

DESARROLLO DE UNA APLICACIÓN DE SOFTWARE PARA LA MIGRACIÓN DE DATOS DE UNA ESTRUCTURA RELACIONAL A UNA ESTRUCTURA DE BIG DATA

Lilian Judith Sandoval

Licenciada en Administración de Empresas, con diplomado en Tecnologías de Software. Docente de la Escuela de Ingeniería en Computación. ITCA-FEPADE, Sede Santa Tecla. E-mail: Lilian.sandoval@itca.edu.sv

Giovanni Tzec Chávez

Técnico en Sistemas Informáticos. Docente de la Escuela de Ingeniería en Computación. ITCA-FEPADE, Sede Santa Tecla. E-mail: Giovanni.tzec@itca.edu.sv

Guillermo Adalberto Jandres

Técnico en Sistemas Informáticos. Docente de la Escuela de Ingeniería en Computación. ITCA-FEPADE, Sede Santa Tecla. E-mail: Guillermo.jandres@itca.edu.sv

Resumen

El presente artículo está basado en los resultados de una investigación aplicada llevada a cabo por docentes investigadores de la Escuela de Ingeniería de Computación de ITCA-FEPADE Santa Tecla. Éste consiste en el desarrollo de una aplicación de software que aborda el conocimiento de una nueva tecnología llamada Big Data para el manejo de información, en sustitución de los sistemas tradicionales y, de una manera más específica, aplica los lineamientos para realizar la migración entre ambas tecnologías. Se mencionan los diferentes componentes de un Sistema Big Data, el proceso de transformación que deben sufrir los datos contenidos en un esquema relacional y, finalmente, la integración a un nuevo esquema Big Data, donde no solo se resolverá la limitante del espacio de almacenamiento, sino también el rendimiento en el procesamiento de los datos, ya que se optimizará de manera notable.

Palabras clave

Big data, bases de datos, sistemas de información, gestión de datos, gestión de sistemas de computación.

Abstract

The present article is based on the results of an applied investigation conducted by the research professors from the Computer Engineering School at ITCA- FEPADE Santa Tecla. It consist about the development of a software application that addresses the study of a new technology called Big Data used for information management, replacing traditional systems and, more specifically, applies the guidelines to perform the migration between both technologies. It will mention the different components of a Big Data System, the transformation process that must undergo the data contained in a relational scheme and, finally, the integration to a new Big Data scheme, where not only the limitation of the storage space will be solved, But also the performance in the processing of the data would be improve, since it will be optimized in a remarkable way.

Keywords

Big data, databases, information systems, data management, computer systems management.

Introducción

Con el nacimiento de las redes sociales y las aplicaciones móviles, surgió la necesidad de manejar volúmenes de información gigantescos y variados que las bases de datos tradicionales ya no pueden soportar, además de lo complejo que se ha vuelto el proceso de búsqueda de información. Por lo anterior, se ha hecho necesario pensar en estructuras de datos completamente distintas, donde

la limitación de espacio no fuera más un problema. Por ello se ha creado la nueva tecnología Big Data para el manejo de volúmenes de datos e interpretación de ellos para diferentes propósitos. Big Data maneja conjuntos de datos enormes que crecen tan rápido que se vuelve muy difícil manipular y analizar a una granularidad tal donde los procesos normales colapsan. [1]

Esta nueva tecnología no solo viene a resolver los problemas de almacenamiento y gestión que plantean las redes sociales, sino que auxilia a otros sectores que también presentaban las mismas dificultades como el científico, el médico, el mercadológico, etc. En este sentido y en vista de la dificultad que se aproxima para la mayor parte de empresas que en la actualidad manejan su información con bases de datos tradicionales (relacionales), se ve la necesidad de implementar una solución que permita migrar esa información de forma rápida y segura a la nueva tecnología Big Data. De esta manera no es necesario comenzar a cargar los datos desde cero, ya que además de generar la estructura de la base de datos Big Data, también se realizará la extracción de los datos relacionales.

1. Marco Teórico

BIG DATA. Es una tecnología que se ocupa de todas las actividades relacionadas con los sistemas que manipulan grandes conjuntos de datos. Las dificultades más habituales vinculadas a la gestión de estas cantidades de datos se centran en la recolección y el almacenamiento, búsqueda, compartición, análisis, y visualización. Big Data maneja conjuntos de datos enormes que crecen tan rápido que se vuelve muy difícil manipular y analizar a una granularidad tal donde los procesos colapsan. Esta nueva tecnología no solo viene a resolver los problemas de almacenamiento y gestión que plantean las redes sociales, sino que también auxilia a otros sectores que también presentaban las mismas dificultades como el científico, el médico, el mercadológico, etc. [4]

MONGO DB. Es un sistema de bases de datos no relacionales, multiplataforma e inspirada en el tipo de bases de datos documental y clave/valor. Su nombre proviene del término en inglés "humongous" que significa "enorme". Está liberada bajo licencia de software libre, específicamente GNU AGPL 3.0.

MongoDB usa el formato **BSON (JSON Compilado)** para guardar la información, dando la libertad de manejar un esquema libre. Este motor de bases de datos es uno de los más conocidos y usados, pudiéndolo comparar en popularidad con MySQL en el caso de las bases de datos relacionales. [5]

JSON - JavaScript Object Notation. Es el formato compacto de representación de objetos para intercambio de datos. Las especificaciones las publicó Douglas Crocford en el documento RFC 6274. JSON es un formato independiente del lenguaje, aunque su uso extendido hasta hace poco era en el lenguaje JavaScript. Actual-

mente se usa JSON en gran cantidad de sistemas para intercambiar información por su simplicidad en comparación con XML. Este formato soporta gran cantidad de tipos de datos, lo que lo hace atractivo para un uso generalizado, y cada vez más lenguajes de programación dan soporte a este formato. [6]

2. Metodología de Investigación

La metodología aplicada a la investigación es exploratoria, ya que la temática en investigación, a pesar de tener ya algunos años de existencia, en nuestro país es relativamente nueva. Se investigarán los diferentes procesos para la migración de datos de una tecnología ya ampliamente utilizada, como es el modelo relacional, hacia la nueva tecnología Big Data, donde no se conoce si existe un método ya definido o es necesario definir uno y determinar las herramientas que pudieran ser de utilidad para lograrlo.

3. Resultados

3.1. Proceso de Migración de Datos

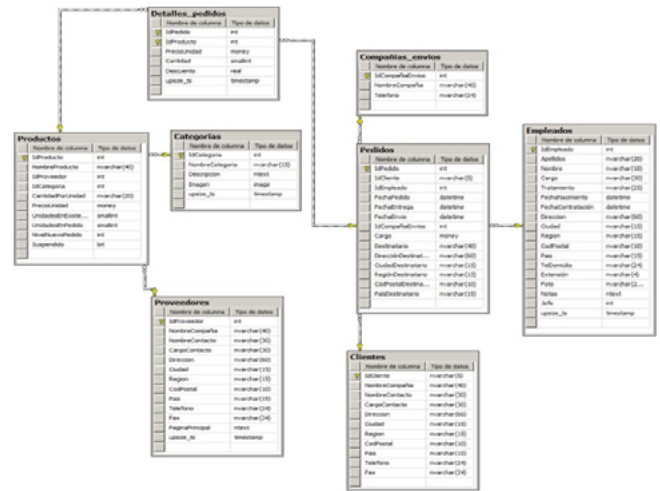


Figura 1. Diagrama Entidad Relación de la Base de Datos Relacional

El proceso para migrar la información de un esquema relacional a BigData, utilizando los gestores de bases de datos, consiste en crear scripts para transformar datos fuentes a una estructura jerárquica en formato JSON que pueda ser importada a MongoDB. Este proceso utiliza subdocumentos y arreglos, que es la forma en como JSON se ajusta a la estructura de objetos a nivel de aplicación. [7]

3.2. Pasos para la Migración

1. Agregar una nueva columna dentro del sistema relacional para el UUID (Identificador Único Universal) creándola en la nueva tabla y poblándola usando la función NEWID().

```
SELECT *, UUID = NEWID( ) INTO NEWTABLE FROM OLDTABLE
```

2. Agregar los índices a la tabla.
3. Renombrar la nueva tabla y eliminar la anterior.
4. En el sistema relacional, se debe crear el script con el formato JSON en un procedimiento almacenado, conteniendo las consultas necesarias para recuperar todos los datos de la base de datos y agregar la cláusula FOR JSON PATH al final. Como en el esquema de BigData no existen relaciones de muchos a muchos, es necesario agregar índices para hacer coincidir la información de una colección a una colección padre. Todos los motores de almacenamiento de MongoDB soportan todos los tipos de índices y los índices pueden ser creados en cualquier parte de documentos JSON, incluyendo subdocumentos internos y elementos de arreglos.
5. En MongoDB, ejecutar los procedimientos almacenados usando el comando BCP para generar el archivo JSON.
6. Una vez generado el archivo JSON dentro de MongoDB, ejecutar el comando **mongoimport**. Se debe usar este comando mientras se está conectado a una instancia de **mongod**.

En el siguiente ejemplo, el comando **mongoimport** importa los datos del archivo con formato JSON **contacts.json** dentro de la base de datos **users**, en la colección **contacts**, en la instancia de MongoDB, corriendo en el localhost en el puerto número 27017. [9]

```
mongoimport --db users --collection contacts --file contacts.json
```

3.3. Diseño de la Aplicación

Escribir la siguiente dirección en cualquier navegador.



Figura 2. URL para ingresar a la aplicación Big Data Converter

Se presentará la pantalla de inicio, donde se ingresarán las siguientes credenciales por defecto: Usuario: **administracion** y contraseña: **123**.

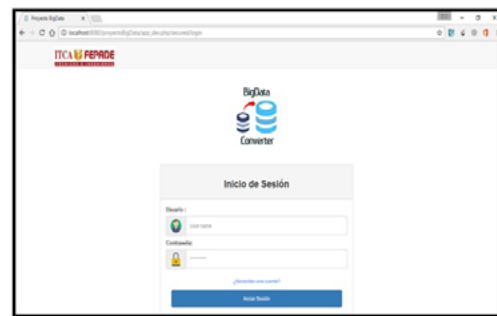


Figura 3. Pantalla de inicio de sesión

Si el usuario y contraseña son correctos, se mostrará la pantalla de inicio. Luego dar click en el link **Entrar** y se mostrará la siguiente pantalla, donde se encuentra un panel principal. Dar click en el botón **Iniciar** para comenzar con la migración. Se mostrará un asistente para iniciar el proceso de la migración. Seleccionar el gestor de base de datos relacional; por el momento el único gestor es SQL Server.

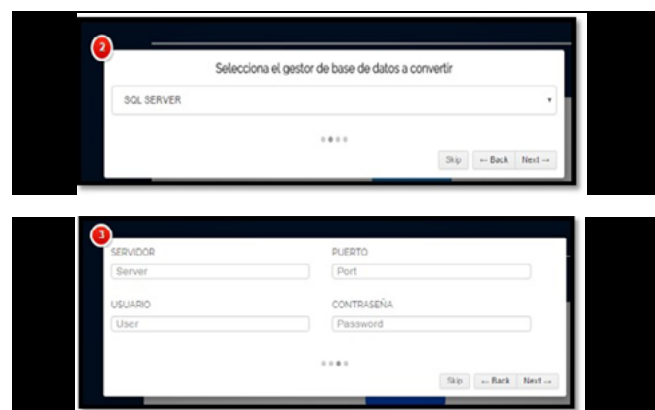


Figura 4. Selección del gestor de base de datos relacional y datos de conexión

Ingresar el nombre del servidor de base de datos SQL Server, el puerto, el nombre del usuario del SGBD y su contraseña. La aplicación mostrará todas las bases de datos que están creadas en el SGBD.



Figura 5. Bases de datos encontradas en el gestor



Figura 6. Pantalla de confirmación de la base de datos

Luego de confirmar la selección de la base de datos, se mostrarán todas las tablas que contiene la base de datos seleccionada.



Figura 7. Lista de tablas de la base de datos seleccionada

Al dar click en el botón **Convertir a BIGDATA**, se mostrará la siguiente pantalla donde se observa el proceso de conversión:

- Se extraen los datos de la base de datos en SQL server a través de la generación de una consulta SQL que incluye todas las tablas.
- Se genera un archivo intermedio con la información, en formato JSON.
- Se ejecuta desde el gestor MongoDB la instrucción para importar los datos que se encuentran en el archivo JSON a BigData. Si la migración ha sido satisfactoria, se presentará el mensaje respectivo.

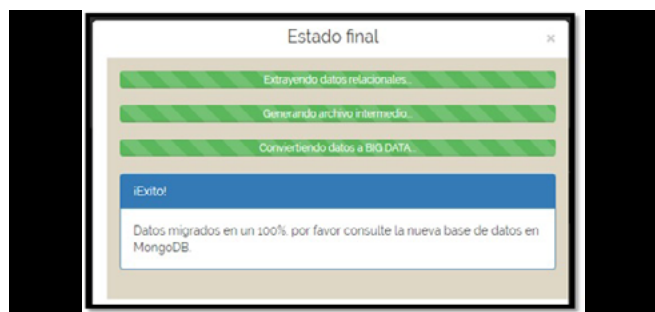


Figura 8. Procesos realizados durante la migración

3.4. Pruebas de la Aplicación

El proceso que lleva a cabo la aplicación **Big Data Converter**, consiste en ejecutar un asistente que permite seleccionar la base de datos en un esquema relacional para ser transformada a un esquema Big Data.

El proceso de transformación consiste en generar una consulta SQL que incluya los datos de todas las tablas en un solo comando. Para el ejemplo, se utilizó la base de datos NEPTUNO.

La consulta que se genera es la siguiente:

```
select c.*,pd.*,pe.*,dp.*,ca.*,pv.*,e.*,ca.* from Pedidos
pe left outer join clientes c on c.idCliente=pe.IdCliente
left outer join Empleados e on e.IdEmpleado=pe.
IdEmpleado left outer join detallesdepedidos dp on
dp.idpedido=pe.IdPedido left outer join productos pd
on pd.idproducto=dp.idproducto left outer join categorías
ca on ca.idcategoria=pd.idCategoria left outer join
proveedores pv on pv.idProveedor=pd.idProveedor for
json auto, root('neptuno')
```

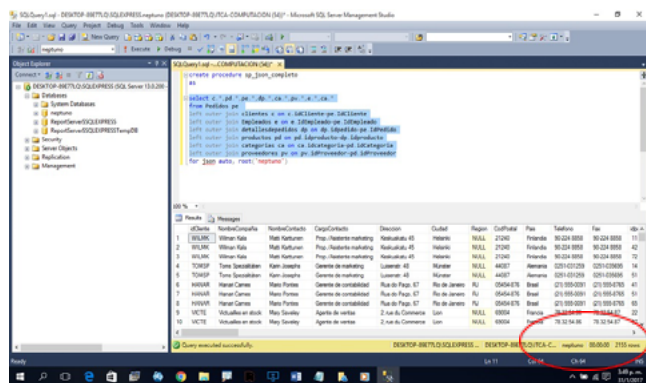


Figura 9. Consulta SQL que recupera todos los datos de la base de datos relacional

La consulta del ejemplo genera 2155 registros, que es el número de registros que contiene la tabla **detallesdepedidos**. Esta es la tabla transaccional ya que es la que tiene el mayor volumen de datos y es alrededor de la cual giran los datos de las demás tablas. Con esta premisa se generará el archivo intermedio en formato JSON. El archivo queda almacenado en la carpeta **jsonFiles**. Luego que la herramienta crea el archivo JSON, ejecuta la importación desde el gestor de datos de Big Data en MongoDB tomando como fuente la información del archivo JSON. Para verificar que los datos han sido migrados es necesario levantar el servidor de MongoDB, desde una ventana de símbolo del sistema con el comando **mongod**.

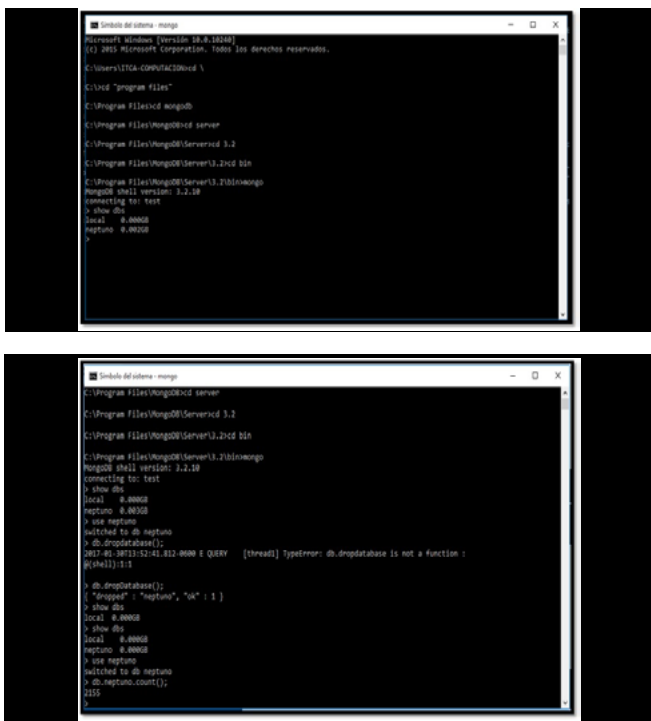


Figura 10. Listado de bases de datos de MongoDB y cantidad de registros

En otra ventana de símbolo del sistema, iniciar MongoDB con el comando **mongo**. Con el comando **show dbs** verificar que se haya creado la base de datos. En nuestro ejemplo, la base de datos se llama Neptuno.

Para acceder a la base de datos, se emite la sentencia **use neptuno**. Luego de abrir la base de datos, ya es posible verificar la cantidad de registros con la función **db.neptuno.count()**. Debe coincidir con la cantidad de registros que se obtuvo en el gestor SQL server con la consulta emitida. En nuestro caso, podemos confirmar que, en efecto, son 2155 documentos. Luego de contar los registros y comprobar que no se ha perdido ninguno, se puede proseguir a ver el contenido de la base de datos en MongoDB usando la sentencia **db.neptuno.find().pretty()**. Si se desea hacer búsquedas específicas de algún documento en especial o grupo de documentos para validar la información, se puede emitir la sentencia **find()** especificando alguna condición de búsqueda.

Por ejemplo, **b.neptuno.find("IdCliente":"CHOPS").pretty()** y luego **db.neptuno.count("IdCliente":"CHOPS")** para confirmar el número de documentos devueltos y comparar que coincidan con los registros devueltos por la consulta SQL.

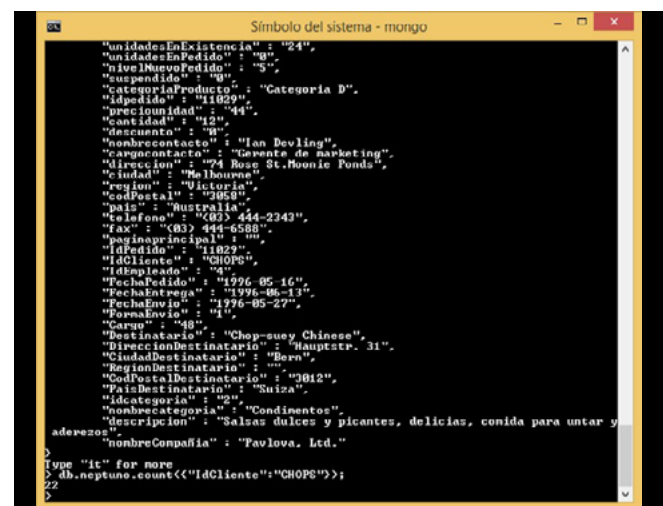
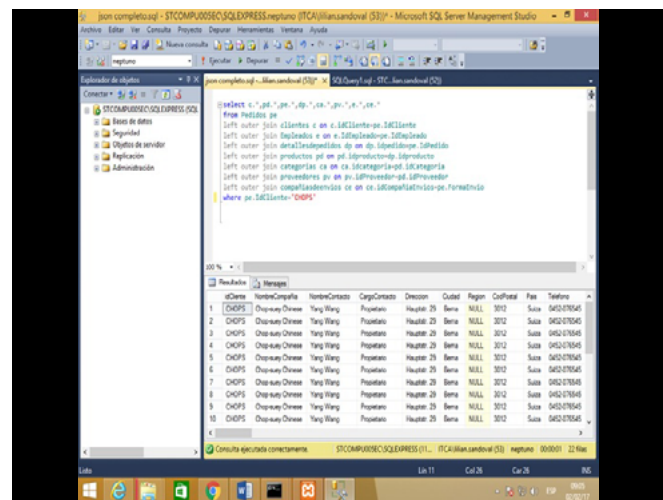


Figura 11. Consulta en SQL Server y en MongoDB para realizar una búsqueda específica

Conclusiones

- Con el desarrollo de la investigación se está brindando a nivel de país el conocimiento básico de la tecnología Big Data, la cual aún no ha sido ampliamente difundida y se está aportando una herramienta que, aunque se encuentra en su primera versión, puede auxiliar a las empresas que tomen la decisión de migrar sus sistemas a Big Data sin necesidad de comenzar la alimentación de sus bases de datos desde cero.
- Se modeló con éxito una base de datos Big Data teniendo como fuente una base de datos con esquema relacional para determinar las sentencias necesarias a utilizar en la migración y luego se incorporaron en la aplicación.
- Se verificó con éxito la integridad de los datos migrados del esquema relacional al esquema Big Data.

- Se desarrolló una aplicación que genera el esquema de cualquier base de datos relacional a una base de datos Big Data y la migración de los datos.
- Se seleccionó una metodología donde las instrucciones se emiten desde los gestores de bases de datos, tanto relacional, como Big Data para realizar la migración.
- Se encontraron las herramientas necesarias para poder recuperar todas las bases de datos existentes en un servidor, sin saber el nombre de éstas ni la ubicación.
- Se generó un asistente para pedir la información sobre la base de datos que se desea migrar, mostrando todas las bases de datos encontradas en el sistema haciendo

uso de las herramientas anteriores.

- En el gestor de base de datos SQL Server 2016 ya trae incorporadas las funciones necesarias para enviar la información recuperada desde la base de datos relacional hacia un archivo intermedio en formato JSON, el que luego se usa como insumo para convertir la información a Big Data. Pero se tuvo el inconveniente que el lenguaje PHP aún no posee la capacidad de ejecutar estas funciones, por lo que no se pudo realizar esta acción desde el gestor SQL Server. La solución fue buscar otras herramientas de PHP que realizaran la acción de crear el archivo JSON a partir de la información recolectada con consultas SQL ejecutadas dentro de PHP.

Recomendaciones

- Contar con los permisos necesarios como administrador o dueño de las bases de datos para poder tener acceso a ellas.
- Es importante depurar la base de datos del esquema relacional y que se cumplan todas las reglas de validación y de integridad referencial.
- Es necesario que todas las tablas de la base de datos

estén relacionadas, pues es a través de las relaciones que se va integrando la información en las consultas para recolectar los datos en forma integral.

- Se recomienda hacer pruebas de stress con bases de datos suficientemente grandes para comprobar tiempos de respuesta y verificar la eficiencia de la base de datos Big Data en comparación con la base de datos relacional.

Referencias

[1] V. Mayer Schonberger y K. Cukier, Big Data: la revolución de los datos masivos. 1ª. ed. Madrid: Turner, 2013.

[2] J. López Quijano, Domine PHP y MySQL: programación dinámica en el lado del servidor. 1ª. ed. México, D.F., MX: Alfaomega, 2007.

[3] A. Gutiérrez Rodríguez y G. Bravo García, PHP 5: a Través de Ejemplos 5. 1ª. ed. México, D.F., MX: Alfaomega, 2005.

[4] "¿Qué es Big Data?" CT316, 18 de junio de 2012. [En línea]. Disponible en: <http://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/index.html>. [Accedido: 15 -feb-2017]

[5] Y. Graterol, "MongoDB en español: T1, El principio". [En línea]. Disponible en : <https://openlibra.com/es/book/mongodb-en-esp>

nol-t1-el-principio. [Accedido: 15 -feb-2017]

[6] "Introduction JSON". [On line]. Available: <http://www.json.org/>. [Accessed: 18 -Feb-2017]

[7] "Rdbms to Mongodb migration". [On line]. Available: <http://xuphol.com/rdbms-to-mongodb-migration-pdf>. [Accessed: 5 Abr- 2017]

[8] "Manual de PHP". [En línea]. Disponible en : <http://php.net/manual/es/index.php>. [Accedido: 17 -febr- 2017]

[9]A. Copquin and V. Carballo, "The Knot : migration from SQL to MongoDB – A Case Study at Theknot" [On line] Available: <https://www.mongodb.com/presentations/migration-from-sql-to-mongodb-a-case-study-at-theknot-com>. [Accessed: 17 -Feb-2017]