

Almacenamiento *raster*

implementando *GridFS* en la base de datos *MongoDB*

Raster Data Storage in Big Data Infrastructures:

Implementing the GridFS Tool for a MongoDB Database

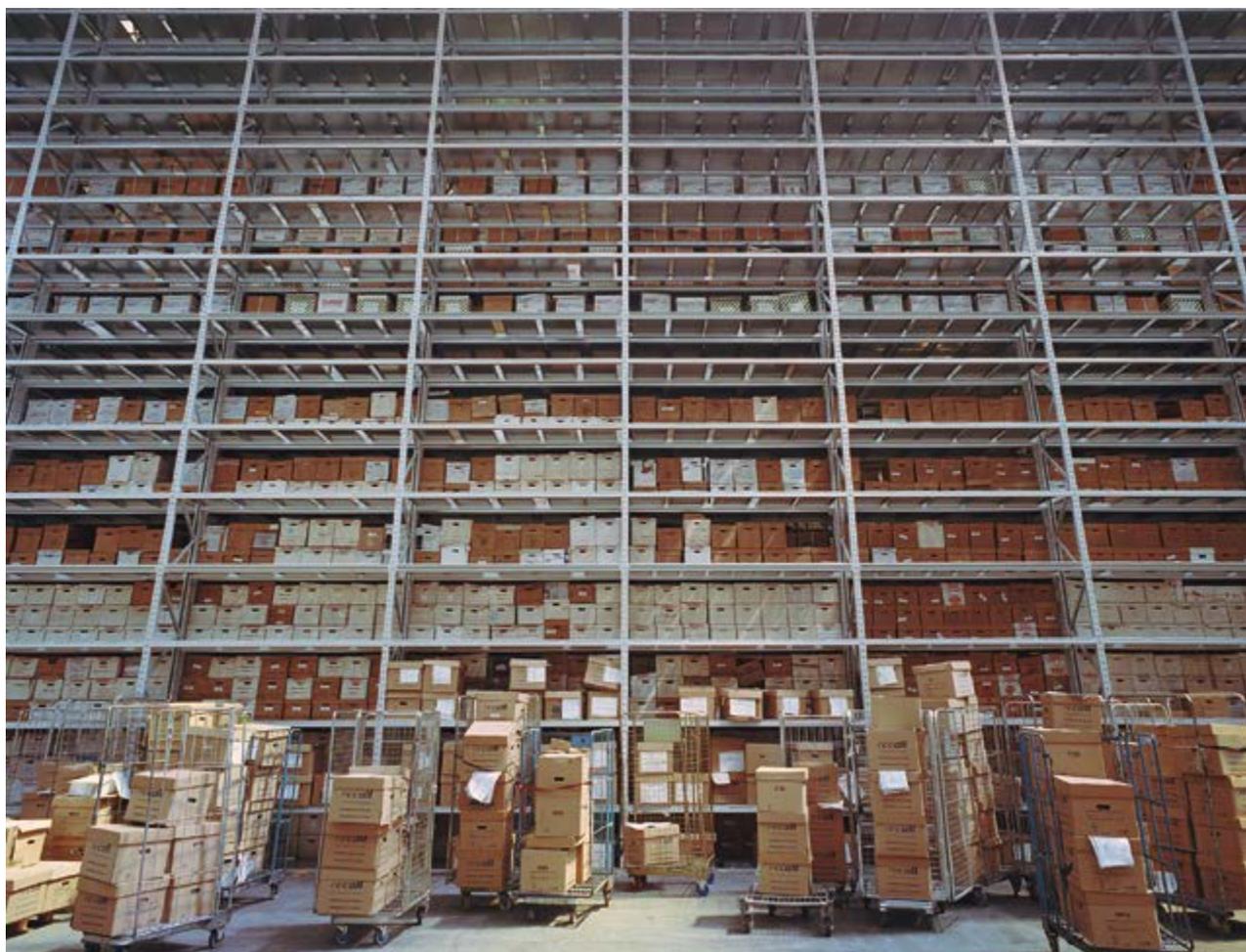
Marco Antonio López Vega*, **Stephane Couturier**** y **Daniel Gonzalo Hernández Rivera*****

*Instituto de Geografía (IGg) de la Universidad Nacional Autónoma de México (UNAM), marco@igg.unam.mx

** IGg de la UNAM, andres@igg.unam.mx

*** IGg de la UNAM, isamux_fca@hotmail.com

Nota: esta investigación se realizó en el marco del proyecto número 208637 de fondos sectoriales CONACYT-INEGI-VISTA-C y del proyecto de Laboratorios Nacionales LN295081, por lo que se agradece a las instituciones mexicanas que favorecen la investigación pública.



Data storage facility/Richard Ross/Getty Images

La modificación del paradigma actual de trabajo con los datos espaciales representa un hito en la evolución de los sistemas de información geográfica. La implementación de nuevos formatos y bases de datos espaciales se derivan de la necesidad de resolver viejos problemas en la gestión de la información geográfica. La integración de tecnologías del tipo *Big Data* ofrece nuevas soluciones y posibilidades en el desarrollo de sistemas que apoyen a quienes toman decisiones. *GridFS* ofrece una manera de almacenar la información del tipo *raster* y permite la interoperabilidad con bases de datos espaciales del tipo *NoSQL*, como lo es *MongoDB*. En este artículo se describe la implementación de esta tecnología y se ofrece una alternativa para la gestión de datos espaciales, en especial del tipo *raster*.

Palabras clave: SIG; *Big Data*; *raster*; *GridFS*; *MongoDB*.

There has been a shift in the paradigm associated to the organization of spatial data, which is currently a challenging topic for the Geographic Information Science. The emergence of new formats and database types is responding to old problems in the management of geographic information. Recently, Big Data technologies offer new opportunities for the development of systems for decision making. In this context, GridFS is a recent tool to store information of the raster type, which ensures interoperability with NoSQL database types, such as MongoDB. This article describes how this technology is implemented in a real project and presents GridFS as an alternative, efficient way for the management of Raster type data.

Key words: GIS; Big Data; Raster; MongoDB.

Recibido: 11 de mayo de 2017.

Aceptado: 23 de agosto de 2017.

I. Introducción

Durante las últimas décadas, las organizaciones que desarrollan y utilizan sistemas de información geográfica (SIG) se han enfrentado a un gran reto debido a la complejidad de los datos espaciales (la combinación de alfanuméricos y multidimensionales). A lo largo de la evolución de los sistemas de almacenamiento espacial, se han creado diversas soluciones para organizar la información, tanto para los datos *vector* como para los *raster*, formadas por una malla de valores con filas y columnas, donde cada pixel representa un valor (Mcinerney y Kempeneers, 2015). Una manera sencilla y estructurada para realizar dicha tarea es la implementación de una base de datos (BD) espaciales relacionales.

Los *raster*, por su tamaño y complejidad, siempre han sido un reto mayor en su almacenamiento y gestión. En efecto, cuando se examinan los diferentes SIG que actualmente están operando, la gestión de estos datos llega a significar problemas de despliegue y procesamiento. Es necesario tomar en cuenta que no es una simple imagen que se puede almacenar como un objeto binario.

Si bien durante muchos años se implementaron bases de datos relacionales —como *ORACLE*, *PostgreSQL (PostGIS)* y *Microsoft SQL*— para almacenar los de tipo *raster* (Yeung y Hall, 2007), éstas han denotado restricciones significativas. El principal reto empieza por el despliegue; por lo

normal, un *raster* es de gran tamaño, y si no se cuenta con un sistema lo suficientemente robusto para procesar el archivo tanto en su consulta en la base de datos como para su despliegue en la aplicación *GIS* o vía web, el desempeño del sistema que esté a cargo de alojarlo será subóptimo.

En esta investigación —y como parte de la construcción de la plataforma Visualizador de Información Satelital, Tendencias Ambientales y Clima (VISTA-C), financiada por el Fondo Sectorial Consejo Nacional de Ciencia y Tecnología (CONACYT)-Instituto Nacional de Estadística y Geografía (INEGI), y desarrollada en el Instituto de Geografía (IGg) de la Universidad Nacional Autónoma de México (UNAM)— implementamos nuevas tecnologías para robustecer el almacenamiento, la gestión y el despliegue de datos *raster* en plataformas nacionales.

En este artículo, describimos los esfuerzos antecedentes a nuestra investigación (sección II) y presentamos las características de la herramienta *GridFS* (sección III); luego, explicamos la tendencia tecnológica en centros de datos a nivel internacional (sección IV) y nuestra plataforma VISTA-C en particular (sección V), la cual incluye los requerimientos del almacenamiento (V.1.), la estructura de *software* del sistema (V.2.), nuestra propuesta para el almacenamiento de los datos *raster* (V.3.), la infraestructura de *hardware* (V.4.) y, finalmente, los impactos que ha tenido la plataforma en funcionamiento (V.5.).

II. Antecedentes

Para la integración de los *raster* en una base de datos, se debe seleccionar un tipo de dato que permita modelar una estructura organizacional. Diferentes tipos se han implementado, como: *Binary Large Object (BLOB)*, *Raster* y *Geomval*, los cuales corresponden a una BD relacional (Yeung y Hall, 2007).

Las bases de datos relacionales se caracterizan por el uso de tablas con información que están relacionadas mediante llaves que permiten tener elementos únicos dentro de la base, las cuales se han desarrollado desde la década de los 70 (Yeung y Hall, 2007), pero no representan una manera óptima de organizar, gestionar y almacenar la información espacial; muestran fallas, como: repetición de ésta y poco poder de procesamiento con archivos de gran tamaño, ya que se presentan cuellos de botella en las entradas y salidas de los datos dentro de los sistemas debido al manejo de datos alfanuméricos y multidimensionales. Es por ello que, más recientemente, se han desarrollado nuevas formas que permiten erradicar dichos problemas de las BD relacionales.

La presente era y cultura digital posibilitan la generación de una gran variedad de datos, de tal manera que nuestra interacción con dispositivos móviles permite la creación de datos en diferentes formatos, por lo normal de gran tamaño, ya que la periodicidad de los mismos puede incrementar significativamente nuestro acervo.

Los *raster* poseen características especiales denominadas espaciales, las cuales no se presentan en los que comúnmente llamamos tipos de datos alfanuméricos; éstas son: georreferencia espacial, resolución, coordenadas extremas, número de pirámides y metadatos (Mcinerney y Kempeneers, 2015).

En las últimas décadas se han incrementado los acervos de datos espaciales, tanto vectoriales

como *raster*, pero estos últimos representan el porcentaje más elevado. El uso de diferentes sistemas satelitales para su adquisición (imágenes *raster*), así como fotografías aéreas, representan siempre un gran reto al momento de su almacenamiento y gestión.

Las nuevas herramientas tecnológicas para almacenamiento y análisis masivo, denominadas *Big Data*, tienen como principal objetivo gestionar datos de gran tamaño, como el de tipo *raster*, ya que en sus características esenciales (volumen, veracidad, velocidad y variedad) lo hacen un elemento idóneo para explotar todas las nuevas características poderosas de dicha tecnología (Bessis y Dobre, 2014). Actualmente, tenemos sistemas que gestionan la información de una forma diferente a las antiguas BD relacionales. Las bases de datos denominadas *NoSQL* —no solo *SQL*— nos permiten organizar los datos de una nueva manera, rompiendo el paradigma establecido desde hace más de 40 años (Yeung y Hall, 2007).

Las BD *NoSQL* nacieron debido a las necesidades de almacenar una gran cantidad de datos. Difieren de las relacionales ya que se pueden organizar por medio de archivos, columnas, gráficos y llaves. Debido a la naturaleza de los datos *raster* (su gran tamaño y periodicidad) hacen que este tipo de BD sea una solución adecuada para su gestión y almacenamiento. La implementación de nuevos formatos y lenguajes de programación de alto nivel proporcionan un comportamiento más ágil tanto en las consultas como en las operaciones de lectura y escritura.

Una de las variantes de este nuevo tipo de bases de datos es la denominada *por documentos*, en la que se establece una estructura organizacional implementando un formato llamado *JavaScript Object Notation (JSON)*. Una de las limitantes para este tipo de datos es el tamaño máximo de 4 gigabytes (GB). *MongoDB* es una BD *NoSQL* que funciona por documentos y proporciona una solución a este problema implementando una manera para almacenar y gestionar archivos de gran tamaño denominado *GridFS* (Dheeraj y Sinha, 2005).

El objetivo de este artículo es relatar la exploración de esta forma y otras herramientas (*Studio 3T*, *Pymongo* y *Python*) de *Big Data* para visualizar, gestionar y distribuir información satelital sobre tendencias ambientales y clima en el ámbito del proyecto institucional VISTA-C —como parte de la estación de adquisición, almacenamiento y procesamiento de imágenes de satélite (estación *ERISA*¹) ubicada en el Laboratorio Nacional de Observación de la Tierra (LANOT) en el IGg de la UNAM, México—, así como para mejorar e implementar un diseño realizado anteriormente con este propósito (López Vega *et al.*, 2015), resolviendo las dificultades encontradas.

III. Características de *GridFS*

Las BD espaciales relacionales implementan tipos de datos (*raster* y *BLOB*) para almacenar información de gran tamaño, pero *GridFS* construye dos tipos de elementos (colecciones), *Chunks* y *Files*, mediante los cuales se pueden almacenar datos *raster* que, regularmente, son de gran tamaño y representan la mayor parte de nuestra carga en un SIG. Los dos elementos funcionales de *GridFS* permiten guardar en su primera parte el nombre del archivo y sus metadatos; en la segunda se integra el dato que, a su vez, es dividido en varias piezas (*Chunks*) para un manejo óptimo en su almacenamiento. Las piezas tienen un tamaño estándar de 256 kilobytes (KB),

¹ <http://www.igg.unam.mx/sigg/investigacion/lage/carrusel/ca3/gomez/1.php>

el cual puede ser modificado por el usuario, dependiendo del tipo de archivo con el que se trabaje. Esta particularidad nos permite trabajar con elementos *raster*, proporcionando una nueva forma de gestionar datos espaciales de una manera más eficiente (Dheeraj y Sinha, 2005).

Algunas ventajas principales de *GridFS* son:

- Acceso directo para optimizar las entradas y salidas de datos.
- Capacidad automática de gestión.
- Bajo costo.

IV. Centros de datos

Actualmente, éstos se componen de una colección de sistemas de almacenamiento. La implementación de virtualización y la red en la *nube* han permitido dar una solución temporal a sus problemas de almacenamiento. Sin embargo, por lo regular, se requieren de grandes inversiones financieras, ya que integran una infraestructura de redes, generadores y sistemas de respaldo de energía. Las nuevas generaciones de los centros de datos se enfocan más en la importancia del *software* que en el *hardware*, así centralizamos un mayor poder en los gestores de BD y en el sistema operativo en sí.

El establecimiento de nuevos paradigmas de funcionamiento en términos de bases de datos espaciales *NoSQL* nos lleva a la implementación de herramientas tecnológicas nuevas, como *MongoDB* y *MongoChef (Studio 3T)*, las cuales interactúan con un sistema operativo *Linux*. Cabe mencionar que toda su plataforma está formada con *software* libre (Plugge *et al.*, 2010). La utilización de *GridFS* en una combinación de *MongoDB* y *Geoserver* en arquitecturas de datos híbridas nos permite procesar y analizar datos de una forma diferente a la que se tiene con sistemas relacionales espaciales, ya que utiliza una manera de organización por medio de archivos que difiere de la estructura de llaves primarias con relaciones entre tablas. Esto hace posible desarrollar una nueva infraestructura de datos, formando una arquitectura más dinámica y rápida. Eso nos faculta a construir soluciones más fácilmente, aumentando la capacidad del análisis de los datos.

V. Proyecto VISTA-C

Éste propone poner a disposición del *Mapa Digital de México* del INEGI un acervo de variables climáticas y de contaminación atmosférica derivadas de imágenes satelitales NOAA y de modelos numéricos climáticos de gran robustez en formato *raster*.

El objetivo de este proyecto es la creación de una plataforma cartográfica en línea para tendencias ambientales y clima, incluyendo un visualizador dinámico (Couturier *et al.*, 2017). Después del análisis de la información con la que se cuenta para su desarrollo, se estableció que 90% de los datos son del tipo *raster*. Los metadatos de dicha información son parte esencial de la estructura del sistema, ya que en ellos se encuentran los elementos descriptivos de los datos.

VISTA-C se diseñó para que utilice solo *software* libre y para que su interfaz principal use una

BD relacional (*PostgreSQL-PostGIS*); se emplea el *software Geoserver* para la implementación de la interfaz de visualización, que es una aplicación ampliamente utilizada en sistemas de esta naturaleza (Lacovella y Youngblood, 2013). De una forma paralela, se desarrolló la infraestructura con una base de datos *NoSQL (MongoDB)*.

V.1 Requerimientos del almacenamiento

El sistema de almacenamiento representa un reto debido a las diferentes tecnologías que hoy en día están disponibles. Los puntos esenciales que éste debe cumplir son:

- Arquitectura flexible.
- Facilidad para escalar.
- Facilidad de gestión.
- Rápida implementación.

El rol de la tecnología está cambiando, actualmente no se puede pensar en un sistema de almacenamiento como un dispositivo aislado, la integración con la arquitectura y la aplicación con la que interactúa el usuario final es esencial.

V.2 Estructura de software

Para este proyecto, se diseñó un sistema de almacenamiento masivo que se encargara de contener la base de datos espacial. El sistema operativo con el que cuenta es *Linux* en su versión *CentOS 7* y el de archivos que se implementa es *Hadoop*, el cual permite la integración y gestión de volúmenes de gran tamaño. Para la construcción y gestión de la BD *MongoDB*, se utiliza la aplicación *MongoChef*.

Si bien *MongoDB* tiene una restricción con los documentos superiores a los 16 megabytes (MB) —*Binary JavaScript Object Notation (BSON)*—, la implementación de *GridFS* nos permite dividir los archivos más grandes (como los *raster*, ver tabla 1). De esta manera, podemos integrar nuestros archivos, que representan la mayor parte de nuestra información.

Tabla 1
Características de los archivos *raster*
AVHRR del LANOT

<i>Raster</i>	Tamaño	Satélite	Resolución	Temperatura
<i>Advanced Very High Resolution Radiometer (AVHRR)</i>	19-35 MB	Serie <i>TIROS-NOAA</i>	1.1 km	Tierra, agua, nubes

Ejemplo de archivos *raster* que se utilizan en este proyecto:

```
n18.081221.2059.ndvi.bil  
n18.081221.2059.ndvi.contents  
n18.081221.2059.ndvi.hdr  
n18.081221.2059.ndvi.sta  
n18.081221.2059.ndvi.xml
```

V.3 Forma de almacenamiento de los datos *raster*

Para la automatización del procesamiento y almacenamiento de la información en la base de datos, se desarrolló un programa para depositar todos los archivos *raster* de un mismo directorio e insertarles los metadatos correspondientes en la BD *MongoDB*. Ya que los datos se reciben con un formato diferente al que utiliza este sistema, se deben realizar tareas adicionales. Las tareas de este programa incluyen la organización de la información, creación de las colecciones en la base y la estructuración de los metadatos.

Por sus ventajas en multiplataformas —se pueden implementar aplicaciones en *Windows* y *Linux* sin necesidad de realizar cambios en el programa—, fue seleccionado el lenguaje de programación *Python*. Al ser un lenguaje interpretado no se necesita compilar el código fuente para poder ejecutarlo, lo que da grandes ventajas, como: una gran rapidez en la ejecución, fácil aprendizaje, está orientado a objetos y es para propósitos generales.

Los módulos necesarios para el funcionamiento del programa son:

1. *Pymongo*: es una distribución de *Python* que contiene herramientas para trabajar con la base de datos *MongoDB*.
2. *GridFS*: es una especificación de almacenamiento para documentos *BSON* que exceden el límite de 16 MB.
3. *OS*: provee una manera de usar las funcionalidades del sistema operativo que se esté utilizando, como *Windows*, *Mac* o *Linux*.
4. *Glob*: encuentra los patrones que se especifiquen para nombres de archivos usando las reglas del *Shell* de *UNIX*.
5. *re*: cuenta con funciones para trabajar con expresiones regulares y cadenas.

Variables:

1. IP: la dirección (xxx.xxx.xxx.xxx) del servidor donde reside la BD *MongoDB*.
2. BASEDATOS: nombre de la BD en *MongoDB*.
3. USUARIO: nombre del usuario de la BD en *MongoDB*.
4. PASSWORD: la palabra clave del usuario de la BD *MongoDB*.
5. COLECCION: el nombre que se le va a dar a la colección de datos a almacenar en la BD *MongoDB* (el programa se encarga de crearla).

La instrucción para efectuar la conexión con la BD *MongoDB* es:

```
from pymongo import MongoClient
client = MongoClient(IP)
db = client[BASEDATOS]
db.authenticate(USUARIO, PASSWORD, source=BASEDATOS)
```

Uso del módulo *GridFS* en *Python*:

```
fs = gridfs.GridFS(db, collection=COLECCION)
```

Instrucción para realizar la búsqueda de todos los archivos con terminación *bil*:

```
for filename in glob.glob('*bil'):
    file = open(filename, 'rb')
```

Se utiliza el mismo nombre de archivo, pero se cambia la terminación a *xml* para leer el archivo de metadatos:

```
file_ext = os.path.splitext(filename)[1]
newfile = filename.replace('.bil', '.xml')
```

Se guarda el archivo *bil* en la BD incluyendo sus metadatos:

```
fid = fs.put(file, filename=filename, content_type='raster', metadata=metadatos)
```

De una forma más sencilla se almacena el archivo de encabezado *hdr*:

```
for filename in glob.glob('*hdr'):
    file = open(filename, 'rb')
    fid = fs.put(file, filename=filename)
```

Siguiendo el mismo método, también se pueden almacenar los archivos *sta* y *contents*.

V.4 Infraestructura *hardware* de VISTA-C

Ésta se refleja en la figura 1 y la tabla 2.

Actualmente, la base de datos *MongoDB* del sistema cuenta con 3 terabytes (TB) de almacenamiento *raster* (ver figura 2).

V.5 Impactos de la plataforma VISTA-C

El cambio de perspectiva en el desarrollo de la base de datos espaciales permitió definir elementos clave para tener ventajas competitivas significativas con respecto a BD anteriores (ver tabla 3). La integración de la nueva tecnología incide en el plan estratégico de nuestra organización —el

Laboratorio de Análisis Geoespacial (LAGE) del IGg de la UNAM—, de tal forma que la estrategia de adquisición de nuevos dispositivos generadores de datos *raster* (p. ej. sistemas de recepción de imágenes satelitales) ya considera este nuevo sistema.

En la medida en que se están superando los paradigmas anteriores de elaboración de sistemas de información en nuestra organización, estamos enfocando los esfuerzos en los nuevos elementos tecnológicos. La parte esencial de este proceso consiste en identificar las áreas de mejora (velocidad, variedad, veracidad y volumen), y cada una de éstas va incidiendo en la forma de trabajar del personal del IGg. Otro de los logros con la implementación de la tecnología es su usabilidad en la construcción de nuevos sistemas de almacenamiento espacial (ver tabla 3), los cuales permiten generar un esquema con la participación de múltiples actores finales.

Figura 1

Infraestructura *hardware* de la plataforma VISTA-C

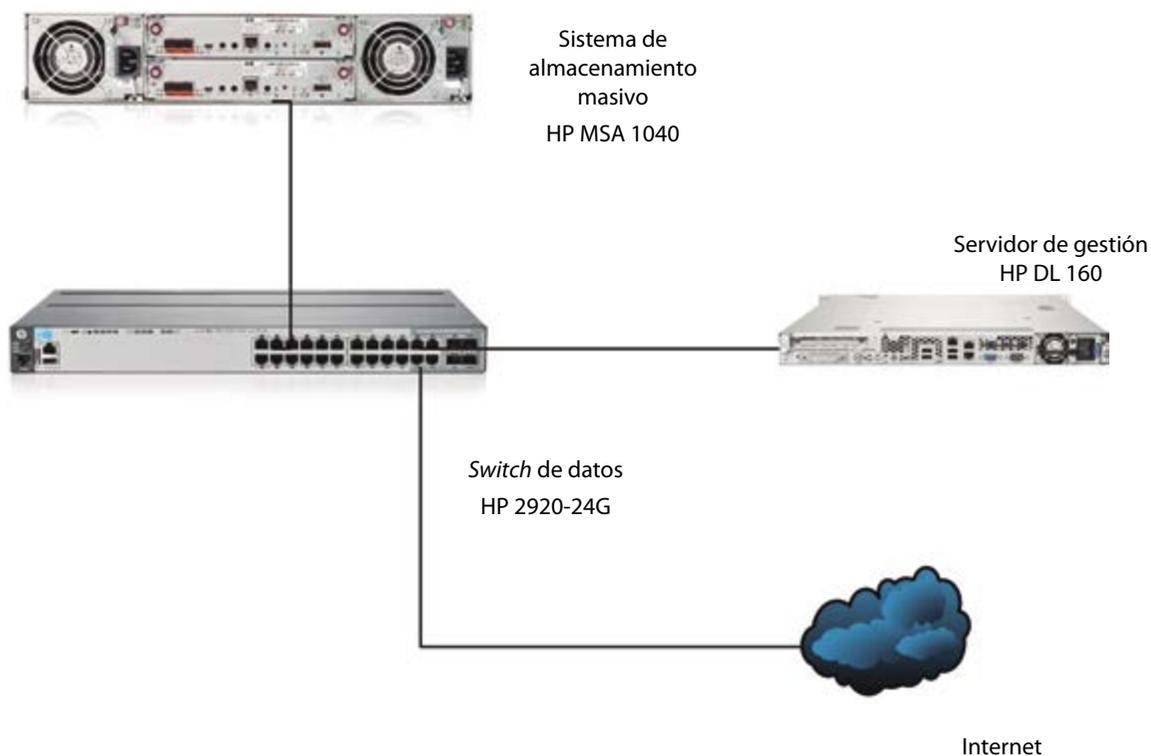


Tabla 2

Características de la infraestructura *hardware* de la plataforma VISTA-C

Equipo	Modelos	Características
Sistema de almacenamiento masivo.	HP MSA 1040	Sistema de almacenamiento 20 Terabytes SAS, controlador de 12 GB/s, discos de 2.5 pulgadas (2), puertos de 8 GB, <i>Fibre Channel</i> por controlador (2), puertos <i>iSCSI</i> de 1 GB por controlador (2), puertos <i>iSCSI</i> de 10 GB/s por controlador (2), puertos SAS de 12 GB/s por controlador.
Servidor de gestión del almacenamiento.	HP DL 160	Procesador Intel® Xeon E5-2600. <ul style="list-style-type: none"> • Dos procesadores. • Ocho core. • 3.0 GHZ. • 64 GB en RAM. • 1 GB 361i Ethernet.
Switch de datos.	HP 2920 24G	IEEE 802.1p GB ports (SFP + and/or 10GBASE-T)24 puertos 10/100/1000.

Figura 2

Visualización de la base de datos *MongoDB*

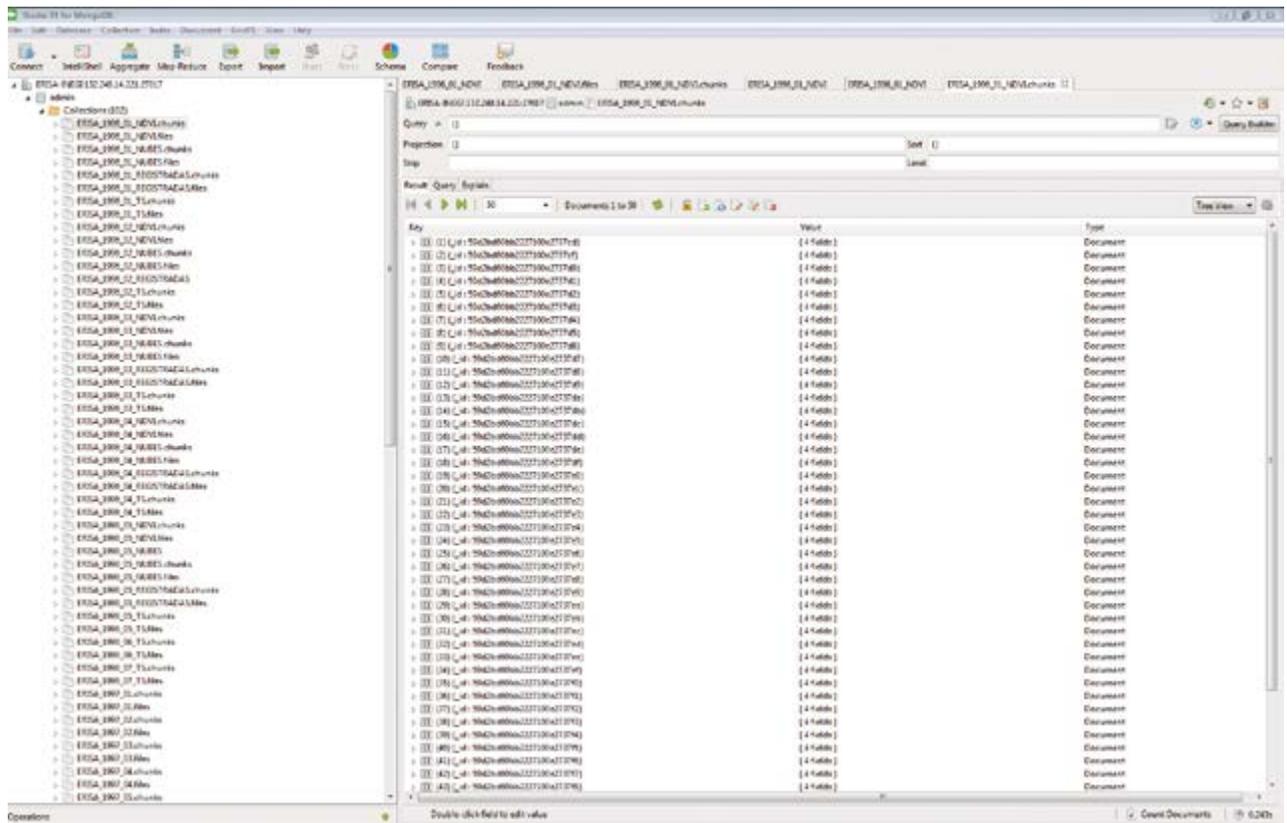


Tabla 3

Evaluación de ventajas y desventajas de la implementación de *MongoDB-GridFS*

Ventajas	Desventajas
El sistema integra <i>software open source</i> , el cual no tiene costo económico.	La curva de aprendizaje se incrementa, ya que algunos de los elementos son nuevos en este tipo de implementación.
Los dispositivos son más rápidos y confiables.	El <i>hardware</i> utilizado sigue siendo costoso, pues los dispositivos integrados son de alta tecnología.
La base de datos es más simple y estructurada.	Los nuevos esquemas de almacenamiento de la información exigen nuevas formas de organización de los datos espaciales.
La velocidad de respuesta de la base de datos es superior a la de tipo relacional.	
Las consultas externas se realizan con mayor velocidad, aun con enlaces de bajo ancho de banda.	
La recuperación de la base de datos en caso de un fallo es mucho más rápido.	
El proceso de integración de los metadatos dentro de la estructura <i>GridFS</i> se puede llevar a cabo mediante un archivo <i>xml</i> , el cual se ajusta al estándar del ISO 19115.	

VI. Conclusión

La transformación digital que se ha presentado en las últimas décadas dentro del área geoinformática ha dado paso a la integración de nuevas tecnologías y paradigmas de trabajo que si bien en un inicio representan un cambio técnico, también proporcionan hitos nuevos que derivan en la solución más óptima de problemas generados en la rápida transformación de nuestro planeta (Osorno Covarrubias *et al.*, 2015). Los datos espaciales, esenciales para la planificación de políticas públicas a varias escalas significan un gran reto en su procesamiento, más aún si nuestra información representa un mayor porcentaje de datos *raster* (p. ej. imágenes satelitales) que de tipo *vector*. La tecnología *Big Data* coloca sobre la mesa nuevas soluciones a la gestión y el procesamiento de grandes volúmenes de datos, pero una gran característica no solo es ésta, además nos permite la integración de variedad y velocidad en la información.

En nuestra investigación para el proyecto VISTA-C, proponemos una nueva forma de estructurar los datos, incrementando las posibilidades de aprovechamiento de la información. La nueva arquitectura basada en la tecnología *MongoDB-GridFS* no solo representa una opción viable para la creación de nuevos sistemas de información, actualmente está implementada en la infraestructura de datos espaciales del IGg de la UNAM, por ejemplo, en la plataforma VISTA-C y, próximamente, en otras plataformas del LANOT. Hoy en día, probamos que el uso de *GridFS* con la BD *MongoDB* constituye una solución factible que incrementa las posibilidades en el desempeño de las bases de datos espaciales de gran tamaño.

Entre las ventajas comparativas de la opción presentada, solo dedicamos presupuesto a la

adquisición de sistemas de almacenamiento de datos, en otras opciones probablemente se requiere la compra de licencias para el gestor de base de datos relacional o el publicador de información en la web. Esta propuesta representa una clara posibilidad de avance en el almacenamiento de información, no solo *raster*, sino que posibilita la unificación de nuevas fuentes de información espacial de gran tamaño.

Una pregunta crítica es: ¿en la práctica, la tecnología *Big Data* es una mejor opción que la ya establecida con las tecnologías relacionales? El desempeño del motor de búsqueda parece favorecer la tecnología *Big Data*, pero un objetivo a mediano plazo es crear una plataforma para la experimentación continua y para establecer mecanismos que midan los resultados. Las organizaciones deben entender que la transformación digital es un proceso continuo y que si se están actualizando sus centros de datos tienen que implementar soluciones tomando en cuenta los estándares internacionales que proporcionan integración e interoperabilidad. El enfoque de estándares abiertos en *software* y *hardware* reduce los costos. Quizás el único costo adicional es el intelectual, ya que requiere de nuevos conocimientos para el uso de esta tecnología.

Fuentes

- Bessis N., y D. Ciprian. *Big Data and Internet of Things: A Roadmap for Smart Enviroments*. Volume 546. Suiza, Springer, 2014, pp 137-168.
- Couturier, S., J. Osorno Covarrubias, V. Magaña Rueda, I. Martínez Zazueta y G. Vázquez Cruz. "Prototype of the Mexican spatial data infrastructure for climate raster models and satellite imagery ("VISTA-C")", en: *IOP Conf. Series: Earth and Environmental Science*. Vol. 54. Reino Unido, Institute of Physics (IOP) Publishing, 2017, pp. 1-6.
- Dheeraj, B. y M. Sinha. "GridFS: Ensuring High-Speed Data Transfer Using Massively Parallel I/O", *Databases in Networked Information Systems*. 4th International Workshop, DNIS 2005 Aizu-Wakamatsu, Japan, March 28-30, 2005 Proceedings. Alemania, Springer, 2005, pp. 280-287.
- Lacovella S., y B. Youngblood. *Geoserver Beginner´s Guide*. Reino Unido, Packt, 2013, pp 33-53.
- López Vega, M. A., S. Couturier y K. Y. Barrera González. "Design scheme for a spatial database of climatic and environmental variables in Mexico, integrating Big Data Technology", en: *Procedia Computer Science*. Vol. 55C. Países Bajos, Elsevier, 2015, pp. 503-513.
- Mcinerney, D. y P. Kempeneers. *Open Source Geospatial Tools*. Part of the series Earth Systems Data and Models. Suiza, Springer 2015, pp 51-60.
- Osorno Covarrubias, J., S. Couturier y M. Ricárdez. "El rol de la Geografía y sus hibridaciones recientes frente a la crisis de sustentabilidad global", en: *Boletín de la Asociación de Geógrafos Españoles (BAGE)*. Vol. 69 (10). España, Asociación de los Geógrafos Españoles, 2015, pp. 93-112.
- Plugge, E., P. Membrey y T. Hawkins. *The Definitive Guide to MongoDB The NoSQL Database for Cloud and Desktop Computing*. EE.UU., Apress, 2010, pp 83-95.
- Yeung A., y G. B. Hall. *Spatial Database Systems Design, Implementation and Project Management*. Países Bajos, Springer, 2007, pp. 22-24, 26-29 y 93-129.