



**UNIVERSIDAD NACIONAL**

**“PEDRO RUIZ GALLO”**

**ESCUELA DE POSTGRADO**



**MAESTRÍA EN INGENIERÍA DE SISTEMAS**

**“Arquitectura de Alta Disponibilidad para la  
Optimización del Servicio de Base de Datos del  
Gobierno Regional Lambayeque”**

**TESIS**

**Presentada para optar el Grado Académico de  
Maestro en Ingeniería de Sistemas con Mención en  
Gerencia de Tecnologías de la Información y Gestión del  
Software**

**AUTOR:**

**Bach. Palacios Ormeño, Julio Cesar**

**ASESOR:**

**Dr. Carrión Barco, Gilberto**

**LAMBAYEQUE - PERÚ**

**2019**

**“ARQUITECTURA DE ALTA DISPONIBILIDAD PARA LA  
OPTIMIZACIÓN DEL SERVICIO DE BASE DE DATOS DEL  
GOBIERNO REGIONAL LAMBAYEQUE”**

**Presentado por:**

---

**Bach. Julio Cesar Palacios Ormeño**  
**Autor**

---

**Dr. Gilberto Carrión Barco**  
**Asesor**

**Aprobado por:**

---

**Dr. Luis Alberto Dávila Hurtado**  
**Presidente**

---

**M.Sc. Pedro Fiestas Rodríguez**  
**Secretario**

**Vocal**

---

**M.Sc. Roberto Arteaga Lora**

## Acta de Sustentación

### ACTA DE SUSTENTACIÓN DE TESIS

038

Siendo las 15:30 horas del día VIERNES 04 de OCTUBRE del año Dos Mil Diecinueve

, en la Sala de Sustentación de la Escuela de Posgrado de la Universidad Nacional Pedro Ruiz Gallo de Lambayeque, se reunieron los miembros del Jurado, designados mediante Resolución N° 0552-2019-EPE de fecha 09/05/2019, conformado por:

<u>DR. LUIS ALBERTO JAIMA HURTADO</u>	PRESIDENTE (A)
<u>MSc. PEDRO FICSTAN RODRIGUEZ</u>	SECRETARIO (A)
<u>MSc. ROBERTO ARTEAGA LORA</u>	VOCAL
<u>MSc. GILBERTO CARRION BARCO</u>	ASESOR (A)



Con la finalidad de evaluar la tesis titulada ARQUITECTURA DE ALTA DISPONIBILIDAD PARA LA OPTIMIZACIÓN DEL SERVICIO DE BASE DE DATOS DEL GOBIERNO REGIONAL LAMBAYEQUE

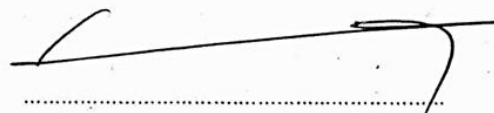

presentado por el (la) Tesista JUAN CESAR PARRALES ORMEÑO  
sustentación que es autorizada mediante Resolución N° 125-2019-EPE de fecha 27 de Setiembre de 2019

El Presidente del jurado autorizó del acto académico y después de la sustentación, los señores miembros del jurado formularon las observaciones y preguntas correspondientes, las mismas que fueron absueltas por el (la) sustentante, quien obtuvo 80 puntos que equivale al calificativo de Bueno

En consecuencia el (la) sustentante queda apto (a) para obtener el Grado Académico de: MAESTRO EN INGENIERIA DE SISTEMAS CON MENCIÓN EN GERENCIA DE TECNOLOGÍAS DE LA INFORMACIÓN Y GESTIÓN DEL SOFTWARE

Siendo las 4:25 p.m. horas del mismo día, se da por concluido el acto académico, firmando la presente acta.

  
PRESIDENTE  
  
VOCAL

  
SECRETARIO  
  
ASESOR

## **Declaración jurada de originalidad**

Yo, Julio César Palacios Ormeño investigador principal y el Dr. Gilberto Carrión Barco, asesor del trabajo de investigación “Arquitectura de Alta Disponibilidad para la optimización del Servicio de Base de Datos del Gobierno Regional Lambayeque”, declaramos bajo juramento que este trabajo no ha sido plagiado, ni contiene datos falsos. En caso se demostrara lo contrario, asumo responsablemente la anulación de este informe, y por ende el proceso administrativo a que hubiere lugar. Que puede conducir a la anulación del título o grado emitido como consecuencia de este informe.

Lambayeque, abril de 2019

Nombre del Investigador: Bach. Julio César Palacios Ormeño

Nombre del Asesor: Dr. Gilberto Carrión Barco

## **Dedicatoria**

A Dios todopoderoso por su grande amor y su bendición sobre mi vida, así como las fuerzas que día a día me entrega para seguir luchando por mis objetivos y metas, y a Él sea la Gloria, Honra y Alabanza.

A mis padres César y Violeta, que son los guías que Dios ha puesto en mi vida, y porque siempre me han apoyado en todas las decisiones que he podido tomar.

A mi esposa Esther, mis hijos Santiago y Mateo por ser el regalo máspreciado que Dios me ha dado, por su amor incondicional hacia mi persona y ser el impulso para seguir creciendo profesionalmente.

## **Agradecimiento**

A mi asesor Dr. Gilberto Carrión Barco por su apoyo incondicional en el desarrollo de mi informe, demostrando su profesionalismo y estima personal.

Al Dr. Freddy Paz por sus enseñanzas en el Seminario de Tesis e incentivarnos a asumir los retos con madurez, convicción y mucha dedicación.

## Índice General

<b>1. Capítulo I. Diseño Teórico.....</b>	<b>14</b>
<b>1.1 Antecedentes de la Investigación.....</b>	<b>15</b>
1.1.1 A nivel internacional.....	15
1.1.2 A nivel nacional .....	16
c) Cajamarca.....	16
<b>1.2 Base Teórica .....</b>	<b>18</b>
1.2.1 Arquitectura de Alta Disponibilidad .....	18
1.2.2 Optimización de Servicios .....	19
1.2.3 Base de Datos .....	19
1.2.4 Replicación de Base de Datos .....	21
1.2.5 Conmutación de Conexiones.....	23
<b>1.3 Definiciones Conceptuales.....</b>	<b>23</b>
1.3.1 Postgresql .....	23
1.3.2 Replication Manager.....	24
1.3.3 PG Bouncer .....	25
1.3.4 Patroni .....	26
<b>1.4 Situación Problemática.....</b>	<b>27</b>
1.4.1 Formulación del Problema.....	28
<b>1.5 Operacionalización de Variables.....</b>	<b>29</b>
1.5.1 Esquema Propuesto .....	29
<b>1.6 Hipótesis.....</b>	<b>30</b>
<b>2. Capítulo II. Métodos y Materiales .....</b>	<b>31</b>
<b>2.1 Tipo de Investigación .....</b>	<b>32</b>
<b>2.2 Método de Investigación .....</b>	<b>32</b>
<b>2.3 Diseño de Contrastación.....</b>	<b>33</b>
<b>2.4 Población, Muestra y Muestreo .....</b>	<b>33</b>
2.4.1 Población .....	33
2.4.2 Muestreo .....	33
2.4.3 Muestra .....	34
<b>2.5 Técnicas, Instrumentos, Equipos y Materiales de Recolección de Datos ...</b>	<b>34</b>
2.5.1 Técnicas: .....	34
2.5.2 Instrumentos de Recolección de Datos: .....	34
2.5.3 Equipos .....	35

2.5.4	Herramientas .....	35
2.5.5	Materiales .....	37
2.6	Procesamiento y Análisis de Datos.....	38
3.	Capítulo III. Resultados.....	39
3.1	Diagnóstico del Servicio de Base de Datos del Gobierno Regional Lambayeque .....	40
3.1.1	Dimensión Administrador de Replicaciones .....	40
3.1.2	Dimensión Conmutación de Conexiones .....	44
3.2	Resultados de cada dimensión después de implementada la propuesta.....	47
3.2.1	Dimensión Administrador de Replicaciones .....	47
3.2.2	Dimensión Conmutación de Conexiones .....	50
3.3	Desarrollo de la propuesta .....	54
3.3.1	Título de la propuesta .....	55
3.3.2	Objetivos de la propuesta .....	55
3.3.2.1	Objetivo General .....	55
3.3.2.2	Objetivos específicos.....	55
3.3.3	Esquema de desarrollo de la propuesta .....	56
3.3.3.1	Topología lógica propuesta de la arquitectura de alta disponibilidad.....	56
3.3.3.2	Motor de base de datos a utilizar .....	57
3.3.3.3	Mecanismo propuesto de replicación de base de datos .....	58
3.3.3.4	Mecanismo propuesto de conmutación de conexiones .....	60
4.	Capítulo IV. Discusión.....	61
4.1	Discusión de resultados de la dimensión Administrador de Replicaciones ..	62
4.2	Discusión de resultados de la dimensión Conmutación de Conexiones .....	62
	Conclusiones.....	65
	Recomendaciones .....	65



## Índice de Tablas

Tabla 1: Operacionalización de la Variable Dependiente .....	30
Tabla 2: Equipos.....	35
Tabla 3: Materiales.....	37
<i>Tabla 4: Resultados de diagnóstico de la dimensión Administrador de Replicaciones .....</i>	<i>42</i>
Tabla 5: Análisis de Fiabilidad de la dimensión administrador de replicaciones.....	42
Tabla 6: Resultados de diagnóstico de la dimensión Conmutación de Conexiones .....	45
Tabla 7: Análisis de Fiabilidad de la dimensión Conmutador de Conexiones .....	46
Tabla 8: Resultados de la dimensión implementada Administrador de Replicaciones.....	48
Tabla 9: Análisis de Fiabilidad de la dimensión implementada Administrador de Replicaciones .....	49
Tabla 10: Resultados de la dimensión implementada Conmutación de Conexiones .....	52
Tabla 11: Análisis de Fiabilidad de la dimensión implementada Conmutación de Conexiones.....	52
Tabla 12: Escala de medición de la dimensión implementada Conmutación de Conexiones.....	53

## Índice de Figuras

Figura 1: Modelo de Arquitectura de Alta Disponibilidad .....	18
Figura 2: Hot Standby .....	19
Figura 3: Base de Datos OLTP .....	20
Figura 4: Base de Datos OLAP .....	21
Figura 5: Eager Replication .....	22
Figura 6: Lazy Replication.....	22
Figura 7: Conmutación de Conexiones.....	23
Figura 8: Modelo de Replicación con Repmgr.....	25
Figura 9: Conmutador de Conexiones .....	26
Figura 10: Modelo de Arquitectura Patroni.....	27
Figura 11: Escala de Medición .....	40
Figura 12: Escala de medición de Dimensión Administración de Replicaciones.....	43
Figura 13: Escala de medición de Dimensión Conmutación de Conexiones .....	46
Figura 14: Escala de medición de la dimensión implementada Administración de Replicaciones .....	50
Figura 15: Topología Lógica Propuesta.....	56
Figura 16: Motor de Base de Datos PostgreSQL.....	57
Figura 17: Replication Manager .....	58
Figura 18: PGBOUNCER - Conmutador de Conexiones.....	60

## **Resumen**

En este estudio se realiza un diagnóstico del servicio de base de datos del Gobierno Regional Lambayeque ubicado en el la provincia de Chiclayo departamento de Lambayeque comprobándose el bajo nivel de disponibilidad cuyas dimensiones son: 1. Existe un alto nivel de interrupciones, 2. Alto tiempo de inactividad (paralización) del servicio de base de datos; todo esto con el fin de implementar una arquitectura de alta disponibilidad basado en un administrador de replicaciones de bases de datos y un conmutador de conexiones.

El estudio fue de tipo cuantitativo y su diseño fue pre-experimental, donde la población estuvo conformada por 500 trabajadores administrativos, de donde se obtuvo 76 trabajadores como muestra para el estudio.

El trabajo de desarrollo en tres etapas, la primera consistió en diagnosticar la variable dependiente, es decir el Servicio de Base de Datos, para lo cual el autor elaboró una encuesta que midiera sus dos componentes, el que determinó un nivel de disponibilidad por debajo de lo normal.

En una segunda etapa se implementó una arquitectura de alta disponibilidad, variable independiente basado en la replicación de base de datos y conmutación de conexiones para finalmente medir los resultados de dicha arquitectura con la misma encuesta antes mencionada, cuyos resultados evidenciaron una significativa optimización en sus dos componentes.

Palabras clave: Arquitectura de Alta Disponibilidad, Servicio de Base de Datos.

## **Abstract**

In this study a diagnosis is made of the database service of the Lambayeque Regional Government located in the province of Chiclayo, department of Lambayeque, checking the low level of availability whose dimensions are: 1. There is a high level of interruptions, 2. High time of inactivity (stoppage) of the database service; all this in order to implement a high availability architecture based on a database replication manager and a connection switch.

The study was quantitative and its design was pre-experimental, where the population was made up of 500 administrative workers, from which 76 workers were obtained as a sample for the study.

The three-stage development work, the first was to diagnose the dependent variable, that is, the Database Service, for which the author developed a survey that measured its two components, which determined an availability level below the normal.

In a second stage, a high availability architecture was implemented, an independent variable based on database replication and connection switching to finally measure the results of said architecture with the same survey mentioned above, whose results showed a significant optimization in its two components.

Keywords: High Availability Architecture, Database Service.

## Introducción

Con el paso de los años la información se ha convertido en el activo más importante para las empresas tanto públicas como privadas, debido a que el éxito en sus operaciones radica en ofrecer la información de forma rápida, bien organizada, segura y sobre todo siempre disponible. (Sánchez, 2003)

Para (Hernández, 2009) la seguridad de la información es muy importante, en el caso de una aplicación presentara problemas, podría afectar a las demás hasta el grado de llegar a perder la información existente en el equipo, es así como surge la necesidad de la disponibilidad de la información.

La falta de mecanismos de replicación de datos y continuidad de servicios, según (Zamora, 2014) aumenta las probabilidades de que ocurran desastres y puedan generarse daños irreparables a la información.

El Gobierno Regional Lambayeque cuenta con un servicio de Base de Datos implementado con un Clúster de Postgresql versión 9.6 que permite gestionar los registros que son generados a través de sus servicios web tales como SIGA (Sistema Integrado de Gestión Administrativa), SISGEDO (Sistema de Gestión Documentaria) y la Administración de Portales Web.

El gran crecimiento de los datos, demanda de un mayor recurso tecnológico en hardware y software para poder organizar la información en bases de datos y que esta sea administrada de una forma eficiente; este aumento de la información exige una mayor capacidad de procesamiento, almacenamiento, mecanismos de contingencia y equipamiento redundante a fin de evitar interrupciones imprevistas en sus motores de base de datos.

Este servicio es de vital importancia para la el Gobierno Regional Lambayeque por ser un eje fundamental para el funcionamiento y operación de los servicios web antes mencionados; pero continuamente viene presentando interrupciones en las horas de más alto tráfico, generando la paralización de los servicios que se le brinda al ciudadano.

A la fecha no existe una arquitectura tecnológica con mecanismos de continuidad de servicio y replicación de base datos que pueda evitar interrupciones, prevea las caídas de este servicio y brinde disponibilidad con una alternativa secundaria de tal manera que los usuarios y el ciudadano no se vean afectados por estos inconvenientes.

# **1. Capítulo I. Diseño Teórico**

## **1.1 Antecedentes de la Investigación**

La realización del estudio tuvo que tener en cuenta los resultados internacionales, nacionales y locales tales como:

### **1.1.1 A nivel internacional**

#### **a) Cuba**

Según (Caro, 2008), la visualización científica ha surgido como una herramienta capaz de procesar y analizar estos grandes volúmenes de datos hasta el orden de los terabytes.

Tradicionalmente se han empleado formatos de archivos binarios secuenciales para gestionar los datos utilizados en la visualización. Sin embargo, el trabajo con estos modelos no resulta suficiente ante las demandas de la comunidad científica actual. Por lo que se necesita una solución integral que combine las ventajas de los Sistemas Gestores de Bases de Datos (SGBD), la eficiencia de los formatos nativos y las potencialidades del procesamiento paralelo.

Se planteó como objetivo general determinar la factibilidad del uso de un sistema de base de datos paralelo para gestionar datos en el área de visualización científica e implementar un caso de estudio en un clúster de computadoras, la muestra no fue probabilística donde se determinaron las herramientas de software libre disponibles para implementar un sistema de bases de datos paralelos que gestione datos científicos y entre ellas se seleccionaron las más consolidadas.

Esta investigación me brinda un valioso aporte porque permite determinar los requerimientos, mecanismos y características de las herramientas disponibles en software libre para la implementación de alta disponibilidad en un servicio de base de datos.

#### **b) México**

Para (Hernández, 2009) su objetivo general fue realizar un esquema de virtualización sobre un equipo de cómputo físico, que permita crear máquinas virtuales utilizadas para el manejo de bases de datos con alta disponibilidad a nivel empresarial. Este trabajo se basó en una investigación documental, con una muestra no probabilística, donde se concluyó que las técnicas de

virtualización permiten el ahorro de licencias de software, en los dos esquemas mostrado en este trabajo, y que la virtualización tiene una mayor ventaja por ser adecuadas a diferentes niveles empresariales, además se ofreció un esquema de alta disponibilidad utilizando un manejador de base de datos que puede ser implementado en diferentes sectores empresariales y educativos.

Esta investigación me permitió identificar las problemáticas que impiden lograr una alta disponibilidad en los servidores de bases de datos empresariales, de esta manera el aporte importante para nuestra investigación es analizar los distintos escenarios que han sido utilizados para lograr un servicio altamente disponible, para así poder tomar uno de esos mecanismos en nuestro proyecto.

### **1.1.2 A nivel nacional**

#### **c) Cajamarca**

(Zamora, 2014) En su investigación; implementa y diseña un clúster de servidores en entorno Linux para brindar un servicio de alta disponibilidad de la información, para tal propósito se utilizaron herramientas como la base de datos PostgreSQL, Pppool 11, Heartbeat. La investigación tuvo como caso de estudios la empresa MINE SENSE SOLUTIONS que implementará un sistema de control de flotas pesadas para el proyecto minero Hudbay-Constancia en la región del Cusco. La empresa en cuestión tiene la necesidad de alta disponibilidad del servicio de información para el sistema que implementará por ser de vital importancia en las operaciones diarias. Para el diseño e implementación del clúster de alta disponibilidad se investigó las herramientas disponibles y que mejor se adaptan a las necesidades antes mencionadas.

La muestra fue no probabilística y concluye que la la necesidad de alta disponibilidad en el caso de estudio es evidente al dar un servicio de información a un sistema de vital importancia en las operaciones de la empresa, por lo cual fue necesario analizar y ver en mercado las mejores opciones y alternativas para el diseño e implementación del clúster que brinde la alta disponibilidad requerida. Además concluye que el diseño e implementación del clúster se dio en un entorno Linux, el cual es muy complejo en comparación con un entorno Windows, para lo cual es necesario tener conocimiento de lenguajes de programación y códigos establecidos para la consola que fue la principal herramienta para la implementación del clúster ya que por medio de esta se accede a los archivos y variables de configuración en la implementación.



Esta investigación brinda un aporte importante respecto a la implementación de Clústeres de servidores en Linux con el propósito de brindar disponibilidad de la información, utilizando además el servicio Postgresql que es el Motor Base de Datos de nuestra investigación en el Gobierno Regional Lambayeque que tiene como directiva el uso de software libre en toda su infraestructura tecnológica. Asimismo nos permite comparar los mecanismos de alta disponibilidad utilizados en esta investigación como lo es Pppol y Heartbeat con los nuevos mecanismos que ofrece Postgresql.

#### **d) Chiclayo**

Por su parte (Guevara, 2017) consideró como objetivo general; analizar, comparar e implementar una infraestructura tecnológica con alta disponibilidad para servidores de la Universidad Señor de Sipán - Lambayeque, donde tuvo como muestra a cinco personas que forman parte del área de Integración de Tecnologías quienes son el jefe de integración de tecnologías, el coordinador de Servicios, el coordinador de gestión de seguridad, el coordinador de infraestructura y telecomunicaciones y el auxiliar operador de central telefónica.

Se aplicó una encuesta para la recopilación de los datos donde se determinó que las interrupciones de los servicios de los sistemas alojados en los servidores campus virtual y base de datos, es de 117 horas al año, llegando a un 98,66% de disponibilidad de los servicios. Según la encuesta planteada, el 60% del personal del Área de Integración de Tecnologías de la Dirección de Tecnologías de la Información opina que la disponibilidad de los servicios mencionados es buena y el 40% opina que es regular.

La presente tesis me permitió contrastar los mecanismos de alta disponibilidad utilizados en un Clúster propietario de Windows Server 2016 con nuestros mecanismos basados en software libre de nuestra investigación, demostrando así las ventajas y desventajas que ofrece una arquitectura propietaria y una arquitectura libre para alta disponibilidad en los servicios de base de datos del Gobierno Regional Lambayeque.

## 1.2 Base Teórica

### 1.2.1 Arquitectura de Alta Disponibilidad

Para la mayoría de las organizaciones, la disponibilidad de los servicios es fundamental para satisfacer las necesidades empresariales. La redundancia es una parte importante del diseño de una arquitectura tecnológica para prevenir o minimizar interrupciones de los servicios de la red al minimizar la posibilidad de un punto único de falla. Un método para implementar la redundancia consiste en instalar equipos duplicados y proporcionar servicios de conmutación por falla para los dispositivos esenciales.

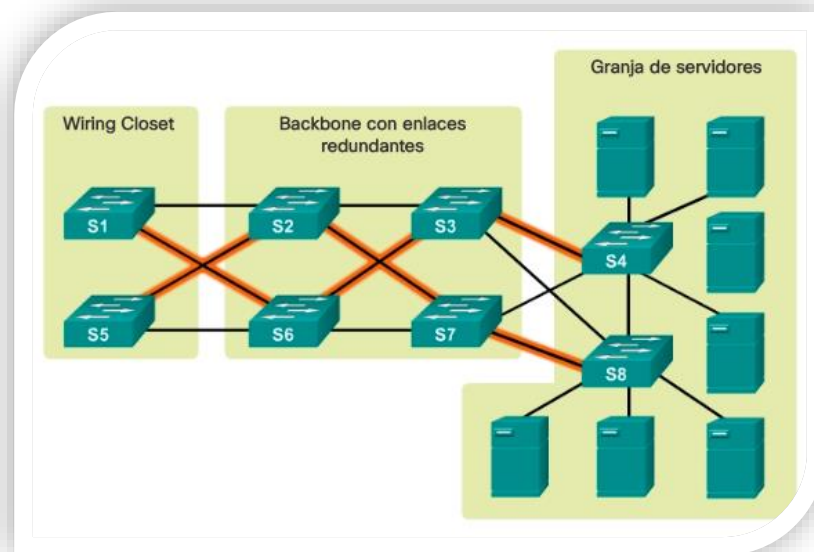


Figura 1: Modelo de Arquitectura de Alta Disponibilidad  
(Cisco, 2018)

Otro método para implementar la redundancia es mediante enlaces redundantes, los enlaces redundantes ofrecen rutas físicas alternativas para que los datos atraviesen la red. En una red conmutada, las rutas redundantes admiten una alta disponibilidad.

“La Alta disponibilidad (High availability) para base de datos como un protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado. Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos. Si un usuario no puede acceder al sistema se

dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible” (Rivero, 2017)

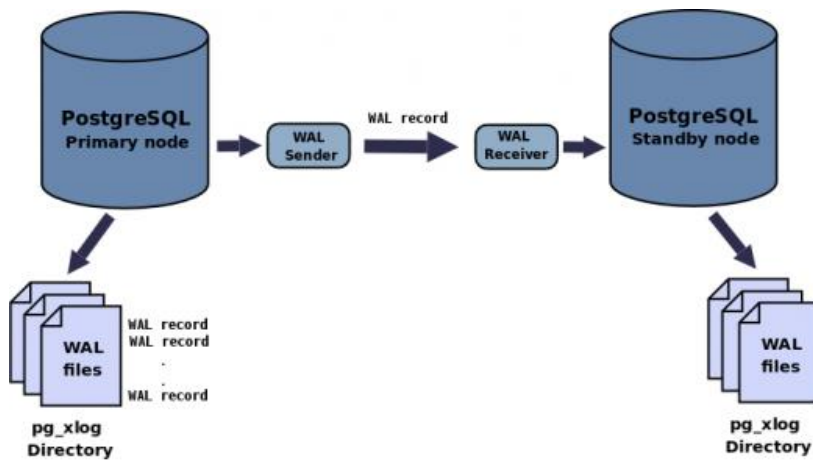


Figura 2: Hot Standby  
(Guerrero, 2016)

### 1.2.2 Optimización de Servicios

Es el proceso de modificar un servicio con el fin de mejorar su eficiencia y hacerlo eficaz mediante mecanismos que permitan ofrecer una calidad de servicio.

Un servicio de base de datos constantemente se ve afectado su rendimiento cuando se realizan grandes cantidades de transacciones de datos enviados y recibidos, esto puede originar fallas en los servidores que albergan las bases de datos.

La optimización de servicios implica calidad, y esto está relacionado estrechamente con la recuperación de desastres (disaster recovery), la alta disponibilidad se encarga de la recuperación de aplicaciones, datos y transacciones de datos enviados hasta el momento de la caída del sistema; no obstante las necesidades operativas de una organización tienen prioridad sobre la sensibilidad de los datos, y el tiempo necesario para recuperar ciertos componentes del sistema puede ser considerablemente menor que aquél en muchos escenarios de recuperación de desastres.

### 1.2.3 Base de Datos

De una manera simple, es un contenedor que permite almacenar la información de forma ordenada con diferentes propósitos y usos. Por ejemplo, en una base de datos se puede almacenar información de diferentes departamentos (Ventas,

Recursos Humanos, Inventarios, entre otros). El almacenamiento de la información por sí sola no tiene un valor, pero si combinamos o relacionamos la información con diferentes departamentos nos puede dar valor.

Existen muchas empresas con diferentes giros y dependiendo del giro será el tipo de procesamiento que se le dará a la información, esto determinará el tipo de base de datos a utilizar. Existen diferentes tipos de bases de datos, pero las más comunes son las OLTP y OLAP.

Las bases de datos de tipo OLTP (On Line Transaction Processing) también son llamadas bases de datos dinámicas lo que significa que la información se modifica en tiempo real, es decir, se insertan, se eliminan, se modifican y se consultan datos en línea durante la operación del sistema. Un ejemplo es el sistema de un supermercado donde se van registrando cada uno de los artículos que el cliente está comprando y a su vez el sistema va actualizando el inventario. (IBM, 2015)

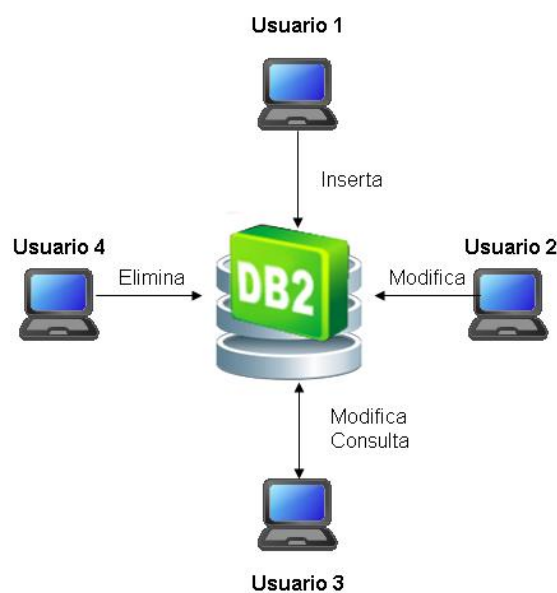


Figura 3: Base de Datos OLTP  
(IBM, 2015)

Las bases de datos de tipo OLAP (On Line Analytical Processing) también son llamadas bases de datos estáticas lo que significa que la información en tiempo real no es afectada, es decir, no se insertan, no se eliminan y tampoco se modifican datos; solo se realizan consultas sobre los datos ya existentes para el análisis y toma de decisiones. Este tipo de bases de datos son implementadas en Business Intelligence para mejorar el desempeño de las consultas con grandes volúmenes de información.

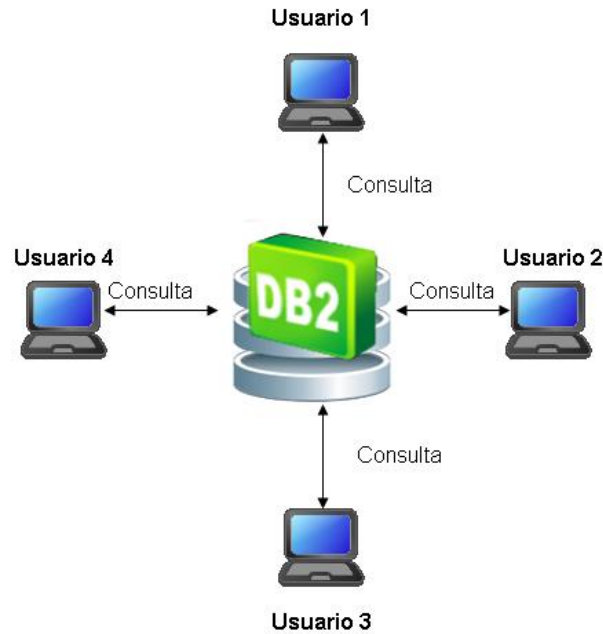


Figura 4: Base de Datos OLAP  
(IBM, 2015)

#### 1.2.4 Replicación de Base de Datos

La replicación de bases de datos forma parte de la comúnmente denominada sincronización de la información, con el único propósito de mantener actualizada dicha información.

Para (Carrascal Reyes & Arciniegas Herrera, 2019) La replicación en bases de datos puede clasificarse de acuerdo a dos aspectos fundamentales, el primero es en qué momento se realiza la replicación y el segundo es quién es capaz de realizar las actualizaciones en las diversas réplicas. Dependiendo del momento en el que se realiza la replicación esta puede clasificarse en eager (temprana) o lazy (perezosa o tardía).

##### a) Replicación eager:

también conocida como replicación síncrona, consiste en que las actualizaciones son propagadas dentro del tiempo de ejecución de la transacción solicitada por el cliente, es decir, la transacción es recibida por un nodo en especial, y este la ejecuta localmente, luego envía las actualizaciones que hayan tenido lugar a las demás réplicas y queda en espera de una respuesta de estas para verificar que no existan problemas de inconsistencia, una vez garantizada la consistencia del sistema, el nodo que recibió la solicitud

entrega una respuesta al cliente, en caso de encontrarse problemas de consistencia la transacción es abortada por completo y ningún cambio surte efecto en ninguna réplica. (Carrascal Reyes & Arciniegas Herrera, 2019)

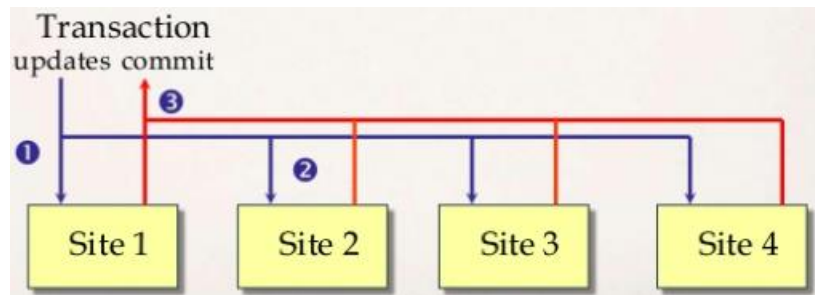


Figura 5: Eager Replication

#### b) Replicación lazy:

Permite que las transacciones se realicen localmente en el servidor que recibió la petición e inmediatamente después de completadas una respuesta es enviada al cliente. La propagación de las actualizaciones hacia las demás réplicas únicamente es enviada después de completada la transacción. Este comportamiento ofrece un bajísimo tiempo de respuesta, algo que, sumado con la simpleza de la técnica, se toma como el mayor fuerte de esta replicación. Sin embargo, en contraste con las técnicas eager, al no existir una fase de acuerdo entre todas las réplicas, la consistencia de la información es el factor débil de este tipo de replicación, y, por ende, muchas veces es necesario hacer uso de técnicas de reconciliación, para corregir los errores presentados durante la replicación. (Carrascal Reyes & Arciniegas Herrera, 2019)

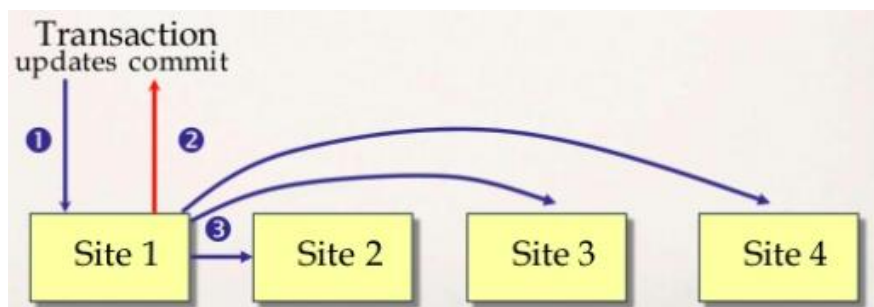


Figura 6: Lazy Replication

### 1.2.5 Conmutación de Conexiones

También conocido como conmutación de paquetes; es una de las formas en la que las redes confiables proporcionan redundancia mediante la implementación de una red conmutada por paquetes. La conmutación por paquetes divide el tráfico en paquetes que se enrutan a través de una red compartida. Un solo mensaje, como un correo electrónico o una transmisión de vídeo, se divide en múltiples bloques de mensajes, llamados paquetes.

A nivel de servidores de base de datos, estos dentro de la red conmutan los paquetes de conexión según la condición de la red en ese momento, esto significa que todos los paquetes en un mismo mensaje pueden tomar diversas conexiones con el fin de establecer un único punto de compilación.



Figura 7: Conmutación de Conexiones

## 1.3 Definiciones Conceptuales

### 1.3.1 Postgresql

PostgreSQL es un poderoso sistema de base de datos relacional de objetos de código abierto que usa y extiende el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central. PostgreSQL se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto de características robustas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer soluciones innovadoras y de alto rendimiento.

PostgreSQL se ejecuta en todos los principales sistemas operativos, cumple con ACID desde 2001 y tiene complementos potentes como el extensor de base de datos geoespaciales PostGIS. No sorprende que PostgreSQL se haya convertido en la base de datos relacional de código abierto elegida por muchas personas y organizaciones.

Posee una amplia reputación gracias a sus características de seguridad, integridad y flexibilidad en comparación con otros servidores de bases de datos.

Postgresql permite escalar con un tamaño de base de datos ilimitado, así como permitir conexiones ilimitadas de usuarios concurrentes.

La principal ventaja de postgres radica en la reducción de costos por implementación y licenciamiento, haciéndolo muy atractivo para las instituciones estatales que buscan minimizar los costos. (Group, 2019)

### **1.3.2 Replication Manager**

Conocido como repmanager (administrador de replicación), es un conjunto de herramientas de código abierto, utilizado para administrar la replicación de bases de datos y realizar el intercambio automático cuando exista un error en el clúster principal (maestro), dirigiendo los datos hacia los clúster standby.

Esta herramienta perteneciente a PostgreSQL optimiza los niveles de disponibilidad configurando servidores en espera, monitorea la replicación y ejecuta tareas administrativas como la conmutación por error o las operaciones de conmutación manual.

Repmaganer realiza la replicación de transmisión para que un servidor en espera se mantenga más actualizado de lo que es posible con el envío de registros basado en archivos. El modo de espera se conecta al primario, que transmite los registros WAL al modo de espera a medida que se generan, sin esperar a que se llene el archivo WAL.

La replicación de transmisión es asíncrona de forma predeterminada, en cuyo caso hay un pequeño retraso entre la confirmación de una transacción en el primario y los cambios que se hacen visibles en el modo de espera. Sin embargo, este retraso es mucho menor que con el envío de registros basado en archivos, generalmente menos de un segundo, suponiendo que el modo de espera sea lo suficientemente potente como para mantenerse al día con la carga. Con la replicación de



transmisión, `archive_timeout` no es necesario para reducir la ventana de pérdida de datos. (2ndQuadrant, 2019)

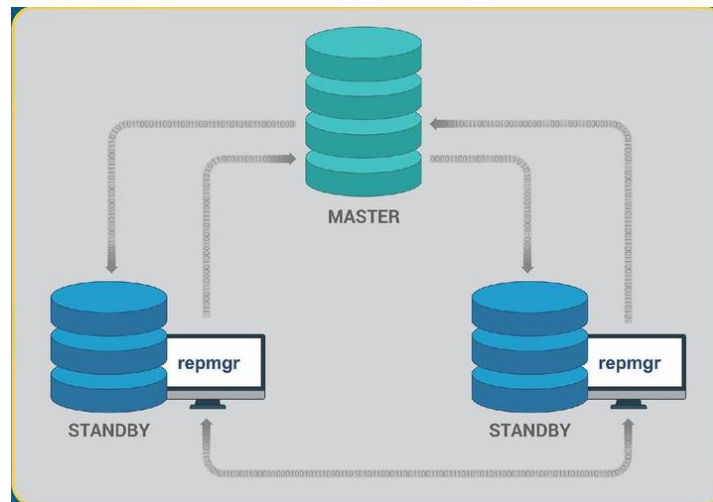


Figura 8: Modelo de Replicación con Repmgr

### 1.3.3 PG Bouncer

Es una herramienta binaria nativa de PostgreSQL que agrupa las conexiones, es de código abierto, ligero y único. Puede agrupar conexiones a una o más bases de datos (de preferencia en servidores diferentes) y atender a clientes a través de sockets de dominio TCP y Unix.

PgBouncer mantiene un grupo de conexiones para cada usuario único, por de bases de datos. Por lo general, está configurado para entregar una de estas conexiones a una nueva conexión de cliente entrante y devolverla al grupo cuando el cliente se desconecta. Puede configurar PgBouncer para agrupar de manera más agresiva para que pueda recoger y devolver la conexión al grupo en los límites de la transacción o la declaración en lugar de los límites de conexión. Sin embargo, hay algunas consecuencias potencialmente indeseables para ellos. (2ndQuadrant, 2019)

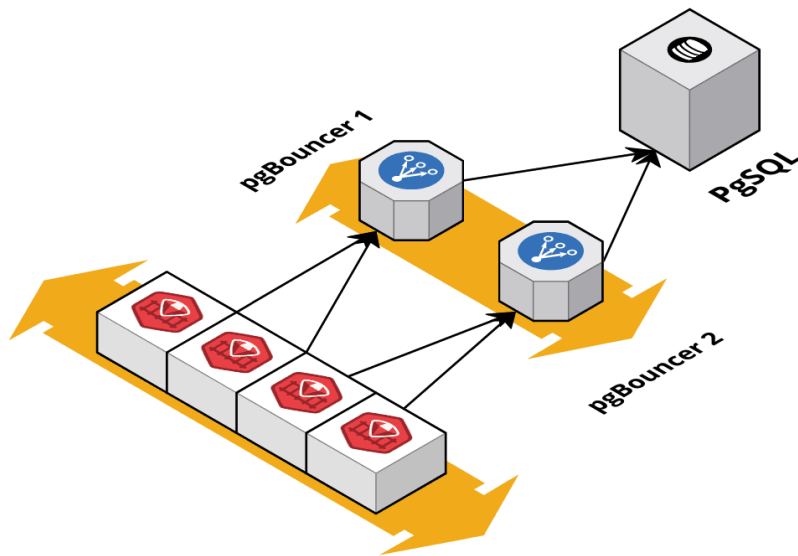


Figura 9: Conmutador de Conexiones

### 1.3.4 Patroni

Patroni es una plantilla para que pueda crear su propia solución personalizada de alta disponibilidad utilizando Python, y para una máxima accesibilidad ofrece una tienda de configuración distribuida como ZooKeeper, etcd, Consul o Kubernetes. Ideal para los ingenieros de bases de datos, DBA, ingenieros de DevOps y SRE que buscan implementar rápidamente HA PostgreSQL en el centro de datos. (SE, 2015)

Principales características:

- Plantilla / marco de gestión de clúster PostgreSQL
- Habla con un almacén de valores clave de consenso distribuido para decidir el estado del Clúster
- Se puede obtener un consenso distribuido usando etcd, ZooKeeper, Consul, etc. para elegir un líder.
- Monitoreo continuo y failover automático.
- Automatización incorporada para recuperar un nodo fallido en el clúster.
- API REST para configuración de clúster y herramientas adicionales.
- Proporciona infraestructura para la conmutación por error transparente de aplicaciones.
- Consenso distribuido para cada acción y configuración.
- Integración con Linux watchdog para evitar el síndrome de cerebro dividido.

- j) Admite conmutación por error manual y automática

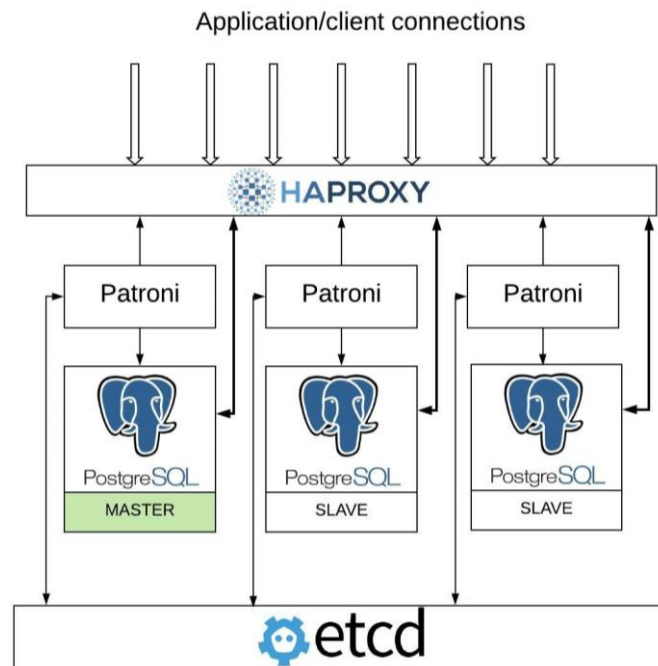


Figura 10: Modelo de Arquitectura Patroni

#### 1.4 Situación Problemática

Con el paso de los años la información se ha convertido en el activo más importante para las empresas tanto públicas como privadas, debido a que el éxito en sus operaciones radica en ofrecer la información de forma rápida, bien organizada, segura y sobre todo siempre disponible. (Sánchez, 2003)

Para (Hernández, 2009) la seguridad de la información es muy importante, en el caso de una aplicación presentara problemas, podría afectar a las demás hasta el grado de llegar a perder la información existente en el equipo, es así como surge la necesidad de la disponibilidad de la información.

La falta de mecanismos de replicación de datos y continuidad de servicios, según (Zamora, 2014) aumenta las probabilidades de que ocurran desastres y puedan generarse daños irreparables a la información.

El Gobierno Regional Lambayeque cuenta con un servicio de Base de Datos implementado con un Clúster de Postgresql versión 9.6 que permite gestionar los registros que son generados a través de sus servicios web tales como SIGA

(Sistema Integrado de Gestión Administrativa), SISGEDO (Sistema de Gestión Documentaria) y la Administración de Portales Web.

El gran crecimiento de los datos, demanda de un mayor recurso tecnológico en hardware y software para poder organizar la información en bases de datos y que esta sea administrada de una forma eficiente; este aumento de la información exige una mayor capacidad de procesamiento, almacenamiento, mecanismos de contingencia y equipamiento redundante a fin de evitar interrupciones imprevistas en sus motores de base de datos.

Este servicio es de vital importancia para la el Gobierno Regional Lambayeque por ser un eje fundamental para el funcionamiento y operación de los servicios web antes mencionados; pero continuamente viene presentando interrupciones en las horas de más alto tráfico, generando la paralización de los servicios que se le brinda al ciudadano.

A la fecha no existe una arquitectura tecnológica con mecanismos de continuidad de servicio y replicación de base datos que pueda evitar interrupciones, prevea las caídas de este servicio (optimizar) y brinde disponibilidad con una alternativa secundaria de tal manera que los usuarios y el ciudadano no se vean afectados por estos inconvenientes.

#### **1.4.1 Formulación del Problema**

¿De qué manera se logrará optimizar el servicio de base de datos del Gobierno Regional Lambayeque?

## 1.5 Operacionalización de Variables

### 1.5.1 Esquema Propuesto

En este esquema se describe la estructura de la propuesta que esencialmente está compuesta por los mecanismos repmanager (administrador de replicaciones) y el pgbouncer (conmutador de conexiones) que permitirán optimizar y obtener una alta disponibilidad del servicio de base de datos con postgresSQL del Gobierno Regional Lambayeque.

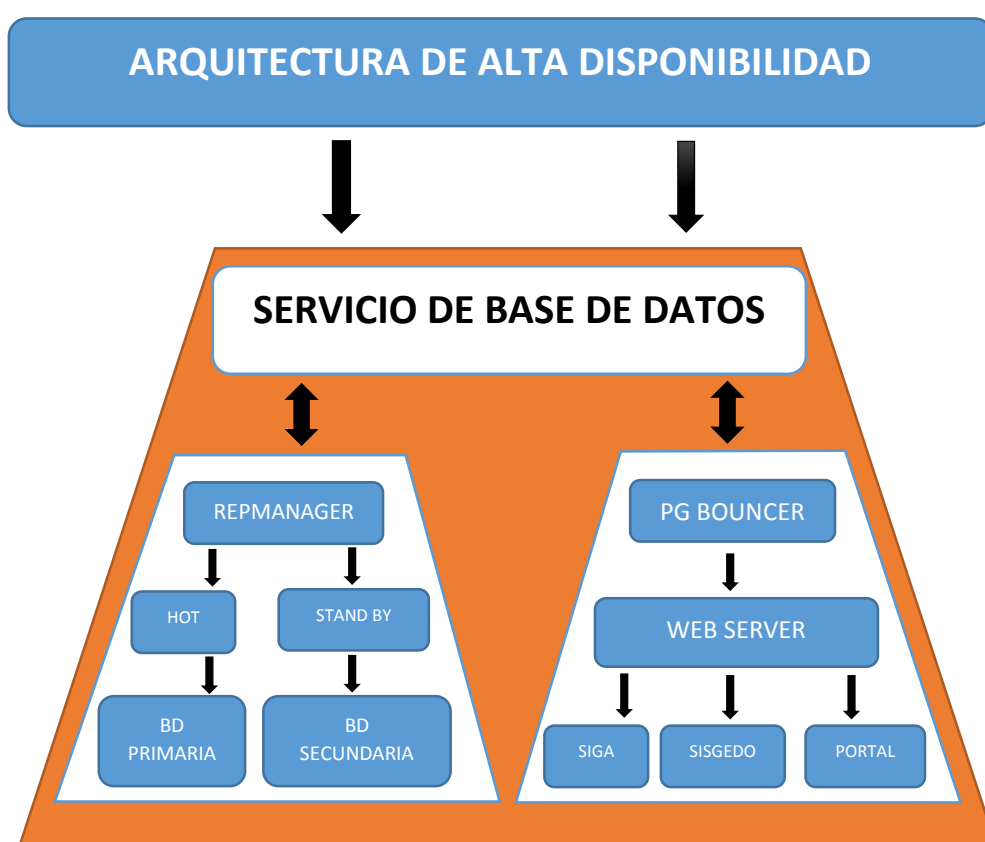


Figura 5: Esquema Propuesto

VARIABLES	DIMENSIÓN	INDICADOR	MEDICIÓN	INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN
DEPENDIENTE				
Servicio de Base de Datos	<b>Replication Manager</b> (Administrador de Replicaciones)	Cantidad de interrupciones	Escala Tipo Likert	- Encuesta
		Frecuencia de la interrupciones		
	<b>Pg_Bouncer</b> (Conmutador de Conexiones)	Tiempo de inactividad	Escala Tipo Likert	- Encuesta
		Riesgo de pérdida de información		
Descripción: contenedor que permite almacenar la información de forma ordenada con diferentes propósitos y usos	Supervisión y Monitoreo	tiempo de reporte de errores	Escala Tipo Likert	- Encuesta - PRTG Network Monitor

Tabla 1: Operacionalización de la Variable Dependiente

## 1.6 Hipótesis

La implementación de una Arquitectura Tecnológica compuesto por la replicación de base de datos y conmutación de conexiones logra una Alta Disponibilidad en el Servicio de Base de Datos del Gobierno Regional Lambayeque.

## **2. Capitulo II. Métodos y Materiales**

## **2.1 Tipo de Investigación**

Según el problema de investigación y los objetivos descritos, el tipo de investigación se determina:

- a. Por su naturaleza o enfoque de investigación:  
La presente investigación es Cuantitativa.
- b. Según su método o diseño de investigación:  
La presente investigación es Pre experimental.
- c. Según su alcance o el nivel de conocimientos que se adquieren:  
La presente investigación es Descriptiva.
- d. Según el propósito o finalidad perseguida:  
La presente investigación es Aplicada.
- e. Según su alcance temporal:  
La presente investigación es Transversal

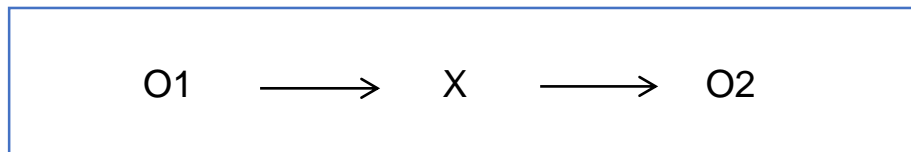
## **2.2 Método de Investigación**

Método experimental, comprendido en dos fases; antes de la implementación de la propuesta y después de la implementación de la propuesta

El instrumento utilizado fue la encuesta, la cual se aplicó en cada una de las fases descritas anteriormente.



## 2.3 Diseño de Contrastación



Donde:

**O1:** Pre observación o diagnóstico del problema

**X:** Es la propuesta

**O2:** Post observación u observación después de implementada la propuesta.

## 2.4 Población, Muestra y Muestreo

### 2.4.1 Población

La Población está conformada por los 500 trabajadores del Gobierno Regional Lambayeque que operan y hacen uso del servicio de Base de Datos a través de los sistemas informáticos. Es decir,  $N = 500$

### 2.4.2 Muestreo

Para el cálculo de la muestra utilizamos la siguiente ecuación para poblaciones finitas:

$$n = \frac{Z^2(pqN)}{Ne^2 + Z^2(pq)}$$

**Donde:**

Z=nivel de Confianza

p=probabilidad a favor

q=probabilidad en contra

N=universo

e=error de estimación

n=tamaño de la muestra

### **Considerando los valores**

$Z=95\%$

$p=50\%$

$q=50\%$

$N=500$

$e=5\%$

$n=?$

El resultado es **76.4506**, lo que indica que la muestra fue de **76** trabajadores. Es decir,  $n = 76$ .

#### **2.4.3 Muestra**

Para nuestro estudio hemos utilizado el muestreo probabilístico con la técnica de muestreo aleatorio simple (MAS).

### **2.5 Técnicas, Instrumentos, Equipos y Materiales de Recolección de Datos**

#### **2.5.1 Técnicas:**

Las técnicas de recolección de datos que se utiliza en el estudio, son:

- a) Entrevista: porque nos permite la obtención de información mediante el diálogo con el jefe y el DBA de la Oficina de Organización y Tecnologías de la Información, en un encuentro formal y planeado.
- b) Encuesta: nos permite obtener información y realizar su tabulación de una manera más sencilla y rápida con informes directos de los usuarios que utilizan el servicio.

#### **2.5.2 Instrumentos de Recolección de Datos:**

- a) Cuestionario de preguntas: Se formó una agenda para visitar la instrucción y entrevistar al Jefe de la Oficina de Organización y Tecnologías de la Información del Gobierno Regional Lambayeque, para poder obtener información acerca de la tecnología que utilizan.
- b) Hoja de Encuesta: Mediante el uso de encuestas, se provee una documentación donde se muestra cuadros estadísticos e indicadores de información en un documento formal para su posterior análisis e interpretación.

### 2.5.3 Equipos

La implementación de la propuesta requirió el uso de una única computadora personal que contaba con las siguientes características técnicas:

Característica Técnica	Detalle
Procesador	CORE I3
Disco Duro	1TB
Memoria RAM	4 GB
Tarjeta de Video	4 GB
Pantalla Led	18''

Tabla 2: Equipos

### 2.5.4 Herramientas

a) Software de Virtualización Oracle Virtualbox

Para la implementación de la arquitectura se utilizó el virtualizador virtualbox de Oracle con tres máquinas virtuales:

- 1. SRV-BD\_MASTER
- 2. SRV-STANDBY
- 3. SRV-APLI

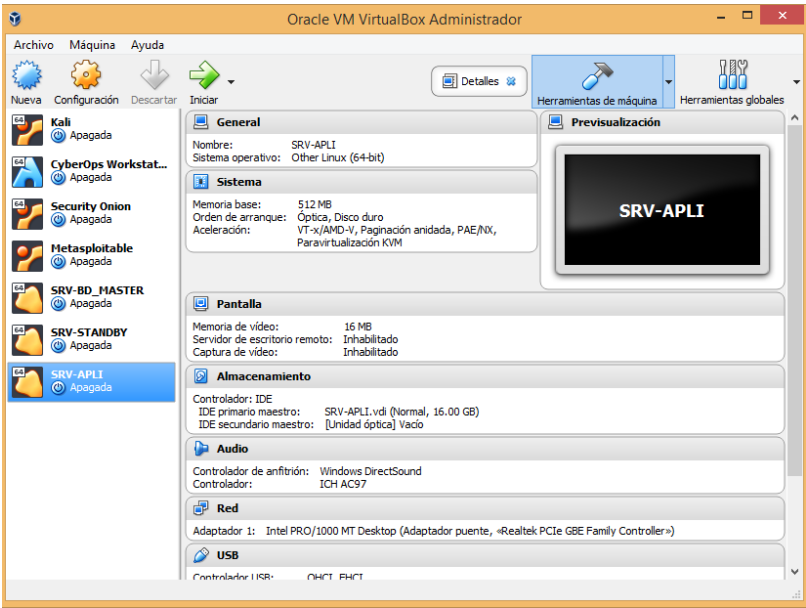


Figura 11: Software de virtualización Oracle Virtualbox

b) Software PRTG Network Monitor

Este software permitió monitorear la disponibilidad del servicio de base de datos en tiempo real, tanto del master como del standby, otorgando al administrador la oportunidad de tomar acciones correctivas según sea el caso.

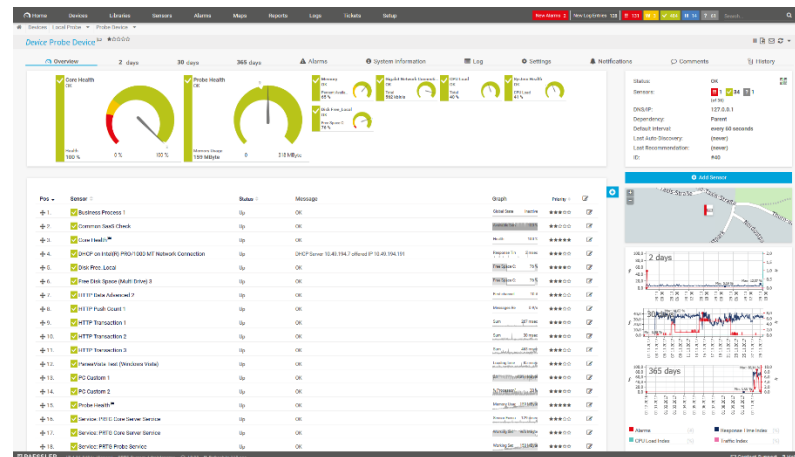


Figura 12: Software de monitoreo de servicio de base de datos

c) Sistema Operativo Centos 7

Para la implementación de la arquitectura, se ha considerado utilizar la última versión del sistema operativo Centos (versión 7.5), software libre que utiliza el Gobierno Regional Lambayeque, el mismo que es compatible con cada una de las aplicaciones que esta institución utiliza.



Figura 13: Sistema Operativo Centos 7

### 2.5.5 Materiales

Para la elaboración del presente informe de tesis, se utilizaron los siguientes materiales.

Descripción	Cantidad	Precio (S/.)
Papel Bond A4	1 ½ millar	37.00
Lapicero	3 unidades	1.80
CD	2 unidades	2.00
Fólder	5 unidades	5.00
Tinta de Impresora	2 frascos	20.00
Libros	2 libros	110.00
<b>Total</b>		<b>S/. 175.80</b>

*Tabla 3: Materiales*

## 2.6 Procesamiento y Análisis de Datos

Para el procesamiento de los datos se utiliza la herramienta IBM SPSS STATISTICS versión 22.0 y Microsoft Excel 2016.

Dichas herramienta permitieron tabular los datos recolectados a través de las encuestas, ordenarlas, obtener el grado de confiabilidad y presentar un gráfico de resultados que fueron determinantes para el logro de objetivos de la propuesta.

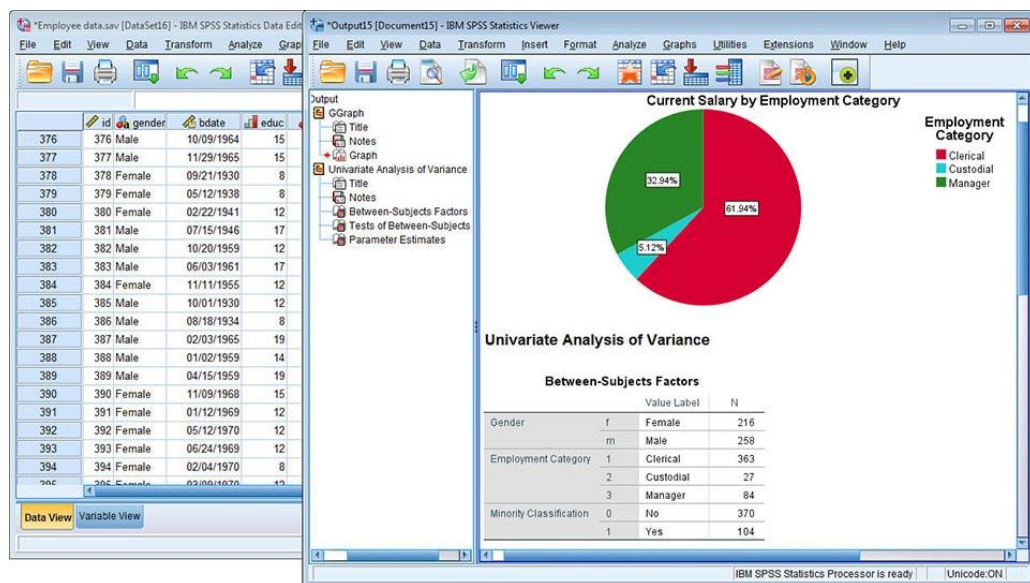


Figura 14: Software IBM SPSS Statistics

### **3. Capitulo III. Resultados**

### 3.1 Diagnóstico del Servicio de Base de Datos del Gobierno Regional Lambayeque

El diagnóstico del variable servicio de base de datos, considera sus dos dimensiones, indicadores, instrumento y escala de medición tipo Likert del 1 al 3.

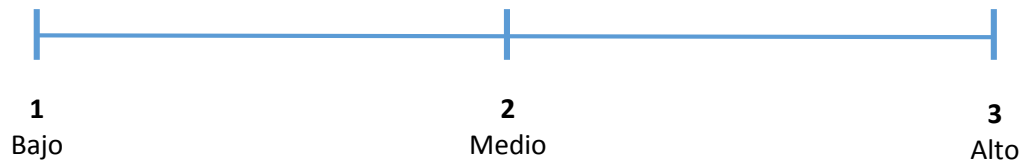


Figura 15: Escala de Medición

#### 3.1.1 Dimensión Administrador de Replicaciones

Para el estudio hemos utilizado como instrumento de medición una encuesta realizada a los 76 trabajadores o usuarios que forman parte de la muestra y que operan los sistemas informáticos que utilizan el servicio de base de datos del Gobierno Regional Lambayeque, cuya matriz de recolección de datos se muestra en la siguiente tabla 5:

	Pregunta 1	Pregunta 2
Encuestado 1	1	2
Encuestado 2	2	2
Encuestado 3	2	2
Encuestado 4	2	2
Encuestado 5	2	3
Encuestado 6	3	2
Encuestado 7	3	2
Encuestado 8	2	2
Encuestado 9	3	3
Encuestado 10	3	2
Encuestado 11	3	3
Encuestado 12	3	3
Encuestado 13	2	2
Encuestado 14	3	3
Encuestado 15	1	2
Encuestado 16	3	3
Encuestado 17	1	3
Encuestado 18	2	1
Encuestado 19	3	2



Encuestado 20	2	2
Encuestado 21	3	3
Encuestado 22	3	3
Encuestado 23	3	3
Encuestado 24	3	3
Encuestado 25	3	3
Encuestado 26	3	3
Encuestado 27	3	3
Encuestado 28	3	3
Encuestado 29	3	3
Encuestado 30	3	3
Encuestado 31	3	3
Encuestado 32	3	3
Encuestado 33	3	3
Encuestado 34	3	3
Encuestado 35	3	3
Encuestado 36	3	3
Encuestado 37	3	3
Encuestado 38	3	3
Encuestado 39	3	3
Encuestado 40	3	3
Encuestado 41	3	3
Encuestado 42	3	3
Encuestado 43	3	3
Encuestado 44	3	3
Encuestado 45	3	3
Encuestado 46	3	3
Encuestado 47	3	3
Encuestado 48	3	3
Encuestado 49	3	3
Encuestado 50	3	3
Encuestado 51	3	3
Encuestado 52	3	3
Encuestado 53	3	3
Encuestado 54	3	3
Encuestado 55	3	3
Encuestado 56	3	3
Encuestado 57	3	3
Encuestado 58	3	3
Encuestado 59	3	2
Encuestado 60	3	3
Encuestado 61	3	3
Encuestado 62	3	3
Encuestado 63	3	3
Encuestado 64	3	3

Encuestado 65	3	3
Encuestado 66	2	3
Encuestado 67	3	3
Encuestado 68	3	3
Encuestado 69	3	3
Encuestado 70	3	3
Encuestado 71	3	3
Encuestado 72	3	3
Encuestado 73	3	3
Encuestado 74	3	3
Encuestado 75	3	2
Encuestado 76	3	3

Tabla 4: Resultados de diagnóstico de la dimensión Administrador de Replicaciones

La tabla de resultados de diagnóstico N° 5 corresponde a las dos primeras preguntas de la encuesta aplicada a los trabajadores y que forman parte de la primera dimensión: administrador de replicaciones.

Realizamos el análisis de fiabilidad de los datos recolectados utilizando SPSS versión 22, y el resultado fue Aceptable con un valor de 0,702, según (George & Mallery, 2003)

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,702	2

Tabla 5: Análisis de Fiabilidad de la dimensión administrador de replicaciones

Para la determinación matemáticamente del problema, utilizamos la siguiente formula:

$$\frac{(\text{Total 1}) \times 1 + (\text{Total 2}) \times 2 + (\text{Total 3}) \times 3}{(N^{\circ} \text{ de preguntas})(\text{muestra})}$$

*Ecuación 1: Cálculo Matemático del Problema*

**Donde:**

Total de 1: son la cantidad de respuestas de escala 1 (bajo)



### 3.1.2 Dimensión Conmutación de Conexiones

Utilizamos la matriz de recolección de datos de las preguntas correspondientes a esta dimensión:

	Pregunta 3	Pregunta 4
Encuestado 1	1	2
Encuestado 2	2	2
Encuestado 3	2	2
Encuestado 4	2	2
Encuestado 5	2	3
Encuestado 6	3	2
Encuestado 7	3	2
Encuestado 8	2	2
Encuestado 9	3	3
Encuestado 10	3	2
Encuestado 11	3	3
Encuestado 12	3	3
Encuestado 13	2	2
Encuestado 14	3	3
Encuestado 15	1	2
Encuestado 16	3	3
Encuestado 17	1	3
Encuestado 18	2	1
Encuestado 19	3	2
Encuestado 20	2	2
Encuestado 21	3	3
Encuestado 22	3	3
Encuestado 23	3	3
Encuestado 24	3	3
Encuestado 25	3	3
Encuestado 26	3	3
Encuestado 27	3	3
Encuestado 28	3	3
Encuestado 29	3	3
Encuestado 30	3	3
Encuestado 31	3	3
Encuestado 32	3	3
Encuestado 33	3	3
Encuestado 34	3	3
Encuestado 35	3	3
Encuestado 36	3	3
Encuestado 37	3	3
Encuestado 38	3	3

Encuestado 39	3	3
Encuestado 40	3	3
Encuestado 41	3	3
Encuestado 42	3	3
Encuestado 43	3	3
Encuestado 44	3	3
Encuestado 45	3	3
Encuestado 46	3	3
Encuestado 47	3	3
Encuestado 48	3	3
Encuestado 49	3	3
Encuestado 50	3	3
Encuestado 51	3	3
Encuestado 52	3	3
Encuestado 53	3	3
Encuestado 54	3	3
Encuestado 55	3	3
Encuestado 56	3	3
Encuestado 57	3	3
Encuestado 58	3	3
Encuestado 59	3	2
Encuestado 60	3	3
Encuestado 61	3	3
Encuestado 62	3	3
Encuestado 63	3	3
Encuestado 64	3	3
Encuestado 65	3	3
Encuestado 66	2	3
Encuestado 67	3	3
Encuestado 68	3	3
Encuestado 69	3	3
Encuestado 70	3	3
Encuestado 71	3	3
Encuestado 72	3	3
Encuestado 73	3	3
Encuestado 74	3	3
Encuestado 75	3	2
Encuestado 76	3	3

*Tabla 6: Resultados de diagnóstico de la dimensión Conmutación de Conexiones*

La tabla de resultados de diagnóstico N° 7 corresponde a las preguntas 3 y 4 de la encuesta aplicada a los trabajadores y que forman parte de la segunda dimensión: conmutación de conexiones.

Realizamos el análisis de fiabilidad de los datos recolectados utilizando SPSS versión 22, y el resultado fue Aceptable con un valor de 0,762 según (George & Mallery, 2003)

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,762	2

Tabla 7: Análisis de Fiabilidad de la dimensión Conmutador de Conexiones

### Cálculo matemático del problema:

$$\frac{(6 \times 1) + (23 \times 2) + (123 \times 3)}{2 \times 76}$$

$$\frac{(6) + (46) + (369)}{152}$$

$$\frac{421}{152} = 2.769736$$

El resultado de la ecuación nos devuelve un valor de 2.769736

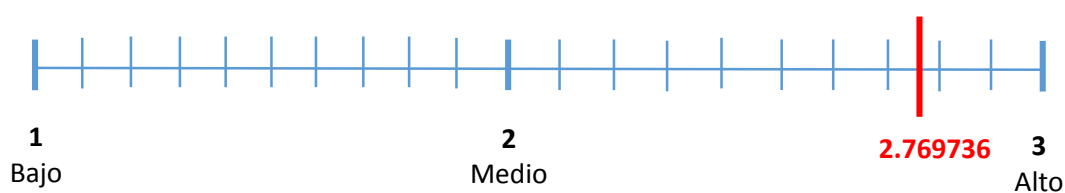


Figura 17: Escala de medición de Dimensión Conmutación de Conexiones

De acuerdo a la escala tipo Likert podemos determinar en base a los indicadores que:

- a) El tiempo de inactividad (paralización de servicios) es alto
- b) El riesgo de pérdida de información es alto

### **3.2 Resultados de cada dimensión después de implementada la propuesta**

#### **3.2.1 Dimensión Administrador de Replicaciones**

Con la implementación de la propuesta, aplicando la misma encuesta a los 76 trabajadores o usuarios que forman parte de la muestra y que operan los sistemas informáticos que utilizan el servicio de base de datos del Gobierno Regional Lambayeque, se obtuvo la siguiente matriz de recolección de datos:

	Pregunta 1	Pregunta 2
Encuestado 1	1	1
Encuestado 2	1	1
Encuestado 3	1	1
Encuestado 4	1	1
Encuestado 5	1	1
Encuestado 6	2	3
Encuestado 7	1	1
Encuestado 8	1	1
Encuestado 9	2	2
Encuestado 10	2	1
Encuestado 11	1	1
Encuestado 12	1	1
Encuestado 13	1	2
Encuestado 14	1	1
Encuestado 15	1	1
Encuestado 16	1	1
Encuestado 17	1	1
Encuestado 18	1	1
Encuestado 19	1	1
Encuestado 20	1	1
Encuestado 21	1	1
Encuestado 22	1	1
Encuestado 23	1	1
Encuestado 24	1	1
Encuestado 25	1	1
Encuestado 26	1	1
Encuestado 27	1	1
Encuestado 28	1	1
Encuestado 29	1	1
Encuestado 30	1	1
Encuestado 31	1	1
Encuestado 32	1	1
Encuestado 33	1	1
Encuestado 34	1	1
Encuestado 35	1	1

Encuestado 36	1	1
Encuestado 37	1	1
Encuestado 38	1	1
Encuestado 39	1	1
Encuestado 40	1	1
Encuestado 41	1	1
Encuestado 42	1	1
Encuestado 43	1	1
Encuestado 44	1	1
Encuestado 45	1	1
Encuestado 46	1	1
Encuestado 47	1	1
Encuestado 48	1	1
Encuestado 49	1	1
Encuestado 50	1	1
Encuestado 51	1	1
Encuestado 52	1	1
Encuestado 53	1	1
Encuestado 54	1	1
Encuestado 55	1	1
Encuestado 56	1	1
Encuestado 57	1	1
Encuestado 58	1	1
Encuestado 59	1	1
Encuestado 60	1	1
Encuestado 61	1	1
Encuestado 62	1	1
Encuestado 63	1	1
Encuestado 64	1	1
Encuestado 65	1	1
Encuestado 66	1	1
Encuestado 67	1	1
Encuestado 68	1	1
Encuestado 69	1	1
Encuestado 70	1	1
Encuestado 71	1	1
Encuestado 72	1	1
Encuestado 73	1	1
Encuestado 74	1	1
Encuestado 75	1	1
Encuestado 76	1	1

Tabla 8: Resultados de la dimensión implementada Administrador de Replicaciones



La tabla de resultados de diagnóstico N° 9 corresponde a las dos primeras preguntas de la encuesta aplicada a los trabajadores después de implementada la propuesta y que forman parte de la primera dimensión: administrador de replicaciones.

Realizamos el análisis de fiabilidad de los datos recolectados utilizando SPSS versión 22, y el resultado fue Aceptable con un valor de 0,777, según (George & Mallery, 2003)

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,792	2

Tabla 9: Análisis de Fiabilidad de la dimensión implementada Administrador de Replicaciones

Para la determinación matemáticamente de la solución del problema aplicando la propuesta, utilizamos la siguiente formula:

$$\frac{(\text{Total 1}) \times 1 + (\text{Total 2}) \times 2 + (\text{Total 3}) \times 3}{(N^{\circ} \text{ de preguntas})(\text{muestra})}$$

*Ecuación 2: Cálculo Matemático de la Solución del Problema*

**Donde:**

Total de 1: son la cantidad de respuestas de escala 1 (bajo)

Total de 2: son la cantidad de respuestas de escala 2 (medio)

Total de 3: son la cantidad de respuestas de escala 3 (alto)

N° de preguntas = 2

Muestra = 76

**Reemplazamos:**

$$\frac{(146 \times 1) + (5 \times 2) + (1 \times 3)}{2 \times 76}$$

$$\frac{(146) + (10) + (3)}{152}$$

$$\frac{159}{152} = 1.046052$$

El resultado de la ecuación nos devuelve un valor de 1.046052

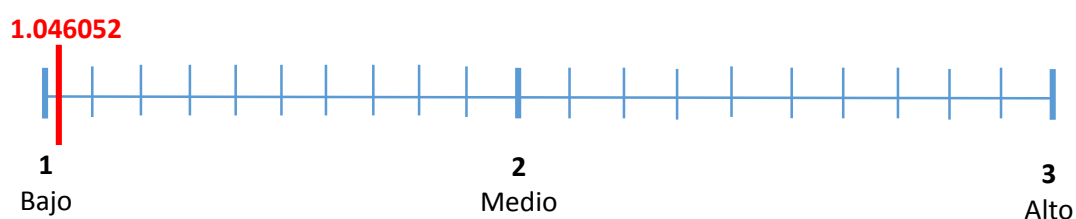


Figura 18: Escala de medición de la dimensión implementada Administración de Replicaciones

De acuerdo a la escala tipo Likert podemos determinar en base a los indicadores que:

- a) Se disminuyó la cantidad de interrupciones hasta el nivel **bajo**.
- b) Se disminuyó la frecuencia de interrupciones hasta el nivel **bajo**.

### 3.2.2 Dimensión Conmutación de Conexiones

Aplicamos la misma encuesta a los 76 trabajadores o usuarios que forman parte de la muestra y que operan los sistemas informáticos que utilizan el servicio de base de datos del Gobierno Regional Lambayeque, de la cual se obtuvo la siguiente matriz de recolección de datos:

	Pregunta 3	Pregunta 4
Encuestado 1	1	1
Encuestado 2	1	1
Encuestado 3	1	1
Encuestado 4	1	1
Encuestado 5	1	1
Encuestado 6	1	1
Encuestado 7	1	1
Encuestado 8	1	1
Encuestado 9	2	2
Encuestado 10	2	3
Encuestado 11	1	1
Encuestado 12	1	1
Encuestado 13	1	2
Encuestado 14	1	1
Encuestado 15	1	1
Encuestado 16	1	1
Encuestado 17	1	1
Encuestado 18	1	1
Encuestado 19	1	1
Encuestado 20	1	1

Encuestado 21	1	1
Encuestado 22	1	1
Encuestado 23	1	1
Encuestado 24	1	1
Encuestado 25	1	1
Encuestado 26	1	1
Encuestado 27	1	1
Encuestado 28	1	1
Encuestado 29	1	1
Encuestado 30	1	1
Encuestado 31	1	1
Encuestado 32	1	1
Encuestado 33	1	1
Encuestado 34	1	1
Encuestado 35	1	1
Encuestado 36	1	1
Encuestado 37	1	1
Encuestado 38	1	1
Encuestado 39	1	1
Encuestado 40	1	1
Encuestado 41	1	1
Encuestado 42	1	1
Encuestado 43	1	1
Encuestado 44	1	1
Encuestado 45	1	1
Encuestado 46	1	1
Encuestado 47	1	1
Encuestado 48	1	1
Encuestado 49	1	1
Encuestado 50	1	1
Encuestado 51	1	1
Encuestado 52	1	1
Encuestado 53	1	1
Encuestado 54	1	1
Encuestado 55	1	1
Encuestado 56	1	1
Encuestado 57	1	1
Encuestado 58	1	1
Encuestado 59	1	1
Encuestado 60	1	1
Encuestado 61	1	1
Encuestado 62	1	1
Encuestado 63	1	1
Encuestado 64	1	1
Encuestado 65	1	1

Encuestado 66	1	1
Encuestado 67	1	1
Encuestado 68	1	1
Encuestado 69	1	1
Encuestado 70	1	1
Encuestado 71	1	1
Encuestado 72	1	1
Encuestado 73	1	1
Encuestado 74	1	1
Encuestado 75	1	1
Encuestado 76	1	1

Tabla 10: Resultados de la dimensión implementada Conmutación de Conexiones

La tabla de resultados de diagnóstico N° 11 corresponde a las preguntas 3 y 4 de la encuesta aplicada a los trabajadores después de implementada la propuesta y que forman parte de la segunda dimensión: conmutación de conexiones.

Realizamos el análisis de fiabilidad de los datos recolectados utilizando SPSS versión 22, y el resultado fue Aceptable con un valor de 0,856, según (George & Mallery, 2003)

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,856	2

Tabla 11: Análisis de Fiabilidad de la dimensión implementada Conmutación de Conexiones

**Cálculo matemático de la solución del problema:**

$$\frac{(147 \times 1) + (4 \times 2) + (1 \times 3)}{2 \times 76}$$

$$\frac{(147) + (8) + (3)}{152}$$

$$\frac{158}{152} = 1.039473$$

El resultado de la ecuación nos devuelve un valor de 1.039473

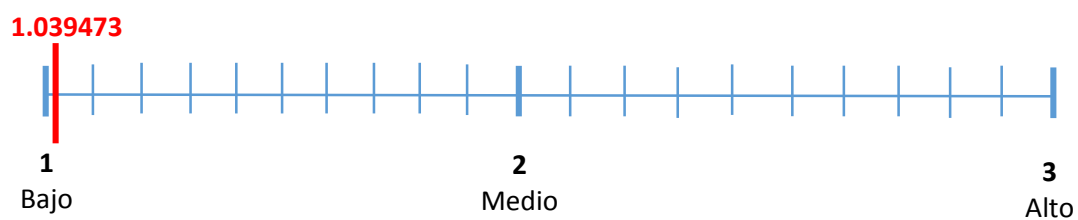


Tabla 12: Escala de medición de la dimensión implementada Conmutación de Conexiones

De acuerdo a la escala tipo Likert podemos determinar en base a los indicadores que:

- a) Se Disminuyó el tiempo de inactividad (paralización de servicios) hasta el nivel **bajo**.
- b) Se Disminuyó el riesgo de pérdida de información hasta el nivel **bajo**.

### **3.3 Desarrollo de la propuesta**

### **3.3.1 Título de la propuesta**

“ARQUITECTURA DE ALTA DISPONIBILIDAD PARA LA OPTIMIZACIÓN DEL SERVICIO DE BASE DE DATOS DEL GOBIERNO REGIONAL LAMBAYEQUE”

### **3.3.2 Objetivos de la propuesta**

#### **3.3.2.1 Objetivo General**

Implementar una arquitectura de alta disponibilidad para optimizar el servicio de Base de Datos del Gobierno Regional Lambayeque

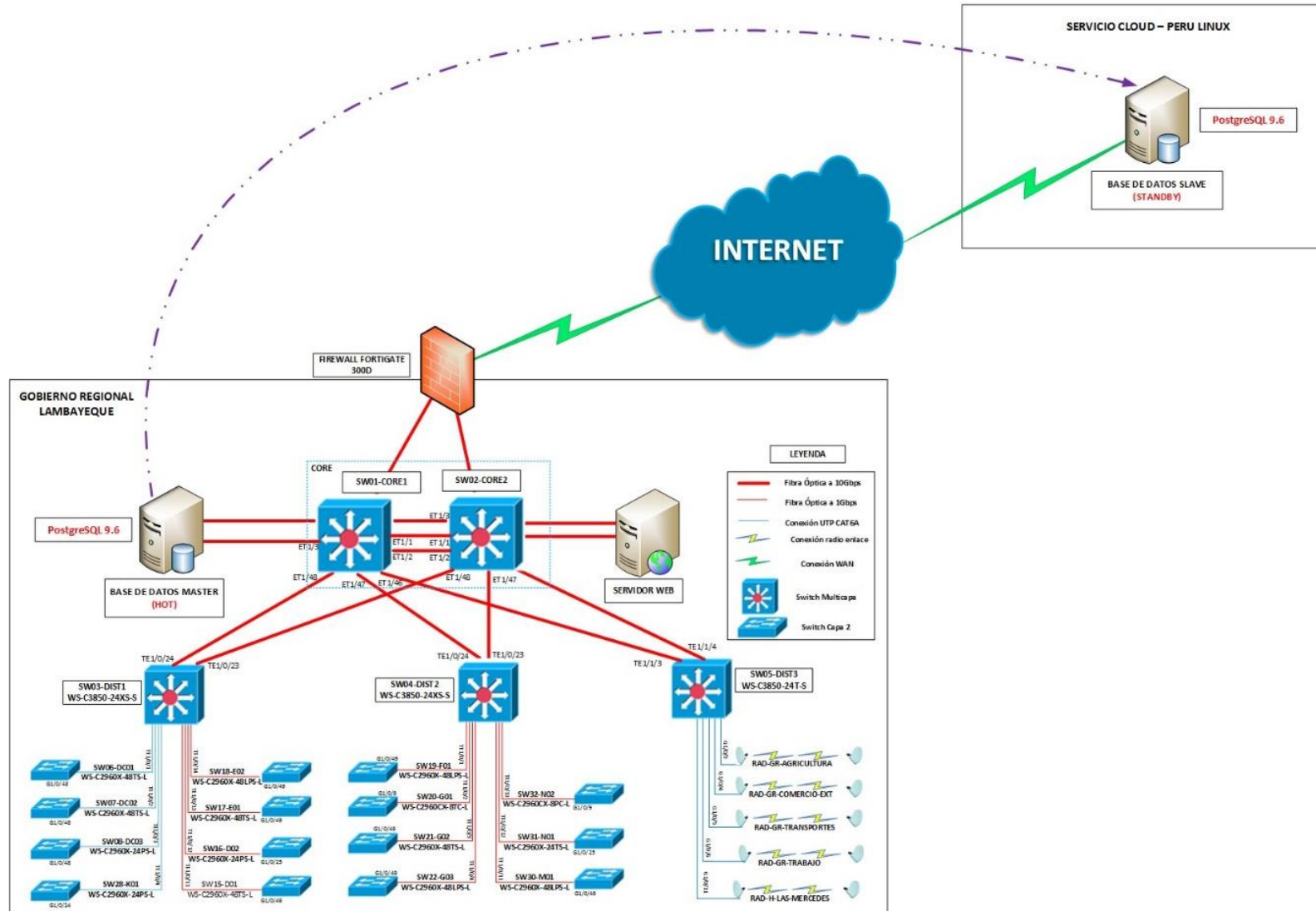
#### **3.3.2.2 Objetivos específicos**

- a) Realizar un diagnóstico de la situación actual del servicio de base de datos
- b) Disminuir las interrupciones imprevistas del servicio de base de datos
- c) Disminuir el tiempo de inactividad del servicio de base de datos.
- d) Implementar un mecanismo de replicación de Base de Datos
- e) Implementar un mecanismo de continuidad del servicio de Base de Datos

### 3.3.3 Esquema de desarrollo de la propuesta

#### 3.3.3.1 Topología lógica propuesta de la arquitectura de alta disponibilidad

Figura 19: Topología Lógica Propuesta





### 3.3.3.2 Motor de base de datos a utilizar

El Gobierno Regional Lambayeque viene utilizando software libre desde el año 2008 en el cual se aprobó su migración a través del Acuerdo Regional N° 199-2008-GR.LAMB/CR, y establecieron sus políticas de incorporación y gestión del software libre a nivel de toda su entidad a través de la Ordenanza Regional N° 019-2008-GR.LAMB/CR.

Por lo tanto su motor de base de datos posee un clúster de **PostgreSQL versión 9.6**, el cual contiene las bases de datos tales como SIGA y SISGEDO2 pertenecientes al Sistema Integrado de Gestión Administrativa (SIGA) y el Sistema de Gestión Documentaria (SISGEDO) versión 2.0; además de ello el portal web de la institución es creado desde la misma base de datos SIGA.

PostgreSQL por ser un sistema de base de datos de código abierto orientado a objetos, este no es dirigido por una empresa o persona específica, sino más bien esta es liderada por profesionales desarrolladores de software que aportan sin ningún interés comercial al avance de este motor de base de datos relacional.

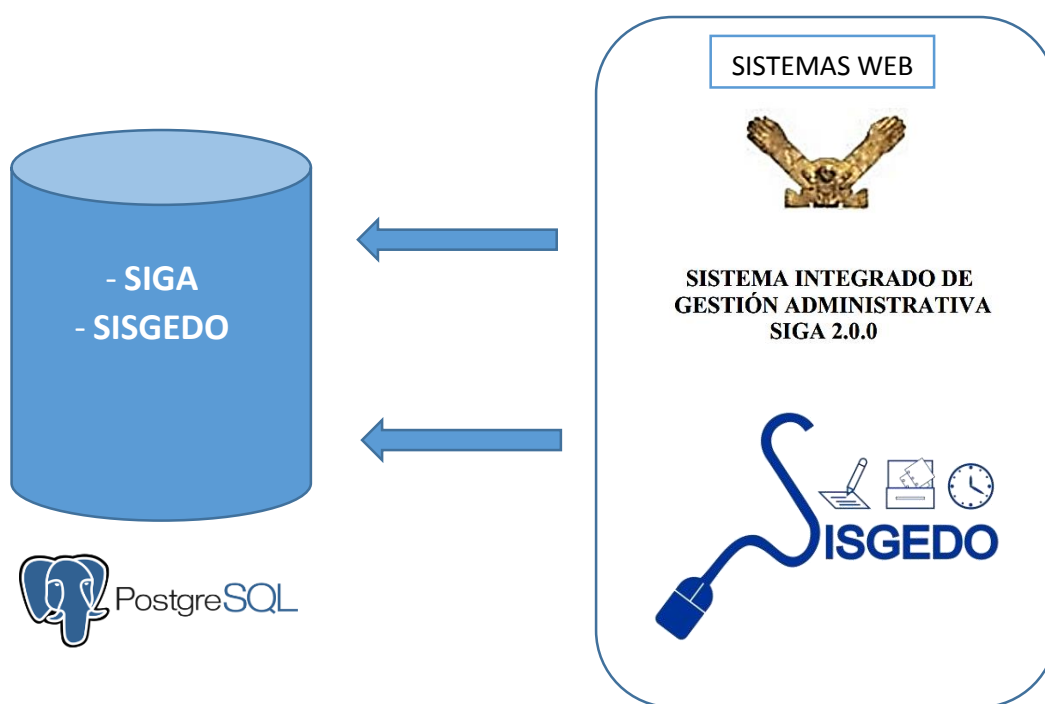


Figura 20: Motor de Base de Datos PostgreSQL

### 3.3.3.3 Mecanismo propuesto de replicación de base de datos

Para la replicación de base de datos utilizamos repmanager versión 9.6 con dos clusters:

- Clúster Master
- Clúster Standby

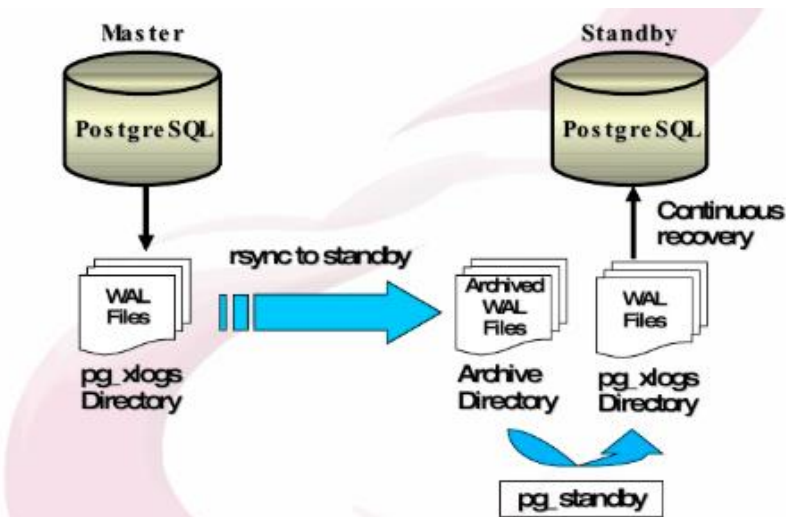


Figura 21: Replication Manager

Comandos de comprobación del funcionamiento del mecanismo de replicación de base de datos:

```
[root@SRV-BD ~]# systemctl status repmgr96.service
```

```
root@SRV-BD-1:~#
-bash-4.2$ psql >C
-bash-4.2$
-bash-4.2$ systemctl start repmgr96.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to manage system services or units.
Authenticating as: administrator
Password:
-bash-4.2$
-bash-4.2$
-bash-4.2$ exit
logout
[root@SRV-BD-1]#
[root@SRV-BD-1]# systemctl status repmgr96.service
● repmgr96.service - A replication manager, and failover management tool for PostgreSQL
Loaded: loaded (/usr/lib/systemd/system/repmgr96.service; disabled; vendor preset: disabled)
Active: inactive (dead)
[root@SRV-BD-1]# systemctl start repmgr96.service
[root@SRV-BD-1]# systemctl status repmgr96.service
● repmgr96.service - A replication manager, and failover management tool for PostgreSQL
Loaded: loaded (/usr/lib/systemd/system/repmgr96.service; disabled; vendor preset: disabled)
Active: active (running) since dom 2019-08-18 15:56:46 -05; 5s ago
Process: 2158 ExecStart=/usr/pgsql-9.6/bin/repmgrd -f $(REPMGRDCONF) -p $(PIDFILE) -d --verbose (code=exited, status=0/SUCCESS)
Main PID: 2162 (repmgrd)
CGroup: /system.slice/repmgr96.service
└─2162 /usr/pgsql-9.6/bin/repmgrd -f /etc/repmgr/9.6/repmgr.conf -...

ago 18 15:56:46 SRV-BD systemd[1]: Starting A replication manager, and failover...
ago 18 15:56:46 SRV-BD repmgrd[2158]: [2019-08-18 15:56:46] [NOTICE] using "...
ago 18 15:56:46 SRV-BD repmgrd[2158]: [2019-08-18 15:56:46] [NOTICE] redirec...
ago 18 15:56:46 SRV-BD systemd[1]: PID file /run/repmgr/repmgrd-9.6.pid not...
ago 18 15:56:46 SRV-BD systemd[1]: Started A replication manager, and failover...
Hint: Some lines were ellipsized, use -l to show in full.
[root@SRV-BD-1]# systemctl status repmgr96.serviceC
[root@SRV-BD-1]#
[root@SRV-BD-1]#
[root@SRV-BD-1]#
[root@SRV-BD-1]#
root@SRV-BD-2:~#
=C>C>postgres
template1 | postgres | UTF8 | es_PE.UTF-8 | es_PE.UTF-8 | =c/postgres
+ | | | | |
=C>C>postgres
(4 files)
postgres=# \q
-bash-4.2$
-bash-4.2$
-bash-4.2$ exit
logout
[root@SRV-BD-2]#
[root@SRV-BD-2]# systemctl status repmgr96.service
● repmgr96.service - A replication manager, and failover management tool for PostgreSQL
Loaded: loaded (/usr/lib/systemd/system/repmgr96.service; disabled; vendor preset: disabled)
Active: inactive (dead)
[root@SRV-BD-2]#
[root@SRV-BD-2]# systemctl start repmgr96.service
[root@SRV-BD-2]#
[root@SRV-BD-2]# systemctl status repmgr96.service
● repmgr96.service - A replication manager, and failover management tool for PostgreSQL
Loaded: loaded (/usr/lib/systemd/system/repmgr96.service; disabled; vendor preset: disabled)
Active: active (running) since dom 2019-08-18 15:57:35 -05; 1s ago
Process: 1641 ExecStart=/usr/pgsql-9.6/bin/repmgrd -f $(REPMGRDCONF) -p $(PIDFILE) -d --verbose (code=exited, status=0/SUCCESS)
Main PID: 1845 (repmgrd)
CGroup: /system.slice/repmgr96.service
└─1845 /usr/pgsql-9.6/bin/repmgrd -f /etc/repmgr/9.6/repmgr.conf -p...

ago 18 15:57:35 SRV-BD systemd[1]: Starting A replication manager, and failover...
ago 18 15:57:35 SRV-BD repmgrd[1841]: [2019-08-18 15:57:35] [NOTICE] using "...
ago 18 15:57:35 SRV-BD repmgrd[1841]: [2019-08-18 15:57:35] [NOTICE] redirec...
ago 18 15:57:35 SRV-BD systemd[1]: PID file /run/repmgr/repmgrd-9.6.pid not...
ago 18 15:57:35 SRV-BD systemd[1]: Started A replication manager, and failover...
Hint: Some lines were ellipsized, use -l to show in full.
[root@SRV-BD-2]#
```

Figura 22: Comandos de verificación del mecanismo Replication Manager

El mecanismo repmanager 9.6 replica cada base de datos en tiempo real, permitiendo que la información pueda salvaguardarse en distintos servidores nacionales así como internacionales; esto otorga a u DBA (Database Administrator) la capacidad de poder redirigir las conexiones de los servidores de aplicaciones y mantener disponible el servicio para los usuarios finales.

The image shows two terminal windows side-by-side, both running PostgreSQL 9.6. The left window is on SRV-BD1 and the right window is on SRV-BD2. Both windows show the same sequence of commands and output, verifying the replication status and listing the databases.

```

root@SRV-BD1:~# systemctl status repmgr96.service
Loaded: loaded (/usr/lib/systemd/system/repmgr96.service; disabled; vendor preset: disabled)
Active: inactive (dead)
[root@SRV-BD1 ~]# systemctl status repmgr96.service
Loaded: loaded (/usr/lib/systemd/system/repmgr96.service; disabled; vendor preset: disabled)
Active: inactive (dead)
[root@SRV-BD1 ~]# su - postgres
Ultimo inicio de sesión:jue jul 18 23:38:46 -05 2019en pts/0
-bash-4.2$
-bash-4.2$
-bash-4.2$
-bash-4.2$
-bash-4.2$
-bash-4.2$ psql
psql (9.6.12)
Digite «help» para obtener ayuda.

postgres=# \l
          Listado de base de datos
+-----+-----+-----+-----+-----+-----+
| Nombre | Dueño | Codificación | Collate | Ctype | Privilegios |
+-----+-----+-----+-----+-----+-----+
| postgres | postgres | UTF8 | es_PE.UTF-8 | es_PE.UTF-8 | postgres |
| siggedo2practica | postgres | UTF8 | es_PE.UTF-8 | es_PE.UTF-8 | postgres |
| template0 | postgres | UTF8 | es_PE.UTF-8 | es_PE.UTF-8 | =c/postgres |
| postgres | postgres | UTF8 | es_PE.UTF-8 | es_PE.UTF-8 | postgres |
+-----+-----+-----+-----+-----+-----+
(4 filas)
postgres=#
  
```

The right window (SRV-BD2) shows identical output, confirming that the databases are replicated in real-time.

Figura 23: Verificación de bases de datos replicadas

### 3.3.3.4 Mecanismo propuesto de conmutación de conexiones

Para la conmutación de conexiones se utiliza PgBouncer versión 9.6, que se encargará de realizar un testeo de la disponibilidad de los servidores de base de datos y determinar el servidor master en operación, y así poder dirigir las conexiones provenientes de los servidores de aplicaciones.

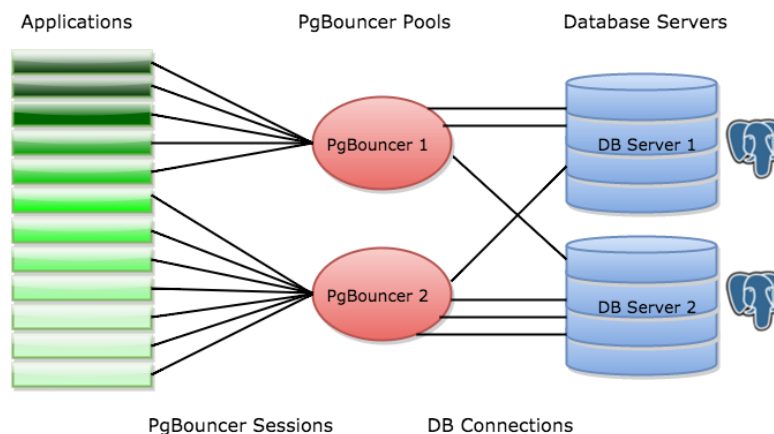


Figura 24: PGBOUNCER - Conmutador de Conexiones

En principio, es necesario verificar que el servicio pgbouncer se encuentre instalado y operativo antes de poner en marcha cualquier

```
root@Srv-Apl1:usr/local/Zend/bin
ago 18 15:45:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:45:12.984 1168 LOG sta...us
ago 18 15:46:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:46:12.985 1168 LOG sta...us
ago 18 15:47:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:47:12.986 1168 LOG sta...us
ago 18 15:48:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:48:12.987 1168 LOG sta...us
ago 18 15:49:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:49:12.988 1168 LOG sta...us
ago 18 15:50:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:50:12.989 1168 LOG sta...us
ago 18 15:50:48 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:50:48.207 1168 LOG S-0...s)
ago 18 15:51:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:51:12.990 1168 LOG sta...us
ago 18 15:52:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:52:12.991 1168 LOG sta...us
ago 18 15:53:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:53:12.992 1168 LOG sta...us
Hint: Some lines were ellipsized, use -l to show in full.
[root@Srv-Apl1 bin]#
[root@Srv-Apl1 bin]# systemctl status pgbouncer
● pgbouncer.service - A lightweight connection pooler for PostgreSQL
   Loaded: loaded (/usr/lib/systemd/system/pgbouncer.service; disabled; vendor preset: disabled)
   Active: active (running) since dom 2019-08-18 15:32:12 -05; 21min ago
     Main PID: 1168 (pgbouncer)
    CGroup: /system.slice/pgbouncer.service
            └─1168 /usr/bin/pgbouncer /etc/pgbouncer/pgbouncer.ini

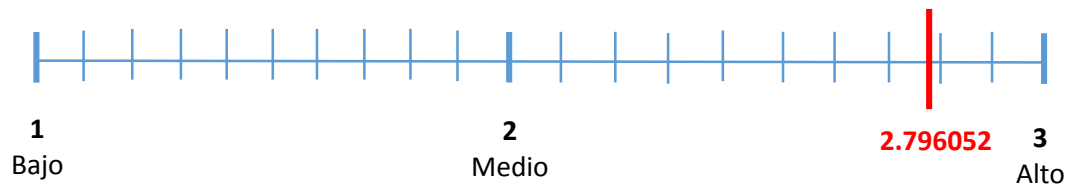
ago 18 15:45:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:45:12.984 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:46:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:46:12.985 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:47:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:47:12.986 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:48:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:48:12.987 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:49:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:49:12.988 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:50:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:50:12.989 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:50:48 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:50:48.207 1168 LOG S-0xafe330: siggedo/postgres@192...022s)
ago 18 15:51:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:51:12.990 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:52:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:52:12.991 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
ago 18 15:53:12 Srv-Apl1 pgbouncer[1168]: 2019-08-18 15:53:12.992 1168 LOG stats: 0 xacts/s, 0 queries/s, in... 0 us
Hint: Some lines were ellipsized, use -l to show in full.
[root@Srv-Apl1 bin]#
```

Figura 255: Verificación del PGBOUNCER

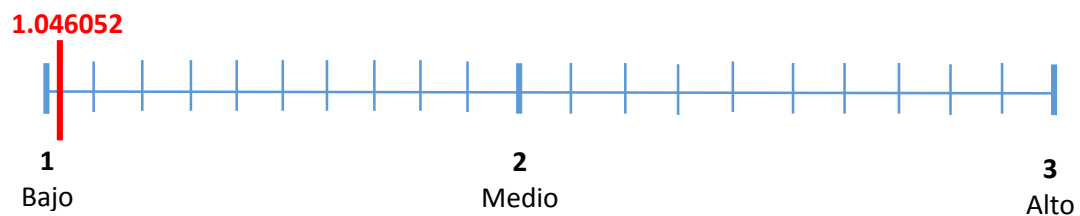
## **4. Capitulo IV. Discusión**

#### 4.1 Discusión de resultados de la dimensión Administrador de Replicaciones

Antes

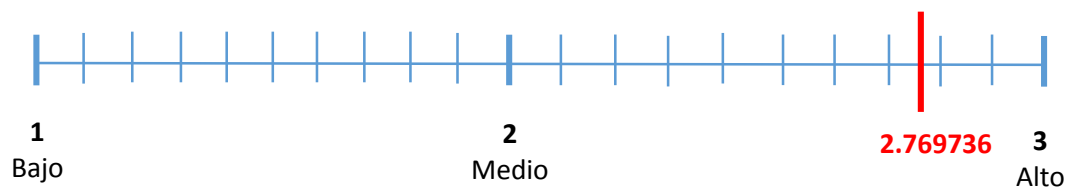


Después

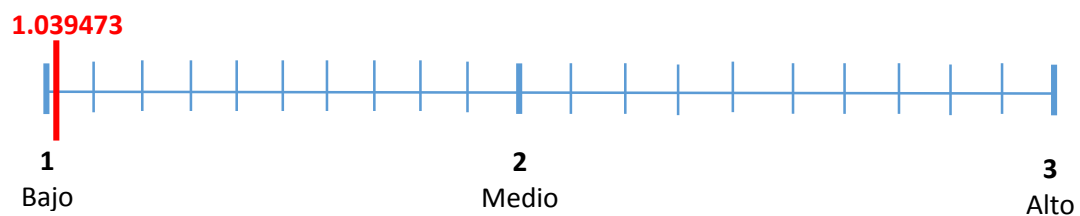


## 4.2 Discusión de resultados de la dimensión Conmutación de Conexiones

Antes



Después



**Interpretación:** el servicio de base de datos ha sido optimizado con la aplicación de nuestra propuesta minimizando:

- a) La cantidad de interrupciones
- b) La frecuencia de interrupciones
- c) El tiempo de inactividad (paralización de servicios)
- d) El riesgo de pérdida de información

Reduciendo de un resultado promedio de 2.782894 (alto) hacia 1.042762 (bajo) en una escala del 1 al 3.

## **Conclusiones y Recomendaciones**



## **Conclusiones**

- a) Se evidenció el estado actual del servicio de base de datos.

Se determinó:

- b) El alto nivel de interrupciones en la base de datos.
- c) El alto nivel de inactividad y paralización de servicios que dependen de la base de datos.

Se implementó:

- d) Un mecanismo de replicación de base de datos para disminuir las interrupciones.
- e) Un mecanismo de continuidad del servicio de base de datos para disminuir el nivel de inactividad y paralización de servicios que dependen de la base de datos.

Se determinó que:

- f) La implementación de un mecanismo de replicación de base de datos disminuye considerablemente las interrupciones.
- g) La implementación de un mecanismo de continuidad del servicio de base de datos disminuyó el nivel inactividad y paralización de servicios que dependen de la base de datos.
- h) Se optimizó el servicio de Base de Datos disminuyendo las interrupciones imprevistas y tiempo de inactividad del servicio de base de datos.
- i) Se logró una alta disponibilidad del servicio de Base de Datos.

## **Recomendaciones**

- a) Aplicar la propuesta considerando los mecanismos de replicación de base de datos y conmutación de conexiones.
- b) Mantener actualizado el mecanismo de replicación de base de datos de la propuesta.
- c) Mantener actualizado el mecanismo de continuidad del servicio de base de datos de la propuesta.
- d) Contar con un personal especializado para el monitoreo y administración de la alta disponibilidad del servicio de base de datos.
- e) Considerar el uso del servicio en cloud (nube) para la replicación de base de datos.

## Referencias Bibliográficas

- 2ndQuadrant. (2019). *Replication Manager for Postgresql Clusters*. Obtenido de Replication Manager for Postgresql Clusters: <https://repmgr.org/>
- Caro, S. E. (2008). *Uso de Sistemas de Gestión de Bases de Datos Paralelas para la*. Obtenido de Uso de Sistemas de Gestión de Bases de Datos Paralelas para la: <http://dspace.uclv.edu.cu/bitstream/handle/123456789/5761/Tesis%20de%20Maestria.pdf?sequence=1&isAllowed=y>
- Carrascal Reyes, M., & Arciniegas Herrera, J. (2019). Sincronización de la información de tasación en entornos convergentes. *Revista UIS Ingenierías*, 21-38.
- Cisco. (2018). *Cisco Networking Academy*. Obtenido de Cisco Networking Academy: <http://cisco.netacad.com>
- George, D., & Mallery, P. (2003). *SPSS for Windows Step by Step, A Simple Guide and Reference Fourth Edition (11.0 update)*. Obtenido de SPSS for Windows Step by Step, A Simple Guide and Reference Fourth Edition (11.0 update): <https://wps.ablongman.com/wps/media/objects/385/394732/george4answers.pdf>
- Group, P. G. (2019). *PostgreSQL: The World's Most Advanced Open Source Relational Database*. Obtenido de PostgreSQL: The World's Most Advanced Open Source Relational Database: <https://www.postgresql.org/>
- Guerrero, R. M. (2016). *Hot Standby Y Streaming Replication*. Obtenido de Hot Standby Y Streaming Replication: <https://e-mc2.net/es/hot-standby-y-streaming-replication>
- Guevara, C. C. (2017). *Implementación de una Infraestructura Tecnológica Virtual con Alta Disponibilidad Basada en Clústers para los Servidores de la Universidad Señor de Sipan - Lambayeque*. Obtenido de Implementación de una Infraestructura Tecnológica Virtual con Alta Disponibilidad Basada en Clústers para los Servidores de la Universidad Señor de Sipan - Lambayeque: [http://alicia.concytec.gob.pe/vufind/Record/UPRG\\_66ae38173dad2c2b3e6df21255c9181f](http://alicia.concytec.gob.pe/vufind/Record/UPRG_66ae38173dad2c2b3e6df21255c9181f)
- Hernández, C. D. (Diciembre de 2009). *Técnicas de Virtualización para Alta Disponibilidad aplicada a Servidores de Bases de Datos Empresariales*. Obtenido de Técnicas de Virtualización para Alta Disponibilidad aplicada a Servidores de Bases de Datos Empresariales: <https://tesis.ipn.mx/jspui/bitstream/123456789/8885/1/TECVIRT.pdf>
- IBM. (2015). *Características y tipos de bases de datos*. Obtenido de Características y tipos de bases de datos: [https://www.ibm.com/developerworks/ssa/data/library/tipos\\_bases\\_de\\_datos/index.html](https://www.ibm.com/developerworks/ssa/data/library/tipos_bases_de_datos/index.html)
- Rivero, E. H. (2017). *Diseño y Configuración de una Arquitectura de Alta Disponibilidad de Base de Datos en Servidores Linux orientada a empresas con una política de continuidad de negocio*. La Paz, La Paz, Bolivia.
- Sánchez, Y. B. (2003). *LA GESTIÓN DEL CONOCIMIENTO Y LAS ORGANIZACIONES MODERNAS*. Obtenido de LA GESTIÓN DEL CONOCIMIENTO Y LAS ORGANIZACIONES MODERNAS:

[http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/administracion/v06\\_n11/gestion.htm](http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/administracion/v06_n11/gestion.htm)

SE, Z. (2015). *Patroni*. Obtenido de Patroni: <https://patroni.readthedocs.io/en/latest/>

Zamora, L. J. (Noviembre de 2014). *Clúster de Servidores Linux para Alta Disponibilidad*. Obtenido de Clúster de Servidores Linux para Alta Disponibilidad: <http://repositorio.unc.edu.pe/bitstream/handle/UNC/547/T%20620.7%20V719%202014.pdf?sequence=1&isAllowed=y>

## Anexos

### 1. Servidor Master

#### A) Instalación de Replication Manager – repmgr

```
[root@SRV-BD ~]# yum install repmgr96 repmgr96-debuginfo
```

```
[root@SRV-BD ~]# su - postgres
```

**Creamos el usuario para replica:**

```
postgres=# create user repmgr replication ;
```

**Asignamos contraseña al usuario:**

```
postgres=# \password repmgr
```

#### Edición del fichero de configuración /etc/repmgr/9.6/repmgr.conf

```
node_id=1
node_name=srv-bd
conninfo='host=192.168.1.110 user=postgres dbname=sisgedo2practica
application_name=repmgr'
data_directory=/var/lib/pgsql/9.6/data
pg_bindir=/usr/pgsql-9.6/bin
replication_user='repmgr'
replication_type=physical
use_replication_slots=yes
async_query_timeout=60
reconnect_attempts=6
reconnect_interval=10
failover=automatic
ssh_options='-q -o StrictHostKeyChecking=no'

log_level=INFO
log_file='/var/log/repmgr/repmgr.log'
monitoring_history=true
promote_command='/etc/repmgr/9.6/promote_command.sh'
follow_command='/etc/repmgr/9.6/follow_command.sh'

service_start_command = sudo systemctl start postgresql-9.6
service_stop_command = sudo systemctl stop postgresql-9.6
service_restart_command = sudo systemctl restart postgresql-9.6
service_reload_command = sudo systemctl reload postgresql-9.6

event_notification_command='/var/lib/pgsql/scripts/events.sh %n %e %s "%t"
"%d"'
```

```
GNU nano 2.3.1                                Fichero: repmgr.conf
node_id=1
node_name=srv-bd
conninfo='host=192.168.1.110 user=postgres dbname=sisgedo2practica application_name=repmgr'
data_directory=/var/lib/pgsql/9.6/data
pg_bindir=/usr/pgsql-9.6/bin
replication_user='repmgr'
replication_type=physical
use_replication_slots=yes
async_query_timeout=60
reconnect_attempts=6
reconnect_interval=10
failover=automatic
ssh_options='-q -o StrictHostKeyChecking=no'

log_level=INFO
log_file='/var/log/repmgr/repmgr.log'

monitoring_history=true

promote_command='/etc/repmgr/9.6/promote_command.sh'
follow_command='/etc/repmgr/9.6/follow_command.sh'

service_start_command = sudo systemctl start postgresql-9.6
service_stop_command = sudo systemctl stop postgresql-9.6
service_restart_command = sudo systemctl restart postgresql-9.6
service_reload_command = sudo systemctl reload postgresql-9.6

# event_notifications = {{ event_notifications }}
# event_notification_command = {{ event_notification_command }}
event_notification_command='/var/lib/pgsql/scripts/events.sh $n $e $s "%t" "%d"'
```

Figura 26: Master: Fichero de configuración /etc/repmgr/9.6/repmgr.conf en el master

## Creación y edición del fichero de configuración /etc/repmgr/9.6/promote.sh

```
#!/bin/sh
```

```
/usr/pgsql-9.6/bin/repmgr standby promote -f "/etc/repmgr/9.6/repmgr.conf"
```

## Edición del fichero de configuración /etc/repmgr/9.6/follow\_command.sh

```
#!/bin/sh
```

```
/usr/pgsql-9.6/bin/repmgr standby follow -f "/etc/repmgr/9.6/repmgr.conf"
```

## 2. Servidor Slave

### A) Instalación de Replication Manager – repmgr

```
[root@SRV-BD2 ~]# yum install repmgr96 repmgr96-debuginfo
```

```
[root@SRV-BD2 ~]# su - postgres
```

**Creamos el usuario para replica:**

```
postgres=# create user repmgr replication ;
```

**Asignamos contraseña al usuario:**

```
postgres=# \password repmgr
```

### Edición del fichero de configuración /etc/repmgr/9.6/repmgr.conf

```
node_id=2
node_name=srv-bd2
conninfo='host=192.168.1.100 user=postgres dbname=sisgedo2practica
application_name=repmgr'
data_directory=/var/lib/pgsql/9.6/data
pg_bindir=/usr/pgsql-9.6/bin
replication_user='repmgr'
replication_type=physical
use_replication_slots=yes
async_query_timeout=60
reconnect_attempts=6
reconnect_interval=10
failover=automatic
ssh_options='-q -o StrictHostKeyChecking=no'

log_level=INFO
log_file=/var/log/repmgr/repmgr.log

monitoring_history=true

promote_command='/etc/repmgr/9.6/promote_command.sh'
follow_command='/etc/repmgr/9.6/follow_command.sh'

service_start_command = sudo systemctl start postgresql-9.6
service_stop_command = sudo systemctl stop postgresql-9.6
service_restart_command = sudo systemctl restart postgresql-9.6
service_reload_command = sudo systemctl reload postgresql-9.6
event_notification_command='/var/lib/pgsql/scripts/events.sh %n %e %s "%t"
"%d"'
```

```
GNU nano 2.3.1                                root@SRV-BD2:~
Fichero: /etc/repmgr/9.6/repmgr.conf

node_id=2
node_name=srv-bd2
conninfo='host=192.168.1.100 user=postgres dbname=sisgedo2practica application_name=repmgr'
data_directory=/var/lib/pgsql/9.6/data
pg_bindir=/usr/pgsql-9.6/bin
replication_user='repmgr'
replication_type=physical
use_replication_slots=yes
async_query_timeout=60
reconnect_attempts=6
reconnect_interval=10
failover=automatic
ssh_options='-q -o StrictHostKeyChecking=no'

log_level=INFO
log_file=/var/log/repmgr/repmgr.log

monitoring_history=true

promote_command='/etc/repmgr/9.6/promote_command.sh'
follow_command='/etc/repmgr/9.6/follow_command.sh'

service_start_command = sudo systemctl start postgresql-9.6
service_stop_command = sudo systemctl stop postgresql-9.6
service_restart_command = sudo systemctl restart postgresql-9.6
service_reload_command = sudo systemctl reload postgresql-9.6

# event_notifications = {{ event_notifications }}
# event_notification_command = {{ event_notification_command }}
event_notification_command='/var/lib/pgsql/scripts/events.sh $n $e $s "$t" "$d"'


```

Figura 27: Fichero de configuración /etc/repmgr/9.6/repmgr.conf en el slave

### Edición del fichero de configuración /etc/repmgr/9.6/promote.sh

```
#!/bin/sh
```

```
/usr/pgsql-9.6/bin/repmgr standby promote -f "/etc/repmgr/9.6/repmgr.conf"
```

### Edición del fichero de configuración /etc/repmgr/9.6/follow\_command.sh

```
#!/bin/sh
```

```
/usr/pgsql-9.6/bin/repmgr standby follow -f "/etc/repmgr/9.6/repmgr.conf"
```

### 3. Servidor de Aplicaciones

#### Edición del fichero de configuración /etc/pgbouncer/pgbouncer.ini

```
:: database name = connect string
::
::
:: connect string params:
:: dbname= host= port= user= password=
:: client_encoding= datestyle= timezone=
:: pool_size= connect_query=
:: auth_user=
[databases]

; foodb over Unix socket
;foodb =

; redirect bardb to bazdb on localhost
;sisgedo = host=192.168.1.110 dbname=sisgedo2practica

; access to dest database will go with single user
;forcedb = host=127.0.0.1 port=300 user=baz password=foo client_encoding=UNICODE
datestyle=ISO connect_query='SELECT 1'

; use custom pool sizes
;nondefaultdb = pool_size=50 reserve_pool=10

; use auth_user with auth_query if user not present in auth_file
; auth_user must exist in auth_file
; foodb = auth_user=bar

; fallback connect string
;* = host=testserver

:: Configuration section
[pgbouncer]

...
::
:: Administrative settings
...
::

logfile = /var/log/pgbouncer/pgbouncer.log
pidfile = /var/run/pgbouncer/pgbouncer.pid

...
::
:: Where to wait for clients
...
::

; IP address or * which means all IPs
listen_addr = *
listen_port = 6432

; Unix socket is also used for -R.
; On Debian it should be /var/run/postgresql
;unix_socket_dir = /tmp
;unix_socket_mode = 0777
;unix_socket_group =

...
::
:: TLS settings for accepting clients
...
::
```



```

;; disable, allow, require, verify-ca, verify-full
;client_tls_sslmode = disable

;; Path to file that contains trusted CA certs
;client_tls_ca_file = <system default>

;; Private key and cert to present to clients.
;; Required for accepting TLS connections from clients.
;client_tls_key_file =
;client_tls_cert_file =

;; fast, normal, secure, legacy, <ciphersuite string>
;client_tls_ciphers = fast

;; all, secure, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
;client_tls_protocols = all

;; none, auto, legacy
;client_tls_dheparams = auto

;; none, auto, <curve name>
;client_tls_ecdhcurve = auto

...
'''
''' TLS settings for connecting to backend databases
'''
'''

;; disable, allow, require, verify-ca, verify-full
;server_tls_sslmode = disable

;; Path to that contains trusted CA certs
;server_tls_ca_file = <system default>

;; Private key and cert to present to backend.
;; Needed only if backend server require client cert.
;server_tls_key_file =
;server_tls_cert_file =

;; all, secure, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
;server_tls_protocols = all

;; fast, normal, secure, legacy, <ciphersuite string>
;server_tls_ciphers = fast

...
'''
''' Authentication settings
'''
'''

; any, trust, plain, crypt, md5, cert, hba, pam
auth_type = md5
auth_file = /etc/pgbouncer/userlist.txt

;; Path to HBA-style auth config
;auth_hba_file =

;; Query to use to fetch password from database. Result
;; must have 2 columns - username and password hash.
;auth_query = SELECT username, passwd FROM pg_shadow WHERE username=$1

```

```

...
... Users allowed into database 'pgbouncer'
...
...

; comma-separated list of users, who are allowed to change settings
admin_users = admin

; comma-separated list of users who are just allowed to use SHOW command
stats_users = admin

...
... Pooler personality questions
...
...

; When server connection is released back to pool:
; session    - after client disconnects
; transaction - after transaction finishes
; statement  - after statement finishes
pool_mode = transaction

;
; Query for cleaning connection immediately after releasing from client.
; No need to put ROLLBACK here, pgbouncer does not reuse connections
; where transaction is left open.
;
;
; Query for 8.3+:
; DISCARD ALL;
;
;
; Older versions:
; RESET ALL; SET SESSION AUTHORIZATION DEFAULT
;
;
; Empty if transaction pooling is in use.
;
server_reset_query = DISCARD ALL

; Whether server_reset_query should run in all pooling modes.
; If it is off, server_reset_query is used only for session-pooling.
;server_reset_query_always = 0

;
; Comma-separated list of parameters to ignore when given
; in startup packet. Newer JDBC versions require the
; extra_float_digits here.
;
;ignore_startup_parameters = extra_float_digits

;
; When taking idle server into use, this query is ran first.
; SELECT 1
;
;server_check_query = select 1

; If server was used more recently that this many seconds ago,
; skip the check query. Value 0 may or may not run in immediately.
;server_check_delay = 30

; Close servers in session pooling mode after a RECONNECT, RELOAD,
; etc. when they are idle instead of at the end of the session.
;server_fast_close = 0

```

```

;; Use <appname - host> as application_name on server.
;application_name_add_host = 0

...
;;
;; Connection limits
...

; total number of clients that can connect
max_client_conn = 100

; default pool size. 20 is good number when transaction pooling
; is in use, in session pooling it needs to be the number of
; max clients you want to handle at any moment
default_pool_size = 20

;; Minimum number of server connections to keep in pool.
;min_pool_size = 0

; how many additional connection to allow in case of trouble
;reserve_pool_size = 5

; if a clients needs to wait more than this many seconds, use reserve pool
;reserve_pool_timeout = 5

; how many total connections to a single database to allow from all pools
;max_db_connections = 0
;max_user_connections = 0

; If off, then server connections are reused in LIFO manner
;server_round_robin = 0

...
;;
;; Logging
...

;; Syslog settings
;syslog = 0
;syslog_facility = daemon
;syslog_ident = pgbouncer

; log if client connects or server connection is made
;log_connections = 1

; log if and why connection was closed
;log_disconnections = 1

; log error messages pooler sends to clients
;log_pooler_errors = 1

;; Period for writing aggregated stats into log.
;stats_period = 60

;; Logging verbosity. Same as -v switch on command line.
;verbose = 0

...
;;
;; Timeouts
...

```

```

;; Close server connection if its been connected longer.
;server_lifetime = 3600

;; Close server connection if its not been used in this time.
;; Allows to clean unnecessary connections from pool after peak.
;server_idle_timeout = 600

;; Cancel connection attempt if server does not answer takes longer.
;server_connect_timeout = 15

;; If server login failed (server_connect_timeout or auth failure)
;; then wait this many second.
;server_login_retry = 15

;; Dangerous. Server connection is closed if query does not return
;; in this time. Should be used to survive network problems,
;; _not_ as statement_timeout. (default: 0)
;query_timeout = 0

;; Dangerous. Client connection is closed if the query is not assigned
;; to a server in this time. Should be used to limit the number of queued
;; queries in case of a database or network failure. (default: 120)
;query_wait_timeout = 120

;; Dangerous. Client connection is closed if no activity in this time.
;; Should be used to survive network problems. (default: 0)
;client_idle_timeout = 0

;; Disconnect clients who have not managed to log in after connecting
;; in this many seconds.
;client_login_timeout = 60

;; Clean automatically created database entries (via "") if they
;; stay unused in this many seconds.
;autodb_idle_timeout = 3600

;; How long SUSPEND/-R waits for buffer flush before closing connection.
;suspend_timeout = 10

;; Close connections which are in "IDLE in transaction" state longer than
;; this many seconds.
;idle_transaction_timeout = 0

...
;; Low-level tuning options
...

;; buffer for streaming packets
;pkt_buf = 4096

;; man 2 listen
;listen_backlog = 128

;; Max number pkt_buf to process in one event loop.
;sbuf_loopcnt = 5

;; Maximum PostgreSQL protocol packet size.
;max_packet_size = 2147483647

;; networking options, for info: man 7 tcp

```

```

;; Linux: notify program about new connection only if there
;; is also data received. (Seconds to wait.)
;; On Linux the default is 45, on other OS'es 0.
;tcp_defer_accept = 0

;; In-kernel buffer size (Linux default: 4096)
;tcp_socket_buffer = 0

;; whether tcp keepalive should be turned on (0/1)
;tcp_keepalive = 1

;; The following options are Linux-specific.
;; They also require tcp_keepalive=1.

;; count of keepalive packets
;tcp_keepcnt = 0

;; how long the connection can be idle,
;; before sending keepalive packets
;tcp_keepidle = 0

;; The time between individual keepalive probes.
;tcp_keepintvl = 0

;; DNS lookup caching time
;dns_max_ttl = 15

;; DNS zone SOA lookup period
;dns_zone_check_period = 0

;; DNS negative result caching time
;dns_nxdomain_ttl = 15

...
...
... Random stuff
...
...

;; Hackish security feature. Helps against SQL-injection - when PQexec is disabled,
;; multi-statement cannot be made.
;disable_pqexec = 0

;; Config file to use for next RELOAD/SIGHUP.
;; By default contains config file from command line.
;conffile

;; Win32 service name to register as. job_name is alias for service_name,
;; used by some Skytools scripts.
;service_name = pgbouncer
;job_name = pgbouncer

;; Read additional config from the /etc/pgbouncer/pgbouncer-other.ini file
%include /etc/pgbouncer/pgbouncer-sisgedo.ini

```

## Edición del fichero de configuración /etc/pgbouncer/pgbouncer-sisgedo.ini

[databases]

```
sisgedo = host=192.168.1.110 user=postgres dbname=sisgedo2practica  
application_name=repmgr
```

## Configuración de la cadena de conexión

```
<?  
//Datos de conexión hacia la Base de Datos  
$dbhost = '192.168.1.120:6432'; // host  
$dbUsuario = 'postgres';        // usuario de conexión a la BD  
$dbpassword = 'postgres';       // password de conexión a la BD  
$dbName='sisgedo'; // Nombre de la Base Datos  
$dbtype = "postgres"; // Tipo de Bd  
?>
```

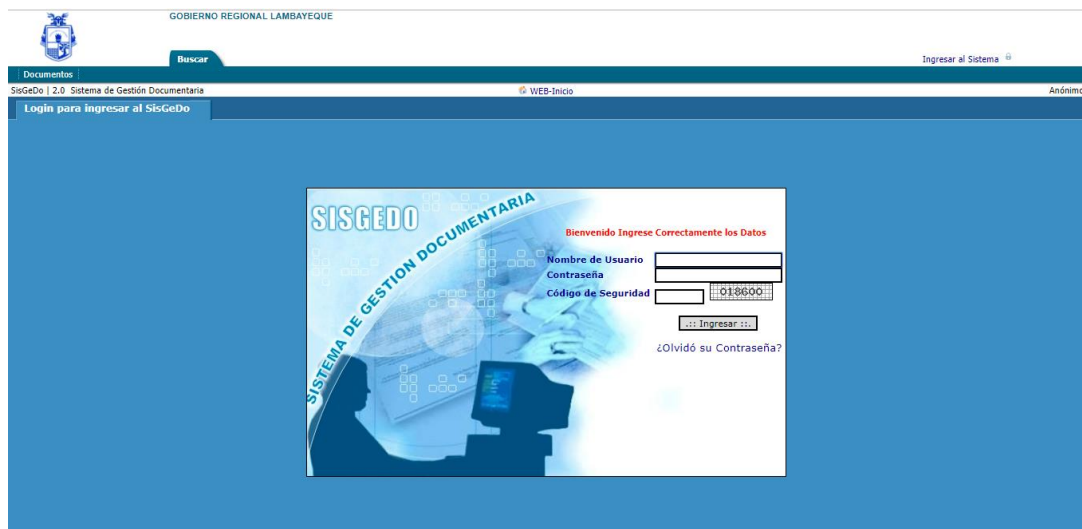


Figura 28: Verificación del Servicio Web Sistema de Gestión Documentaria

## ENCUESTA DIRIGIDA A LOS TRABAJADORES PÚBLICOS DEL GOBIERNO REGIONAL LAMBAYEQUE

**OBJETIVO:** Medir el nivel de disponibilidad de los servicios informáticos que hacen uso del motor de base de datos que posee el Gobierno Regional Lambayeque

### DATOS GENERALES:

Encuestado:	
Cargo:	
Fecha:	
Encuestador:	Julio César Palacios Ormeño

En la Universidad Nacional Pedro Ruiz Gallo, se está realizando una investigación dirigida a realizar la propuesta de ARQUITECTURA DE ALTA DISPONIBILIDAD PARA LA OPTIMIZACIÓN DEL SERVICIO DE BASE DE DATOS DEL GOBIERNO REGIONAL LAMBAYEQUE, por tal motivo se requiere de su tiempo, para corroborar que la propuesta de esta investigación genera los resultados establecidos en la hipótesis. Su información será estrictamente confidencial. Se agradece por el tiempo invertido.

### Indicaciones:

1. En la hoja de encuesta se presenta una escala del 1 al 3 en orden ascendente, marque con una "X" conforme considere su conocimiento sobre el tema de la investigación.

ESCALA DE MEDICIÓN		
1	2	3
BAJO	MEDIO	ALTO

2. A continuación se presentará la hoja de encuesta, sírvase marcar con una “X” las opción que considera de acuerdo a su percepción del tema de acuerdo al enunciado propuesto.

### I. Hoja de Encuesta

OPTIMIZACIÓN DEL SERVICIO DE BASE DE DATOS				
N°	PREGUNTA	1	2	3
Considera Usted que:				
1	¿La cantidad de interrupciones de los sistemas informáticos es?			
2	¿La frecuencia de las interrupciones de los sistemas informáticos es?			
3	¿El tiempo de inactividad (paralización) de los servicios informáticos es?			
4	¿El riesgo de pérdida de información es?			