases de datos del SGBD origen, ver anexo F y G.

---

<sup>34</sup> DATE C. J., Introducción a los sistemas de Bases de Datos, Séptima Edición 2001, pp.28

- b) Realiza la instalación y configuración del servidor de SGBD origen en sus diferentes ambientes (desarrollo y certificación)
- c) Realiza la configuración del SGBD origen (desarrollo y certificación), para que quede, lo más parecido posible al ambiente de producción.
- d) Realiza el backup de las bases de datos del ambiente de producción, así como los objetos que pueda tener el SGBD origen y restaura estas bases de datos y objetos en los ambientes de desarrollo y certificación en los SGBD origen.
- e) Realiza la instalación y configuración del servidor de SGBD de destino en sus diferentes ambientes (desarrollo y certificación).
- f) Realiza la configuración del SGBD destino (desarrollo, certificación y producción), considerando que esta configuración podrá ser la adoptada para la configuración del SGBD destino en producción.
- g) Define los accesos tanto en los SGBD origen como en el destino para las diferentes usuarios que van a participar en la migración, por ejemplo los desarrolladores deben tener acceso a nivel administrador al SGBD destino, pero no así al SGBD origen, donde deben tener privilegios de acceso de lecturas a todas las bases y objetos del SGBD origen.
- h) Realizar la documentación de las configuraciones realizadas en los SGBD destino.
- i) Participa como ejecutor del proceso de migración en el ambiente de producción.

#### **4.4.3 Desarrolladores**

El o los desarrolladores tienen como función principal realizar y desarrollar la migración a SGBD de software libre y estándares abiertos, por esta razón los desarrolladores deben ser profesionales expertos en DDL y DML, tanto en el SGBD origen como en el SGBD destino. Los desarrolladores deberán cumplir con las siguientes tareas o actividades en el proceso de migración a SGBD de software libre y estándares abiertos:

- a) Toma como línea base el inventario de servidores y bases de datos, realizados a través de los anexos F y G.
- b) Realiza su trabajo en el entorno de desarrollo preparado para tal efecto.
- c) Extrae los objetos a nivel base de datos (tablas, vistas, procedimientos almacenados, índices, triggers, etc.) y a nivel servidor del SGBD origen, para llevarlos al SGBD destino y probar su compatibilidad.
- d) Realiza cambios a aquellos objetos que sean incompatibles en el SGBD destino.
- e) Realiza pruebas de aquellos objetos que tuvieron que ser modificados por incompatibilidad en el SGBD destino, compara el resultado contra el SGBD origen.
- f) Realiza pruebas de migración de datos entre el SGBD origen y el SGBD destino, considerando volúmenes y tamaños de la información de las diferentes tablas.
- g) Compara el SGBD destino contra el SGBD origen con el propósito de determinar los cambios realizados y que no falten objetos por migrar.
- h) Realiza el cambio de la cadena de conexión en los programas y aplicativos, para que apunten al SGBD destino.
- i) Realiza la documentación de las pruebas realizadas en el SGBD destino.
- j) Realiza la documentación de las incompatibilidades encontradas entre el SGBD origen y el SGBD destino, así como los cambios realizados.

- k) Participan en el proceso de migración en el ambiente de producción comparando los objetos del servidor origen contra el servidor destino y dando el soporte necesario en caso de algún inconveniente en el proceso de migración.

Es necesario que los desarrolladores tengan acceso al entorno de producción, esto con el propósito de comparar resultados o verificar logs de ejecución, actividad que debe ser solicitada y coordinada con el administrador de bases de datos.

#### **4.4.4 Certificadores**

Son los usuarios que utilizan la información contenida en los SGBD desde estaciones de trabajo a través de aplicaciones o programas. En el proceso de migración a SGBD de software libre y estándares abiertos tienen la responsabilidad de realizar pruebas de los programas y aplicativos que certifiquen que funcionaran con el SGBD de software libre y estándares abiertos. Realiza las siguientes tareas y actividades:

- a) Realiza su trabajo en el entorno de certificación preparado para tal efecto.
- b) Realiza las pruebas de los programas y aplicativos contra el SGBD de destino, con la finalidad de encontrar errores.
- c) Realiza la documentación de las pruebas realizadas y sus resultados.
- d) Certifica que los programas y aplicativos funcionan con el SGBD Destino basados en las pruebas realizadas.

### **4.5 Herramientas a desarrollar**

Para el desarrollo de herramientas que ayuden y soporten el proceso de migración a SGBD de software libre y estándares abiertos, se puede considerar utilizar lenguajes de programación como JAVA o C# de .NET, que son las que proporcionan acceso y trabajan con distintos SGBD tanto desde la perspectiva del servidor origen como la perspectiva del servidor destino.

En general para poder desarrollar las herramientas que soporten el proceso de migración a SGBD de software libre y estándares abiertos, se debe considerar el acceso a la meta data del SGBD del servidor origen a nivel servidor y a nivel base de datos, para ello puede recurrir a los apéndices del presente trabajo, donde se define la arquitectura de las principales SGBD comercial, propietario o privativo, así como los principales SGBD de software libre y estándares abiertos.

Para la especificación de las herramientas a desarrollar, se definirán las entradas que van a utilizar ya sea como parámetros o a través de entrada de datos. Los proceso o pasos que deben seguir, Y las salidas ya sea como reporte u otro tipo de archivo.

#### **4.5.1 Inventario de objetos**

Básicamente el inventario de objetos se realiza con la finalidad de contar los objetos y mostrar su detalle, es decir cómo está compuesto, de forma que posteriormente se pueda comparar con los objetos que son llevados al servidor destino y verificar que no estamos olvidando ningún objeto. Otro objetivo importante del inventario es poder identificar aquellos objetos puedan provocar incompatibilidad en el servidor destino o sean complicados en su tratamiento de llevar al servidor destino, como tablas de gran tamaño.

Entradas, recibe dos entradas como parámetros. El Nombre del servidor del cual se obtendrá el inventario y las credenciales (usuario y password) de acceso al servidor.

Procesos, son los siguientes:

- a) Determina el SGBD y su versión, por ejemplo Microsoft SQL Server 2008 R2.
- b) Dependiendo del SGBD se arma la cadena de conexión considerando las entradas y se conecta al SGBD.
- c) En la metadata del SGBD se buscan los objetos que sean bases de datos, por cada objeto base de datos se debe realizar lo siguiente:
  - a. Se accede a la metadata de la base de datos.
  - b. Por cada objeto (Tablas, vistas, procedimientos almacenados, índices, triggers, etc), se procede a contar el número de objetos en cada caso.
  - c. Para el caso de los objetos tablas se debe obtener el número de filas de cada tabla como así el tamaño en MB que tiene.
- d) Si existen otros objetos que no sean de bases de datos, deben ser considerados y enumerados de acuerdo a su tipo de objeto, también es importante poder obtener el detalle de cada objeto que no sea de bases de datos.

Salidas, Produce un reporte basado en los Anexo D para objetos a nivel servidor y otro reporte basado en el Anexo F para los objetos a nivel base de datos. El reporte puede ser generado en formato HTML.

#### **4.5.2 Extraer objetos del origen**

En la figura 4 de la sección 4.4.1 Ambiente de desarrollo, se definió el proceso el proceso de sacar objetos del servidor origen para poder llevarlos al servidor destino y probar su compatibilidad, este proceso puede ser bastante moroso y complicado si se considera que cada objeto de base de datos puede contener de cien (100) a quinientos (500) objetos, entre tablas, vistas, procedimientos almacenados, etc. Lo cual le tomaría al desarrollador un tiempo considerable sacar cada objeto y llevarlo al servidor destino, para acelerar este proceso es que se propone la herramienta de extraer objetos.

Entradas, recibe como parámetros el nombre de servidor origen, sus credenciales, el nombre de servidor destino y sus credenciales respectivas.

Procesos, realiza lo siguiente:

- a) Determina el SGBD origen y su versión, así como el SGBD destino y su versión.
- b) Se arma las cadenas de conexión para el servidor origen, así como para el servidor destino, se procede a conectar al servidor origen como al servidor destino.
- c) En la metada del SGBD origen se buscan los objetos que sean bases de datos y se obtiene en forma de script sus características, como nombre de la base de datos, su tamaño y donde está almacenado.
- d) Para cada objeto base de datos, se accede a su metadata, para buscar los diferentes objetos que tiene y obtener su script (sentencias SQL que componen el objeto, vea la figura 7)
  - a. Si es Tabla, se debe sacar el nombre de la tabla, las columnas que las componen, sus tipos de datos de cada columna y su tamaño.

```

SET ANSI_NULLS OFF
SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tabla] (
    [numero_doc] [char](11) NOT NULL,
    [codigo_tipo] [char](2) NOT NULL,
    [codigo_correlativo] [char](2) NOT NULL,
    [nro_crv_trf_pgm] [numeric](9, 0) NOT NULL,
    [nro_crv_det_aga] [numeric](9, 0) NOT NULL,
    [descripcion_dia] [char](10) NOT NULL
)
ON [PRIMARY]
WITH (DATA_COMPRESSION=PAGE)
GO

```

Figura 8. Ejemplo de script de una tabla

- b. Si es una vista, se saca la consulta/query que compone la consulta.
- c. Si es un Procedimiento Almacenado, se debe sacar las sentencias SQL que componen el procedimiento almacenado.
- d. Si es un índice, se saca a que tabla hace referencia y que columna(s) utiliza en el índice.
- e) Para cada objeto extraído, se procede a probar el objeto en el servidor destino.
- f) El orden de creación de los objetos en el servidor destino será: los objetos base de datos, luego las tablas y vistas, seguido por los procedimientos almacenados, en lo posible se debe tratar de evitar la creación de los índices antes de la copia de los datos.
- g) Cada objeto extraído, se guarda el script en un archivo que tenga como nombre el nombre del objeto y tenga como extensión .sql.
- h) Se debe llevar un log detallado de la ejecución de los objetos en el servidor destino, esto con la finalidad de poder documentar en caso de error de compatibilidad.
- i) Para cada objeto de tipo tabla se debe copiar su información del servidor origen al servidor destino.

Salidas, para poder sacar los objetos del servidor origen se propone la siguiente estructura de carpetas, vea figura 8, formado por el nombre de base de datos y dentro de cada carpeta de base de datos las diferentes carpetas para cada objeto de la base de datos, donde estarán almacenados los objetos y su script de cada tabla, vista, procedimiento almacenado, etc. La carpeta principal (origen) podría ser el nombre del servidor origen.

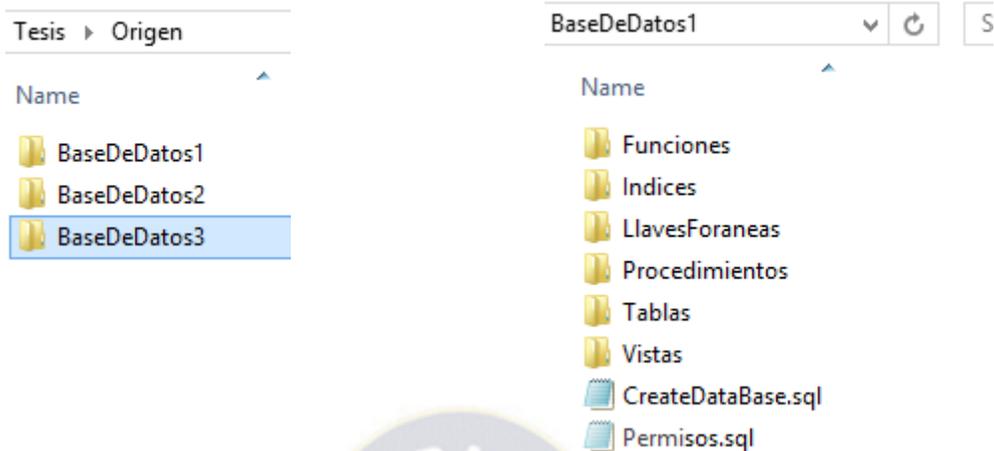


Figura 9, Estructura de carpetas para la extracción de objetos

### 4.5.3 Comparación de objetos

La idea de tener una herramienta que me permita comparar objetos, es poder determinar que objetos no se están llevando al servidor destino, o dicho de otra forma que objetos se están olvidando durante el proceso de migración.

Se deben realizar dos tipos de comparaciones, una relacionada con los procedimientos almacenados y su contenido, esto para determinar qué cambios se realizaron en estos y que sean fácilmente identificables. La segunda comparación se realiza a nivel objetos de cada base de datos, es decir se identifica un objeto, digamos una tabla en el servidor origen, esta tabla debe estar en el servidor destino, así como se cuenta sus número de filas de cada tabla y se compara contra el número de filas de la misma tabla en el servidor destino. En caso que no exista el objeto se debe registrar la tabla con su observación respectiva que no existe en el servidor destino.

Entradas, recibe como parámetros el nombre de servidor origen, sus credenciales, el nombre de servidor destino y sus credenciales respectivas, en total 6 parámetros.

Procesos,

- a) Determina el SGBD origen y su versión, así como el SGBD destino y su versión.
- b) Se arma las cadenas de conexión para el servidor origen, así como para el servidor destino, se procede a conectar al servidor origen como al servidor destino.
- c) En la metada del SGBD origen se buscan los objetos que sean bases de datos y se busca en el servidor destino si existe dicho objeto.
- d) Para cada objeto de base de datos del servidor origen, se procede a buscar su correspondiente objeto en el servidor destino. En caso que no exista se lo registra en el reporte como no existe en servidor destino.

Salidas, se debe producir un reporte, que puede ser de tipo texto, donde se considera los anexos G y H para el formato y nombre de las columnas.

## *Capítulo V: Marco de resultados*

En este capítulo se encuentran los resultados cualitativos generales de la investigación (desde el diagnóstico, pasando por el proceso y la etapa de resultados). Asimismo, se detalla la validación de la propuesta, el estado de los objetivos, la comprobación de la hipótesis, conclusiones y recomendaciones.

### *5.1 Estado de los objetivos*

De los objetivos planteados se procede a describir cada uno de ellos y se describe en qué medida fueron cumplidos.

**Objetivo Específico.** “Determinar la factibilidad de la migración de los SGBD licenciados a SGBD de software libre y estándares abiertos, en las instituciones y entidades públicas del estado”.

De acuerdo a la sección 3.4 Propuesta de factibilidad, del capítulo 3, se puede determinar la factibilidad en tres áreas o niveles, Operativa, Técnica y Económica. Siendo de todas ellas la más importante la factibilidad operativa, ya que esta factibilidad indaga por la propiedad del código fuente de los aplicativos o sistemas de información que utilizan los SGBD a ser migrados a SGBD de software libre y estándares abiertos.

**Objetivo Específico.** “Diseñar y describir métodos, procedimientos y técnicas que permitan realizar la migración de forma segura de los SGBD licenciados a SGBD software libre y estándares abiertos en las instituciones y entidades públicas del estado y otras que así lo consideren”.

En el capítulo 4, en la sección 4.2 La migración, se diseñó y describió métodos, procedimientos y técnicas. En la sección 4.1 Ambientes de trabajo se definen los ambientes necesarios para realizar una migración de SGBD a SGBD de software libre y estándares abiertos.

**Objetivo Específico.** “Diseñar y describir métodos, técnicas y procedimientos que permitan probar y validar las aplicaciones que utilizan los SGBD que se van a migrar a software libre y estándares abiertos”.

En el capítulo 4, en la sección 4.3 Pruebas de aplicaciones, se diseñaron y describieron métodos, técnicas y procedimientos para probar y validar aquellas aplicaciones que utilizan los SGBD que se van a migrar a SGBD de software libre y estándares abiertos.

**Objetivo Específico.** “Definir y describir roles y responsabilidades de los participantes en el proceso de migración de los SGBD licenciados a SGBD de software libre y estándares abiertos”.

De acuerdo al capítulo 4, sección 4.4 Roles y Responsabilidades, se describen los roles y responsabilidades de cada uno de los participantes en el proceso de migración de SGBD comerciales, privativo o propietario a SGBD de software libre y estándares abiertos.

**Objetivo Específico.** “Establecer un marco de trabajo para la elaboración de herramientas que soporten el proceso de migración de los SGBD licenciados a SGBD de software libre y de estándares abiertos”.

En la sección 4.5 Herramientas a desarrollar del capítulo 4, se establece las herramientas que serán necesarias, para acelerar el proceso de los SGBD comerciales, privativo o propietario a SGBD de software libre y estándares abiertos.

**Objetivo General.** “Diseñar una metodología, que determine la factibilidad para realizar la migración de forma segura de los SGBD licenciados a SGBD de software libre y estándares abiertos en las instituciones y entidades públicas del Estado”.

En relación al objetivo general se puede concluir que, como los objetivos específicos fueron cumplidos, el objetivo general también están siendo cumplidos, debido a que se propone en el capítulo 4 métodos, procedimientos y técnicas, que inicialmente ayudarán a determinar si la migración a SGBD de software libre y estándares abiertos es factible, para posteriormente proponer una metodología en el proceso de migración.

## ***5.2 Estado de la hipótesis***

Una de las funciones de hipótesis es probar teorías, si aporta evidencia en su favor. Cuando varias hipótesis de una teoría reciben evidencia en su favor, la teoría va haciéndose más robusta. Desde el punto de vista técnico no se acepta una hipótesis a través de un estudio, sino que se aporta evidencia en su favor o en su contra. En ese sentido la hipótesis planteada:

*“El Diseño de una metodología, para realizar la migración de los SGBD licenciados a SGBD de software libre y estándares abiertos en las instituciones públicas del Estado, aumentara la factibilidad con aseguramiento de los datos.”*

Es apoyada por la siguiente evidencia:

- a) De los Cuadros 1 y 2 usos de SGBD, se evidencia que las instituciones y entidades más representativas del Estado utilizan SGBD propietario, privativo y comercial. De acuerdo a la Ley N° 164 y el Decreto Supremo N° 1793, será necesario realizar la migración de SGBD propietario a SGBD de software libre y estándares abiertos.
- b) De acuerdo a la Tabla 5 y los cuadros 33, 34 y 35 de la sección 3.4.1 Factibilidad Operativa, las instituciones y entidades más representativas del Estado, cuentan con el código fuente de las aplicaciones que utilizan SGBD, así como su documentación en general está debidamente actualizada, por lo que la migración de SGBD licenciados a SGBD de software libre y estándares abiertos es factible de realizarse.
- c) Actualmente las instituciones y entidades del Estado no cuentan con una metodología para realizar la migración de SGBD licenciados a SGBD de software libre y estándares abiertos. Como se aprecia en el cuadro número 36, donde las instituciones y entidades más representativas del Estado que cuentan con un plan de migración (41%) lo realizaron de iniciativa propia sin contar con conocimiento o experiencia propia.

- d) Como se muestra en el siguiente cuadro número 36, donde se aprecia que la mayoría de las instituciones y entidades más representativas del Estado No tienen (59%) un plan de migración de SGBD licenciados a SGBD de software libre y estándares abiertos. Por lo que al diseñarse una metodología aumentara la factibilidad de realizar la migración de SGBD.



Cuadro 36, Plan de migración a SGBD de software libre y estándares abiertos. Fuente: Diagnóstico e-gobierno DGGE, BID, REDES. Diciembre 2014.

- e) Al diseñarse roles y responsabilidades de los participantes en el proceso de migración de SGBD propietario, comercial o privativo a SGBD de software libre y estándares abiertos, se aumenta la seguridad de los datos.

## 5.3 Conclusiones y recomendaciones

### 5.3.1 Conclusiones

La primera conclusión del presente estudio, es que la migración de SGBD que tienen licencias, son privativos o comerciales a SGBD de software libre y estándares abiertos, es posible, al contar con métodos, técnicas y procedimientos que guíen el proceso de migración, esto recordando que existen entidades e instituciones del estado que ya tienen experiencia en procesos de migración como el Servicio de Impuestos y la migración realizada de sistemas de archivos con COBOL en sistemas WANG a sistemas basados en Unix-Informix, implementándose el SIIF, Sistema Integrado de Información Financiera.

- Las instituciones y entidades más representativas del Estado prefieren utilizar actualmente SGBD propietarios, comerciales y privativos y que la mayoría de los sistemas de información o aplicativos utilizados son enfocados en sistemas especializados y que los lenguajes de programación preferidos son el PHP y JAVA.
- Las instituciones y entidades más representativas del Estado no cuentan con soporte técnico para los SGBD que utilizan, que en su mayoría son de tipo propietario.
- Los funcionarios de las instituciones y entidades públicas más representativas Del Estado, cuentan con poca capacitación en administración de SGBD y que en general la capacitación es financiada con fuentes propios, por existir poco o ningún presupuesto.

- d) Existe muy poco profesional en el mercado de Bolivia que tenga una certificación internacional en software libre y estándares abiertos, tenemos el conocimiento que solo existe una persona certificada en sistemas operativos de software libre y estándares abiertos.
- e) Se desconoce de la existencia de un repositorio, listado o base de datos de profesionales que cuenten con una certificación internacional de software libre y estándares abiertos.
- f) El área de sistemas dentro del organigrama de la entidad o institución del Estado, generalmente está ubicado en 4to o 5to nivel, por lo que sus labores asignadas son de soporte y no de gestión de información, de este conclusión están excluidas entidades como el RUA, BCB, SIN y Aduana Nacional.
- g) Las instituciones y entidades más representativas del estado utilizan servidores para virtualizar, los sistemas operativos más utilizados son de tipo propietario desde el punto de vista del usuario y que por lo general no tienen soporte técnico contratado.
- h) El licenciamiento de uso de SGBD, que las instituciones y entidades más representativas del estado no tiene en general, licencia de uso de SGBD, de las cuales son de tipo propietario y de esta último concluimos que las licencias de uso no adquiridas corresponden en su mayoría al fabricante Microsoft.
- i) Como consecuencia del párrafo anterior, se concluye que el Estado considera al fabricante de SGBD Microsoft como imperialista y no así al fabricante de SGBD ORACLE, a pesar que ambos fabricantes provienen de Estados Unidos, país que es considerado imperialista por el Estado.
- j) Las instituciones y entidades del Estado cuentan con su licencia de uso de los sistemas operativos de servidores y escritorios, los que no cuentan con licencia de uso corresponden a Microsoft Windows 7 y Microsoft Windows XP, este último ya sin soporte por parte del fabricante.
- k) No existe un listado, repositorio o base de datos de que instituciones y entidades del Estado ya tienen un plan de migración, en qué etapa están, o que porcentaje de avance tienen, por ejemplo: una institución del Estado de 50 computadoras de escritorios que tiene ya tiene migrada 25 computadoras de escritorios a software libre y estándares abiertos, es decir 50% de avance en la migración de las computadoras de escritorio.
- l) Hasta la fecha de culminación de la investigación para el presente trabajo, se desconoce si en el Estado existe una migración exitosa de SGBD propietarios a SGBD de software libre y estándares abiertos.

### 5.3.2 *Recomendaciones*

- a) Realizar una categorización de las entidades o instituciones del Estado que manejan SGBD que pueden ser migrado a SGBD de software libre y estándares abiertos, con la finalidad de determinar la disponibilidad y criticidad de la información que administran para el Estado, esto para determinar con qué entidad o institución del Estado iniciar un proceso de migración a SGBD de software libre y estándares abiertos.
- b) Iniciar las migraciones a SGBD de software libre y estándares abiertos con aquellas entidades o instituciones del estado que no sean críticas en su funcionamiento para el Estado, de tal forma de ir adquiriendo confianza, seguridad y experiencia en el proceso de migración a SGBD de software libre y estándares abiertos. Para luego encarar las entidades o instituciones del estado como Servicio de Impuestos Nacional o Aduana Nacional.
- c) Cada entidad o institución del Estado que vaya a migrar a SGBD de software libre y estándares abiertos debe contemplar en su personal del área de sistemas a un administrador de bases de datos (DBA) experto y de ser posible certificado en el SGBD que va utilizar la entidad o institución del Estado. Juntamente con asignar mayor presupuesto para la capacitación de los administradores de SGBD.
- d) El Estado debe ser el principal promotor, para que los profesionales de las diferentes universidades del país cuenten con una certificación internacional relacionada con software libre y estándares abiertos. Como cursos de postgrados a nivel de especialidades.
- e) El Estado a través de la agencia de información (ADSIB) debe mantener un listado, repositorio o base de datos de profesionales que tenga una certificación internacional en software libre y estándares abiertos.
- f) Creación de una empresa de desarrollo de software libre y estándares abiertos, que entre otras funciones y responsabilidades será la de realizar y desarrollar migraciones a SGBD de software libre y estándares abiertos. Así como desarrollar herramientas para la migración a SGBD de software libre y estándares abiertos.
- g) Se debe mejorar el nivel dentro del organigrama de las instituciones y entidades del Estado de las áreas de sistemas, de forma que puedan tomar decisiones relacionadas con la migración de SGBD a software libre y estándares abiertos.
- h) Una vez migrada el SGBD a software libre y estándares abiertos, se debe considerar la interoperabilidad entre SGBD de software libre y estándares abiertos contra SGBD propietarios, privativos y comerciales, para no perjudicar el intercambio de información entre las instituciones y entidades del Estado.

**NOTA:** A la fecha de publicación del presente trabajo, tanto el plan de gobierno electrónico, así como el plan de implementación de software libre se encuentran en la Unidad de Análisis de Políticas Sociales y Económicas (UDAPE), y en septiembre del 2015 el Gobierno procedió con la creación de la Agencia de Gobierno Electrónico y Tecnologías de la Información y Comunicación<sup>35</sup> (AgeTIC) con el fin de facilitar y simplificar a la población los trámites en entidades del Estado, al hacer la gestión desde su casa.

La AgeTIC es una institución pública descentralizada y autónoma de gestión para elaborar e implementar políticas, planes y estrategias de Gobierno Electrónico y de Tecnologías de la Información y la Comunicación (TIC), para las entidades del Estado<sup>36</sup>.

El trabajo de la AgeTIC será coordinado con la Agencia para el Desarrollo de la Sociedad de la Información en Bolivia (ADSIB), dependiente de la Vicepresidencia. Y su objetivo es implementar el Plan de Gobierno Electrónico y el Plan de Migración a Software Libre. Las estrategias para ambos están en discusión en la Unidad de Análisis de Políticas Sociales y Económicas (Udape) como ya se mencionó.



---

<sup>35</sup> Decreto Supremo 2514, promulgado en Sep/2015, por el presidente Evo Morales.

<sup>36</sup> <http://www.la-razon.com/sociedad/Creacion-conforman-Agencia-Gobierno-Electronico-0-2343365680.html>

# *Bibliografía*

## **Entrevistas y Encuestas**

Encuesta realizada a través del “Formulario de Gobierno Electrónico” a instituciones y entidades estatales en junio de 2014 que reúne 76 entidades.

Encuesta realizada a través del “Formulario de Software Libre” a instituciones y entidades estatales en junio de 2014 que reúne 84 entidades.

## **Libros y Tesis**

CARAGUAY MARTINEZ Susana Beatriz, Metodología para la migración de datos que permita asegurar y conservar la integridad y consistencia de la información administrada por la empresa vsystems, Universidad Técnica de Ambato Ecuador, Septiembre del 2012.

DATE C. J., Introducción a los sistemas de Bases de Datos, Séptima Edición 2001, Pearson Educación.

ESCALERA Saúl, Técnicas de Investigación y Desarrollo en Ciencias y Tecnologías, Segunda Edición 2006, Cochabamba - Bolivia.

HERNÁNDEZ S. R., FERNÁNDEZ C. C y BAPTISTA L.P., Metodología de la Investigación. Quinta Edición 2010, McGraw Hill - México.

SENN James, Análisis y Diseño de Sistemas de Información, 1987, McGraw Hill - México.

SOMMERVILLE Ian, Ingeniería del Software, Séptima Edición 2005, Pearson Educación.

PRESSMAN Roger, Ingeniería del Software, Un enfoque práctico, Tercera Edición 1993, McGraw Hill.

TAMAYO Mario, El proceso de la investigación científica, Quinta Edición 2013, LIMUSA.

## **Leyes, Decretos. Reglamentos y Normas**

Ley N° 164, Ley general de telecomunicaciones, tecnologías de información y comunicación, 8 de agosto del 2011.

Decreto Supremo N° 1793, Reglamento a la Ley N° 164, 13 noviembre del 2013.

Decreto Supremo N° 1874, Creación de la “*Dirección de Gobierno Electrónico (DGE)*”, 23 de enero de 2014.

Norma ISO IEC 17799, Código de práctica para la administración de la seguridad de la información, 2003.

## **Ensayos y Artículos**

Manuel Dávila Sguerra, El software libre en Colombia, Colombia Digital,

Director del Departamento de Informática Redes y Electrónica de Uniminuto  
[manuel\\_davila@grupolinux.net](mailto:manuel_davila@grupolinux.net)

Javier Fernandez Javier y Juan Camilo Giraldo, Metodología para la construcción de un migrador universal de bases de datos, Universidad Cooperativa de Colombia, Seccional Medellín, Mayo 2010.

Synapsi, Plan de Migración del Sistema de Información (SIM) de la Procuraduría General de la Nación, Colombia, 2007.

Corona Correa Susana Laura, Factores críticos de éxito en el proceso de migración de bases de datos relacionales, Universidad Nacional Autónoma de México, Dirección de Sistemas.

### **Internet**

ADSIB <http://www.adsib.gob.bo>

Sistema de Gestión de Bases de Datos, <http://es.wikipedia.org>.

Sistema operativo desarrollado en Cuba [https://es.wikipedia.org/wiki/Nova\\_\(sistema\\_operativo\)](https://es.wikipedia.org/wiki/Nova_(sistema_operativo))

STRATEGY 7 Migration Services, <http://www.s7.com/S7-Migration-Services.php>

SQL Server Migration Assistant, <http://www.mssqltips.com/sqlservertip/2230/sql-server-migration-assistant-2008-ssma/>

SQL Server 2012 Upgrade and Application Compatibility, Choose an Upgrade Strategy, <http://social.technet.microsoft.com/wiki/contents/articles/7009.sql-server-2012-upgrade-and-application-compatibility.aspx>

ORACLE, <https://azerdark.wordpress.com/2010/05/12/oracle-database-structure/>

Microsoft SQL Server, [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)

MySQL <http://es.wikipedia.org/wiki/MySQL>

Instalaciones de MySQL <http://www.grnlight.net/index.php/programming-articles/95-dispelling-the-myths>

COPLUTIC <http://coplusic.gob.bo/>

Diccionario de la lengua española (DRAE), Edición 22, 2001. [www.rae.es](http://www.rae.es)

## Anexos

### Anexo A: Cuestionario Gobierno, 001 - SISTEMAS

#### INFRAESTRUCTURA TECNOLÓGICA DE SU OFICINA

##### Hardware

Realice una breve descripción de la Infraestructura tecnológica que tiene el Centro de Cómputo o de Sistemas

.....

.....

Indique que clase de servidores tiene

Servidores	SI	NO	Cantidad
Virtualización			
Correo Electrónico			
Bases de Datos			
Página Web			
Aplicaciones			
Archivos/Impresión			
Otros*			

\*Otros: Servidores DNS, DHCP, Firewall, Certificados, etc.

En caso de existir, por favor describa las características de los servidores que tiene, describiendo el servidor que tiene mayor capacidad en cada caso:

Servidores	RAM (Gb)	CPU (Ghz)	Disco (Gb)
Virtualización			
Correo Electrónico			
Bases de Datos			

Servidores	RAM (Gb)	CPU (Ghz)	Disco (Gb)
Página Web			
Aplicaciones			
Archivos/Impresión			
Otros			

¿Cuántas computadoras tienen?

EQUIPOS	CANTIDAD	CAPACIDAD	Software/Configuración
Computadoras de escritorio			
Laptops (portátiles)			
Impresoras			
Dispositivos móviles			
Otros (especificar)			

## Software

Por favor realice una descripción de los Sistemas Operativos utilizados y si cuentan con soporte técnico

Sistemas Operativos	Cantidad	Licencia (Si/No/NA)	Soporte Técnico (Si/No)
Windows XP			
Windows 7			
Windows 8/8.1			
Windows Server 2008			

<b>Sistemas Operativos</b>	<b>Cantidad</b>	<b>Licencia (Si/No/NA)</b>	<b>Soporte Técnico (Si/No)</b>
Windows Server 2012			
Linux			
UNIX			
Mac OS			
Otros_____			

Por favor realice una descripción de los Sistemas Gestores de Bases de Datos\* utilizados y si cuentan con soporte técnico.

<b>Gestor</b>	<b>Cantidad</b>	<b>Licencia (Si/No/NA)</b>	<b>Soporte Técnico (Si/No)</b>	<b>Versión/Edición</b>
ORACLE				
Microsoft SQL Server				
DB2				
Informix				
MySQL				
Postgres				
Microsoft Access				
Otros_____				

\*El Gestor de Base de Datos hace referencia a un servidor

Por favor indique el volumen de información que maneja con los Sistemas de Gestores de bases de Datos

<b>Frecuencia</b>	<b>Cantidad (Gb)</b>
Anual	
Mensual	

Frecuencia	Cantidad (Gb)
Semanal	
Diaria	

Tiene el código fuente de las aplicaciones que acceden a sus Gestores de Bases de Datos? SI\_\_\_\_\_ NO\_\_\_\_\_

Tiene la documentación de las aplicaciones que acceden a sus Gestores de Bases de Datos? SI\_\_\_\_\_ NO\_\_\_\_\_

La documentación de las aplicaciones que acceden al Gestor de Bases de Datos está actualizada? SI\_\_\_\_\_ NO\_\_\_\_\_

Por favor indique si tiene algún plan de migración de los gestores de bases de datos, de acuerdo al artículo 77 de la ley 164 y su decreto supremo 1793? SI\_\_\_\_\_ NO\_\_\_\_\_

Qué tipo de conectividad utilizan sus aplicaciones a sus Gestores de Bases de Datos?

Tipo Conectividad	SI/NO
ODBC	
JODBC	
OLEDB	
ADO.NET	
Otros_____	

Por favor realice una descripción de los Servidores de Aplicaciones y si cuentan con soporte técnico.

Aplicaciones	Cantidad	Licencia (Si/No/NA)	Soporte Técnico (Si/No)	Versión/Edición
WEBLOGIC				
Microsoft IIS				
Websphere (IBM)				
TOMCAT				
APACHE				

Aplicaciones	Cantidad	Licencia (Si/No/NA)	Soporte Técnico (Si/No)	Versión/Edición
JBOSS				
Otros _____				

Mencione los lenguajes de Programación Utilizados.

Lenguajes	SI/NO	Licencia (Si/No/NA)	Soporte Técnico (Si/No)	Versión/Edición
JAVA				
JAVA Script				
JSP				
C++, C#				
ASP/ASP.NET				
PHP				
Otros _____				

Realice una breve descripción de los Sistemas de Información en general y de las Aplicaciones Críticas incluyendo:

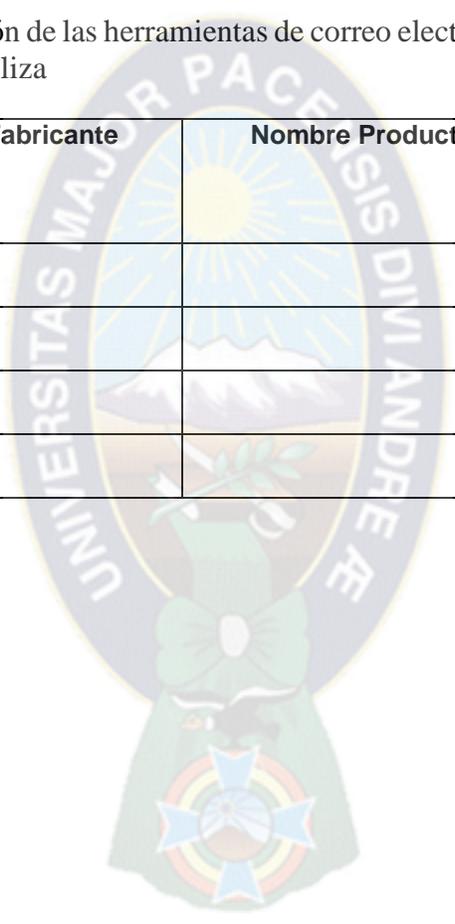
Nombre de la Aplicación:	
Funcionalidad o Uso:	
Con que Herramienta/Lenguaje fue desarrollada: (Java, C++, .Net, etc)	
Número de Usuarios Aproximados	
La aplicación es accedida vía Web?	
Cuál sería el costo estimado de tener esta aplicación crítica fuera de línea?	
Tienen actualmente algún sistema de contingencia contra fallas o desastres? (Cluster, servidor de respaldo, plan de contingencia, etc.)	

Qué tipo de Almacenamiento (storage) utiliza?	
Existe alguna necesidad actual de mejorar los niveles de Alta Disponibilidad de esta aplicación?	

¿Utiliza herramientas de Colaboración? SI\_\_\_\_\_ NO\_\_\_\_\_

Por favor realice una descripción de las herramientas de correo electrónico, mensajería instantánea, mensajería de voz, etc. Que utiliza

Descripción	Fabricante	Nombre Producto y Versión	Soporte Técnico (Si/No)
Correo Electrónico			
Mensajería Instantánea			
Mensajería de Voz			



## **Anexo B: Lista de instituciones encuestadas**

Lista de entidades o instituciones del estado plurinacional que participaron de la encuesta.

<b>MINISTERIOS</b>	
1	Ministerio de Relaciones Exteriores (358)
2	Ministerio de Educación (368)
3	Ministerio de Defensa (360)
4	Ministerio de Justicia (364)
5	Ministerio de Economía y Finanzas Públicas (361)
6	Ministerio de Desarrollo Productivo y Economía Plural (362)
7	Ministerio de Salud (366)
8	Ministerio de Desarrollo Rural y Tierras (369)
9	Ministerio de Transparencia Institucional y Lucha Contra la Corrupción (371)
10	Ministerio de Culturas y Turismo (370)
11	Ministerio de Autonomías (359)
12	Ministerio de Planificación del Desarrollo (Nota Interna 075)
13	Ministerio de Trabajo, Empleo y Previsión Social (365)
14	Ministerio de Obras Públicas, Servicios y Vivienda (363)
15	Ministerio de Medio Ambiente y Agua (367)
16	Ministerio de Hidrocarburos y Energía (372)
17	Ministerio de Minería y Metalurgia (373)
18	Ministerio de Deportes (374)
<b>SERVICIOS Y DIRECCIONES</b>	
1	Aduana Nacional (386)
2	Controlaría General del Estado (375)
3	Servicio General de Identificación Personal (376)
4	Seduca (393)
5	Servicio de Registro Civil-La Paz (400)
6	Impuestos Nacionales (387)
7	RUAT (388)
8	Banco Central de Bolivia (385)
9	Agencia Nacional de Hidrocarburos (382)
10	Derechos Reales (399)
11	SERNAP (394)
12	SEDEM (395)
13	SENAPE (396)
14	SENAPI (377)
15	INASES (398)
16	SENAMHI (397)
17	SENASIR ((378)

18	PROMUEVE (379)
<b>EMPRESAS</b>	
1	ENTEL (389)
2	EMPRESA PÚBLICA "QUIPUS" (381)
3	Empresa Estatal de Transporte por Cable "Mi Teleférico" (383)
4	YPFB (390)
5	COMIBOL (384)
<b>AUTORIDADES</b>	
1	Autoridad del Juego (394)
2	Autoridad de Electricidad (392)
3	Autoridad de Telecomunicaciones y Transporte (380)
4	Autoridad de Supervisión del Sistema Financiero (377)
5	Autoridad de Fiscalización y Control Social de Empresas (391)



## ***Anexo C: Factibilidad***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

### **Resumen de Evaluación**

Evaluación de la factibilidad de realizar la migración a SGBD de software libre y estándares abiertos, se debe evaluar si los siguientes componentes basados en la factibilidad operativa, factibilidad técnica y factibilidad económica.

DESCRIPCIÓN EVALUACIÓN	SATISFACTORIO	
	SI	NO
Factibilidad Operativa		
Factibilidad Técnica		
Factibilidad Económica		

Basado en nuestra consideración de los componentes anteriores, nuestra evaluación de la factibilidad de realizar la migración a SGBD de software libre y estándares abiertos:

Factible	No Factible
----------	-------------

---

Firma Responsable Evaluación

## ***Anexo D: Factibilidad Operativa***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

### **Resumen de Evaluación**

Evaluación de la factibilidad operativa de realizar la migración a SGBD de software libre y estándares abiertos, se debe evaluar si los siguientes elementos o componentes operativos son considerados.

DESCRIPCIÓN EVALUACIÓN	SATISFACTORIO	
	SI	NO
Existe acceso al código fuente de los aplicativos y programas que acceden o utilizan los SGBD o bases de datos a migrar.		
Existe documentación de los aplicativos y programas.		
Se mantiene una relación de control de versiones de la documentación.		
El área de sistemas, dentro del organigrama institucional, es de dirección de sistemas o superior.		
El personal del área de sistemas, se encuentra capacitado en SGBD de software libre y estándares abiertos.		
El personal del área de sistemas, cuenta con alguna clase de certificación internacional en SGBD de software libre y estándares abiertos.		

## ***Anexo E: Factibilidad Técnica***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

### **Resumen de Evaluación**

Evaluación de la factibilidad técnica de realizar la migración a SGBD de software libre y estándares abiertos, se debe evaluar si los siguientes elementos o componentes técnicos son considerados.

DESCRIPCIÓN EVALUACIÓN	SATISFACTORIO	
	SI	NO
Existe inventario de objetos a nivel servidor		
Existe inventario de objetos a nivel bases de datos		
Existe infraestructura adicional para desarrollar ambientes de pruebas		

## ***Anexo F: Inventario del Servidor***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Sistema Operativo: .....

Gestor de Base de Datos: .....

Número de Procesadores: .....

Memoria RAM: .....

Collation/Character Set: .....

Número Bases de Datos: .....

Bases de Datos u Objetos	Data File MB	Log File MB	Número de Tablas	Número de Vistas	Número de SP

Para poder emitir este inventario de servidor de forma más fácil y automática, se puede utilizar o desarrollar una herramienta que acceda a la metadata del gestor de base de datos del servidor origen y saque esta información.

## ***Anexo G: Inventario de la Base de Datos***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

Nombre Bases de Datos: .....

Collation/Character Set: .....

Tipo de Objeto	Nombre Objeto	Número de Filas

La columna Tipo de objeto puede contener: Tabla, Vista, Procedimiento Almacenado, Función, Índices, Triggers, etc.

La columna Nombre Objeto tiene el nombre del objeto que puede ser el nombre de una Tabla, Vista, Procedimiento Almacenado, Función, Índices, Triggers, etc.

La columna Número de Filas, solo aplica cuando el objeto es del tipo tabla.

Para agilizar el inventario de bases de datos se puede utilizar o desarrollar una herramienta que acceda a la metadata de la base de datos para emitir la información que se necesita en el inventario.

## ***Anexo H: Modificación de objetos***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

Base de Datos: .....

Las siguientes fueron modificaciones, para soportar la migración a SGBD de software libre y estándares abiertos. Se realizaron modificaciones a la sintaxis de las tablas, vistas, índices y procedimientos almacenados para hacerlas compatibles con SGBD de software libre y estándares abiertos.

### **Tablas, Vistas e índices**

Nombre del Objeto	Tipo Objeto	Modificación Realizada

La columna Tipo de objeto puede contener los valores: Tabla, Vistas o Índices.

### **Procedimientos Almacenados**

Procedimiento	Modificación Realizada

## ***Anexo I: Comparación de Objetos***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

Nombre Base de Datos: .....

Aplica solo a objetos de tipo Tablas.

Nombre Tabla	Nro. Filas Origen	Nro. Filas Destino	Observaciones

La columna Nro. Filas Origen, es el número de filas de la tabla respectiva en el SGBD de origen o servidor origen.

La columna Nro. Filas Destino, es el número de filas de la tabla en SGBD destino o servidor destino.

La columna Observaciones, puede contener información relacionada con:

- Si el número de filas de la tabla del SGBD origen no es igual al número de filas de la tabla del SGBD, puede indicar un error en el número de filas.
- Si el número de filas de la tabla del SGBD origen es igual al número de filas de la tabla del SGBD, puede indicar un estado en el número de filas, como por ejemplo OK.

## **Anexo J: Comparación de Objetos (Continuación)**

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

Nombre Base de Datos: .....

Aplica a Tablas, vistas, índices, triggers y procedimientos almacenados.

Nombre Objeto	Estado	Observaciones

La columna Estado puede contener los siguientes valores:

- OK, Significa que no fue necesario realizar cambios, para llevar el objeto al servidor destino.
- Cambiado, Que se realizó alguna modificación en el servidor destino, respecto al servidor origen.
- No encontrado, Que el objeto está en el servidor origen, pero no se encuentra en el servidor destino

La columna Observaciones, puede contener los siguientes valores:

- En el caso que el Estado sea Cambiado, se puede incluir en esta columna el cambio realizado, así como su sentencia original.
- Cualquier nota u observación que se crea pertinente.

## ***Anexo K: Secuencia de Pasos Migración***

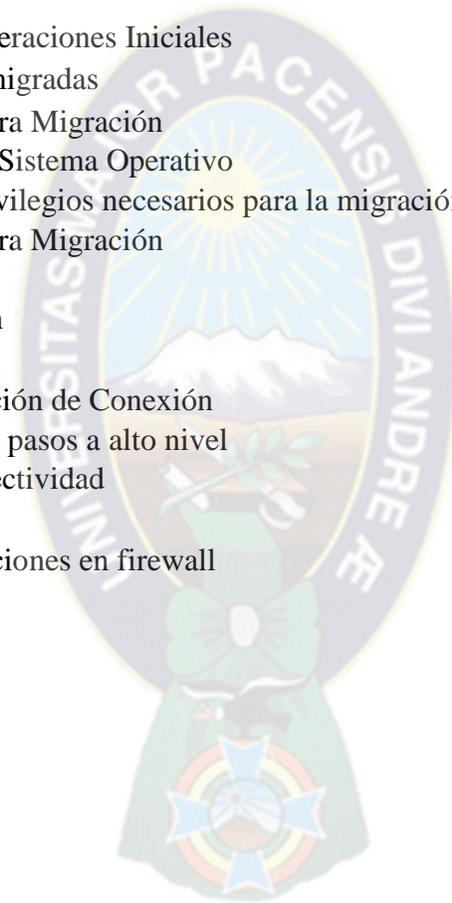
Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

El documento de Secuencia de pasos para la migración a SGBD de software libre y estándares abiertos debe considerar los siguientes puntos:

- Introducción y Consideraciones Iniciales
- Bases de Datos a ser migradas
- Secuencia de Pasos para Migración
  - Requisitos del Sistema Operativo
  - Permisos y Privilegios necesarios para la migración
  - Preparación para Migración
  - Migración
  - Post Migración
- Cambio de Configuración de Conexión
  - Descripción de pasos a alto nivel
  - Prueba de conectividad
- Habilitación de excepciones en firewall



## ***Anexo L: Pruebas de Componentes***

Institución o Entidad: .....

Gestión (Año): .....

Nombre Servidor: .....

Gestor de Base de Datos: .....

Nombre Base de Datos: .....

Nombre del Objeto	Tipo Objeto	Estado Prueba	Observaciones

En la columna:

- Tipo Objeto, será Tabla, Vista, Procedimiento Almacenado, etc.
- Estado Prueba, OK si la prueba fue satisfactoria, No OK si la prueba fallo o presenta observaciones.
- Observaciones, cualquier comentario que ayude a describir el resultado de la prueba.

# Apéndices

## Apéndice A: Gestor ORACLE

El Gestor de base de datos es un software que permite crear, mantener y administrar la información de una base de datos. Oracle es un gestor de base de datos objeto-relacional. En su característica de gestor relacional incorpora estructuras estáticas de datos, conocidas como tablas. En su enfoque orientado a objetos permite trabajar de forma relacional y añadir cualquier mejora orientada a objetos<sup>37</sup>.

El gestor de base de datos Oracle a pesar de poseer características de orientación a objetos, mantiene su núcleo enfocado a lo relacional. Los elementos que forman parte del servidor Oracle son:

- Instancia de Oracle.
- Gestor de almacenamiento.

El gráfico siguiente describe la estructura del servidor de base de datos Oracle.

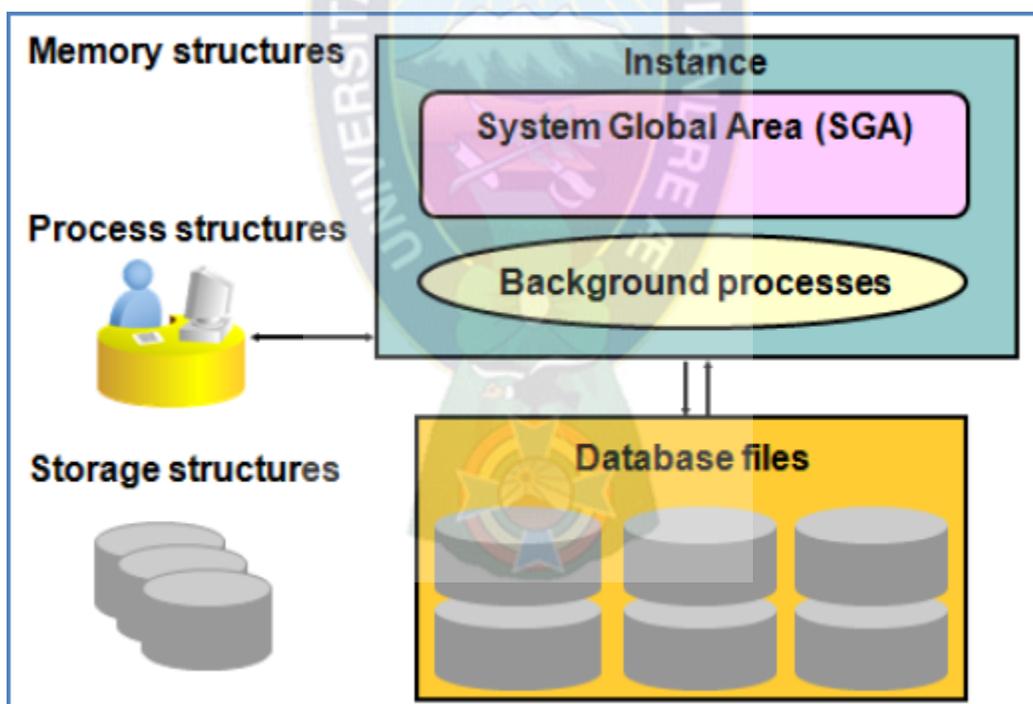


Figura 10, Estructura del servidor Oracle

Una instancia de Oracle comprende estructuras de memoria conocidas como SGA (System Global Area) y los procesos background de Oracle. La instancia de base de datos Oracle contiene también

<sup>37</sup> Todo el apéndice fue extraído de <https://azerdark.wordpress.com/2010/05/12/oracle-database-structure/>

los archivos de datos (datafiles), archivos de recuperación de operaciones (redo log files) y los archivos de control (control files). El gráfico muestra las estructuras de memoria y procesos para acceder a la base de datos Oracle.

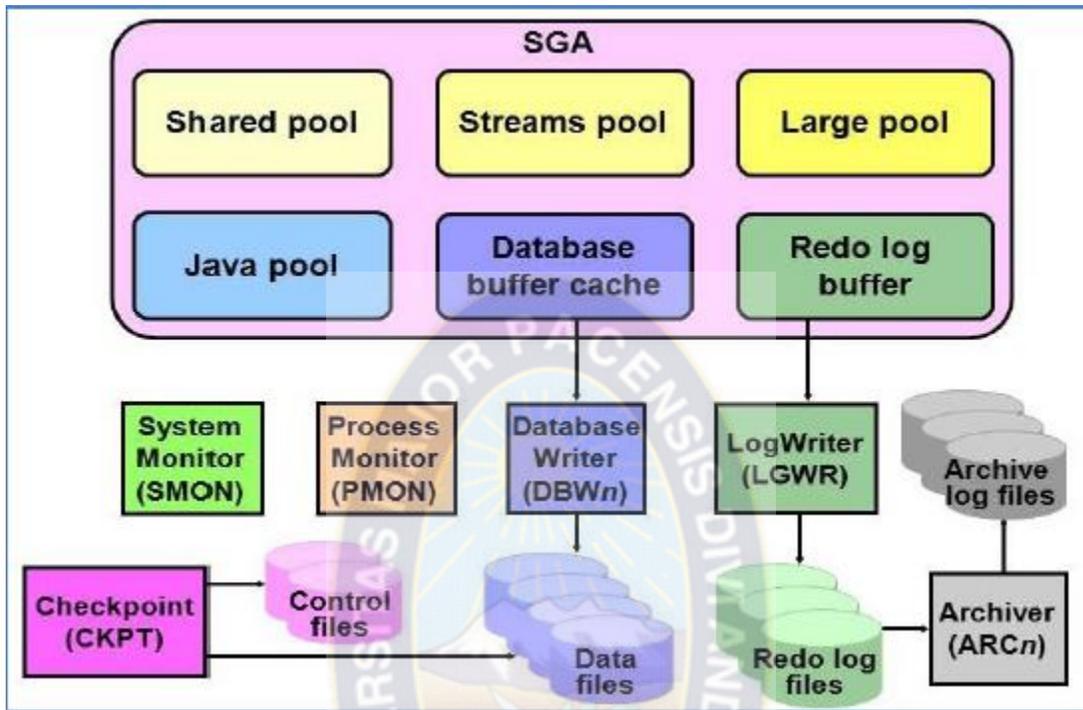


Figura 11, Estructuras de memoria y procesos para acceder a la base de datos Oracle

El gestor de almacenamiento es el módulo de software que se encarga de interactuar con el gestor de archivos para efectuar operaciones de lectura y escritura sobre el disco del computador. El gestor de almacenamiento de Oracle consta de:

- Estructuras lógicas o internas: Tablespaces (Espacios de tablas), objetos del esquema de usuario.
- Estructuras físicas o externas: Los ficheros de datos almacenados en disco. Los ficheros de datos (asociados a los tablespaces), los ficheros redo log y los ficheros de control.

Las bases de datos relacionales incorporan sólo estructuras estáticas de datos, las llamadas tablas. Esto permite para modelar problemas de datos muy grandes, pero no permite aplicar técnicas de programación orientada a objetos. Las estructuras internas de Oracle son:

- Tabla: Una tabla representa la unidad lógica básica de almacenamiento de información. Se distribuye en conjunto de filas y columnas. Cada columna tiene un nombre y un tipo de dato asociado a ella.
- Esquema: El esquema es una colección de objetos lógicos, utilizados para organizar de manera más comprensible la información, es decir se denomina esquema al conjunto de objetos de una cuenta de usuario.

- **Índices:** Un índice es una estructura de datos cuya función es agilizar la recuperación de los registros, generando búsqueda de una manera más rápida y eficiente.
- **Clúster:** El clúster es un grupo de tablas almacenadas en conjunto físicamente como una sola tabla que comparten una columna en común, esto generalmente se realiza con tablas que son accedidas juntas y frecuentemente, minimizando el número de eventos E/S.
- **Vista:** Es una tabla temporal que surge de la selección de información de una o varias tablas. No pueden ser indexadas ni genera almacenamiento de datos.
- **Secuencias:** Oracle posee un generador de secuencia que se emplea para crear valores enteros secuenciales únicos.
- **Procedimientos y Funciones:** Rutinas escritas en PL/SQL que devuelven un valor escalar o realizan un procedimiento con los datos sin devolución de valor.
- **Triggers:** Los triggers son rutinas que se activan al ejecutar un evento específico sobre una tabla.
- **Sinónimos:** Los sinónimos son objetos del sistema que apuntan a otros objetos. Implementan alias de tablas, vistas, secuencias o unidades de programas. Por lo general se utilizan para esconder ciertos detalles del objeto que representan al usuario final.
- **Segmento de Rollback:** Son áreas lógicas de la base de datos que contienen información de las transacciones que se encuentran en curso y que aún no han sido confirmadas o deshechas.
- **Espacios de tablas (Tablespaces):** Un tablespace es una división lógica de la base de datos. Es un contenedor para tablas, índices y más objetos de base de datos. Cada base de datos tiene al menos uno y un tablespace puede pertenecer sólo a una Base de Datos. Son usados para mantener juntos los datos de usuarios o de aplicaciones para facilitar su mantenimiento o mejorar las prestaciones del sistema.

Dentro de los espacios de tablas (tablespaces) y archivos de datos (datafiles), el espacio utilizado para almacenar datos es controlado por el uso de tres estructuras:

- **Segmentos:** Es un grupo de extensiones utilizados para almacenar un tipo particular de datos. Existen 4 tipos de segmentos: datos, índices, rollback y temporales.
- **Extensiones:** Es un grupo de bloques de datos. Se establecen en un tamaño fijo y crecen a medida que van almacenando más datos. También se pueden redimensionar para aprovechar mejor el espacio de almacenamiento.
- **Bloques:** Es la unidad de almacenamiento más pequeña en una base de datos Oracle. Contiene una pequeña porción de información (header) referente al bloque en sí y el resto a los datos que guarda. Generalmente, un bloque de datos ocupará aprox. 2 KB de espacio físico en el disco (asignación típica).

Las estructuras externas de Oracle están contenidas por ficheros que utiliza el servidor de la base de datos. Estos ficheros guardan información tanto de los datos almacenados como de la información necesaria para gestionar la propia base de datos. Los ficheros correspondientes a la estructura externa de Oracle son:

- Ficheros de BD
- Ficheros redo log

- Ficheros de control
- Ficheros de Traza

Para poder ejecutar SQL sobre la base de datos, hay que conectar con la instancia Oracle de la base de datos, lo cual requiere la comunicación entre un proceso cliente y el servidor. Los componentes utilizados por Oracle para procesar el SQL dependen del código enviado y los eventos que se podrían generar son:

- Las consultas devuelven filas.
- Las instrucciones DML (Lenguaje de Manipulación de Datos) graban cambios.
- La instrucción commit asegura el proceso de la transacción.

La conectividad de red con la base de datos Oracle es manejado por un proceso de servidor llamado “Listener” que está configurado para escuchar la conexión en un puerto específico en el servidor de base de datos. Por defecto el puerto de Oracle es 1521.



## Apéndice B: Gestor Microsoft SQL Server

SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL<sup>38</sup>.

Sus principales características son:

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE (Microsoft Desktop Engine) con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

Tabla 11, Información general del Microsoft SQL Server

Microsoft SQL Server	
Desarrollado por	
Microsoft	
<a href="http://www.microsoft.com/sql/">www.microsoft.com/sql/</a>	
Información General	
Última Versión	2012 (11) 06 de marzo de 2012
Género	RDBMS
Sistema operativo	Microsoft Windows
Licencia	Microsoft EULA

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

En el manejo de SQL Server se utiliza mediante líneas de comando el SQLCMD, osql, o PowerShell. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos de Microsoft.

El tipo NUMERIC fue mejorado para ser usado como identificador de columna a partir de la versión 2008 R2.

### Programación T-SQL Nativo

T-SQL (Transact-SQL) es el principal medio de interacción con el Servidor. Permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de la base

<sup>38</sup> El apéndice fue extraído de [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server), el 5/MAR/2014

de datos, la introducción y edición de los datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias de T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente.

### **Ciente Nativo de SQL**

Ciente Nativo de SQL es la biblioteca de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa nativamente soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos. Cliente Nativo de SQL se utiliza como extensión de SQL Server plug-ins para otras tecnologías de acceso de datos, incluyendo ADO u OLE DB. Cliente Nativo de SQL puede también usarse directamente, pasando por alto las capas de acceso de datos.

### **Manejo de memoria**

En versiones de 32 bits, SQL Server usa Address Windowing Extension (AWE) para hacer el direccionamiento por encima de 4 GB. Esto le impide usar la administración dinámica de memoria, y sólo le permite alojar un máximo de 64 GB de memoria compartida. Esta limitación es exclusiva de sistemas operativos 32 bits; en sistemas operativos 64 bits, la memoria máxima que se puede direccionar en Edición Estándar es 64 Gb y en Edición Enterprise 4Tb.

Historia de versiones			
Versión	Año	Nombre de la versión	Nombre clave
1 (OS/2)	1989	SQL Server 1-0	SQL
4.21(WinNT)	1993	SQL Server 4.21	SEQUEL
6	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra
7	1998	SQL Server 7.0	Sphinx
	1999	SQL Server 7.0 OLAP Tools	Plato
8	2000	SQL Server 2000	
8	2003	SQL Server 2000 64-bit Edition	Liberty
9	2005	SQL Server 2005	Yukon
10	2008	SQL Server 2008	Katmai
10.5	2010	SQL Server 2008 R2	Kilimanjaro
11	2012	SQL Server 2012	Denali
12	2014	SQL Server 2014	SQL14 (antes Hekaton)

*Tabla 12, Historial de versiones Microsoft SQL Server*

## Apéndice C: Gestor PostgreSQL

PostgreSQL es un SGBD objeto-relacional (ORDBMS de acuerdo a sus siglas en inglés), con licencia BSD y con su código fuente disponible libremente. Además es un gestor de base de datos multiplataforma.

Originalmente fue desarrollado por un equipo de la Universidad de Berkeley (California) y se denominó POSTGRES, su diseño poseía soporte parcial a la orientación de objetos y conceptos avanzados de base de datos. A partir de 1994 se incluye SQL a Postgres y se libera éste proyecto con muchas mejoras en relación al proyecto original. En 1996 cambia su nombre a PostgreSQL.

Este gestor de base de datos utiliza un modelo cliente/servidor, el programa servidor se llama Postgres y usa multiprocesos para garantizar la estabilidad del sistema.

En un host únicamente se puede ejecutar un único proceso postmaster y se pueden conectar varios clientes vía TCP/IP de manera remota.

Al ser de licencia BSD se permite utilizar, modificar y distribuir a terceros el código fuente y el código binario del programa de software original con o sin modificaciones. Los trabajos derivados pueden optar a licencias de código abierto o comercial. Las distribuciones del software bajo licencia BSD deben incluir copias literales de la licencia, anuncio de copyright y una "negación de responsabilidad". La figura siguiente describe los principales componentes de PostgreSQL:

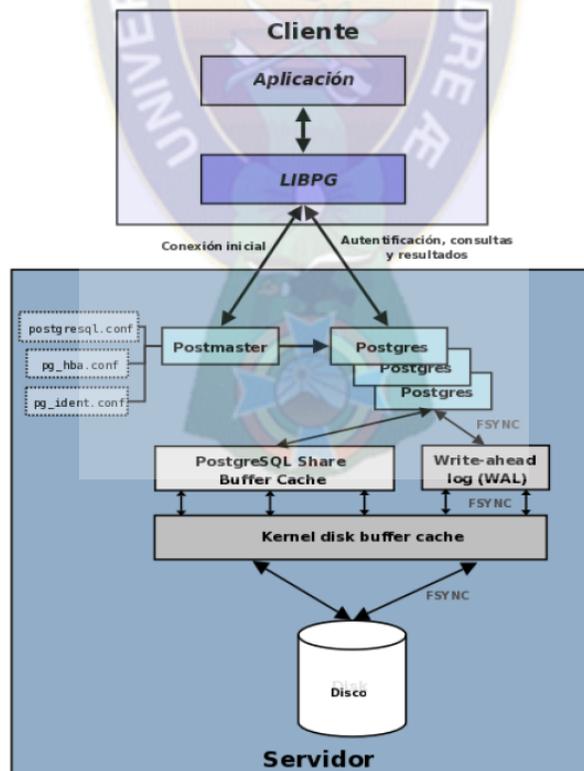


Figura 12, Principales componentes del PostgreSQL

Existe un único proceso postmaster y pueden haber varios procesos de servidor postgres. Cada proceso servidor postgres únicamente puede atender a un cliente, es decir pueden existir varios procesos servidor dependiendo del número de clientes que haya. Los componentes internos del servidor de postgresQL son:

- Ficheros de configuración: Sirve para especificar parámetros para la base de datos, los tres principales son postgresql.conf, pg\_hba.conf y pg\_ident.conf.
- PostgreSQL share buffer cache: Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- Write-Ahead Log (WAL): Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
- Kernel disk buffer cache: Caché de disco del sistema operativo.
- Disco: Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

PostgreSQL dispone de versiones para mayoría de los sistemas operativos y cumple totalmente con ACID (del inglés, Atomicity, Consistency, Isolation, Durability) que hacen referencia al procesamiento de transacciones de base de datos que cumplen con las características de atomicidad, consistencia, aislamiento y perdurabilidad. Además tiene soporte para claves foráneas, joins, vistas, disparadores, procedimientos almacenados (en múltiples lenguajes de programación) e integridad transaccional.

La figura siguiente representa un esquema de conexión básico de PostgreSQL.



Figura 13, Conexión básica de PostgreSQL

PostgreSQL ofrece características tales como:

- Control concurrente multiversión (MVCC) permite el control de concurrencia en su mayor parte sin requerir bloqueos.
- Optimizador de consultas: permite la resolución exitosa de consultas complejas en tiempos adecuados comparables a las bases de datos propietarias.
- Puntos de recuperación en el tiempo (point in time recovery, PITR) que consiste en almacenar y copiar continuamente todas las transacciones de PostgreSQL desde el último backup realizado a nivel de respaldo en disco. Permitiendo la restauración de la base de datos a un determinado punto de tiempo en caso de ser necesario.
- Transacciones anidadas: Mediante uso de savepoints que son marcas internas en el procesamiento de una transacción, se puede deshacer partes específicas de transacción y no toda la transacción por medio del comando ROLLBACK.
- Permite almacenamiento agrupado en tablespaces.
- Posee replicación asíncrona y copias de seguridad en caliente (en línea).
- Es altamente escalable tanto en la cantidad de datos que puede manejar como en el número de usuarios concurrentes que puede atender.



## ***Apéndice D: Gestor MySQL***

MySQL<sup>39</sup> es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones<sup>40</sup>. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr, y YouTube.

### **Lenguajes de Programación**

Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una interfaz de programación de aplicaciones específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP.

MySQL está escrito en una mezcla de C y C++. Hay un documento que describe algunas de sus estructuras internas en <http://dev.mysql.com/doc/internals/> en (en inglés).

### **Aplicaciones**

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la

---

<sup>39</sup> Extraído todo el apéndice de <http://es.wikipedia.org/wiki/MySQL>

<sup>40</sup> Según <http://www.grnlight.net/index.php/programming-articles/95-dispelling-the-myths>

modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

## **Plataformas**

MySQL funciona sobre múltiples plataformas, incluyendo:

- AIX
- BSD
- FreeBSD
- HP-UX
- Kurisu OS
- GNU/Linux
- Mac OS X
- NetBSD
- OpenBSD
- OS/2 Warp
- QNX
- SGI IRIX
- Solaris
- SunOS
- SCO OpenServer
- SCO UnixWare
- Tru64
- eBD
- Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8 y Windows Server (2000, 2003, 2008 y 2012).
- OpenVMS18

## **Características**

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...

- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

### **Características adicionales**

- Usa GNU Automake, Autoconf, y Libtool para portabilidad
- Uso de multihilos mediante hilos del kernel.
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice
- Tablas hash en memoria temporales
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL.
- Completo soporte para operadores y funciones en cláusulas select y where.
- Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).
- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.

- En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.
- MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

### **Características distintivas**

Las siguientes características son implementadas únicamente por MySQL:

- Permite escoger entre múltiples motores de almacenamiento para cada tabla. En MySQL 5.0 éstos debían añadirse en tiempo de compilación, a partir de MySQL 5.1 se pueden añadir dinámicamente en tiempo de ejecución: Los hay nativos como MyISAM, Falcon, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example
  - Desarrollados por partners como solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (antes Brighthouse), Kickfire, XtraDB, IBM DB2. InnoDB Estuvo desarrollado así pero ahora pertenece también a Oracle.
  - Desarrollados por la comunidad como memcache, httpd, PBXT y Revision.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

### **Tipos de compilación del servidor**

Hay tres tipos de compilación del servidor MySQL:

- Estándar: Los binarios estándares de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento.

## *Glosario*

ADO.NET	Activex Data Object
ADSIB	Agencia de Desarrollo para la Sociedad de la Información en Bolivia.
Cadenas de Conexión	de Contiene información de inicialización que se transfiere como un parámetro desde un proveedor de datos a un origen de datos. La sintaxis depende del proveedor de datos.
COBOL	Common Bussines Oriented Languaje
COPLUTIC	Comité Plurinacional de Tecnologías de la Información y Comunicación
DDL	Data Definition Languaje
DML	Data Manipulation Languaje
DTSX	Data Transformation Services XML en inglés, o Servicios de Transformación de datos en XML
SAFCO	Ley de Administración y Control Gubernamental
SICOES	Sistemas de Contrataciones Estatales
SIGMA	Sistema de Control Gubernamental
SIRAT	Sistema Integrado de Recaudación y Administración Tributaria
SGBD	Sistemas de Gestión de Bases de Datos
SSMA	SQL Server Migration Assistant
ODBC	Open Data Base Conectivity
JODBC	Java Open Data Base Conectivity
OLE DB	Object Linking and Embedding, Database, es un API diseñado por Microsoft, permite acceder a una variedad de fuentes de datos de forma uniforme
TIC	Tecnologías de la Información y Comunicación
T-SQL	Transac SQL