



Módulo de replicación de datos para la Arquitectura Xalix

Data replication module for the Xalix architecture

Reiman Alfonso Azcuy¹

Leduan B. Rusell²

Yosleidy Arteaga Gómez³

¹Universidad de las Ciencias Informáticas. razcuy@uci.cu

²Universidad de las Ciencias Informáticas. leduanb@uci.cu

³Universidad de las Ciencias Informáticas. yosleidy@uci.cu

Resumen

Con el avance de las tecnologías de la información y las comunicaciones ha aumentado la demanda por parte de las empresas de las nuevas tecnologías, principalmente en lo referido al almacenamiento de información en bases de datos. Varias de estas empresas funcionan de forma geográficamente distribuida como sucursales, por lo que requieren que la información esté actualizada en cada una de ellas, teniendo en cuenta los cambios que ocurren en otras, es por ello, que se hace necesario la utilización de herramientas de réplica de datos. Ese es el caso de las plataformas desarrolladas en el Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas. En FORTES, para el trabajo con réplicas, se utiliza la herramienta SymmetricDS por su eficiencia y bajos requisitos de hardware, pero esta tiene la desventaja de que las operaciones sobre ella se tornan complejas dado que deben ejecutarse mediante la escritura de comandos en la consola y la modificación manual de ficheros de texto. La herramienta SymmetricDS cuenta con una interfaz visual liberada bajo licencia privativa por lo que para poder utilizar dicha herramienta el país tendría que incidir en importantes gastos económicos. En función de lo descrito se plantea una propuesta de solución que permite facilitar el trabajo con el software mencionado, sin que el país tenga que incurrir en dichos gastos. La propuesta consiste en un módulo que permite la replicación con SymmetricDS desde la Arquitectura Xalix utilizada para las plataformas desarrolladas por FORTES.

Palabras clave: SymmetricDS, Xalix, bases de datos, FORTES.



Abstract

With the advancement of information and communication technologies, the demand from companies for new technologies has increased, mainly in relation to the storage of information in databases. Several of these companies work in a geographically distributed way as branches, so it requires that the information be updated in each one of them, taking into account the changes that occur in others, that is why it is necessary to use tools of data replication. This is the case of the platforms developed at the FORTES Training Technology Center of the University of Computer Sciences. In FORTES, to work with replicas, the SymmetricDS tool is used because of its efficiency and low hardware requirements, but this has the disadvantage that the operations on it become complex since they must be executed by writing commands in the console and the manual modification of text files. The SymmetricDS tool has a visual interface released under a private license, so in order to use this tool, the country would have to have significant financial expenses. Based on what has been described, a solution proposal is proposed that makes it easier to work with the software mentioned above, without the country having to incur in high costs. The proposal consists of a module that allows the replication with SymmetricDS from the Xalix Architecture used for the platforms developed by FORTES.

Keywords: SymmetricDS, Xalix, databases, FORTES.

Introducción

En la última década, el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) se ha enfocado en la informatización de los distintos sectores de la sociedad, incluyendo los procesos en diferentes entidades y empresas, lo que ha conllevado al almacenamiento masivo de información en Bases de Datos (BD).

Las BD se han convertido en una fuente confiable de almacenamiento de grandes cantidades de información para cualquier sector, sirviendo de soporte para diversos procesos vinculados a la toma de decisiones en sistemas empresariales y a la consulta rápida de información. Las mismas son utilizadas en los servidores administrativos de todo el mundo en apoyo a servicios como correo electrónico y Sistemas de Nombre de Dominio (DNS según sus siglas en inglés), siendo el centro del funcionamiento en los sistemas de dichos servidores por lo que se han hecho imprescindibles (Monge, 2005). Sin embargo, con dichos avances se hace necesario que la información almacenada en una BD sea accesible desde otros centros de información (CI).

La necesidad de acceso a información que se encuentra en diferentes BD, se hace visible en diversas empresas que se distribuyen en varias sucursales (Morales, 2011) y se necesita que la información registrada en cada una de esas sucursales se actualice en el servidor central de la empresa (Morales, 2011). Un ejemplo de ello es la red de Universidades del Ministerio de Educación Superior o la empresa GeoCuba, ya que las mismas poseen 16 sucursales distribuidas a lo largo del país. También ocurren los casos en los que por el volumen de la BD y las propiedades del hardware con que cuenta el servidor de BD el mismo no puede soportar las operaciones de lectura y escritura. Debido a esto es necesario separar la BD en dos o más servidores para lectura y escritura, manteniéndose, la necesidad de la actualización de los datos en diferentes servidores mencionada anteriormente.



Generalmente se requiere tener una BD centralizada, y hacer copias de la información desde ella hacia otras BD secundarias. La forma adecuada para solucionar esta situación es lo que se denomina replicación de datos o réplicas, lo cual permitiría el envío de información entre Bases de Datos distribuidas geográficamente a través de la red, haciendo uso de algún mecanismo o herramienta de replicación.

En Cuba existen varias entidades que utilizan replicación para garantizar la accesibilidad de la información, ejemplo de ello son las aplicaciones que se desarrollan en varios centros de producción de la Universidad de las Ciencias Informáticas (UCI).

En el centro de tecnologías para la formación FORTES de la UCI, se desarrollan plataformas educativas bajo la arquitectura Xalix. Dentro de estas plataformas se encuentran Zera y el Sistema de Gestión de Ingreso a la Educación Superior (SIGIES). Dichas plataformas deben funcionar de forma distribuida, manteniendo actualizada la información en un nodo central según las modificaciones realizadas en los diferentes servidores, por lo que se requiere la utilización de réplicas. Sin embargo, para cumplir con las políticas de migración a software libre que se están llevando a cabo en el país, se decide utilizar por parte del equipo de desarrollo la herramienta libre SymmetricDS.

La herramienta SymmetricDS es efectiva para la configuración y monitorización de réplicas (C. H. Eric Long, 2016) sin embargo, configurarla es un proceso bastante engorroso. Debido a que se deben modificar, de forma manual, una serie de ficheros correspondientes a cada una de las BDs entre las cuales se desea establecer un sistema de réplicas. Además de que todas las instrucciones le deben ser escritas por medio de la consola. Existe una herramienta desarrollada por la compañía Oracle llamada SymmetricDS-Pro (C. H. Eric Long, 2016) la cual funciona como interfaz gráfica para la herramienta SymmetricDS, permitiendo realizar de forma simple la planificación, configuración, administración y monitorización de réplicas (C. H. Eric Long, 2016).

Esta herramienta, tiene la agravante de que es un producto liberado bajo licencia privativa con un costo de \$3850. Dicha agravante implica que, si es instalado sin pagar la licencia correspondiente, la utilización de la misma no puede ser superior a quince días. Esto significa que para poder utilizar este producto el país tendría que incurrir en gastos económicos significativos para el pago de la licencia según el valor antes mencionado.



Figura 1. Interfaz de SymmetricDS-Pro herramienta privativa de SymmetricDS.

A raíz de la investigación no se encontraron otras herramientas no privativas que además fueran compatibles con las tecnologías utilizadas por la arquitectura Xalix utilizadas por el centro FORTES.

A partir de la situación descrita anteriormente, se plantea el siguiente **problema a resolver**:

¿Cómo facilitar la monitorización y configuración de réplicas con la herramienta SymmetricDS, en los productos desarrollados en FORTES sin que el país tenga que incurrir en gastos económicos por el pago de una licencia?

Contenido

Replicación de datos

Para poder implementar un módulo que permita configurar réplicas primeramente se debe tener conocimiento de qué es una réplica y lo que representa. Se debe tener claro que las mismas constituyen una forma más rápida y confiable de esparcir datos entre distintas bases de datos en un sistema que trabaja de forma distribuida. A continuación, se enuncian algunas definiciones.

“La replicación copia y mantiene los objetos de las BD en las múltiples bases de datos que levantan un sistema distribuido. Puede mejorar el funcionamiento y proteger la disponibilidad de las aplicaciones porque alterna opciones de acceso a los datos existentes (Monge, 2005).”

“La replicación es un conjunto de tecnologías destinadas a la copia y distribución de datos desde una base de datos hacia otra o múltiples bases de datos, para luego sincronizarlas y mantener su coherencia. Además, permite distribuir datos entre diferentes ubicaciones y entre usuarios remotos o móviles mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet” (Morales, 2011).

“La replicación de datos consiste en el transporte de datos entre dos o más servidores, permitiendo que ciertos datos de las BD estén almacenados en más de un sitio y así aumentar la disponibilidad de los datos y mejorar el rendimiento de las consultas globales” (C. H. Eric Long, 2016).

“El sistema conserva varias copias o réplicas idénticas de una tabla. Cada réplica se almacena en un nodo diferente” (Community, S. SymmetricDS. 2013).

A partir de los conceptos anteriores se puede concluir que la replicación es un recurso que permite intercambiar información entre bases de datos geográficamente distribuidas, de forma segura. Una réplica, se puede definir como una acción del concepto de la replicación.

Ventajas de la replicación de datos

Dentro de las ventajas de la utilización de réplicas se puede definir (Morales, 2011):

Efectividad: depende de la forma en la que los datos son distribuidos y almacenados. A mayor efectividad, mayor será la disponibilidad de datos para ejecutar procesos paralelos.



Alta Disponibilidad: es la razón de tiempo prudente en la que un servicio puede ser accedido. En el mejor de los casos puede ser de un 100%, a pesar de los fallos que se puedan presentar en el servidor, ya que debe existir un servidor adicional que posea alguna técnica de replicación que lo pueda suplantar en caso de ser necesario.

Tolerancia a fallos: garantiza un comportamiento correcto, donde efectivamente pueden existir un número finito de fallos y tipos de fallos.

Coordinación: cada una de las partes que conforman la BD distribuida acuerda un consenso para realizar las invocaciones de los servicios a los objetos, que al final de la transacción debe realizarse tal y como fue solicitada, para lo cual debe utilizar algún tipo de ordenamiento.

Herramienta SymmetricDS

Una vez que se tiene dominio sobre lo que representa la replicación se puede pasar al análisis de la herramienta SymmetricDS.

SymmetricDS es un software de replicación de datos asíncrona que permite subscriptores múltiples y sincronización bidireccional. Utiliza tecnologías web y de BD para replicar tablas entre BD relacionales, casi en tiempo real. El software fue diseñado para escalar a un gran número de BD, trabajar con conexiones de bajo ancho de banda y resistir a periodos de inoperatividad de la red (Morales, 2011).

Una única instalación de SymmetricDS se denomina un Nodo. Un Nodo es inicializado mediante un fichero *properties* y es configurado insertando datos de configuración en una serie de tablas de BD (C. H. Eric Long, 2016). A continuación, el Nodo crea disparadores de BD en las tablas de aplicación especificadas, de modo que los eventos de BD son capturados para ser entregados a otros Nodos SymmetricDS.

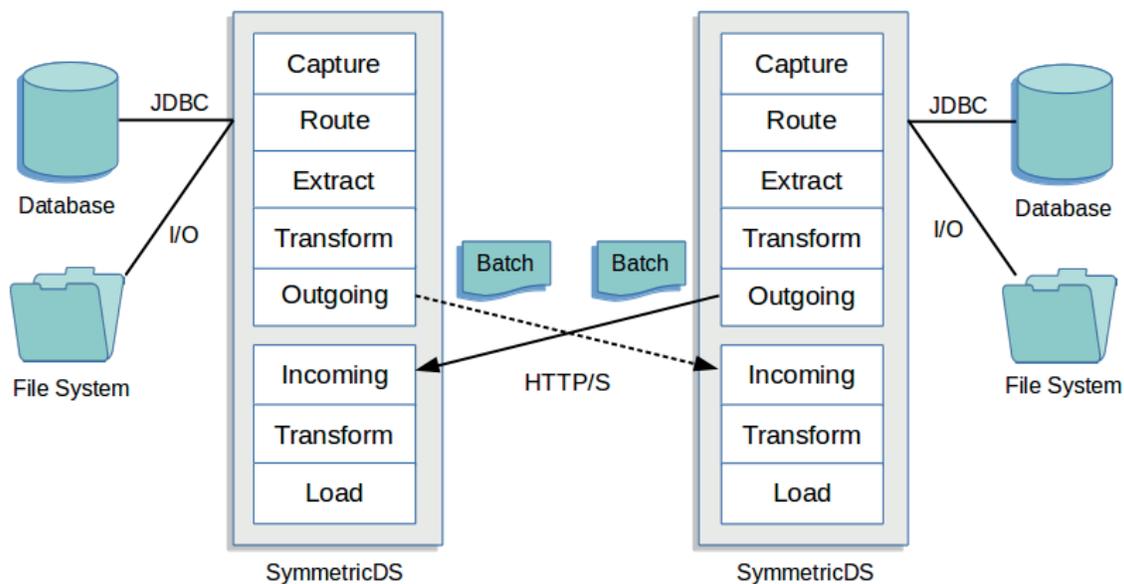


Figura 2. Flujo de trabajo de SymmetricDS.

SymmetricDS está escrito en Java 5 (Community, S. SymmetricDS. 2013). y requiere Java SE Runtime Environment (JRE) o Java SE Development Kit (JDK) versión 5.0 o superior. Soporta la sincronización entre diferentes plataformas de BD, mediante el concepto de dialectos de base de datos. Un dialecto de BD es una capa de abstracción con la cual interactúa SymmetricDS para aislar la lógica de sincronización de los detalles de implementación específicos de cada BD (Community, S. SymmetricDS. 2013). Las principales características por lo cual se eligió SymmetricDS sobre las demás herramientas son (Morales, 2011), (C. H. Eric Long, 2016), (Community, S. SymmetricDS. 2013):

- » Licencia de código abierto GPL, multiplataforma.
- » Pensado para trabajar con conexiones de bajo ancho de banda.
- » Pensado para tener nodos desconectados por largos periodos de tiempo.
- » Es capaz de replicar los cambios estructurales de la BD que puedan surgir.
- » Flexibilidad a la hora de declarar las reglas de replicación de los datos.
- » Facilidad de configuración.
- » Amplia documentación, comunidad, soporte y ejemplos.
- » Alto rendimiento en ambientes con bajo ancho de banda y problemas de conexión.

A continuación, se describen los pasos a seguir para configurar una réplica con la herramienta SymmetricDS (C. H. Eric Long, 2016):

1. Configurar un fichero de extensión .properties para cada nodo físico incluido en sistema de réplica.
2. Ejecutar el comando para generar las tablas en cada uno de los nodos.
3. Ejecutar el comando para iniciar el servicio en cada uno de los nodos.
4. Insertar mediante una consulta SQL en el servidor maestro los canales a utilizarse.
5. Insertar mediante una consulta SQL en el servidor maestro los Grupos de Nodos.
6. Insertar mediante una consulta SQL en el servidor maestro los Nodos.
7. Insertar mediante una consulta SQL en el servidor maestro los Disparadores por cada tabla.
8. Insertar mediante una consulta SQL en el servidor maestro los Enrutadores por cada tabla.
9. Relacionar los Enrutadores con los disparadores.
10. Insertar la identidad de los nodos.



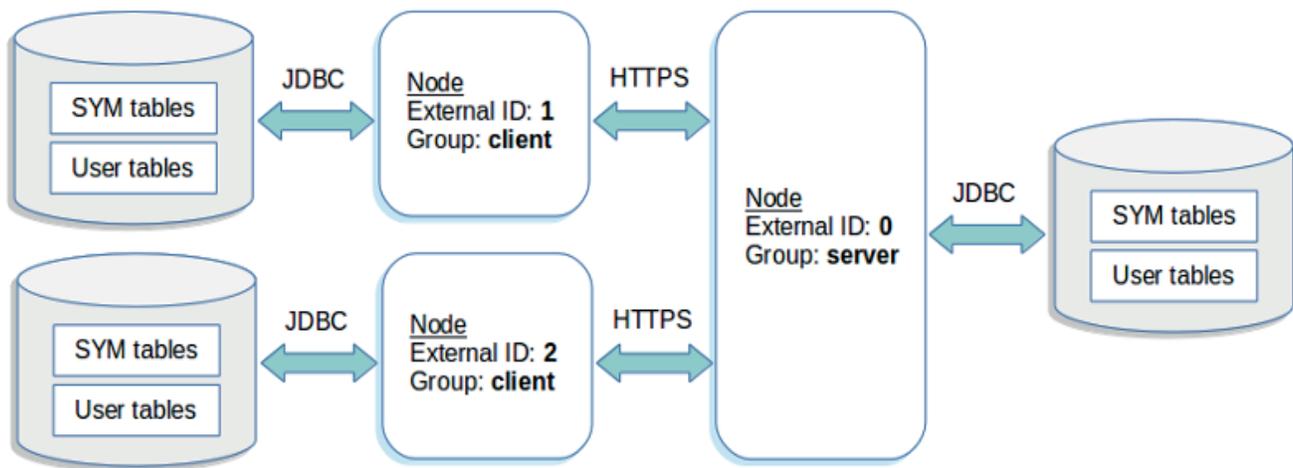


Figura 3. Arquitectura de SymmetricDS.

Arquitectura Xalix

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema (C. H. Eric Long, 2016). Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software (C. H. Eric Long, 2016) y (Bourque, P., & Dupuis, R. 2004).

Una arquitectura de software se selecciona y diseña con base en objetivos (requerimientos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información (C. H. Eric Long, 2016).

Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos (Jacobson, I., Booch, G., & Rumbaugh, J., 1999). Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar, un cierto número de elementos arquitectónicos de forma adecuada, para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad (Lawrence-Pfleeger, & Shari., 1998).

El centro FORTES de la UCI desarrolla una serie de proyectos enmarcados en la línea Xauce dedicada a la educación, dentro de ellos se encuentran la plataforma educativa Zera, y el Sistema de Gestión de Ingreso a la Educación Superior, a los cuales se hacía referencia en la introducción del documento. Estos proyectos, o estas plataformas deben ser desplegados en todas las provincias del país o al menos en la mayoría. Esto implica que se requiera replicación para la actualización de los mismos dado que tendrán un servidor o nodo central y varios hijos que dependan de él. Para con la utilización de algún mecanismo de

necesarias, y que recoja los comandos que brinda la herramienta como comandos del marco de trabajo siendo reconocidos por la interfaz web para comandos que posee cada proyecto de tipo plataforma que es desarrollado por FORTES.

Se propone el desarrollo de un módulo que consta con 55 requisitos funcionales (Martin, 2012) que incluyen la instalación de SymmetricDS, así como la configuración y monitorización. Todo bajo las tecnologías por las que se rige el marco de trabajo Xalix para plataformas web utilizado en FORTES. Dicho módulo está constituido interfaces web que permiten al usuario de forma simple aplicar todas las operaciones que ofrece SymmetricDS sin tener que editar ficheros de configuración o escribir comandos en la consola.

El módulo fue implementado utilizando el framework de desarrollo Symfony en su versión 2.7.1 para PHP, que incluye a la herramienta Doctrine para el mapeo relacional de objetos con PostgreSQL como gestor de bases de datos. Todo fue montado mediante el servidor web Nginx. Para el tratamiento de la interfaz visual fueron utilizadas las tecnologías del lado cliente css, y javascript mediante los frameworks Bootstrap y jQuery. La estructura física del módulo se detalla en la figura 4.

A continuación, se listan algunos de los requisitos que más inciden en el proceso de replicación:

1. Instalar y desinstalar SymmetricDS.
2. Iniciar, detener, y consultar estado del servicio SymmetricDS.
3. Configurar físicamente los servidores de SymmetricDS de forma básica y avanzada.
4. Gestionar Canales de Comunicación.
5. Gestionar Nodos, Grupos de Nodos, Seguridad de Nodos, Identidad de Nodos Disparadores y Enrutadores.
6. Gestionar enlaces entre Grupos de Nodos y Disparadores-Enrutadores.
7. Monitorizar datos mediante las consultas a las tablas Sym_Data, Sym_Conflict, y Sym_Error.

Al módulo le fue implementado además un mecanismo de selección de tablas con el objetivo de evitar errores por parte del usuario haciendo uso de un recorrido a lo ancho sobre la BD modelada en forma de Grafo. Esto no impide que el usuario entre una configuración errónea por lo que aún continua el riesgo de que en las réplicas se produzcan conflictos (Wilson, S. F., 1999) y (Martin, 2012). Por ello se considera que el módulo es administrativo. Actualmente se trabaja en un modo de recomendar configuraciones al usuario mediante el uso de árboles de decisión.

Pruebas

Las pruebas unitarias permitieron ir comprobando el correcto funcionamiento de determinadas funciones implementadas durante el desarrollo del sistema. Para llevar a cabo estas pruebas se empleó el método de caja blanca utilizando además el framework de pruebas PHPUnit (Moreno, 2012). Estas no se



planificaron ni fueron registradas ya que fueron realizadas a medida que avanzaba el proceso de desarrollo de la aplicación (Pressman R., 2010).

Con el objetivo de verificar el cumplimiento de los requisitos funcionales establecidos para la presente investigación se hace uso de las Pruebas de Caja Negra, teniendo en cuenta la técnica de partición por equivalencia. Dicha técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software y descubrir de forma inmediata los errores presentes en el sistema (Reingart, 2016). La partición equivalente se basa en la definición de casos de pruebas que descubran diferentes tipos de errores, reduciendo así en número de clases de prueba que hay que desarrollar (Replicación de SQL Server, 2014).

Además, se hace uso de los casos de prueba generados durante este flujo de trabajo con el fin de detectar la mayor cantidad de no conformidades posibles en las funcionalidades del sistema realizándose cuatro iteraciones de prueba (Martin, 2012). Para el seguimiento de todo el proceso de corrección de no conformidades se realiza una tabla, la misma contará con la cantidad de no conformidades determinadas en cada iteración clasificadas de acuerdo a su nivel de significación alto, medio, o bajo (Bourque, P., & Dupuis, R. 2004). Se define nivel de significación alto para aquellas no conformidades que impactan directamente en el éxito de funcionalidades importantes para el trabajo del sistema; medio para aquellas que no afectan la lógica del sistema, pero si la forma en que el usuario interactúa con ellas, y bajo para aquellas referidas a problemas de diseño, redacción, ortografía o traducción.

Tabla 1. Tabla de no conformidades detectadas en las iteraciones

Significación de la no conformidad	Iteración I	Iteración II	Iteración III	Iteración IV
Baja	16	11	4	0
Media	8	6	2	0
Alta	11	8	3	0
Total	35	25	9	0

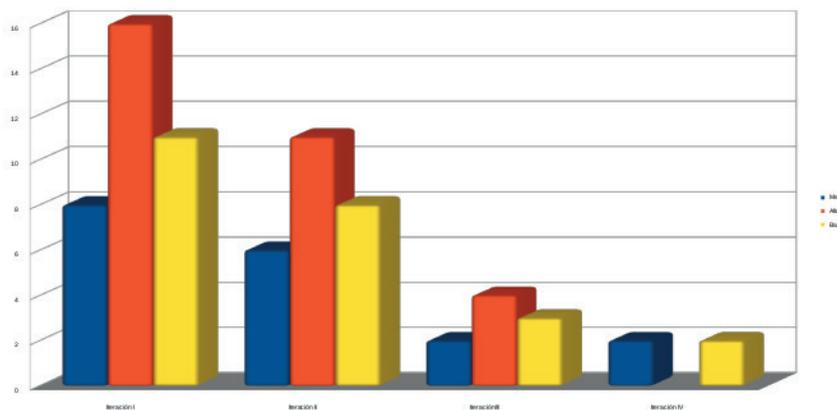


Figura 5. Comportamiento de las no conformidades durante las pruebas.

Conclusiones

Con la presente investigación se puede concluir que:

Se confeccionó la base teórica de la investigación donde fueron realizadas las siguientes acciones:

1. Análisis detallado de las ventajas de la utilización de réplicas.
2. Análisis de la herramienta para réplicas SymmetricDS.
3. Descripción del procedimiento para el trabajo con SymmetricDS.
4. Se realizó el análisis, diseño e implementación de un módulo para la monitorización y configuración de réplicas con la herramienta SymmetricDS para la arquitectura Xalix, lo cual incluye:
5. Descripción de la propuesta de solución resumiendo lo esencial de los requisitos funcionales, así como los pasos para configurar una réplica haciendo uso de la misma.
6. Se obtiene un módulo de código abierto que facilita el trabajo con la herramienta SymmetricDS que abre una nueva posibilidad para el país y evita que el mismo tenga que incurrir en gastos económicos.
7. Con la presente solución se obtiene un sistema que permite resumir los pasos mencionados anteriormente al trabajo con una interfaz web sin tener que utilizar la consola o directamente un sistema gestor de bases de datos para la escritura de consultas. Dichas consultas son generadas automáticamente por la aplicación.

Referencias

- Monge, R. (2005): Sistemas Distribuidos de Computación. Trabajo Investigativo “Base de Datos Distribuidas:Replicación”. Valparaiso.
- Morales, V. T. (2011): Bases de Datos Distribuidas.
- Community, S. SymmetricDS. 2013. (2013). Retrieved 10 8, 2016, from <http://www.SymmetricDS.org/>
- Eric Long, C. H. (2014): SymmetricDS User Guide. JumpMind, Inc.
- Bourque, P., & Dupuis, R. (2004): Guide to the Software Engineering Body of Knowledge.
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999): The Unified Software Development.
- Lawrence-Pfleeger, & Shari. (1998): Software Engineering: Theory and Practice.
- Wilson, S. F. (1999): Analyzing Requirements and Defining Solution Architectures. Redmond: Microsoft Press,.
- Manso Guerra, Y. (2016, 11 16): Arquitectura Xalix. (R. Alfonso Azcuy, Interviewer)



Martin, D. (2012): Aplicación para resolución de conflictos en bases de datos. Bogotá.

Martin, R. C. (2003): Agile Software Development, Principles, Patterns, and Practices.

Moreno, G. (2012): Replicación en PostgreSQL 9.0. Universidad Nacional de Salta .

Reingart, M. (2016): Sistema de replicación simple para PostgreSQL programado en Python. Retrieved 12 9, 2016, from <http://code.google.com/p/pyreplica/>.

Replicación de SQL Server. (2014). Retrieved 11 24, 2016, from <http://msdn.microsoft.com>.

