

Tipo de artículo: Artículo original

Temática: Desarrollo de aplicaciones informáticas

Recibido: 03/10/2016 | Aceptado: 20/10/2016 | Publicado: 30/10/2016

JEASYDB: Herramienta para migrar bases de datos MySQL, SQLite Y dBase A PostgreSQL

JEASYDB: Tool to migrate MySQL, SQLite and dBase databases to PostgreSQL

Félix González Martínez ^{1*}, Karel Rodríguez Carmenates ¹

¹ Centro de Tecnologías de Gestión de Datos (DATEC). Universidad de las Ciencias Informáticas. Carretera a San Antonio, km 2 ½, La Lisa, La Habana.

* Autor para correspondencia: felix@uci.cu

Resumen

En el presente trabajo se realiza un estudio de diferentes herramientas que existen para el proceso de migración de bases de datos entre los gestores MySQL, SQLite y dBase a PostgreSQL, el cual permitió detectar características y funcionalidades que atentan contra la migración exitosa de bases de datos (BD) en proyectos reales. Se identificó además que la mayoría de las herramientas son propietarias y algunas introducen errores en el proceso de migración, entre los que destacan la falta de integridad referencial y el cambio de nombres en tablas y tipos de datos. Debido a estas limitantes surge la necesidad de la creación de una herramienta para la migración de BD desde los gestores MySQL, SQLite y dBase a PostgreSQL con tecnología libre, teniendo en cuenta la necesidad del traspaso paulatino de software propietario a software libre en la Universidad de las Ciencias Informáticas (UCI). Para el desarrollo de la herramienta se utilizó el lenguaje de programación Java en su versión 1.7, aprovechando la facilidad multiplataforma que brinda, posibilitando que la aplicación pueda ser ejecutada en cualquier sistema operativo. La importancia de la herramienta desarrollada se basa en la necesidad de migrar las bases de datos en los gestores antes mencionados a PostgreSQL, teniendo en cuenta que realizar el proceso de forma manual sería engorroso debido a propiedades y funciones propias de cada gestor.

Palabras clave: bases de datos; dbase; gestores; migración; mysql; postgresql; sqlite.

Abstract

This paper is a study of various tools available to the migration process between the databases managers MySQL, SQLite and dBase III to PostgreSQL, which allowed detecting features and functionalities that threaten the successful migration of databases (DB) in real projects. It was also identified that most are proprietary tools or introduce errors in the migration process such as the lack of referential integrity and changes of names in tables and data types. Due to these limitations arises the need for the creation of a tool for migrating databases from MySQL, SQLite and dBase to PostgreSQL as free technology, having into account the need for gradual transfer from proprietary to free software in the University of Informatics Sciences (UCI). For the development of the tool, was used Java programming language in its version 1.7, taking advantage of the multiplatform ease that provides, allowing the application to be run on any operating system. The importance of it is based on the need to migrate the databases stored in the database managers MySQL, SQLite and dBase to PostgreSQL, considering that perform the process manually would be virtually impossible due to the large amount of stored data, as well as other properties and functions of each manager.

Keywords: *databases, dbase, migration, mysql, postgresql, sqlite.*

Introducción

En la actualidad existe un auge vertiginoso en el desarrollo tecnológico del campo de la informática. Una de las actividades más importantes en esta rama es el desarrollo de las bases de datos (BD). Estas facilitan el manejo de grandes volúmenes de datos sobre los cuales se aplican técnicas y procedimientos para llevar a cabo el tratamiento automático de la información, logrando un mejor desempeño en los directivos y especialistas, derivado de la toma de decisiones que inciden en procesos claves de las empresas u organizaciones.

Gran parte de los sistemas informáticos incluyen en su desarrollo bases de datos para guardar determinados indicadores, logrando que los valores perduren en el tiempo. Para la gestión de estas bases de datos se emplean los Sistemas Gestores de Bases de Datos (SGBD), cuyo propósito general es llevar a cabo el tratamiento de un conjunto de datos que posteriormente serán información importante para las organizaciones o entidades. En el mercado de software existen variadas opciones propietarias que permiten el manejo de los datos, sin embargo, existen opciones libres, que pueden ser usadas de forma gratuita por cualquier persona con conocimientos sobre programación y trabajo con base de datos.

Entre los SGBD se pueden mencionar algunos como MySQL, el cual es propietario en algunas de sus versiones y está patrocinado por una empresa privada (Corporación de Oracle) que posee el derecho de autor de la mayor parte del código. DBase, comercializado desde 1999 por dataBased Intelligence, Inc quien poseó los derechos de autor, fue el primer sistema de gestión de base de datos usado ampliamente para microcomputadoras, cuyo formato “dbf”, es ampliamente utilizado en muchas otras aplicaciones que necesitan un formato simple para almacenar datos estructurados. Además, destaca SQLite con dominio público y libre para cualquier uso, ya sea comercial o privado, el cual es actualmente utilizado en gran cantidad de aplicaciones, sin embargo, no es recomendable para aplicaciones donde varios clientes necesitan acceder y utilizar la misma base de datos, ni para cuando la aplicación crece y requiere de altos volúmenes de escritura, pues SQLite solo permite una operación de escritura en un momento dado limitando así el rendimiento. Por otro lado, se encuentra PostgreSQL; un sistema estable y de alto rendimiento que permite desarrollar aplicaciones sin necesidad de incurrir en costos adicionales de licenciamiento por estar incluido en la categoría de software libre. Existen variadas circunstancias en las que una organización o empresa hace uso de un SGBD y por cuestiones de cambios y mejoras necesitan migrar de un gestor a otro.

La migración a Software Libre en Cuba es una necesidad por lo que esta representa para los cubanos en los aspectos políticos, económicos y sociales. La Universidad de las Ciencias Informáticas (UCI) se ha insertado en el proceso de migración de aquellas aplicaciones que todavía usan software propietario. En general varias entidades en Cuba emplean como gestor de base de datos a PostgreSQL, pero a pesar de esto aún existe un grupo grande que continúa usando el gestor MySQL y es por ello que dentro de sus necesidades está el poder migrar de forma íntegra todas aquellas bases de datos almacenadas en MySQL. Hay que destacar que es inminente el traspaso hacia la tecnología libre, pero como proceso debe hacerse de forma gradual y con previa planificación.

Debido al grado de dificultad que presentan los procesos de migración entre distintos gestores de bases de datos se deberían emplear herramientas de migración. En un estudio realizado a las principales herramientas que realizan este proceso, se pudo observar como en su mayoría no llegan a migrar cada uno de los elementos necesarios tales como: tablas, datos, funciones, vistas e integridad referencial entre tablas, elementos importantes en el propio diseño. Cada gestor hace propia cada una de estas características, aun haciendo uso todos ellos del lenguaje SQL estándar. Por otro lado, muchas de estas herramientas no siguen la filosofía del software libre y para obtener resultados positivos en el proceso de migración, se debe incurrir en gastos de licencias que están atadas a precios por tiempo de consumo y versiones de la herramienta.

No emplear en el proceso de migración herramientas que sean propietarias o libres implica hacer uso de recursos humanos. Esto trae consigo que el proceso de migración se realice de forma manual y que posteriormente se incluyan

revisiones y chequeos para validar el trabajo realizado. Este método actualmente no es una opción viable debido a que las BD varían en estructura y tamaño en dependencia de la empresa o entidad, lo que incurriría en gastos por concepto de pago de mano de obra, además de que el proceso de migración puede tardar semanas y hasta meses en completarse. Debido a las dificultades mencionadas anteriormente se plantea el siguiente: Problema de Investigación: ¿Cómo migrar bases de datos desde los gestores MySQL, SQLite y dBase hacia el gestor PostgreSQL? En búsqueda de la solución se establece como objetivo de la investigación: Desarrollar una herramienta que permita la migración de bases de datos desde los gestores MySQL, SQLite y dBase hacia el gestor PostgreSQL.

El resultado esperado de este trabajo es el desarrollo de una herramienta que permita migrar bases de datos desde los gestores MySQL, SQLite y dBase hacia el gestor PostgreSQL añadiendo además nuevos elementos que mejoren la manipulación de los datos.

Materiales y métodos

La migración de una base de datos de un SGBD a otro es un proceso complejo pues cada gestor tiene sus características particulares. Al transferir la información entre los diferentes gestores de base de datos existe una complejidad en los distintos tipos de datos de las tablas en las que se esté trabajando. Por ejemplo, los campos de tipo fecha, numéricos, booleanos pueden recopilarse de diferentes formas, además de las relaciones presentes entre tablas y las referencias de llaves que esto provoca. Ante estas dificultades, muchos emprenden su investigación en la búsqueda de una solución y algunos logran encontrarlas, pero las mismas presentan características propias del problema que resuelven, además del interés personal de quienes las crean. En el mercado internacional y preferentemente potenciado por el comercio electrónico se habilitan un conjunto de herramientas para la migración de bases de datos. Estas se reúnen en dos grupos con intereses diferentes entre las que se encuentran las libres y las propietarias, pero ambos grupos con el fin de realizar la migración de BD entre gestores.

Herramientas para la migración entre SGBD

ESF Database Migration Toolkit Professional Edition (SOFTPEDIA, 2016) es un programa que permite convertir entre diferentes formatos de base de datos. Puede conectarse directamente a MySQL, SQL Server, PostgreSQL, SQLite, Oracle, Access, Excel, Paradox, Lotus, dBase, Visual FoxPro, texto y otros formatos y convertir entre estos formatos (SQL Server a MySQL, MySQL a Access, entre otros). Permite conectarse a la base de datos origen, seleccionar las tablas y a continuación, convertir y copiar al destino. También ofrece todas las opciones de conversión

necesarias teniendo en cuenta las peculiaridades de los dos formatos de entrada y salida de la base de datos, además de incluir compatibilidad para las claves primarias de las tablas, índices e identificación automática. Es compatible con todos los caracteres Unicode (UTF-8, LATIN, CP1250, ASCII, entre otros).

Esta herramienta al migrar la estructura de la base de datos no mantiene los campos que se encuentran con letras mayúsculas, cambiándolos a minúscula. Debido a su carácter propietario, para aquellos usuarios que no posean licencias, la herramienta migra los campos de tipo *cadena* con una T en la primera letra.

DBConvert (DBCONVERT, 2016): es un conjunto de herramienta de migración de base de datos que permite convertir bases de datos entre diferentes gestores. Opera con una base de datos completa o solo con las tablas necesarias, campos, índices y llaves foráneas. Esta herramienta no está basada en software libre, por lo que en su versión de evaluación solamente permite migrar 50 registros por tabla.

Mysql2pgsql.perl (PostgreSQL, 2016): es un script basado en el lenguaje Perl usado para convertir de un dump MySQL a un formato compatible de PostgreSQL. Tiene la desventaja que no migra la estructura de un atributo con comentarios, además de que los atributos no deben tener caracteres extraños, como por ejemplo la ñ. Tras realizar algunas pruebas con la herramienta se detectaron algunas dificultades en el proceso de migración, las mismas son:

- Convierte el IF a “if” como si fuera el nombre de una tabla.
- Elimina las comillas “” de los nombres de las tablas, lo que provoca que los nombres de los campos se migren en minúsculas.
- En las sentencias INSERT al nombre de la tabla le pone una comilla doble y una simple.

My2pg.perl (MY2PG.PERL, 2016): es un script escrito en Perl, con licencia gratis enfocado a la conversión de formato MySQL en PostgreSQL. La misma al generar el archivo .sql presenta errores en la migración como son:

- Coloca los comentarios a continuación del campo, lo cual no es una sintaxis válida en PostgreSQL.
- Los nombres de las tablas y los campos los genera con comillas simples, lo que no es permisible en PostgreSQL.
- En la declaración que hace del CONSTRAINTS la palabra que precede al CONSTRAINTS debe estar sin comillas.
- En las sentencias ADD CONSTRAINT agrega ON DELETE CASCADE ON UPDATE CASCADE, las cuales deben ir en los campos que lo lleven.

MySQL-to-PostgreSQL 1.0 (MYSQL-TO-POSTGRESQL 1.0, 2016): es una herramienta para la migración entre los gestores de MySQL a PostgreSQL. Dentro de sus principales características se pueden mencionar que soporta todas

las versiones de PostgreSQL, rapidez en la copia de datos, convierte los índices de todos los atributos, soporta múltiples conjuntos de caracteres, entre otras. Sin embargo, presenta algunas limitaciones como son:

- No migra las llaves foráneas.
- No migra las vistas, procedimientos almacenados y disparadores.
- La versión de prueba solo migra 5 filas por tabla.

A todas ellas se suma el hecho de que es una herramienta que necesita ser pagada para poder hacer uso de todas sus funcionalidades.

Navicat Premium 9.0 (Navicat, 2016): es una herramienta que une todos los miembros de la familia Navicat para administración de base de datos y desarrolladores. Permite que los usuarios se conecten a cuatro de las bases de datos más comunes: MySQL, Oracle, PostgreSQL y SQLite desde una única aplicación. Entre una de sus funcionalidades se encuentra la transferencia de datos entre los diferentes gestores a los que se conecta, permitiendo migrar bases de datos entre ellos. Sin embargo, presenta algunas limitaciones entre las que destacan:

- No migra las llaves foráneas.
- No migra los índices.
- Es una herramienta propietaria.

Después de realizar un estudio de las diferentes herramientas que existen para la migración de base de datos entre diferentes gestores de bases de datos se decide no utilizar ninguna de estas herramientas en la investigación; por un lado, algunas son herramientas propietarias y otras presentan problemas con algunas funciones y sentencias que no son reconocidas en PostgreSQL. De manera general ninguna de estas realiza un proceso íntegro de migración, demostrado anteriormente. Agregar además que ninguna de las herramientas migra las vistas que puedan estar presentes en las bases de datos.

A raíz de esto se decide implementar una herramienta que reúna todos los pasos de la migración y así poder lograr un proceso más completo, incluyendo la migración de las vistas. Sin embargo, se toman algunos ejemplos de buenas prácticas en el desarrollo de software como elementos de interfaz centrado en el diseño de interfaces gráficas de usuario.

Pasos para la migración de bases de datos a PostgreSQL

Para llevar a cabo la migración de una base de datos de un SGBD a otro se debe tener en cuenta diferentes aspectos y características de cada uno de los gestores. Por ejemplo, los diferentes tipos de datos que se almacenen en cada uno de estos, así como las diferentes funciones que existen debido a que cada SGBD las implementa diferente.

El proceso de migración de datos es bastante complejo en cuanto a las particularidades entre los diferentes tipos de datos, tablas involucradas y sus relaciones, entre otros aspectos. A continuación, se enuncian los elementos fundamentales o pasos a tener en cuenta en el proceso de migración (ver figura 1). Siguiendo cada uno de estos pasos se destaca el hecho que es posible migrar si se desea, solo algunos elementos y no todos. Esto es debido a que se puede seleccionar hasta donde migrar comenzando con la estructura de la BD y finalizando en las Vistas. Siempre teniendo en cuenta que para pasar a un nivel se debe completar exitosamente los anteriores. Por lo que se define como solución:

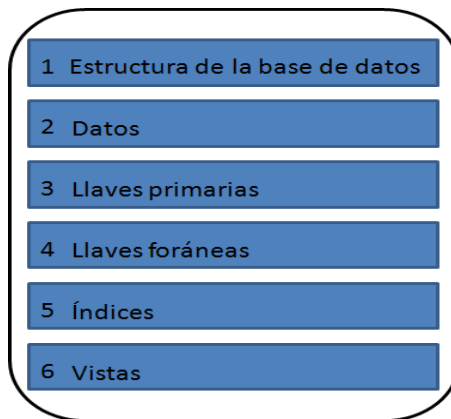


Figura 1. Elementos fundamentales para la migración de una base de datos.

- Migrar la estructura completa de la base de datos sin tener en cuenta los datos y la integridad referencial. En este paso se procesan todas las tablas presentes en el diseño de la base de datos. Para cada una de estas se ha de realizar el mapeo de los tipos de datos presentes en cada uno de los gestores. En este sentido se deben reemplazar los tipos de datos de MySQL, SQLite y dBase con su correspondiente en PostgreSQL. En la tabla 1 se muestra un ejemplo de la correspondencia entre algunos tipos de datos de MySQL y PostgreSQL mencionados (AJDB SOFT, 2016).

Tabla 1. Correspondencias entre algunos tipos de datos de MySQL y PostgreSQL.

MySQL	PostgreSQL
int(10)	integer
tinyint	smallint
mediumint	integer
bigint unsigned	numeric(20)
double	double precision
float	real

- Al mismo tiempo se ha de verificar que los nombres de las tablas y de los campos se encuentren entre comillas dobles, de esta forma PostgreSQL mantiene los nombres en su formato original. De no realizar este paso PostgreSQL almacena todos los nombres en minúscula.
- Realizado el paso anterior se procede a migrar los datos de cada una de las tablas de la base de datos. Para esto se hace una consulta sobre cada tabla donde se obtienen todos los datos de la misma. Vale destacar que aquí no se deben migrar datos sobre un diseño que presente las relaciones entre las tablas, con vistas a evitar cualquier error de copia de llaves duplicadas.
- Posteriormente se ha de crear, en PostgreSQL, las llaves primarias y foráneas para cada tabla de manera que se garantice la integridad referencial entre las tablas. Para esto se emplea el catálogo al cual se le realizan consultas con el objetivo de buscar las llaves primarias y foráneas de cada tabla.
- Siguiendo la estrategia del paso anterior sobre el catálogo se buscan por cada tabla los índices. A continuación, se realizan varias consultas con el objetivo de crearlos en PostgreSQL según las tablas que correspondan.

Finalmente se procede a migrar las vistas presentes en la base de datos. En este paso es muy importante tener en cuenta todas las características presentes en una consulta *SELECT*. Esto quiere decir que se han de tener en cuenta las funciones propias de cada gestor, conociendo como estas funcionan y qué parámetros reciben, así como la forma en que estos son declarados en PostgreSQL (Anthony R, S, L y otros, 2015).

Resultados y discusión

Tras realizar una descripción acerca de los pasos que se tuvieron en cuenta para la implementación del proceso de migración de bases de datos a PostgreSQL, se prosigue con una explicación detallada de la herramienta desarrollada.

Esta tiene como nombre JEasyDB presenta internacionalización para el idioma español e inglés. A continuación en la figura 2 se visualiza la interfaz principal de la herramienta de migración.



Figura 2. Interfaz inicial de la herramienta de migración JEasyDB.

En la interfaz mostrada en la figura 2 se introducen los parámetros necesarios para poder realizar como primer paso la conexión a los gestores MySQL, SQLite y dBase. La interfaz modifica dinámicamente los parámetros en dependencia del gestor seleccionado y después de establecida la conexión permite listar las bases de datos donde se seleccionará la BD a migrar. Una vez efectuado este paso de configuración de forma exitosa se habilita el botón siguiente accediendo a la configuración para PostgreSQL.

En el segundo paso se procede a llenar los parámetros necesarios para establecer la conexión con PostgreSQL y una vez establecida la conexión se procederá a escoger el nombre de la base de datos hacia donde se realizará la migración de la base de datos seleccionada (ver figura 3). Hay que tener presente que para poder avanzar hacia el próximo paso se hace necesario que la base de datos esté creada en PostgreSQL. La herramienta permite limpiar la BD seleccionada y además crear una nueva en PostgreSQL a solicitud del usuario.

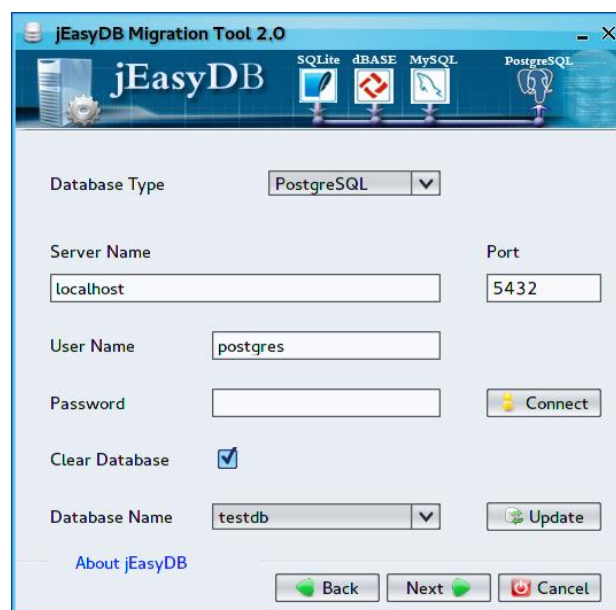


Figura 3. Interfaz de JEasyDB para la conexión al gestor PostgreSQL.

Posteriormente se le muestra al usuario el total de tablas y vistas que se migrarán. De cada una de ellas se puede ver cuáles son los nombres de las tablas, así como de las vistas. Para ello se ha de desplegar el árbol a través de un clic sobre el signo +. En la Figura 4 se visualiza un ejemplo donde los elementos a migrar son marcados mediante cajas de selección múltiple.



Figura 4. Interfaz de JEasyDB para seleccionar los elementos a migrar.

En este paso se puede acceder al botón Opciones, el cual visualiza una nueva ventana que permite especificar elementos genéricos de configuración para el proceso de migración. Entre los elementos fundamentales destaca la selección de objetos a migrar, especificaciones sobre el mapeo de datos, permitiendo al usuario redefinir elementos del mapeo incluyendo la creación de nuevos tipos de datos y finalmente la configuración de los registros de proceso de migración.

El siguiente paso comenzará el proceso de migración. En esta ventana se le muestra al usuario las tablas que están siendo migradas con la cantidad de registros por cada una de ellas, así como las llaves primarias y foráneas, y finalmente las vistas. Los resultados mostrados del proceso dependen de la configuración general establecida en el paso anterior (ver figura 5).

De existir algún error durante el proceso de migración se le muestran al usuario los registros de dónde y cuál fue la causa visualizado la información en el cuadro de texto Salida. Los registros muestran en detalle el proceso de migración de carácter informativo para que el usuario conozca de la evolución del proceso. Sin embargo, en caso de registrar errores se mostrará la consulta sql ejecutada y los objetos de la base de datos involucrados facilitado la toma de decisiones para la futura corrección. Finalmente se puede observar el resultado del proceso de migración de cada uno de los pasos descritos anteriormente en el gestor PostgreSQL.

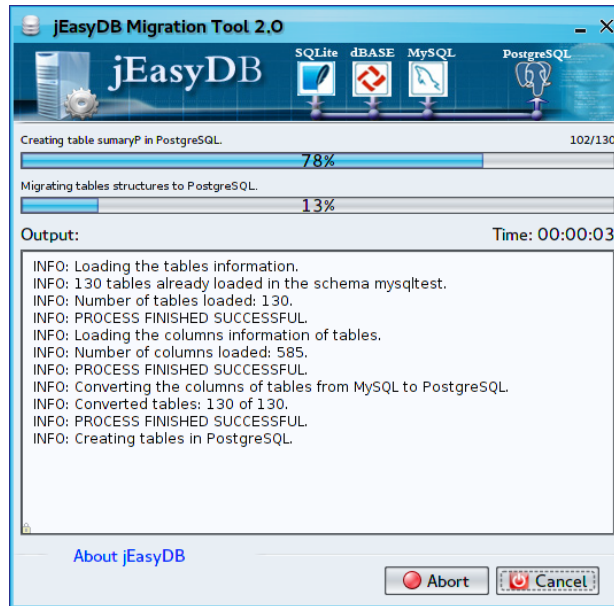


Figura 5. Interfaz de JEasyDB migrando base de datos entre los gestores.

Pruebas a la herramienta JEasyDB

Para la realización de las pruebas funcionales a la herramienta se utilizaron algunas bases de datos conformadas a partir de la unión de varias bases de datos e insertándole datos aleatoriamente. La figura 6 muestra los resultados al realizar el proceso de migración de las bases de datos seleccionadas analizando objetos de la BD como: total de tablas, registros, llaves primarias y foráneas además de los índices, las vistas y el tiempo máximo empleado en el proceso de migración.

SGBD	Tablas	Registros	Llaves primarias	Llaves foráneas	Índices	Vistas	Tiempo de Migración
MySQL	16	47273	16	21	25	7	00:00:14
	130	1946372	130	175	179	23	00:07:18
SQLite	44	3374485	35	32	63	-	00:15:27
	16	35345	16	21	24	-	00:00:10
dBase	1	1036	-	-	1	-	00:00:01

Figura 6. Resultados de la migración a PostgreSQL.

Valoración económica y aporte social

La herramienta JEasyDB obtenida facilita la personalización del proceso de migración permitiendo al usuario seleccionar las funcionalidades que se ajusten a las necesidades del negocio. Se desarrolló con tecnologías libres y se encuentra en la clasificación de herramientas de software libre. Este constituye un importante aporte en el ámbito económico pues evita la adquisición de herramientas propietarias que poseen altos costos adquisitivos en el mercado internacional.

También permitirá de manera gradual un avance en el proceso de migración a software libre con el objetivo de facilitar la independencia tecnológica de la UCI y del país. En conjunto a este proceso de migración se prepararán las bases para el uso del gestor PostgreSQL permitiendo que se aprovechen las ventajas del mismo y agilizándose el periodo de transición del empleo del antiguo gestor al nuevo.

En la actualidad realizar el proceso de migración sin automatizar es técnicamente complejo debido a que se necesita el empleo de recursos humanos capacitados y de períodos de tiempos variables en dependencia de la complejidad estructural de la base de datos que se quiere migrar. JEasyDB permite automatizar el proceso, así como una reducción sustancial en el tiempo de migración.

Esta solución informática trae consigo el beneficio del mínimo esfuerzo para la entidad, aportando una reducción en costos por pago de licencias de otras herramientas con el mismo fin, el empleo de un pequeño equipo por lo que disminuye el gasto por salario y esfuerzo del personal de la entidad con la tarea de migración de la BD, contribuyendo directamente en el cumplimiento de los cronogramas de trabajo.

Conclusiones

El estudio realizado de las diferentes herramientas que permiten llevar a cabo un proceso de migración entre los SGBD MySQL, SQLite y dBase a PostgreSQL, permitió concluir que en su mayoría no llegan a realizar un proceso completo de migración de cada uno de los elementos necesarios en el diseño de la base de datos, además de ser herramientas que no están basadas en su totalidad en software libre.

El desarrollo de la herramienta para migrar bases de datos JEasyDB permitió la migración de los datos contenidos en el SGBD MySQL, SQLite y dBase a PostgreSQL permitiendo la visualización del proceso mediante trazas.

La aplicación del proceso de migración propuesto permite la migración íntegra de la estructura de la base de datos, datos, llaves primarias y foráneas, índices y vistas de los SGBD MySQL, SQLite y dBase a PostgreSQL.

Referencias

1. AJDB SOFT. [En línea] [Citado el: 27 de Septiembre de 2016.]
<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=500>.
2. Anthony R, S, L y Yudisney, V, O. PL/PGSQL y otros Lenguajes Procedurales en PostgreSQL. ISBN: 9781312994898, La Habana. Universidad de las Ciencias Informáticas 2015.
3. DB CONVERT. [En línea] 2001-2016. [Citado el: 15 de Septiembre de 2016]
<http://dbconvert.com/convert-mysql-to-postgresql-pro.php>.
4. GUSTAVO RUIZ, CARLOS “Mejores Prácticas para Migrar su Base de Datos”. Centro Nacional de Tecnologías de Información de Venezuela (CNTI). 2016.
5. MYSQL-TO-POSTGRESQL 1.0. [En línea] 2006-2016. [Citado el: 21 de Septiembre de 2016]
<http://www.convert-in.com/sql2pgs.htm>.
6. MY2PG.PEEL. [En línea] 2006-2016. [Citado el: 23 de Septiembre de 2016]
<http://my2pg.script.soft32download.com/>.
7. Navicat. [En línea] 1999 – 2016. [Citado el: 23 de Septiembre de 2016].
<http://www.navicat.com/es/whatisnavicat>
8. POSTGRESQL. [En línea] 1996-2016. [Citado el: 18 de Septiembre de 2016]
<http://archives.postgresql.org/pgsql-es-ayuda/2009-01/msg00375.php>.
9. SOFTPEDIA. [En línea] 2001-2016. [Citado el: 10 de Septiembre de 2016]
<http://www.softpedia.es/programa-ESF-Database-Convert-Professional-105802.html>.