ISSN 2007-9737

Framework for Heterogeneous Data Management: An Application Case in a NoSQL Environment from a Climatological Center

Alicia Margarita Jiménez-Galina*, Aide Aracely Maldonado-Macías,
Karla Miroslava Olmos-Sanchez, Israel Hernández,
Fernando Estrada-Saldaña, Felipe Adrián Vázquez-Gálvez

Universidad Autónoma de Ciudad Juárez,
Chihuahua,
Mexico

al206560@alumnos.uacj.mx, {amaldona, kolmos,
israel.hernandez, festrada, fvazquez}@uacj.mx

Abstract. Processing, visualizing and understanding data from meteorological networks can present several challenges due to the variety and complexity of the data and must be accessible in real time and in different formats, protocols and standards. This paper presents the development of an innovative technological framework for handling heterogeneous climatological data in a NoSQL environment. The framework was developed following the Action Research methodology and enables the extraction of heterogeneous data, their homogenization, and the creation of a dataset. Its real-case application took place in data repositories used for climatological data management in a specialized regional center in Ciudad Juarez, México. The main repository use MongoDB and contain 631,202 documents with data from several meteorological stations. A 70% reduction in data processing time is evidence that the methodology and framework developed were effective in the case of the application. In addition, the generated data sets are homogenized and in formats compatible with advanced analysis tools.

Keywords. Heterogeneous data, homogenization, action research.

1 Introduction

Nowadays, organizations around the world generate information resulting from diverse activities; such information is composed of different kinds of data. This data can be structured, unstructured, or semi-structured, but it is mostly heterogeneous and comes from several sources.

Proper data processing and understanding are crucial as they lead to better predictions and decision making. Therefore, companies need to handle data efficiently, effectively, and reliably, seeking to use low-cost and optimization alternatives to safeguard them [17, 24].

1.1 Big Data Solutions for Data Management and Migration

One alternative to solve these organizational needs can be found in relational database systems, which have been used for more than 40 years for similar purposes [3].

Although currently storing large amounts of information in relational database systems is inexpensive, such systems have limitations in instances that require handling unstructured data and horizontal scaling, which makes it impossible to partition data on different computers.

That is why new technologies have emerged to help manage data effectively, for example MapReduce / Hadoop, and NoSQL, among others [4]. Likewise, cloud computing and new big data system-related applications have appeared, both of which can handle massive data, thus allowing organizations to improve their understanding of stored information.

Big data systems have the infrastructure, technology, and service capacities to manage large amounts of data.

Manage implies entry, storage, analysis, search, exchange, and transfer of data. Furthermore, data visualization, consultation, and actualization are important in maintaining their privacy, origin, veracity, and value of data [4].

Among these technologies, NoSQL databases provide flexible structures and enable the horizontal scaling of large amounts of data and users. Unstructured databases can be displayed in several forms, for example documents, key-values, column-widths, and graphs.

The best-known NoSQL database of the document type is MongoDB, which, in addition to other features, stores data such as objects in a JSON (JavaScript Object Notation) format and allows each record to become a document with an independent structure from the others [3].

The tendency in organizations is to take advantage of these new information management technologies to enhance their decision-making processes and promote data-based innovative solutions [7]. However, for those cases where data is scattered though different repositories and relational databases, information management process can be difficult and laborious.

Currently, it is necessary to migrate information to unstructured schemes in an optimal way while preserving the basic principles of integrity, confidentiality, and availability. Thus, migration from a structured database like SQL to an unstructured NoSQL like MongoDB is increasingly frequent and necessary [21, 31, 1].

However, the process must be careful and efficient since it is time-consuming and, most of the time, underestimated [27]. In addition, it requires exhaustive data origin analysis [8]. Some cases have been found where from data contained in the SQL database, a record migrated efficiently as a document to MongoDB [3, 10].

Therefore, effective data migration requires the use of data mining techniques such as ETL (Extract-Transform-Load) processes, which are useful and convenient for data source analysis and which also make the cleaning, transforming, or reformatting processes possible [12, 13, 27].

Other advantages of ETL processes are the establishment of a central repository, as well as decision-making processes based on the analysis of data concentrated in a new database. They also aid in various processes such as data migration between different applications, as well as their synchronization and consolidation [22]. These processes extract data from one or more sources, transform them or clean them if necessary, and load them into another database, called Data Warehouse (DW), for later analysis.

However, migration processes also present difficulties; among them are the migration of structured SQL databases to unstructured NoSQL in MongoDB [1], the homogenization of heterogeneous data [17], and the creation of a dataset from MongoDB databases [5].

1.2 Solutions for Migrating Data Between Databases

Regarding the difficulty of migration between databases, some authors have proposed some solutions to improve the implementation of ETL processes [13]. These authors have also carried out in-depth studies on aspects such as elasticity, dynamism, and the cost of resources.

Additionally, they have analyzed ETL solutions for the domain of big data in the cloud through task or programming parallelization [12] and have proposed an alternative solution based on a new architecture that eliminates the “buffer zone” to cut storage space in half, in addition to reducing data-processing time.

Further solutions have been proposed where semantic technology techniques were used based on data in the cloud and big data characteristics such as speed, variety, and volume [13]. Finally, some solutions proposed improvements to the ETL process by combining the Query Cache and Scripting methods [27].

1.3 Miscellaneous Data-Management Studies

This section presents several studies on heterogeneous data management, SQL queries, the use of metadata, and other frameworks developed to migrate and homogenize data. Regarding solutions for managing heterogeneous data, the literature has shown efficient use of framework development.

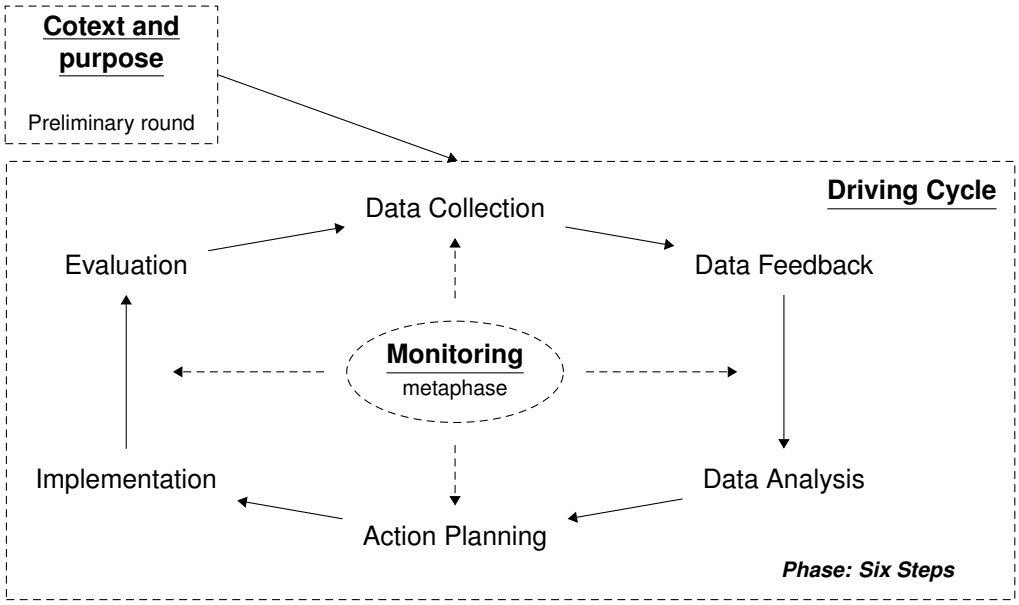

Fig. 1. Action research cycle [14]

For example, the HDS Analytics framework created a heterogeneous dataset to feed an analysis model that located the shortest route in a public transport domain [18].

Another framework was developed to detect medical events and create trends by correlating physical sensors such as temperature, air pressure, wind, and rain with suspended particles and a social sensor [11].

One further study [32] referred to a semi-structured query engine through which SQL queries were optimized according to the model. In addition, several proposals were found for the handling of non-relational data, and the use of metadata and their integrity.

For example, [30] managed data effectively through the use of an R-tree structure for operations in MongoDB. Another solution proposed by [23] improved the dataset homogenization process by incorporating metadata to optimize queries and data integration.

Finally, [20] developed a solution in order to take care of information integrity, which created a framework that, together with the metadata, enabled the extraction of the information to be analyzed.

Finally, some authors [19] presented a project created in a NoSQL environment, whose process of quality evaluation, homogenization, and visualization of climatological data precedes the development of this framework.

As can be seen, the implementation of new technologies in the areas of dataset migration, homogenization, and generation contributes to a better understanding of data. However, special care must be taken during data source analysis to identify valuable content and convert it to a JSON document for effective migration to MongoDB.

1.4 Action Research Methodology

The Action Research (AR) methodology aims to address a problem in an organization, whether it relates to a research topic or an organizational challenge, and solve it in a cooperative and participatory way [15, 26, 6].

Another research study added the participatory and simultaneous elements to the characteristics of the action looking for innovative solutions [9]. As shown in Figure 1, the AR methodology consists of a preliminary round that includes the driving cycle and the monitoring metaphase [14].

In the preliminary round, the objectives and the context are established and understood. Then, the Driving Cycle takes place; it involves a six-step phase that focuses initially on data and then on the action.

Thus, it first collects, gives feedback on, and analyzes data, and then plans, implements, and evaluates the action. The monitoring metaphase is the follow-up phase, in which the results of each of the steps are verified.

1.5 Paper Contribution and Organization

The literature review shows that, thus far, only partial solutions for migration processes have been offered. Thus, the development of a complete and comprehensive solution can be considered an open problem or an opportunity for innovation.

Because innovations in these processes are necessary for a better understanding of data, this paper presents a development of an innovative technological framework applied to an environmental data and information management case with the following characteristics:

Handling of heterogeneous data in a NoSQL environment, an initial storage procedure, data extraction and transformation methods, and dataset creation in three different formats for subsequent analysis.

The framework was developed using the Action Research methodology, which has the advantages of providing effective solutions for improving processes, practices, and strategies [15, 26, 6, 9, 14]. This paper is organized as follows.

The introduction includes the problem statement and the literature review. Section 2 describes the methodology used for the creation of the framework, as well as the use of other studies.

Section 3 explains the case of application in the climatological center, including the context and purpose, and an explanation of the driving cycle with its five processes: metadata, integral solution, uploading, development, and evaluation. Finally, Section 4 discusses the conclusions as well as some future research initiatives.

2 Methodology

This section describes how the AR methodology was used, as well as complementary studies in the real application case for the framework developed.

2.1 Application of the AR Methodology

The implementation of the AR methodology in the development of the framework took place in two main parts. Part one consisted of the preliminary round, which included the context and the purpose. Part two consisted of the driving cycle, composed of the six steps in the monitoring metaphase.

The monitoring metaphase supervised and verified each of the steps in collaboration with the experts. The application and results of the steps in the application case in the climatological center are also described.

2.2 Miscellaneous Proposals for Data Management

In addition to the AR methodology used, this section presents some proposals for managing heterogeneous data that were considered for the framework development: Investigations related to ETL processes were used at different stages of the framework for cleaning, loading, and transforming data; during the phase of initial loading of SQL to MongoDB; and later in the homogenization and dataset creation phase [12, 13].

The investigations by [18, 11, 32] were taken into consideration to improve heterogeneous data understanding and management, while the studies by [30, 23] influenced the development of the structure and use of metadata to support the management framework.

3 Application Case: Climatological Center

This section explains the implementation of the two main parts of the AR methodology in the application case.

3.1 Part One. Context and Purpose

The importance of having historical climate bases has been highlighted by several authors. Some authors, promoted their use in order to improve climate predictions [16]; others proposed them to support agriculture [29]; some others used them to analyze energy, health, and insurance [28]; and others analyzed the impact of climatic variability on natural gas [25].

The case chosen for the application of this framework was the Centro de Estudios Atmosféricos y Tecnologías Verdes, CECATEV (Center for Atmosphere Studies and Green Technologies, for its Spanish acronym), which is located at the Universidad Autónoma de Ciudad Juárez (Autonomous University of Ciudad Juarez, UACJ for its Spanish acronym) as part of a collaboration agreement between the UACJ and the Instituto Nacional de Ecología y Cambio Climático (National Institute for Ecology and Climate Change).

CECATEV was created as a scientific reference laboratory for the Ciudad Juarez atmospheric basin air quality program and oversees the maintenance of the climatological network as well as the study of air pollution. CECATEV has worked on different projects to increase the meteorological stations in the city. To carry out its work, the center must create big data systems to concentrate the climate variables in the region's climatological network in databases.

These meteorological data are temperature, direction, wind speed, relative humidity, evaporation, rainfall, and solar radiation. Once the information is gathered, it must be shared with experts, different users, and several universities inside and outside the country.

This application case was carried out using a central repository in MongoDB containing 631,202 documents with data from five meteorological stations and one station for gases and suspended particles. To fully understand the impact of this application case, the following sections will describe its manual process as well as the problem studied.

3.1.1 Manual Process Description

The following manual process was carried out in the meteorological station: Every day, users downloaded files in csv (Comma Separated Values) format from a repository on the web.

Then, they conducted a data cleaning, or pre processing procedure, to eliminate hyphens, invalid characters, and non corresponding columns; this process was carried out in an Excel file. Once data was pre-processed, it was loaded onto a tool called "R" used to create graphs and analyze data.

If users found any human error at any step of data processing, they had to restart, which delayed the process. Another way to develop the dataset was to run a query directly on the SQL server, yet this put the integrity of data at risk due to direct manipulation.

In addition, the staff lacked the knowledge to generate SQL queries and troubleshoot any kind of errors. After the query was successfully generated, it was exported to a csv file and users could go through the cleanup process described above.

The previous cleaning and loading processes also applied to the gas and suspended particles station, except for the generation of the file since the laboratory staff would have had to enter the CECATEV site to access the server and download the files, and that would have represented a risk to physical security and data integrity.

Thus, the creation of a dataset of non homogenized data from a station considering one month of data took about 40 to 60 minutes.

3.1.2 Problem Description and Purpose

There were different problems in the manual process described, such as the time used, the risk of integrity, the management of heterogeneous data and the number of stations.

With respect to time and integrity, by including information from specific sensors and multiple stations, it involved several complex manual processes that required a lot of time and represented a significant risk factor to data integrity.

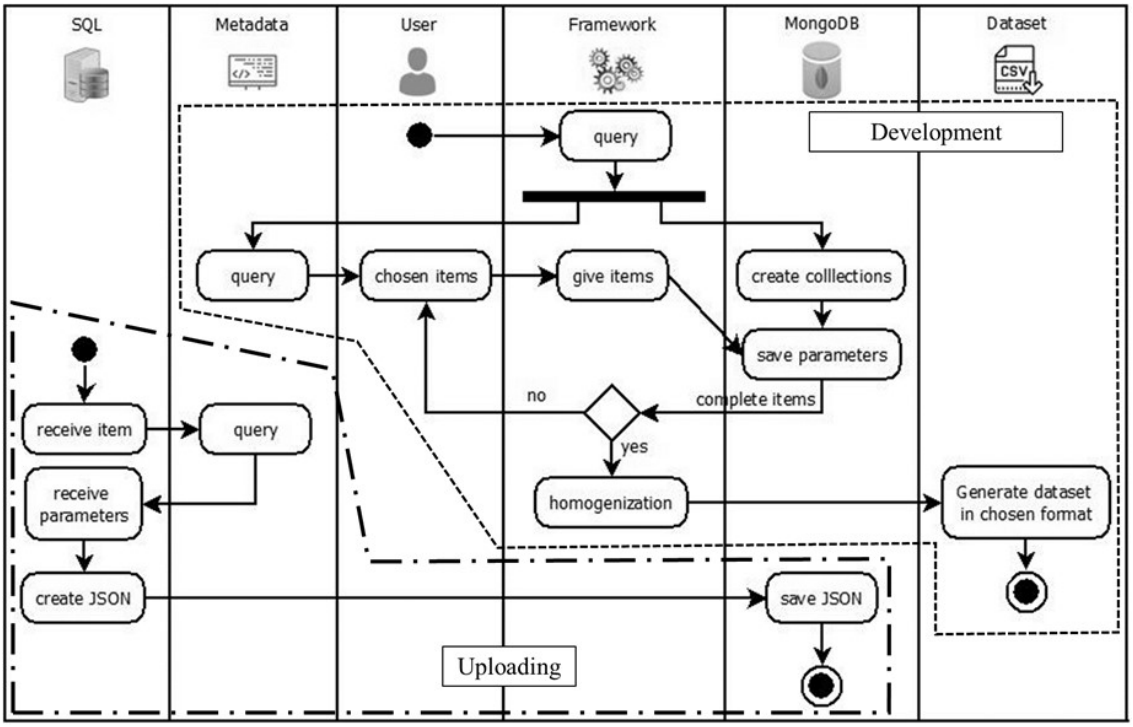

Fig. 2. Diagram of the solution

Regarding the heterogeneity of the data, it is due to various factors, such as the weather network is made up of weather stations of different brands and models. Another factor is that the stations for gases and suspended particles are of different types.

In addition, each of the stations can contain a different number of sensors and of a different brand; finally, the readings collected may be in different units of measure. Due to these factors, the resulting data set was not homogenized. Finally, this process was carried out individually by weather station.

Meteorological data analysis is used to support decision making, product development and a better understanding of radioactive and contaminating processes in our region, but data visualization has been a constant challenge.

Therefore, it was essential to design a tool that would allow data processing, time minimization, and data homogenization so that they could be assimilated into predictive atmosphere models.

3.2 Part two. Driving Cycle

This section will describe the driving cycle, which includes the 6 steps of the monitoring metaphase that are embedded in the processes. The use of italics emphasizes these steps in the text. The first process describes the definition and construction of the metadata, followed by the development of a comprehensive solution.

The initial data loading is described later, and the development of the framework is explained at the end, along with the evaluation of the framework in the application case.

3.2.1 Metadata

The steps of the monitoring metaphase, data collection, data feedback and data analysis, involved a review of the origin of data, hence it was necessary to migrate it efficiently. The information that is migrated from a relational database to a NoSQL can only include the valuable data instead of the entire record; that is, in NoSQL databases

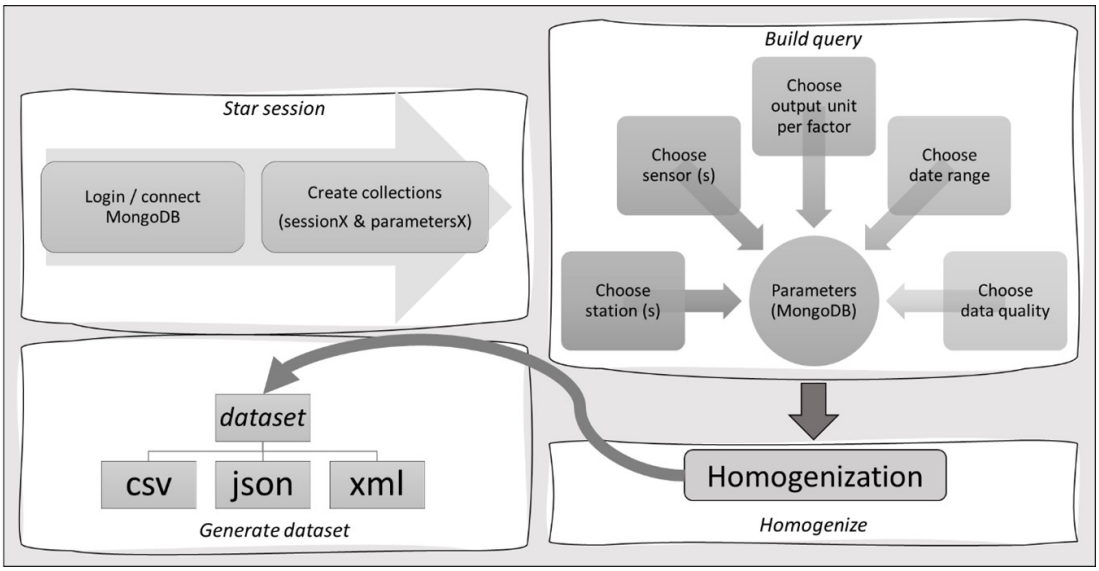

Fig. 3. Framework components diagram

it is possible to have documents with independent structures. That is the reason why the metadata was defined and constructed in XML format. The metadata was made up of four groups.

The first group contained the information related to the server; the second group was formed by the conversion factors, which made it possible to choose the output unit to homogenize the sensor values; the third group included the stations' profile; and the fourth group contained the sensors belonging to each station.

The use of metadata had two purposes: the first was to provide the elements that would make up the structure of the document in JSON, which would be built from the fields with SQL values, inserted into MongoDB. The second purpose was to provide the elements to be included in the dataset, as well as the information needed to homogenize and generate dataset in the output.

In this application case, data collected by the weather stations was stored in a SQL server with a database of 65 fields. However, the stations had an average of 19 sensors taking readings, therefore there was storage waste. To solve this, the metadata provided by the elements was used to build the JSON document for migration.

3.2.2 Integral Solution

In the action planning and implementation steps, the integral solution was designed. It was made up of the process of uploading SQL to MongoDB (see section 3.2.3) and the development of the framework (see section 3.2.4). Figure 2 shows the activity diagram of the solution.

3.2.3 Uploading

This process was carried out collaboratively. The name of the station to migrate was provided. Then the metadata was checked to identify the fields with values. Finally, the query was built to generate the JSON file to load to MongoDB.

3.2.4 Development

This process led to the development of the described framework as a solution to the process of homogenization and generation of datasets in the csv, JSON, and XML formats. The central repository used for this project was a NoSQL database on MongoDB, and the framework was designed using the Python language because of its advantages in the use of mathematical functions and its compatibility with MongoDB [2].

Table 1. Homogenized data processing times per station using the framework

Process of Framework													
							Time in seconds						
Line	Station	# Sensors	Start Date	End Date	Days	Documents MongoDB	Generation Dataset	Generation File:				Total time all files	Total time in minutes
								readme	json	csv	xml		
1	Estacion 05	14	2019-06-27 18:10	2019-10-04 18:05	98.9965278	17,358	75.3255	0.0034	1.9741	0.7102	1.1697	79.1830	1.3197
2	Estacion 05	14	2019-08-04 18:05	2020-01-15 01:05	163.291667	31,936	132.5774	0.0041	3.9222	1.3321	2.6404	140.4762	2.3413
3	Estacion 05	14	2019-06-27 18:10	2020-01-15 01:05	201.288194	39,780	169.1441	0.0030	4.9403	1.6928	3.2616	179.0418	2.9840
4	Estacion 09	19	2019-03-08 19:50	2019-06-08 19:50	92	18,980	79.3514	0.0045	2.5711	0.9402	1.7722	84.6394	1.4107
5	Estacion 09	19	2019-03-08 19:50	2019-09-04 18:05	179.927083	36,914	156.2165	0.0029	5.1794	1.8674	3.2699	166.5360	2.7756
6	Estacion 09	19	2019-03-08 19:50	2019-11-04 18:05	240.927083	54,467	233.9262	0.0029	7.4542	2.6587	4.8130	248.8551	4.1476
7	Estacion 25	19	2019-10-01 00:00	2019-11-01 00:00	31	8,929	40.2444	0.0038	1.3998	0.4805	0.8378	42.9664	0.7161
8	Estacion 25	19	2019-01-01 00:00	2020-01-01 00:00	365	80,353	358.0659	0.0030	11.9192	4.3548	7.7943	382.1372	6.3690
9	**Estacion 25**	**19**	**2017-04-03 20:55**	**2020-01-15 00:40**	**1016.15625**	**194,377**	**861.0886**	**0.0041**	**29.6684**	**10.6237**	**18.9564**	**920.3411**	**15.3390**
10	Estacion 26	14	2019-03-28 19:40	2019-06-28 19:40	92	26,460	114.6024	0.0036	3.1081	1.0885	1.9389	120.7416	2.0124
11	Estacion 26	14	2019-03-28 19:40	2019-10-15 00:40	200.208333	54,119	232.4990	0.0032	6.5583	2.3220	4.1233	245.5058	4.0918
12	Estacion 26	14	2019-03-28 19:40	2020-01-15 00:40	292.208333	79,223	344.7402	0.0034	9.7701	3.3503	6.2492	364.1132	6.0686
13	Estacion 101	19	2018-09-25 20:10	2018-12-31 20:10	97	25,682	113.9750	0.0047	3.7560	1.3825	2.4682	121.5864	2.0264
14	Estacion 101	19	2018-09-25 20:10	2019-05-28 18:40	244.9375	64,921	282.4692	0.0030	9.7583	3.4286	6.3382	301.9973	5.0333
15	Estacion 101	19	2018-09-25 20:10	2019-08-28 18:40	336.9375	90,384	399.1689	0.0039	13.3276	4.8884	8.8727	426.2614	7.1044
16	Teledyne	18	2018-09-01 00:00	2019-03-01 00:00	181	88,379	391.0676	0.0042	8.3926	3.4936	5.2528	408.2109	6.8035
17	Teledyne	18	2018-09-01 00:00	2019-08-15 01:00	348.041667	132,427	589.4341	0.0039	12.3581	5.1125	7.1101	614.0187	10.2336
18	Teledyne	18	2018-09-01 00:00	2020-01-15 01:00	501.041667	172,971	757.7355	0.0044	15.9425	6.4051	9.4482	789.5356	13.1589

The Python license was certified as Open Source[1] and was compatible with the GPL[2]. Figure 3 shows a diagram of the framework's component operation.

The framework was built as a set of libraries, which performed specific functions; thus, when combined, they generated the homogenized output dataset with user selections. The operation was divided into four processes: start session, build query, homogenize, and generate dataset. These processes are detailed below.

Start Session. Initially, on the MongoDB server, the user collection was created to manage users and their working collections, which would be used during their session in the framework.

Later in the start session process, the user was registered and validated; the session was created and closed, and the working collections (sessionX and parametersX) were generated.

These working collections were maintained during the user's session. The parametersX collection stored the user selections in each of the levels represented in the XML metadata; thus, it saved the parameters needed to build the query with which the MongoDB information would be extracted.

The sessionX collection, on the other hand, contained the resulting homogenized dataset to be exported to csv, JSON, or XML. Note: The X at the end of the collections is a random number between 1 and 1000. That number was verified in MongoDB before creating the collections to avoid collisions.

Build Query. Once the session started, the elements to be included in the dataset had to be chosen. The build query process contained the set of libraries that made up data extraction query. Each library displayed each of the groups and subgroups of metadata items to choose from.

This way, the user first selected the stations, then the sensors to include per station, and finally the output's unit of measurement for each group of factors (each sensor belonged to a group of

[1] News from the blog — Open Source Initiative. opensource.org
[2] The GNU General Public License v3.0 - GNU Project - Free Software Foundation. www.gnu.org/licenses/gpl-3.0.html

Table 2. Frame homogenization process times for 6 stations and 103 sensors

Process of Framework												
Line	Start date	End date	Days	Documents MongoDB	Embedded Documents	Time in Feconds						Total time in minutes
						Generation Dataset	Generation File:				Total time all files	
							readme	json	csv	xml		
1	2019-10-01 00:00	2019-11-01 00:00	31.0000	43,295	8,929	41.1790	0.0039	4.8485	1.9920	3.3504	51.3739	0.8562
2	2018-09-01 00:00	2019-03-05 00:00	185.0000	144,188	90,143	391.2329	0.0037	20.6551	15.2575	14.4693	441.6185	7.3603
3	2019-02-08 02:05	2019-11-04 00:00	268.9132	327,738	77,445	361.3077	0.0040	39.0183	16.4658	27.0040	443.7999	7.3967
4	2018-09-25 00:00	2019-11-04 00:00	405.0000	444,441	146,663	664.7569	0.0034	57.4077	28.5741	39.5156	790.2577	13.1710
5	2018-01-27 00:00	2020-01-15 01:00	718.0417	571,358	219,505	991.5286	0.0046	73.1795	39.9118	49.6220	1,154.2465	19.2374
6	**2017-04-03 20:55**	**2020-01-15 01:05**	**1,016.1736**	**631,202**	**279,349**	**1,260.0749**	**0.0039**	**81.1401**	**48.3309**	**55.0631**	**1,444.6129**	**24.0769**
Average			437.3547	360,370	137,006	618.3467	0.0039	46.0415	25.0887	31.5041	720.9849	12.0164

conversion factors). The date range and the quality of data were part of data requested. This is how the selections and parameters were stored in the parametersX collection.

At this point, it was possible to change the chosen elements as many times as the user wished. Finally, data was extracted through the query to go on to the homogenization process.

Homogenization. During this phase, each of the documents extracted was analyzed. The value of the item by chosen quality was used as the input unit to be transformed into the chosen output unit. The homogenized result was stored in the sessionX collection.

Generate Dataset. For the generate dataset process, the user had already selected the output format for the dataset, which could be csv, JSON or XML, and the sessionX information had undergone a dataset construction process into the desired format.

In the generated dataset, the stations were embedded by datetime. Finally, another file was generated along with the dataset. It was a Readme.txt file which contained detailed information on the dataset content.

3.2.5 Evaluation

In the evaluation step, the framework execution times for dataset generation were shown, including their homogenization. As can be seen in line 9 of Table 1, the framework took 15,339 minutes to generate the queries, homogenize data, create the Readme.txt file, and generate the dataset in csv, JSON and XML. All data from Station 25 were included in this process: a total of 19 sensors and 194,377 documents, which corresponded to 1,016.15 days. Table 2 shows the frame run times including all 6 stations, all sensors for each station, and all time periods.

As can be seen in line 6, the total time in minutes used by the framework for the query-making and homogenized dataset generation processes was less than 24.0769 minutes. Although a total of 631,202 documents MongoDB were analyzed, when generating the dataset, only 279,349 embedded documents were created; this is because the documents were aligned by a timestamp.

In order to compare the manual process with the one carried out using the proposed framework, the following aspects were considered: the dataset generation time in minutes, the number of stations, the analysis time in days, and the homogenized data.

For the manual process, it can take 60 minutes to generate a single station dataset including 30 days of non-homogenized data. In contrast, the proposed framework used half the time to generate a dataset for six stations including up to 1,200 days of homogenized data. This represents a significant increase in data throughput.

Once the results were tested, it was observed that the framework is indeed an efficient solution since it decreases the dataset generation times considerably in comparison to the manual process.

In addition, it is a tool that can homogenize data, generate datasets in different formats that can be adapted to other advanced analysis tools, provide a profile of the generated dataset content, maintain data integrity by eliminating direct contact with them, and be implemented in a user-friendly environment.

4 Conclusions and Feature Research

It can be concluded that the methodology and the framework developed were effective in the case of application as they enabled efficient data loading and showed a considerable reduction in processing times while including the homogenization and generation of datasets in formats that are compatible with advanced analysis tools.

Finally, the proposed methodology developed a framework that contributes to several technological aspects, which will be explained in the following paragraphs. The framework provides a methodology for data management, including efficient extraction and loading, as well as for data conversion factors using metadata from an unstructured database.

In this case, MongoDB was used since it takes advantage of a dynamic structure to align the records by timestamp. Additionally, the framework achieves a considerable reduction in dataset generation times, including the homogenization process for ensuing analyses.

Furthermore, it creates datasets in different formats such as csv, XML and JSON, which were validated by experts. In addition, it ensures data integrity by avoiding their direct manipulation. The framework also contributes in terms of physical security by eliminating having to enter restricted spaces to obtain the required information.

Additionally, the solution was developed in the open-source Python language applying the AR methodology. Furthermore, it was developed using a standardized coding pattern, so new libraries can be easily added.

Likewise, it was proven portable since, due to the language used in its development, it was implemented in two environments such as command line and Django.

One advantage in using this framework is that it can be extended to other domains since this architecture design allows for its adaptation through the definition of metadata.

For example, it can migrate from structured to unstructured data and be implemented as a template in scenarios that require handling and transforming heterogeneous data, as well as providing files in different formats for further advanced analyses.

Regarding further research, several opportunities were identified. In defining metadata, a tool can be developed to simplify maintenance and to easily include additional information for any group.

As for the framework's areas of opportunity, libraries could be added for the creation of datasets in other output formats to support different time zones. Additional libraries can also feature other functions to meet the needs of the CECATEV meteorological center.

References

1. **Arora, R., Aggarwal, R. R. (2013).** An algorithm for transformation of data from MySQL to NoSQL (MongoDB). International Journal of Advanced Studies in Computer Science and Engineering, Vol. 2, No. 1, pp. 6–12.
2. **Brink, H., Richards, J. W., Mark, F. (2017).** Real-world machine learning. Manning Publications.
3. **BĂZĂR, C., IOSIF, C. S. (2014).** The transition from RDBMS to NoSQL. A comparative analysis of three popular non-relational solutions cassandra, MongoDB and couchbase. Database Systems Journal, Vol. 5, pp. 49–59.
4. **Camargo-Vega, J. J., Camargo-Ortega, J. F., Joyanes-Aguilar, L. (2015).** Conociendo big data. Facultad de Ingeniería, Vol. 24, No. 38.

5. **Chauhan, D., Bansal, K. L. (2017).** Using the advantages of NOSQL: A case study on MongoDB. International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 5, No. 2, pp. 90–93.

6. **Checkland, P., Holwell, S. (1998).** Action research: Its nature and validity. Systemic Practice and Action Research, Vol. 11, No. 1, pp. 9–21. DOI: 10.1023/a:1022908820784.

7. **Chen, M., Mao, S., Liu, Y. (2014).** Big data: A survey. Mobile Networks and Applications, Vol. 19, No. 2, pp. 171–209. DOI: 10.1007/s11036-013-0489-0.

8. **Chicco, G. (2021).** Data consistency for data-driven smart energy assessment. Frontiers in Big Data, Vol. 4, pp. 1–19. DOI: 10.3389/fdata.2021.683682.

9. **Coughlan, P., Coghlan, D. (2002).** Action research for operations management. International Journal of Operations and Production Management, Vol. 22, No. 2, pp. 220–240. DOI: 10.1108/01443570210417515.

10. **Cruz, A., Antaño, M., Mario, J., Martínez-Castro, J. M., Cuevas-Valencia, R. (2014).** Migración de bases de datos SQL a NoSQL. Revista Tlamati Sabiduria, Vol. 5, pp. 144–148.

11. **Dao, M. S., Zettsu, K. (2015).** Discovering environmental impacts on public health using heterogeneous big sensory data. IEEE International Congress on Big Data, BigData Congress, pp. 741–744. DOI: 10.1109/BigDataCongress.2015.122.

12. **Diouf, P. S., Boly, A., Ndiaye, S. (2018).** Performance of the ETL processes in terms of volume and velocity in the cloud: State of the art. 4th IEEE International Conference on Engineering Technologies and Applied Sciences, ICETAS 2017, pp. 1–5. DOI: 10.1109/ICETAS.2017.8277875.

13. **Diouf, P. S., Boly, A., Ndiaye, S. (2018).** Variety of data in the ETL processes in the cloud: State of the art. IEEE International Conference on Innovative Research and Development, pp. 1–5. DOI: 10.1109/ICIRD.2018.8376308.

14. **Dresch, A., Pacheco-Lacerda, D., Cauchick-Miguel, P. A. (2015).** A distinctive analysis of case study, action research and design science research. Revista Brasileira de Gestao de Negocios, Vol. 17, No. 56, pp. 1116–1133. DOI: 10.7819/rbgn.v17i56.2069.

15. **Eden, C., Huxham, C. (1996).** Action research for management research. British Journal of Management, Vol. 7, No. 1, pp. 75–86. DOI: 10.1111/j.1467-8551.1996.tb00107.x.

16. **Giffard-Roisin, S., Yang, M., Charpiat, G., Kumler-Bonfanti, C., Kégl, B., Monteleoni, C. (2020).** Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data. Frontiers in Big Data, Vol. 3, pp. 1–13. DOI: 10.3389/fdata.2020.00001.

17. **Jäger, S., Allhorn, A., Bießmann, F. (2021).** A Benchmark for data imputation methods. Frontiers in Big Data, Vol. 4, pp. 1–16. DOI: 10.3389/fdata.2021.693674.

18. **Jaybal, Y., Ramanathan, C., Rajagopalan, S. (2018).** HDSanalytics: A data analytics framework for heterogeneous data sources. Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pp. 11–19. DOI: 10.1145/3152494.3152516.

19. **Jiménez, A., Nieve, J., Estrada, F., Vázquez-Gálvez, F. A., Hernández, I. (2019).** Management of heterogeneous data in the red climatológica UACJ in a NoSQL environment. IEEE International Autumn Meeting on Power, Electronics and Computing, pp. 1–6. DOI: 10.1109/ROPEC48299.2019.9057068.

20. **Liu, Q., Guo, X., Fan, H., Zhu, H. (2018).** A novel data visualization approach and scheme for supporting heterogeneous data. Proceedings of the 2nd IEEE Information Technology, Networking, Electronic and Automation

Control Conference, pp. 1259–1263. DOI: 10.1109/ITNEC.2017.8284978.

21. **Patil, M. M., Hanni, A., Tejeshwar, C. H., Patil, P. (2017).** A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing — Sharding in MongoDB and its advantages. International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), pp. 325–330. DOI: 10.1109/i-smac.2017.8058365.
22. **PowerData (2013).** Procesos ETL: Definición, características, beneficios y retos.
23. **Steinacker, A., Ghavam, A., Steinmetz, R. (2001).** Metadata standards for web-based resources. IEEE MultiMedia, Vol. 8, No. 1, pp. 70–76. DOI: 10.1109/93.923956.
24. **Stevenson, R. D., Suomela, T., Kim, H., He, Y. (2021).** Seven primary data types in citizen science determine data quality requirements and methods. Frontiers in Climate, Vol. 3. DOI: 10.3389/fclim.2021.645120.
25. **Stuivenvolt-Allen, J., Wang, S. S. Y. (2019).** Data mining climate variability as an indicator of US natural gas. Frontiers in Big Data, Vol. 2, pp. 1–6. DOI: 10.3389/fdata.2019.00020.
26. **Thiollent, M., Colette, M. (2020).** Pesquisa-ação, universidade e sociedade. Revista Mbote, Vol. 1, No. 1, pp. 042–066. DOI: 10.47551/mbote.v1i1.9382.
27. **Tiwari, P. (2017).** Improvement of ETL through integration of query cache and scripting method. Proceedings of the International Conference on Data Science and Engineering. DOI: 10.1109/ICDSE.2016.7823935.
28. **Volpi, D., Meccia, V. L., Guemas, V., Ortega, P., Bilbao, R., Doblas-Reyes, F. J., Amaral, A., Echevarria, P., Mahmood, R., Corti, S. (2021).** A novel initialization technique for decadal climate predictions. Frontiers in Climate, Vol. 3, pp. 1–14. DOI: 10.3389/fclim.2021.681127.
29. **Wurster, P. M., Maneta, M., Kimball, J. S., Endsley, K. A., Beguería, S. (2021).** Monitoring crop status in the continental United States using the SMAP level-4 carbon product. Frontiers in Big Data, Vol. 3, pp. 1–17. DOI: 10.3389/fdata.2020.597720.
30. **Xiang, L., Huang, J., Shao, X., Wang, D. (2016).** A MongoDB-based management of planar spatial data with a flattened R-tree. ISPRS International Journal of Geo-Information, Vol. 5, No. 7, pp. 119. DOI: 10.3390/ijgi5070119.
31. **Zeng, N., Zhang, G. Q., Li, X., Cui, L. (2017).** Evaluation of relational and NoSQL approaches for patient cohort identification from heterogeneous data sources. Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, pp. 1135–1140. DOI: 10.1109/BIBM.2017.8217817.
32. **Zhong, M., Liu, M. (2009).** 3Se: A semi-structured search engine for heterogeneous data in graph model. Proceedings of the 18th ACM conference on Information and knowledge management. DOI: 10.1145/1645953.1646131.

Article received on 2023/01/09; accepted on 20/01/2024.
** Corresponding author is Alicia Margarita Jiménez-Galina.*