Revista Colombiana de Computación
Vol. 23, No. 1. January – June 2022, pp. 34-43
e-ISSN: 2539-2115, https://doi.org/10.29375/25392115.4228

REVISTA COLOMBIANA DE COMPUTACIÓN

Evaluation of the response time of a geoservice using a hybrid and distributed database

Evaluación del tiempo de respuesta de un geoservicio utilizando una base de datos híbrida y distribuida

Marlen Treviño-Villalobos[1], Leonardo Víquez-Acuña[1], Rocío Quirós-Oviedo[1], Oscar Víquez-Acuña[1]

[1]Instituto Tecnológico de Costa Rica, Carrera de Ingeniería en Computación, San Carlos, Costa Rica.
mtrevino@tec.ac.cr, lviquez@tec.ac.cr, rocio.quiros@tec.ac.cr, oviquez@tec.ac.cr

(Received: 3 September 2021; accepted: 11 January 2022; Published online: 30 June 2022)

Abstract. Web mapping services provide information directly to users and other software programs that can consume and produce information. One of the main challenges this type of service presents is improving its performance. Therefore, in this research, a new geoservice integrated into GeoServer was developed, called GeoToroTur, with an OWS implementation of vector layers that consumes the information from a hybrid and distributed database that was implemented with PostgreSQL and MongoDB, making use of ToroDB for document replication. This geoservice was evaluated by executing geographic and descriptive attribute filter queries. Based on the results, we can conclude that the response time for GeoToroTur is shorter than that for Geoserver.

Keywords: Database, Geoservice, Response time, SQL, NoSQL.

Resumen. Los servicios de cartografía Web proporcionan información directamente, no sólo a los usuarios, sino también a otros programas de software que pueden consumir y producir información. Uno de los principales retos que presentan este tipo de servicios es mejorar su rendimiento. Por ello, en esta investigación se desarrolló un nuevo geoservicio integrado a GeoServer, denominado GeoToroTur con una implementación OWS de capas vectoriales que consume la información de una base de datos híbrida y distribuida que fue implementada con PostgreSQL y MongoDB haciendo uso de ToroDB para la replicación de documentos. Este geoservicio fue evaluado mediante la ejecución de consultas geográficas y de filtro de atributos descriptivos. Los resultados obtenidos permiten concluir que el geoservicio GeoToroTur tiene un menor tiempo de respuesta que Geoserver.

Palabras clave: Base de datos, Geoservicio, Tiempo de respuesta, SQL, NoSQL

Paper type: Research paper.

1 Introduction

The size of spatial data grows every day, and its management is increasingly complicated, being more frequent the generation of unstructured data (Moreno Jiménez, 2004; Růžička, 2016; Schmid et al., 2015; X. Zhang et al., 2014). This causes it to be difficult to index and respond to spatial queries by visualizing geographic layers through web mapping services or geoservices as they are also known (Bai et al., 2013).

Web cartography services provide cartographic information directly, not only to users but also to other software programs that can consume and produce information (Veenendaal et al., 2017). An important contribution to geoservices was the development of standards and specifications by the Open Geospatial Consortium (OGC). Some of the first standards issued by the OGC include Web Mapping Services (WMS), with the first version launched in 2000 (Open Geospatial Consortium, 2000). Since then, the number of geospatial standards has increased to more than 50 specifications published by OGC (Apostolopoulos et al., 2019).

Also, the Web Services Context Document (OWS) through XML files allows the distributed spatial processing system to interact with the HTTP protocol and provides an interoperability framework for discovery, access, combination, analysis, exploitation, and visualization based on online spatial data resources, sensor information, spatial processing services, and location services (Wenjue et al., 2004).

One of the challenges that has generated this increase in spatial data and computer resources is obtaining geospatial data, information, and knowledge from large volumes of geospatial sources distributed in an interoperable manner (Yuan et al., 2009). Another challenge is performance evaluation for measuring the efficiency of execution time, which is an essential quality factor for geoservices (OASIS, 2005; Schmid & Reinhardt, 2015). The use of metrics can provide guidelines for the selection of web services and server-side improvements for consumers and providers of these (Gui et al., 2016). Also, a high-performance response to a web request is a mandatory requirement (Loechel & Schmid, 2013). Therefore, improving the performance of these geoservices is becoming increasingly important (European Parliament, 2007; Gui et al., 2016).

Some authors have developed several research methods to solve the challenges posed in relation to geoservices. One of the topics studied is the use of techniques and retrieval of map image mosaic storage in caches, such as mosaic caches and reverse proxy caches, to obtain a faster response from a WMS (Y. Zhang et al., 2008). The proposed system organizes map image mosaics of the same layer in the same file with the relationships between the corresponding map image mosaics at different zoom levels.

GIS providers have implemented their own service management approaches and have not provided a unified interface, which brings many difficulties to users in the use, integration, and dynamic management, among others. In this context, a specification was proposed for managing WMS and WFS services based on a Service-Oriented Architecture (SOA) called Geographic Information Service Management (MSGIS) (Shen et al., 2010).

A study was conducted based on the premise that WMS clients are not well designed to support multiple connections to servers. This consisted of analyzing the performance patterns of different servers and based on these patterns, proposing the design of a new WMS client (Yang et al., 2007).

The scalability of a WMS service can be affected because the probability of receiving two exact mapping requests is very low and forces images to be dynamically generated on the fly every time a request is received. To illustrate this, a client can request arbitrarily sized map images from the server, overlay multiple layers in different coordinate reference systems, and apply specific background styles and colors. Therefore, to improve cache optimization and management, tessellate schemes were implemented, developing algorithms for initialization that fill the cache based on previous access history and for cache replacement based on neural networks (Garcia et al., 2012).

Similarly, in relation to the problem of scalability, a cloud computing implementation of a scalable argument mapping tool was proposed (Sani & Rinner, 2011).

The search for resources on the web gave rise to the development of a search engine that identifies and indexes all geospatial resources. It is called Geoweb Crawler and uses the MapReduce concept to improve its performance (Huang & Chang, 2016). This system responds to the problem of resource dispersion on the web, causing users to be unable to find the resources of their interest efficiently, and provides a specialized service unlike the generalized search engines.

Also, we have worked on the semantic search of distributed resources with the implementation of a mapping model for the conversion of the geoservice description information to the value of corresponding tags of the service profile through the web ontology language (OWL) (Miao et al., 2016).

The evaluation of geoservices using different Database Management Systems (DBMS) has also been studied (Růžička, 2016; Schmid et al., 2015). In the same way, dynamic web map services supported by a dynamic database connection have been analyzed (Zhao et al., 2006).

However, this article hypothesizes that a web map service has a shorter response time using a hybrid database. The scientific community has not analyzed this topic. It takes as a basis that currently requires databases capable of processing large amounts of data quickly and flexibly, requiring systems with higher performance (Puangsaijai & Puntheeranurak, 2017) and the existence of experiences that demonstrate the usefulness of a hybrid SQL and NoSQL database (Wu et al., 2017). Another factor to consider is that studies have shown that PostgreSQL with PostGIS has a limited visualization of simple and fast queries (Kepka & Ježek, 2013).

To respond to the hypothesis, a new geoservice was developed called GeoToroTur with an OWS implementation of vector layers that makes use of a hybrid database and contains a set of data corresponding to the Northern Huetar Region of Costa Rica. The hybrid databases work as an abstraction layer that is located on top of both the SQL and NoSQL databases (Goyal et al., 2015). The PostgreSQL and MongoDB database management systems were used to build this hybrid database (Colorado Pérez, 2017). The goal was to combine the best parts of the SQL and NOSQL paradigms while minimizing their flaws.

The selection of PostgreSQL was since it was one of the first databases to adopt the spatial theme (The PostgreSQL Global Development Group, 2022). Therefore, its PostGIS extension is highly optimized for spatial queries (Agarwal & Rajan, 2016; POSTGIS, 2022). It also has a large number of spatial functions that make it relevant. In addition, it is a stable and reliable database and offers great extensibility (Goyal et al., 2015).

MongoDB was chosen because it is the only document-based NoSQL database to date that supports line intersection and point containment queries (Agarwal & Rajan, 2016). In addition, a number of studies have determined that MongoDB has the lowest average query response time for the reading process among the DBMS NoSQL (Chopade & Dhavase, 2017; Gunawan et al., 2019; Kumar et al., 2017; Pramukantoro et al., 2019; Trevino Villalobos et al., 2018; Treviño-Villalobos et al., 2019)

Another feature that was considered is that both DBMS are open source and Geoserver has support for them since version 2.11.4 of Geoserver incorporated a component for the connection and publication of data from MongoDB (Open Source Geospatial Foundation, 2022).

To evaluate the developed geoservice, 75 queries were defined, 25 perform operations of different access patterns and complexity levels associated with geographical processes, such as: intersections, within, near and crosses. The remaining 50 queries, only perform basic retrievals to the geographical tables. The test consisted in obtaining the response time of the execution of the 75 queries as a whole by repeating the process 35 times in each geoservice. The results obtained allow us to conclude that the GeoToroTur geoservice has a shorter response time than Geoserver.

2 Methodology

2.1 Type and level of research

Given that the objective of this study is to evaluate the GeoToroTur geoservice with respect to that of the Geoserver, this study is comparative (A. Díaz, 2009). It is also experimental because tests were carried out in the laboratory to obtain the data. This research is classified as prospective because, from the point of view of planning the collection of data, these were obtained specifically for the writing of this article. Likewise, due to the number of measurements of the variable, the study is longitudinal since each geoservice was analyzed with 75 different consultations in 35 repetitions. In addition, the study is univariate because only one response variable is analyzed, and it is balanced because the treatments have the same number of repetitions. Likewise, the research is classified as analytical since the behavior of the geoservices was examined with the purpose of detecting possible relationships among them. Finally, the level of this research is explanatory since it is oriented to establishing the cause-effect relationships between the variables analyzed from the results obtained through the experiment.

2.2 Collection of information

75 queries were defined in the Common Query Language (CQL) for the Geoserver geoservice. The 75 queries were then translated into the format required by the GeoToroTur geoservice. Of the 75 queries, 25 perform operations of different access patterns and levels of complexity associated with geographic processes, such as: intersections, within, near, and crosses. The remaining 50 only perform basic retrievals of the geographic tables (see Table 1). The above, with the objective of evaluating the behavior of the engines both in simple data recovery transactions as well as in the application of filters and geographic processing.

Each test consisted of obtaining the response time of the execution of the 75 queries. This process was repeated 35 times in each geoservice, with the aim of minimizing the variation in response times caused by the assignment of operating system processes.

For the data collection process, the JMeter tool was used, which allows the measurement of the behavior of both geoservices according to the process described above (Apache Software Foundation, 2022). This application has a simple graphic user interface, offers a great capacity for load generation, is open source and implemented in Java (Patel et al., 2014). Likewise, it is an environment that allows the control of variables; that is, the operations are designed and managed by the test team, and the database used corresponds to a real data sample of the project. To evaluate the results, the component provided by the tool was used, which is called Summary Report, and allows the results of the test performed to be displayed in a table. The datasets presented by this component are (F. J. Díaz et al., 2008):

- Label: sample label
- # Samples: the number of threads used
- Average: average response time in milliseconds for a set of results.
- Min: the minimum time it takes a thread to access a page.
- Max: the maximum time it takes a thread to access a page
- Performance: performance measured in requirements per second / minute / hour.
- Kb/sec: performance measured in Kilobytes per second.
- Average in bytes: average server response size (in bytes).

Table 1. Types of queries.

Types of queries	Description
Geographic Filter	Operations of different access patterns and levels of complexity associated with geographical processes are performed, such as: intersect, within, near and crosses.
Filter on descriptive attributes	Within this type, basic retrieval queries of geographic tables or information retrieval based on their descriptive attributes were performed.

The data set used for the execution of the tests corresponds to the North Huetar Region of Costa Rica, contemplating the geometric structures of points, lines, or polygons that correspond to vectorial data. It worked with the 61 geographical data layers that are published on the IDEHN website (Infraestructura de Datos Espaciales de la Región Huetar Norte) (Instituto Tecnológico de Costa Rica, 2022). Subsequently, these geographic data layers were converted to GeoJSON format using the QGIS tool (Internet Engineering Task Force, 2016; QGIS, 2022). Then, the data was loaded into the MongoDB database and synchronized to PostgreSQL using the ToroDB tool (8Kdata, 2016) coordinate reference system associated with the SRID (Spatial Reference System Identifier) number 4326. Also, once we were done importing the dataset into MongoDB, we moved on to making the spatial indexes for the geometry fields.

The disk space used by the dataset on each platform, as well as the GeoJSON source and shapefiles, are presented in Table 2. The Geoserver database is considerably smaller in size than the hybrid database. The hybrid database in PostgreSQL has 294 tables and 61 materialized views (one view per geographic data layer). While the hybrid database in MongoDB has 61 collections and 72 geographic indexes.

Table 2. Disk space of the data set

Storage type	Data	Index
PostgreSQL Geoserver Database	147M	17M
PostgreSQL Hybrid Database	1268M	1730M
MongoDB Hybrid Database	395M	4M

Regarding the geographic data layers used, the representations used were points, lines, and polygons. The Forest Cover layer of the Northern Huetar Region for the year 2005 is the largest, since it has 21616 records. Another of the layers that has an important size is the contour lines layer, with 15552 records.

2.3 Test environment

There is a test server on which the two geoservices are installed. Its main features are an Intel core i7 processor with the Ubuntu 16.04 LTS 64-bit operating system and 32 GB of RAM. In addition, the versions of the DBMS are MongoDB Server 3.4.10 (MongoDB, 2015), and PostgreSQL 9.6 (The PostgreSQL Global Development Group, 2022). Also, version 1.0.0 of ToroDB Stampede was used. Finally, the version of the tool used for the evaluation and comparison of performance was Apache JMeter 3.2 (Apache Software Foundation, 2022). The drivers used for the connection with each DBMS are PostgreSQL 42.1.4 JDBC driver, MongoDB 2.11.3 Java driver.

Similarly, for the execution of performance tests in the two geoservices, we used a computer with an Intel core i7 processor, Windows 10 operating system 64 bit and 16 GB of RAM.

The connection used was wireless with an estimated speed of 100 MBPS/5 MBPS.

2.4 Statistical analysis

The analysis was realized for the quantitative variable response time by geoservice, evaluating for each case the normality criterion using the Anderson-Darling statistical test (Anderson & Darling, 1952, 1954). Also, the homogeneity of variances was verified by means of the Levene test and the Fisher exact test to determine that the samples are independent (Gastwirth et al., 2009; Raymond & Rousset, 1995). These tests are input for the combined analysis of variance (ANOVA) of a factor for independent samples (Terrádez & Juan, 2003). Finally, the Tukey test and the descriptive statistics for the geoservice response time were calculated (Abdi & Williams, 2010). All tests were validated with a significance level of = 0.05.

3 Results and discussion

3.1 GeoToroTur architecture

An OWS was implemented for the construction of this new geoservice (see Figure 1). This is because it allows a set of configured information resources (a set of services) to be passed between applications. This geoservice stores the data in a hybrid and distributed database and uses a synthetic-semantic analyzer or parser to talk to the two database management systems that integrate the hybrid and distributed database.

Figure 1. GeoToroTur Geoservie.

For the development of the hybrid database, a systematic mapping was made based on empirical studies with an open-source DBMS that allows an integration of the NoSQL paradigms with the PostgreSQL database engine. The systems ArangoDB, CouchBase, and MongoDB were tested using the tools Apache JMeter and Yahoo! Cloud Services Benchmark (Trevino Villalobos et al., 2018; Treviño Villalobos et al., 2020; Treviño-Villalobos et al., 2019). From the evaluation process, MongoDB was identified as the best performing DBMS of the three evaluated, so the database was built using PostgreSQL and MongoDB. Subsequently, it was necessary to identify the performance of each DBMS (MongoDB and PostgreSQL) according to the type of query for the selection of the DBMS responsible for executing the query (Treviño Villalobos et al., 2020).

The implementation of the hybrid and distributed database has a sense of synchronization from MongoDB to PostgreSQL, taking advantage of the ToroDB tool that performs a decomposition process from documents to relational tables. To unify the data in a single table, triggers were implemented to keep the data synchronized in materialized views that are consumed by the GeoToroTur geoservice (Herrera-Ramírez et al., 2021).

The geoservice involved the development of a synthetic-semantic analyzer or parser (see Figure 2), with the purpose of translating the specific query language with the JSON format that it receives from GeoToroTur to another intermediate type (in this case, to SQL and NoSQL languages).

The parser consists of a lexical analyzer that reads the characters to group them into meaningful sequences; a syntactic analyzer that obtains a string of tokens from the lexical analyzer to verify that the string of names can be generated by the grammar of the source language; and a semantic analyzer that uses a syntactic tree and the information in the symbol table to check the semantic consistency (A. Díaz & Granados, 2018). To define whether the translation is into the SQL language or NoSQL, the response times of PostgreSQL and MongoDB were analyzed by query type. In this way, it was decided that the geographic queries would be translated into the SQL language and the rest into the NoSQL language (Treviño Villalobos et al., 2020). Figure 3 shows the integration of the geoservice developed with Geoserver. The result of the developed geoservice is a GeoJSON format file.

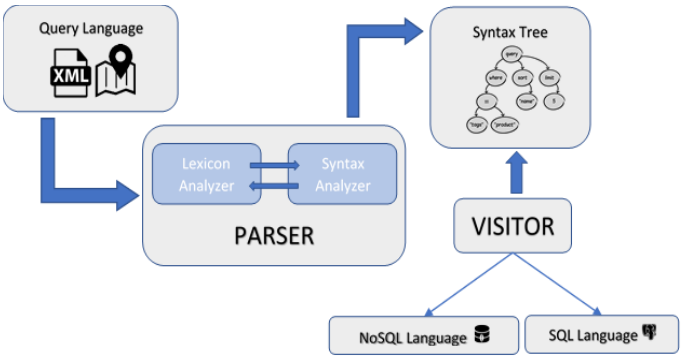

Figure 2. Parser for GeoToroTur Geoservice.

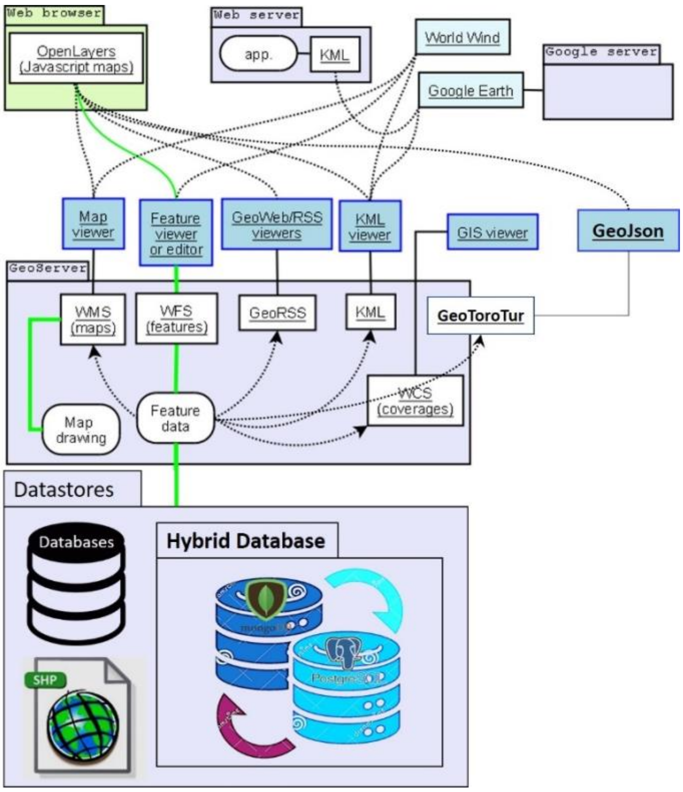

Figure 3. Geoserver architecture with GeoToroTur geoservice

3.2 Evaluation

The response time variable quantitative for the Geoserver and GeoToroTur geoservices is normal as the p-value for the Anderson-Darling test is 0.6581 and 0.3786 respectively. The Levene test to determine the homogeneity of variances in both geoservices yielded a p-value of 0.746, so the data is homogeneous. The p-value for Fisher's exact test is 2.2e-16, which means that the samples are independent.

In Table 3, the ANOVA results for the geoservices response time variable are shown. The p-value calculated for the geoservices is lower than the significance level, so the null hypothesis of equality of means is rejected, and it is admitted that there are significant differences in the response time between the geoservices. Furthermore, according to Tukey's test, it is observed that there is a difference of -21.89829 milliseconds between the GeoToroTur and Geoserver averages (see Table 4). Finally, Table 5 shows that the GeoToroTur mapping web service has a shorter response time than the Geoserver.

Tabla 3. ANOVA results for the geoservices response time variable.

	Sum of squares	DF	F	Pr(>F)
Geoservice	74.50	1	380.7	1.48e-13
Residual	3.52	18	-	-

Tabla 4. Tukey’s test results for the geoservices response time variable.

Response time	Mean difference I-J	P	Confidence interval	
			Upper limit	Lower limit
GeoToroTur-Geoserver	-21.89829	0	-25.3976	-18.39897

Tabla 5. Mean and standard deviation for the geoservices response time variable.

Geoservice	Mean	Standard deviation
Geoserver	308.4946	7.139
GeoToroTur	286.5963	7.527

4 Conclusions

With the development of this research, it was possible to obtain a robust and scalable architecture for a hybrid and distributed database that was implemented with PostgreSQL and MongoDB, making use of ToroDB for document replication. In addition, a geoservice was implemented that consumes the information in that database and is integrated with the GeoServer map server.

The main result achieved was the demonstration, from the statistical point of view, that there is a significant difference between the response times of the GeoToroTur and Geoserver geoservices. It was also found that the GeoToroTur geoservice responds to requests faster than the Geoserver.

At the time of execution of the tests, some limitations of the GeoToroTur geoservice were identified in the execution of the queries, among which the transformation of coordinates between reference systems in an explicit manner and the fact that the limit function was not incorporated.

The unidirectional synchronization of MongoDB to PostgreSQL with ToroDB generates an important process for the restructuring of documents into relational tables and a duplication of information in PostgreSQL. Since this problem was solved by using materialized views, it is suggested that in the future, synchronization should be done from PostgreSQL to MongoDB.

In this study, only the geoservice was evaluated with vector data and query operations, so in future research, other types of transactions such as modifications and deletions should be evaluated.

Statement of conflict of interest

The authors declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Marlen Treviño-Villalobos https://orcid.org/0000-0002-1135-0650
Leonardo Víquez-Acuña https://orcid.org/0000-0002-0604-7219
Rocío Quirós-Oviedo https://orcid.org/0000-0003-1532-1512
Oscar Víquez-Acuña https://orcid.org/0000-0001-8896-4813

References

8Kdata. (2016). *ToroDB*. https://www.torodb.com/

Abdi, H., & Williams, L. J. (2010). Tukey's honestly significant difference (HSD) test. *Encyclopedia of Research Design*, *3*(1), 1–5.

Agarwal, S., & Rajan, K. S. (2016). Performance analysis of MongoDB versus PostGIS/PostGreSQL databases for line intersection and point containment spatial queries. *Spatial Information Research*, *24*(6), 671–677. https://doi.org/10.1007/s41324-016-0059-1

Anderson, T. W., & Darling, D. A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, *23*(2), 193–212.

Anderson, T. W., & Darling, D. A. (1954). A Test of Goodness of Fit. *Journal of the American Statistical Association*, *49*(268).

Apache Software Foundation. (2022). *Apache JMeter™*.

Apostolopoulos, K., Kakaletris, G., & Koltsida, P. (2019). Extending OGC Standards for Supporting Big-Earth Data Retrieval and Analytics. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 5559–5562. https://doi.org/10.1109/IGARSS.2019.8897888

Bai, J.-W., Wang, J.-Z., & Huang, J.-L. (2013). Spatial Query Processing on Distributed Databases. In *Advances in Intelligent Systems and Applications* (Vol. 1, pp. 251–260). https://doi.org/10.1007/978-3-642-35452-6_27

Chopade, Mrs. R. M., & Dhavase, N. S. (2017). Mongodb, couchbase: Performance comparison for image dataset. *2017 2nd International Conference for Convergence in Technology (I2CT)*, 255–258. https://doi.org/10.1109/I2CT.2017.8226131

Colorado Pérez, M. A. (2017). NoSQL: ¿es necesario ahora? . *TIA*, *5*(2), 174–179.

Díaz, A. (2009). *Diseño estadístico de experimentos* (Segunda). Universidad de Antioquia.

Díaz, A., & Granados, A. (2018). Analizador sintáctico-semántico para la enseñanza de la programación de software. *Revista Tecnocientífica URU*, *14*, 11–18.

Díaz, F. J., Banchoff Tzancoff, C. M., Rodríguez, A. S., & Soria, V. (2008). Usando Jmeter para pruebas de rendimiento. *XIV Congreso Argentino de Ciencias de La Computación*.

European Parliament. (2007). *Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)*.

Garcia, R., de Castro, J. P., Verd, E., Jess, M., & Mara, L. (2012). Web Map Tile Services for Spatial Data Infrastructures: Management and Optimization. In *Cartography - A Tool for Spatial Analysis* (pp. 26–48). InTech. https://doi.org/10.5772/46129

Gastwirth, J. L., Gel, Y. R., & Miao, W. (2009). The Impact of Levene's Test of Equality of Variances on Statistical Theory and Practice. *Statistical Science*, *24*(3), 343–360. https://doi.org/10.1214/09-STS301

Goyal, S., Srivastava, P. P., & Kumar, A. (2015). An overview of hybrid databases. *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 285–288. https://doi.org/10.1109/ICGCIoT.2015.7380474

Gui, Z., Cao, J., Liu, X., Cheng, X., & Wu, H. (2016). Global-Scale Resource Survey and Performance Monitoring of Public OGC Web Map Services. *ISPRS International Journal of Geo-Information*, *5*(6), 88. https://doi.org/10.3390/ijgi5060088

Gunawan, R., Rahmatulloh, A., & Darmawan, I. (2019). Performance Evaluation of Query Response Time in The Document Stored NoSQL Database. *2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering*, 1–6. https://doi.org/10.1109/QIR.2019.8898035

Herrera-Ramírez, J. A., Treviño-Villalobos, M., & Víquez-Acuña, L. (2021). Hybrid storage engine for geospatial data using NoSQL and SQL paradigms. *Revista Tecnología En Marcha*, *34*(1), 40–54. https://doi.org/10.18845/tm.v34i1.4822

Huang, C.-Y., & Chang, H. (2016). GeoWeb Crawler: An Extensible and Scalable Web Crawling Framework for Discovering Geospatial Web Resources. *ISPRS International Journal of Geo-Information*, *5*(8), 136. https://doi.org/10.3390/ijgi5080136

Instituto Tecnológico de Costa Rica. (2022). *IDE Región Huetar Norte*. https://www.tec.ac.cr/

Internet Engineering Task Force. (2016). *The GeoJSON Format*. https://www.rfc-editor.org/rfc/rfc7946

Kepka, M., & Ježek, J. (2013). Web client for PostGIS—the concept and implementation. *Geoinformatics FCE CTU*, *11*, 63–76. https://doi.org/10.14311/gi.11.5

Kumar, K. B. S., Srividya, & Mohanavalli, S. (2017). A performance comparison of document oriented NoSQL databases. *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 1–6. https://doi.org/10.1109/ICCCSP.2017.7944071

Loechel, A., & Schmid, S. (2013). Comparison of different caching techniques for high-performance web map services. *International Journal of Spatial Data Infrastructures Research*, *8*, 43–73.

Miao, L., Jing Guo, Wenchao Cheng, & Zhou, Y. (2016). A novel model to support OGC Web Services semantic search using OWL-S. *2016 24th International Conference on Geoinformatics*, 1–4. https://doi.org/10.1109/GEOINFORMATICS.2016.7578973

MongoDB. (2015). *The MongoDB 3.4 Manual*. https://www.mongodb.com/docs/v3.4/

Moreno Jiménez, A. (2004). Nuevas tecnologías de la información y revalorización del conocimiento geográfico. *Geo Crítica / Scripta Nova. Revista Electrónica de Geografía y Ciencias Sociales*, *VIII*(170). http://www.ub.edu/geocrit/sn/sn-170-62.htm

OASIS. (2005). *Quality Model for Web Services*.

Open Geospatial Consortium. (2000). *OpenGIS® Web Map Server Interface Implementation Specification. Revision 1.0.0*.

Open Source Geospatial Foundation. (2022). *Geoserver*. https://geoserver.org/

Patel, B., Parikh, J., & Shah, R. (2014). A review paper on comparison of SQL performance analyzer tools: Apache JMeter and HP LoadRunner. *International Journal of Current Engineering and Technology*, *4*(5), 3642–3645.

POSTGIS. (2022). *About PostGIS*. https://postgis.net/

Pramukantoro, E. S., Primanita Kartikasari, D., & Siregar, R. A. (2019). Performance Evaluation of MongoDB, Cassandra, and HBase for Heterogenous IoT Data Storage. *2019 2nd International Conference on Applied Information Technology and Innovation (ICAITI)*, 203–206. https://doi.org/10.1109/ICAITI48442.2019.8982159

Puangsaijai, W., & Puntheeranurak, S. (2017). A comparative study of relational database and key-value database for big data applications. *2017 International Electrical Engineering Congress (IEECON)*, 1–4. https://doi.org/10.1109/IEECON.2017.8075813

QGIS. (2022). *Bienvenido Al Proyecto QGIS!* https://www.qgis.org/es/site/

Raymond, M., & Rousset, F. (1995). An exact test for population differentiation. *Evolution*, *49*(6), 1280–1283. https://doi.org/10.1111/j.1558-5646.1995.tb04456.x

Růžička, J. (2016). Comparing speed of Web Map Service with GeoServer on ESRI Shapefile and PostGIS. *Geoinformatics FCE CTU*, *15*(1), 3–9. https://doi.org/10.14311/gi.15.1.1

Sani, A., & Rinner, C. (2011). A Scalable GeoWeb Tool for Argumentation Mapping. *GEOMATICA*, *65*(2), 145–156. https://doi.org/10.5623/cig2011-023

Schmid, S., Galicz, E., & Reinhardt, W. (2015). WMS performance of selected SQL and NoSQL databases. *International Conference on Military Technologies (ICMT) 2015*, 1–6. https://doi.org/10.1109/MILTECHS.2015.7153736

Schmid, S., & Reinhardt, W. (2015). Towards a general evaluation procedure for Geo Web Services. *Proceedings of the ICC 2015 International Cartographic Conference*, 23–28.

Shen, L., Duan, W., Ren, Y., & Yang, C. (2010). Management Service on WMS/WFS services. *2010 18th International Conference on Geoinformatics*, 1–5. https://doi.org/10.1109/GEOINFORMATICS.2010.5567481

Terrádez, M., & Juan, A. A. (2003). *Análisis de la varianza (ANOVA)*. http://www.uoc.edu/in3/emath/docs/ANOVA.Pdf

The PostgreSQL Global Development Group. (2022). *PostgreSQL: The World's Most Advanced Open Source Relational Database*. https://www.postgresql.org/

Treviño Villalobos, M., Víquez Acuña, L., & Quirós Oviedo, R. (2020). Comparison of the Response Times of MongoDB and PostgreSQL According to Type of Query in Geographical Databases. *Computación y Sistemas*, *24*(4), 1461–1469. https://doi.org/10.13053/cys-24-4-3292

Trevino Villalobos, M., Viquez Acuna, L., Quiros Oviedo, R., & Esquivel Vega, G. (2018). Una comparación de rendimiento entre MongoDB, ArangoDB y CouchBase para la operación lectura sobre bases de datos geográficas [Not available in English]. *2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII)*, 1–6. https://doi.org/10.1109/CONCAPAN.2018.8596387

Treviño-Villalobos, M., Víquez-Acuña, L., Quirós-Oviedo, R., & Esquivel-Vega, G. (2019). Una comparación de rendimiento entre bases de datos NoSQL: MongoDB y ArangoDB. *Revista Tecnología En Marcha*. https://doi.org/10.18845/tm.v32i6.4223

Veenendaal, B., Brovelli, M. A., & Li, S. (2017). Review of Web Mapping: Eras, Trends and Directions. *ISPRS International Journal of Geo-Information*, *6*(10), 317. https://doi.org/10.3390/ijgi6100317

Wenjue, J., Yumin, C., & Jianya, G. (2004). Implementation of OGC web map service based on web service. *Geo-Spatial Information Science*, *7*(2), 148–152. https://doi.org/10.1007/BF02826653

Wu, C., Zhu, Q., Zhang, Y., Du, Z., Ye, X., Qin, H., & Zhou, Y. (2017). A NoSQL–SQL Hybrid Organization and Management Approach for Real-Time Geospatial Data: A Case Study of Public Security Video Surveillance. *ISPRS International Journal of Geo-Information*, *6*(1), 21. https://doi.org/10.3390/ijgi6010021

Yang, P., Cao, Y., & Evans, J. (2007). Web Map Server Performance and Client Design Principles. *GIScience & Remote Sensing*, *44*(4), 320–333. https://doi.org/10.2747/1548-1603.44.4.320

Yuan, J., Yue, P., & ong, J. (2009). Integration of geospatial Web services and Web portal technologies for geospatial Information sharing and processing. *2009 17th International Conference on Geoinformatics*, 1–4. https://doi.org/10.1109/GEOINFORMATICS.2009.5293518

Zhang, X., Song, W., & Liu, L. (2014). An implementation approach to store GIS spatial data on NoSQL database. *2014 22nd International Conference on Geoinformatics*, 1–5. https://doi.org/10.1109/GEOINFORMATICS.2014.6950846

Zhang, Y., Li, D., & Zhu, Z. (2008). A Server Side Caching System for Efficient Web Map Services. *2008 International Conference on Embedded Software and Systems Symposia*, 32–37. https://doi.org/10.1109/ICESS.Symposia.2008.39

Zhao, F., Zhang, J., & Cao, D. (2006). Dynamic Web Map Service for Web Publishing System of Mass Remote Sensing Images. *2006 IEEE International Symposium on Geoscience and Remote Sensing*, 858–860. https://doi.org/10.1109/IGARSS.2006.220