WikiOlapBase: A collaborative tool for open data processing and integration

Pedro M. Bernardo[1], Ismael S. Silva[1], Glívia A. R. Barbosa[1], Flávio R. S. Coutinho[1], Evandrino G. Barros[1]

[1]Departamento de Computação – CEFET–MG
Av. Amazonas, 7675 – Nova Gameleira – Belo Horizonte – MG – Brazil

pedromagalhaesbernardo@gmail.com,

{ismaelsantana, gliviabarbosa, coutinho, ebarros}@decom.cefetmg.br

Abstract. *The technological advances have made data sharing and knowledge generation possible in several areas. In order to support information extraction and knowledge generation, several datasets have been made publicly available, giving rise to the concept of open data. However, while such data are available, the processing, visualization, and analysis of them by society, in general, can be considered difficult tasks. Data are available to a great volume, in different files and formats, making it difficult to cross-reference and analyze them to obtain relevant information without the support of appropriate tools. Inspired by this scenario, this paper presents WikiOlapBase, a collaborative tool capable of processing, integrating and making feasible the analysis of open data from different sources, even by people without technical knowledge. WikiOlapBase contributes to the expansion of open data analysis, since it favors a greater information sharing and knowledge dissemination.*

1. Introduction

Technological evolution has enabled an increase in the speed and quantity of data that are produced daily and can be used to extract information and generate knowledge in different areas (e.g., economics, sociology, computing, engineering, biology and political science). Currently, this data can be produced and extracted from sources such as transactional information systems, e-commerce websites, social networks, mobile devices, sensors, governmental registries, among others. This phenomenon became known as Big Data [Sagiroglu and Sinanc 2013].

With the goal of sharing information and knowledge, several datasets have been made publicly available, giving rise to the concept of open data, which is available to anyone independently of their technical expertise [Hilbert 2013]. An example of that is data provided by the government, also known as open government data (OGD) [Vaz et al. 2010].

However, data availability alone is not enough for most people to process and analyze open data (whether governmental or not). As the data generally is heterogeneous, available in a variety of formats, in large volume and not always of easy understanding for the interested people, the lack of technical knowledge becomes a hindrance for its consumption [Graves and Hendler 2013].

This context raises two challenges: the first concerns the demand for an infrastructure capable of processing and integrating open data from different sources, making it possible to explore and analyze these datasets. The second refers to the demand for a tool, powered by this infrastructure, capable of generating visualizations and analysis without the need of technical knowledge from the user [Graves and Hendler 2013], [Sagiroglu and Sinanc 2013].

Encouraged by these challenges, this work proposes WikiOlapBase, a collaborative tool, publicly available, which is able to process and integrate open data. The goal of this tool is to provide an infrastructure so that contributors can (1) insert large volume of data from different sources, (2) manipulate and integrate data from different sources through filtering, grouping and joining operations, and (3) make available the processed and integrated data in a single format so they can be consumed by data visualization and analysis tools.

This work was conducted in two phases. The first one consisted of reviewing the literature on approaches for data processing, storage, and integration in the open data scenario, as well as in the definition of the tool's requirements. In the second phase the tool's architecture was proposed, followed by its development and evaluation.

The proposed tool contributes so that users, from different areas of knowledge can process and integrate open data of interest. In a complementary way, WikiOlapBase may support the expansion of open data analysis, thus contributing to a greater information sharing and knowledge dissemination [Graves and Hendler 2013], [Hilbert 2013].

2. Related Work

In the literature, it is possible to find works (e.g., [Graves and Hendler 2013], [Hoxha and Brahaj 2011], [Ding et al. 2010]) that describe the architecture of systems for open data integration and analysis. For example, Graves and Hendler (2013) introduce the prototype of a tool, OpenData-Vis, to integrate open data and make it readable for humans and computers. The authors argue about how this type of tool can benefit the population interested in the integration and analysis of large data volume and reinforces the demand to implement it and make it available for public use.

In turn, Hoxha and Brahaj (2011) propose the use of semantic web technologies to integrate data from different governmental organizations. The paper presents a conceptual approach composed of three modules. The first one is responsible for modeling an ontology and converting the raw data. The second module consists of an interface for querying the generated knowledge base and the third specifies an information visualization tool. Similarly, Ding et al. (2010) are developing an initiative to integrate and make available data from the United States government. The authors show how semantic web technologies can be used to convert and integrate this data.

Another related research line consists of works that describe system architectures to support data visualization. Viegas et al. (2007) introduce ManyEyes, a collaborative website in which users could submit data, as well as create and analyze interactive visualizations. The work done by Tang et al. (2004) addresses the challenges of designing an architecture for data visualization systems. According to the authors, in order to create a visualization system that is suitable for use, it is necessary to define a transparent infrastructure that describes: (1) the data model, (2) the way data is sent, and (3) the possibilities of transformation for the generated visualizations.

Although the related work discusses the challenges, requirements, and/or models of architectures for data processing, integration, and visualization systems, the authors do not detail the designs of the architectures so that they can be implemented and made publicly available. In addition, in spite of praising the importance of the collaborative aspect in the process of processing, visualizing and analyzing open data, the architectures presented in previous works do not offer collaboration mechanisms for the steps of data processing, integration and crossing [Graves and Hendler 2013], [Hoxha and Brahaj 2011], [Ding et al. 2010].

The tool proposed here differs from the others since it offers an infrastructure that supports processing and integration of open data in a collaborative way. Thus, users are able to collaborate in the insertion, processing, integration and cross-referencing of data from different sources. In addition, WikiOlapBase presents a robust architecture, designed to support open data visualization systems, and extend collaboration capability beyond the generation and analysis of such visualizations.

3. Methodology

As highlighted by Tang et al. (2004) and Graves and Hendler (2013), to architect and develop an infrastructure for open data processing and integration it is necessary to specify the requirements and architecture of the tool, develop the interaction modules, and evaluate and make it available. Thus, the methodology for conducting this work consisted in initially identifying the necessary requirements for open data processing and integration from different sources.

The requirements were identified from a review in of literature on Google Scholar and other major research repositories related to Computing: IEEE Xplore, ACM Digital Library, and Springer. The search string included the following terms: "open data processing tools", "collaborative open data analysis", and "open data visualization". Following that, the identified requirements were validated with three experts with more than eight years of experience with data processing and analysis.

Later, as suggested by Graves and Hendler (2013), the architecture of WikiOlapBase was defined in terms of technologies, data model, and operations to support the processing, integration and cross-referencing of different data sources. Then, WikiOlapBase was implemented from the requirements and the architecture defined in the previous steps. After its development, the tool was evaluated with users to verify its suitability for use through a Usability Test [Rubin and Chisnell 2008]. Next, each step of the methodology will be detailed and its main results will be presented and discussed.

4. Proposed Tool for Open Data Processing and Integration

This section presents the requirements that guided the development of WikiOlapBase, as well as the architecture and the tool created

4.1. WikiOlapBase's Requirements

Since this work proposes a tool for processing, integrating and cross-referencing open data from different sources, through a literature review (e.g, works such as those performed by

Tang et al. (2004) and Graves and Hendler (2013)) and validation of specialists in the area of data processing and analysis, it was possible to identify that WikiOlapBase should include the following features:

1. The tool must maintain the meaning of the original data.
2. The tool should convert different formats to the defined data model.
3. The tool should allow users to access the data present in its integrated database.
4. The tool should allow the definition of metadata to a given dataset.
5. The tool must be able to establish a relationship between different datasets.
6. The tool must accept compressed files.
7. The tool should allow the division of datasets into multiple files for submission.
8. The tool must provide an interface so that other applications can access the data present in the integrated database.
9. The tool must be able to store data on large scale.
10. The tool should optimize the data query time.

4.2. Architecture

The architecture of WikiOlapBase was specified, from its objective and requirements, in order to define: (1) the programming language used, (2) the data model and the Database Management Systems used, (3) the form of data access and (4) any design decisions regarding data processing and integration.

The Model-View-Controller (MVC) architecture standard was used for the development of the tool. In this pattern, the data model, user interface and control logic are separated into three components: (1) the model, which represents the data structure and business rules of the application, (2) the view, which presents the model for the user and (3) the controller, which interprets the user input and communicates with the model to make the necessary changes [Plekhanova 2009]. This pattern has been chosen because it allows the development and testing of each component independently, which facilitates and accelerates development. In addition, MVC also favors the evolution of web application functionalities [Gupta et al. 2012].

Python and the Django framework were used for the coding of the tool. Python is a popular programming language that supports integration with other languages and tools, in addition to a variety of libraries. Django is an open-source framework that seeks to automate development by adhering to the principle "do not repeat yourself"[Plekhanova 2009]. Also, the user interfaces were developed using HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript.

Since WikiOlapBase should work as a base infrastructure for open data visualization tools, and these tools typically make use of OLAP operations ([Viegas et al. 2007], [Tang et al. 2004], [Graves and Hendler 2013]), the WikiOlapBase data model should enable operations provided in such tools. Therefore, WikiOlapBase makes use of the column family model, which was designed to be horizontally scalable and it optimizes OLAP operations, that typically involve complex queries on large data volume [Sorjonen 2012], [Moniruzzaman and Hossain 2013].

In addition to storing datasets, WikiOlapBase also records the metadata that characterizes these sets. The storage of this metadata is relevant so that data from different sources can be structured, retrieved, and integrated into the tool [Turner 2002].

WikiOlapBase adopts the document-oriented data model for storing the metadata. This model was chosen because it does not have a defined structure, which allows adding new attributes that describe metadata on demand [De Diana and Gerosa 2010].

Cassandra was used to implement and manage the column-family model, for the document-oriented model, MongoDB was used. According to the website db-engines.com (2016), this two DBMS are among the ten most used, being the first place in their data model categories. This shows the popularity and acceptance of the community in relation to these tools, which justified their choices. In addition, the Apache Spark platform was used to perform more complex data operations (e.g., join) and to make reading and writing data faster on Cassandra [Kolaczkowski 2014].

Finally, to access the data, a REST API was made available, due to its simplicity and natural suitability for the web [Maleshkova et al. 2010]. Figure 1 shows the implemented architecture diagram.

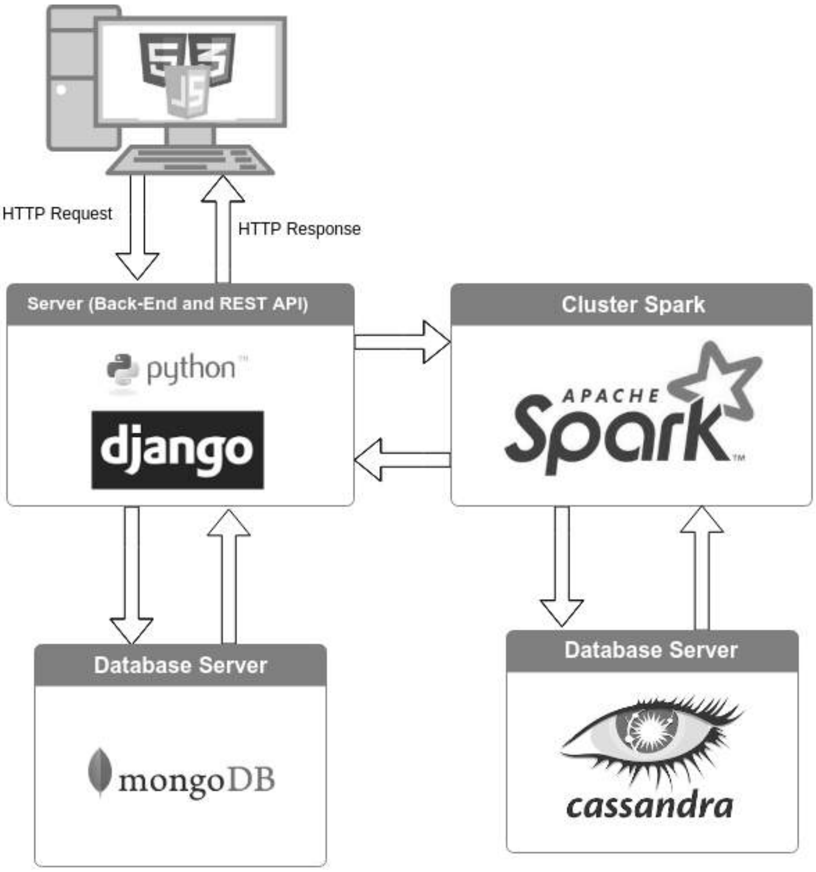


Figura 1. WikiOlapBase's Architecture

4.3. WikiOlapBase

The proposed tool has two modules, the first one is responsible for receiving, characterizing and integrating datasets sent by users. In this process the dataset is processed and stored in Cassandra and the metadata is stored in MongoBD.

The main flow of execution of the first module of WikiOlapBase consists of four steps. As shown in Fig. 2, the first step is to select and send the desired dataset, it is worth mentioning that in this first version only CSV files are accepted.

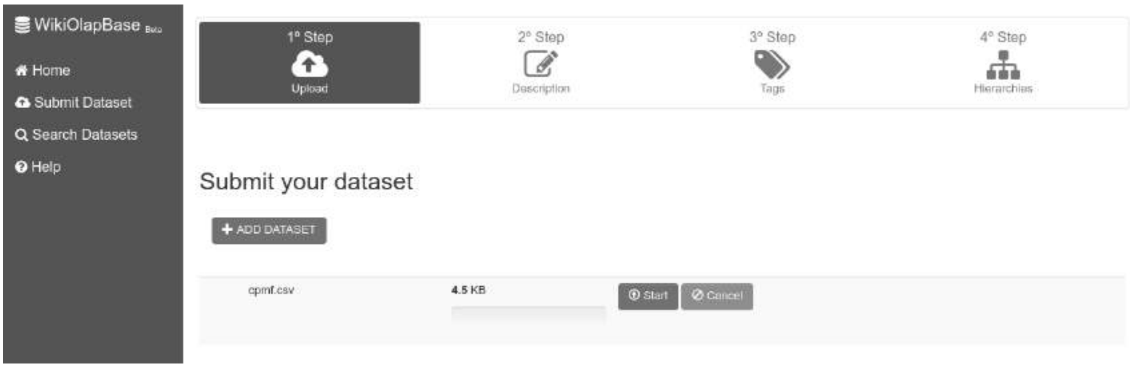


Figura 2. Upload interface - 1st step

From the data sent, the user must fill in the corresponding metadata. As shown in Figs.3, 4 and 5 this procedure encompasses the execution of three steps - although a sequence is suggested, the user can perform in the desired order.

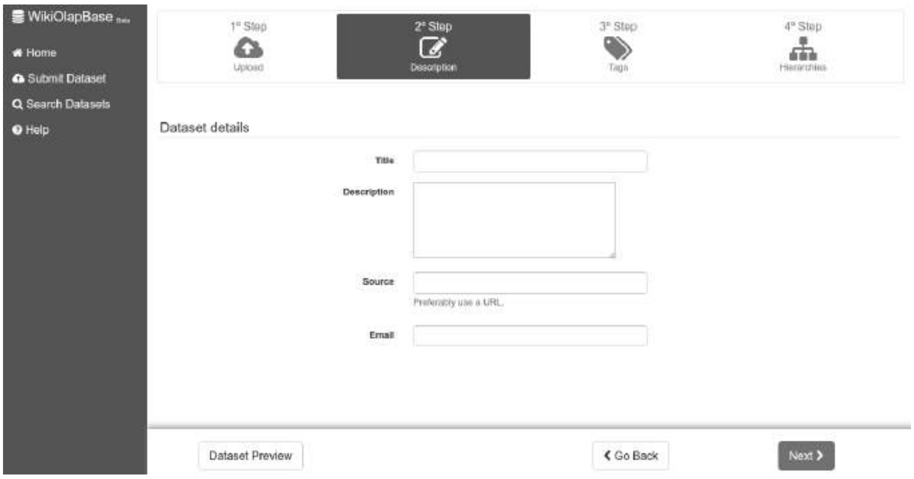
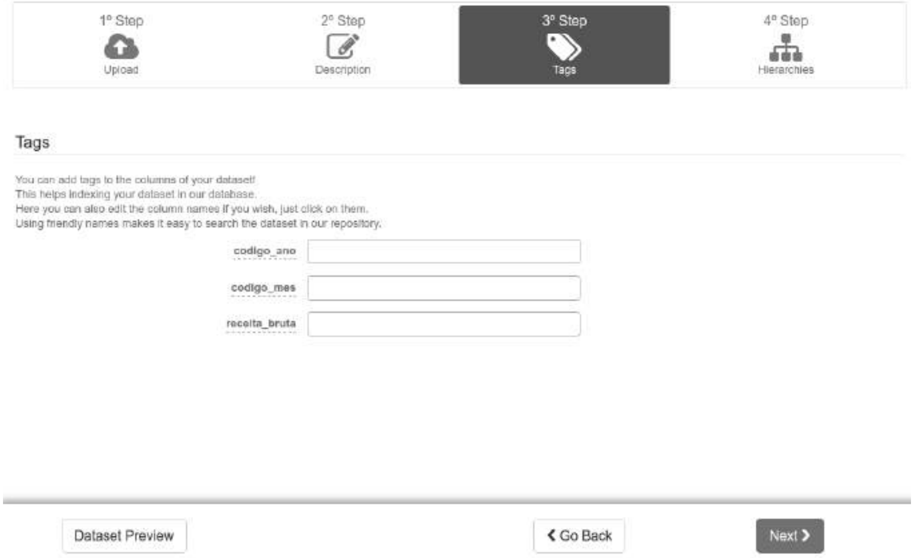


Figura 3. 2nd Step

Following the suggested sequence, basic dataset information such as source, title, and description should initially be filled. This information enables indexing inside the repository, allowing later, other users to search that data.

In sequence, the user can add tags to the columns of the dataset. In addition to helping in data indexing, tags also enable cross-referencing between different datasets, since they allow the discovery of datasets that have attributes in common. If desired, the user can also rename the dataset columns.

Finally, the user can identify data hierarchies within the dataset. This information can be used to generate visualizations that use OLAP operations such as drilling down and up [Graves and Hendler 2013]. In addition, at any time, the tool gives the user a preview of the dataset that he/she is sending, so it is possible to check if there are any errors in the datset before submitting it.



Tags

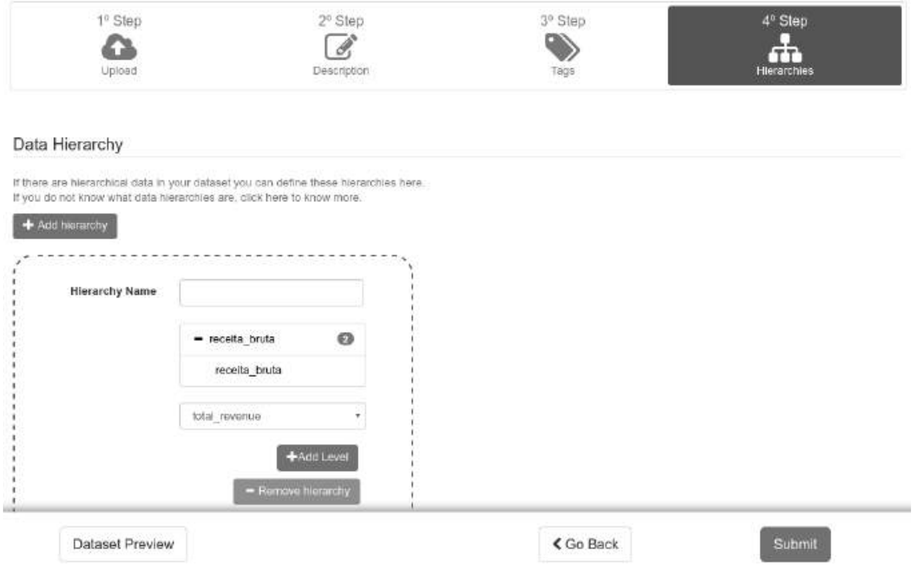
You can add tags to the columns of your dataset!
This helps indexing your dataset in our database.
Here you can also edit the column names if you wish, just click on them.
Using friendly names makes it easy to search the dataset in our repository.

codigo_ano

codigo_mes

receita_bruta

Figura 4. 3rd Step



Data Hierarchy

If there are hierarchical data in your dataset you can define these hierarchies here.
If you do not know what data hierarchies are, click here to know more.

Hierarchy Name

receita_bruta 2

receita_bruta

total_revenue

Figura 5. 4th Step

This process of loading and characterizing the dataset by the user enables the integration between the dataset sent by it and others already in the repository. In addition, the conscious filling of the metadata supports the collaborative aspect of the tool, since this enables the reuse of the datasets sent by any user that so wishes. The source code of WikiOlapBase is available at (*https://github.com/pedromb/wikiolapbase*).

The second module of WikiOlapBase provides access to the repository of integrated data through a REST API. This REST API has methods that can be requested to perform operations such as: retrieval of data, retrieval of metadata, and joining between different datasets. Its documentation is available at the address (*http://docs.wikiolapapi.apiary.io/*).

To better illustrate the differences between WikiOlapBase and similar tools presented in the Related Work section, Table 1 shows a comparison between these tools and WikiOlapBase.

Tabela 1. Comparison between the systems found in the literature and WikiOlapBase

Reference	Data Model	Data Access	Data importing format	Users able to send data	Access to other users datasets	Metadata	Cross-referencing datasets[1]
OpenData Vis - Graves and Hendler (2013)	Linked Data	GUI	N/S	N/S	N/S	Yes	No
ODA - Hoxha and Brahaj (2011)	Linked Data	GUI or SPARQL	XML, CSV, text	No	No	Yes	No
Data-Gov Wiki - Ding et al. (2010)	Linked Data	SPARQL web service	CSV	No	No	Yes	No
Many Eyes - Viegas et al. (2007)	Tables/Unstructured text	GUI	Tab delimited text.	Yes	Yes	Yes	No
Rivet - Tang et al. (2004)	Relational	REST API	CSV, MDX and SQL connections	Yes	No	Yes	No
WikiOlap Base	Column-family	REST API	CSV	Yes	Yes	Yes	Yes

WikiOlapBase has two distinguishing features in relation to the previous tools, the first being its collaborative aspect, as the loading and characterization of the datasets are carried out by the users themselves. The second feature is the possibility of relating and joining different datasets that are available in the repository. One aspect to be improved in WikiOlapBase is the availability of submitting data in different formats, since the initial version only allowed CSV files.

WikiOlapBase was evaluated for its usability to verify its suitability for use by the target audience. The methodology and main results of this evaluation are presented in the next section.

[1] Using data sent by other users

5. WikiOlapBase Evaluation

WikiOlapBase has been evaluated from the perspective of users regarding usability and collaboration. To that end, we conducted a Usability Test [Rubin and Chisnell 2008]. This test consists of an evaluation method that involves the participation of users in a controlled environment and involves the following phases: preparation, execution, and analysis [Rubin and Chisnell 2008].

The preparation phase is subdivided into the following steps: (1) determining the test objectives; (2) definition of the tasks to be performed; (3) selection of participants; (4) considerations about ethical aspects; and (5) execution of the pilot test. These steps generate artifacts that are subsequently used during the execution step of the Usability Test [Rubin and Chisnell 2008].

The execution represents the phase in which the evaluation of the system happens from the perspective of users. The evaluator conducts this phase, following the steps: (1) receiving the user; (2) presentation of the system; (3) the consent of the users, using the consent term; (4) pre-test questioning; (5) observation of tasks performed by users and (6) interview or post-test questionnaire [Rubin and Chisnell 2008]. In the third phase of the method, the data collected is analyzed by the evaluator [Rubin and Chisnell 2008].

Having exposed how the Usability Test is conducted, we now report how the evaluation of WikiOlapBase was carried out. In the preparation phase, after establishing the goal of the test (i.e., evaluating the usability and collaboration mechanisms of WikiOlapBase), the artifacts that would be used during the evaluations were developed. They are: the evaluation script, the participation consent term, the task description scenarios, the evaluation control sheet and the questionnaire regarding the degree of usability and collaboration of the tool [Rubin and Chisnell 2008].

With regards to the tasks, it is important to emphasize that the main interaction scenarios with WikiOlapBase were taking into account: (T1) Learning to use the tool from the instructions presented in the help section; (T2) Sending a dataset in CSV format; (T3) Observing the preview of the dataset; (T4) Filling in the basic information regarding the dataset; (T5) Defining tags for the columns of the uploaded file and renaming them; (T6) Defining a data hierarchy within the dataset; (T7) Submitting the metadata; (T8) Checking that the dataset was included in the repository using the search feature; (T9) Using the available API to retrieve the data that was sent and generate visualizations and; (T10) Using the API to join two distinct datasets and to generate visualizations.

From these tasks three different scenarios were defined, that involved one or more tasks: (C1) Send and generate the visualization of a dataset. This scenario (C1) involves tasks T1 to T9; (C2) Send a dataset and cross-reference that dataset with another already in the repository to generate a visualization. This scenario (C2) demands the execution of tasks T1 to T8, and T10 and; (C3) Use two datasets already in the repository and generate a visualization from them. This scenario (C3) involves performing T10. The users used two test datasets, these datasets were made available previously.

The execution phase of the usability test of WikiOlapBase was attended by 6 users. Of these, 5 were professionals in the area of computing (Computer Engineering or Information Systems), the last one had a degree in Mechanical Engineering. All of them worked with software development and had experience with data mining, analysis, and visualization. It is important to emphasize that this amount of users taking part in the evaluation is justified, since according to Nielsen (2000) [Nielsen 2000] and Rubin and Chisnell (2008)[Rubin and Chisnell 2008], usability tests must be executed by 3 to 5 users.

Each proposed scenario was tested by two different users. For each task performed by a user, the evaluator considered the time spent in its execution and, moreover, observed and noted how the task was completed (i.e., completed without error, completed with error, or not completed). It was not allowed, during the execution, that the evaluator answered questions related to the interface, or to some functionality of WikiOlapBase. This type of question would be answered only in the period after each task, when the doubts, difficulties, and suggestions of the users would also be discussed [Rubin and Chisnell 2008].

The Usability Tests, with the 6 users, took place in a period of 3 days, between September 27 and 29, 2016. Each test was performed individually and had a maximum duration of one hour. From the obtained data, the results were analyzed in order to characterize the indicators of completion of the tasks by the users; the mean elapsed time for each of the tasks as well as the overall mean time (i.e., average completion time of a scenario); and the degree of adequacy of WikiOlapBase to the principles of usability and collaboration. Through these measures it was possible to characterize the usability and collaboration of WikiOlapBase from the perspective of its users. The results obtained are discussed below.

5.1. Discussion of Results

Regarding the execution of the tasks, the graph in Fig. 6 shows the percentage of completion of each task. It is possible to check that all the tasks were completed by the users, with only 20% finished with an error.

Table 2 shows the execution time of each of the tasks by the users and the average time spent. It also displays the total execution times of tasks for each user (U). Users U1 and U2 performed scenario C1 while, users U3 and U4 performed scenario C2 and users U5 and U6 executed scenario C3. In this way it was also possible to calculate the average time per scenario. C1 was found to have an average running time of 11 minutes and 44 seconds, while the mean time to complete C2 was 12 minutes and 01 second, and the mean time to complete C3 was 05 minutes and 25 seconds.

Task T1 presented a similar execution time among most users. This task involves accessing the instruction page of the tool to learn about its operation. Although this task did not generate difficulties or doubts, it was possible to realize that most users prefer to learn how to use the tool during the execution itself. This explains the discrepancy that occurred, because a user was more interested in understanding the instructions on how to use the tool before actually exploring it. The task T2 was executed without problems and without great variation of time between the users.

Task T3 presented a small variation in execution times between users. In this task users should check the dataset sent from the "preview"functionality. During the evaluation, 3 users reported difficulties in locating this functionality and, in addition, presented suggestions for improvements in the visibility of this function. Although this has been considered a cosmetic problem, the suggestions pointed out by users will be implemented in the next version of WikiOlapBase.

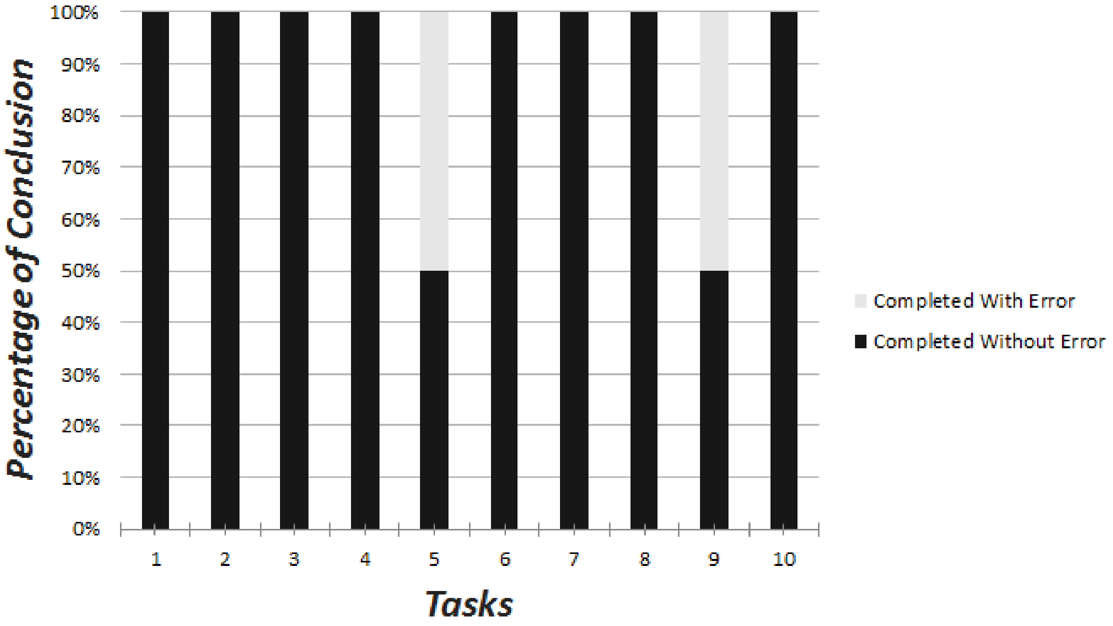


Figura 6. Percentage of completion of tasks by users.

In turn, in the execution of task T4 there was no doubt or difficulty. A task that presented great variation between the execution times was T5, in addition it was concluded with error by 2 users. In this task, users should insert tags for each column of the dataset sent and rename their columns. The errors occurred for two reasons: (1) the possibility of editing the column names and how to do this modification were unclear to users in the interface, and (2) the process of assigning tags was considered confusing by a user.

In the execution of tasks T6, T7, T8 and T10 there were no problems. Task T9 was completed with error by a user. In this task, the users should use the available API to retrieve the sent dataset and create a visualization. One possible explanation for the results obtained in this task is the lack of user experience in the use of web services. However, it is important to note that the WikiOlapBase (WOB) architecture will have integration with other data visualization tools. Therefore, direct access to this API by a user is not necessary, since the creation of visualizations, from the data available in the tool, will be executed through a graphical user interface (GUI) in future versions.

As mentioned before, after performing the tasks, each user evaluated the tool from the perspective of the 07 usability principles [Nielsen 1994], besides the 02 collaboration principles defined specifically for this tool, (1) passive collaboration and (2) active collaboration. The term passive collaboration refers to the possibility of using an existing dataset, that is, the degree to which the system allows users to use data sets that have been sent by other users. The term active collaboration refers to the possibility of sending a dataset so another person can use it.

The answers were grouped according to the interaction scenario that the user performed. Figure 7 summarizes the answers from the users who performed scenario C1, Fig. 8 from the users who performed scenario C2 and Fig. 9 from the users who performed the C3 scenario.

Tabela 2. Time elapsed in minutes for each task

	Scenario 1		Scenario 2		Scenario 3				
Task	U1	U2	U3	U4	U5	U6	Total Number of Users	Average Time	Standard Deviation
T1	01:07	03:09	01:56	01:20	00:00	00:00	4	01:53	00:55
T2	00:18	00:31	00:27	00:20	00:00	00:00	4	00:24	00:06
T3	01:18	00:40	00:29	00:15	00:00	00:00	4	00:40	00:27
T4	01:22	01:01	00:51	01:38	00:00	00:00	4	01:13	00:21
T5	02:23	00:36	01:18	02:01	00:00	00:00	4	01:34	00:47
T6	02:06	00:41	02:07	00:57	00:00	00:00	4	01:27	00:45
T7	00:24	00:04	00:10	00:05	00:00	00:00	4	00:10	00:09
T8	00:52	00:35	01:13	01:29	01:11	00:54	6	01:02	00:19
T9	03:18	03:04	00:00	00:00	00:00	00:00	2	03:11	00:10
T10	00:00	00:00	03:35	03:52	04:33	04:13	4	04:03	00:25
Total	13:08	10:21	12:06	11:57	05:44	05:07	-	-	-

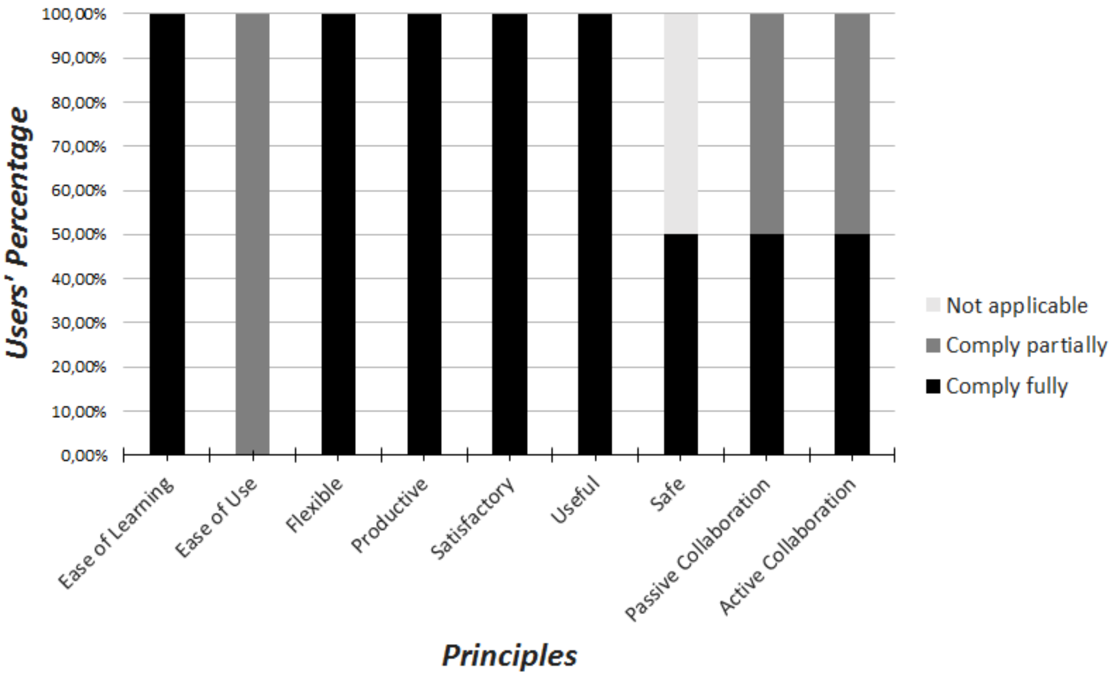


Figura 7. Degree of suitability of WOB, by principle of usability and collaboration in the users' perspective - Scenario C1

From the presented data it is possible to observe that, in the three scenarios, no principle was judged as "does not comply"from the perspective of users. In other words, for all users, all principles are either met or not applicable. It was possible to notice that for 66.67% of users, WOB fully complies with the principle of ease of learning and for

16.67% WOB fully complies with the principle ease of use. This indicates that learning how to use WOB is a simple task.

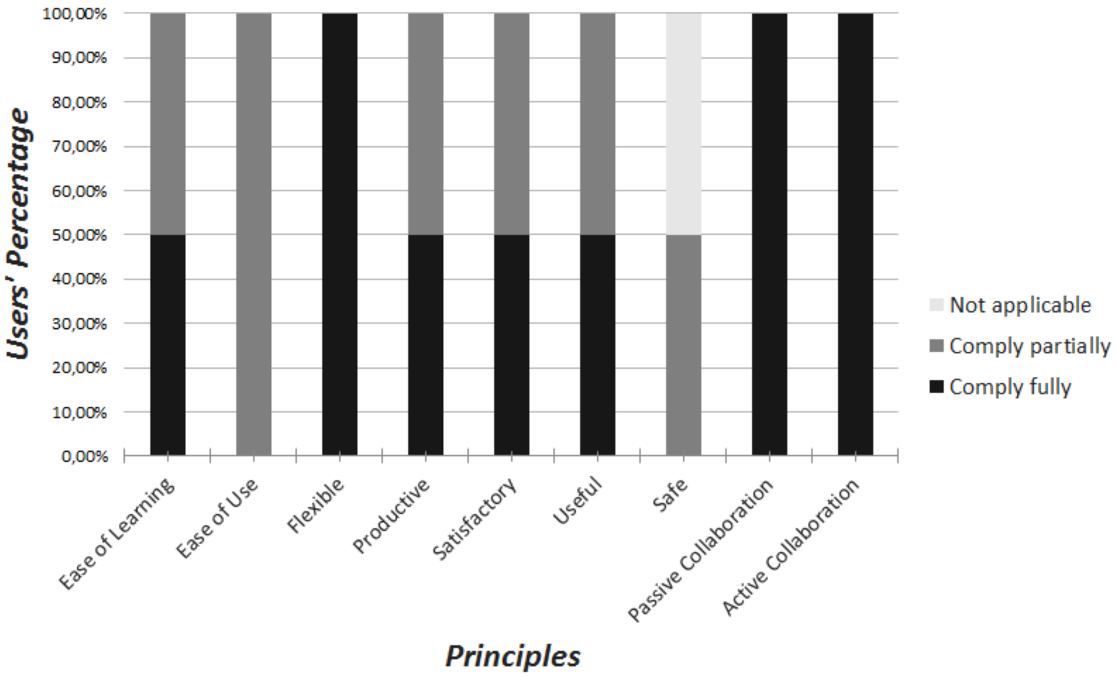


Figura 8. Degree of suitability of WOB, by principle of usability and collaboration in the users' perspective - Scenario C2

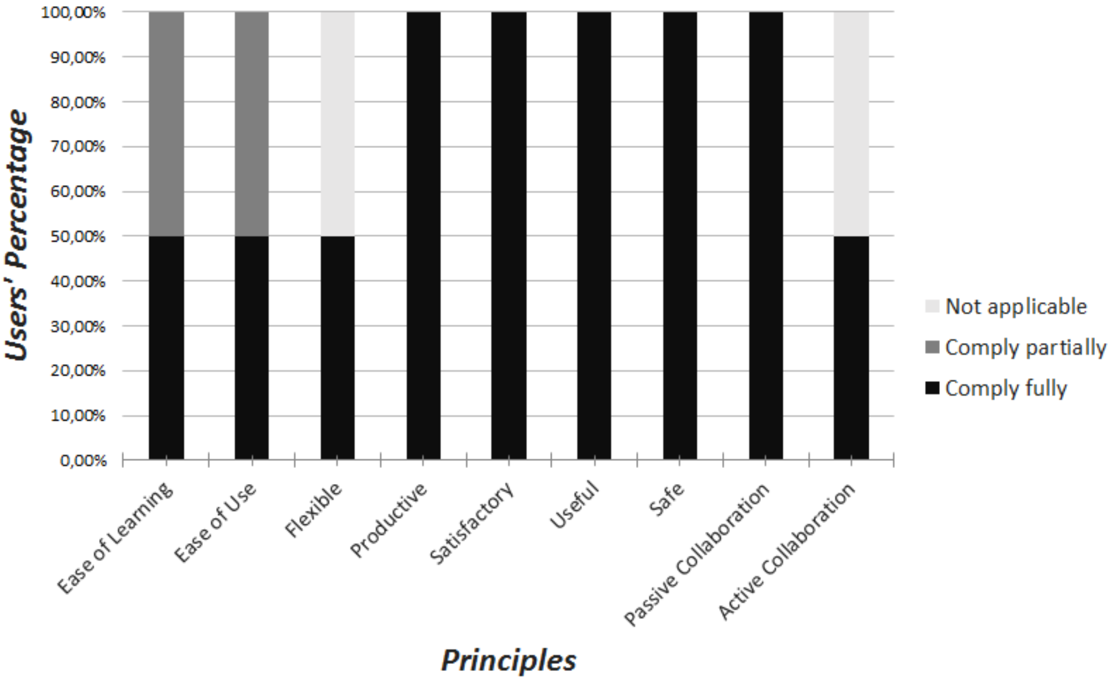


Figura 9. Degree of suitability of WOB, by principle of usability and collaboration in the users' perspective - Scenario C3

It is also possible to note that three principles were judged to be inapplicable. In scenario C1 and C2 was the principle "Safe". According to users, this principle did not apply to these scenarios because the tool is open to any interested users. In scenario C3 users judged that the principles "Flexible"and "Active Collaboration"did not apply. This scenario involves only the search for two datasets already in the repository and the use of the API to join the data, thus not involving alternative paths or sending datasets for other users to sue. This explains the interpretation by the users when answering that these principles do not apply.

Although there are adjustments to be implemented, through the tests with users it was possible to demonstrate that WikiOlapBase is suitable for use. This is corroborated by the speech of users throughout the tests, which approved the objective of the tool, as well as its flow of execution.

The next section presents our conclusions, as well as future work.

6. Conclusions and Future Work

This paper presented WikiOlapBase, a collaborative tool that allows processing, integration and cross-referencing of open data from different sources.

This tool allows (1) the insertion of large data from different sources, (2) data integration and manipulation from different sources through filter, cluster and join operations and (3) availability of processed and integrated data in a single format so that it can be consumed by data visualization and analysis tools.

In addition, WikiOlapBase features a robust architecture designed to support open data visualization systems and to extend collaboration capabilities beyond the generation and analysis of those visualizations. The results obtained, from the Usability Test, show that WikiOlapBase is a useful tool, satisfactory and suitable for users.

The proposed tool helps interested users to collaborate to process, integrate and cross-reference open data of interest. In a complementary way, WikiOlapBase can support the expansion of open data analysis, thus contributing to a greater information sharing and knowledge dissemination [Graves and Hendler 2013], [Hilbert 2013].

A second phase is planned as future work, in which a tool for open data visualization will be developed and integrated with WikiOlapBase. In addition, it would be interesting to conduct comparative analyzes to outline the advantages and disadvantages of WikiOlapBase over other existing tools.

Referências

De Diana, M. and Gerosa, M. A. (2010). Nosql na web 2.0: Um estudo comparativo de bancos não-relacionais para armazenamento de dados na web 2.0.

Ding, L. et al. (2010). Data-gov wiki: Towards linking government data.

Graves, A. and Hendler, J. (2013). Visualization tools for open government data. In *Proceedings of the 14th Annual International Conference on Digital Government Research*, dg.o '13, pages 136–145, New York, NY, USA. ACM.

Gupta, P. et al. (2012). Utilizing asp.net mvc in web development courses. *J. Comput. Sci. Coll.*, 27(3):10–14.

Hilbert, M. (2013). *Big data for development: From information-to knowledge societies*.

Hoxha, J. and Brahaj, A. (2011). Open government data on the web: A semantic approach. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2011 International Conference on*, pages 107–113. IEEE.

Kolaczkowski, P. (2014). Lightning fast cluster computing with spark and cassandra. Presentation at the CodeMesh-London event, available at https://www.infoq.com/presentations/spark-cassandra, visited in 18-October-2016.

Maleshkova, M. et al. (2010). Investigating web apis on the world wide web. In *Web Services (ECOWS), 2010 IEEE 8th European Conference on*, pages 107–114.

Moniruzzaman, A. and Hossain, S. A. (2013). Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*.

Nielsen, J. (1994). Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414. ACM.

Nielsen, J. (2000). Why you only need to test with 5 users. [Online; visited 23-October-2016].

Plekhanova, J. (2009). Evaluating web development frameworks: Django, ruby on rails and cakephp. *Institute for Business and Information Technology*.

Rubin, J. and Chisnell, D. (2008). *Handbook of usability testing: how to plan, design and conduct effective tests*. John Wiley & Sons.

Sagiroglu, S. and Sinanc, D. (2013). Big data: A review. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 42–47. IEEE.

Sorjonen, S. (2012). Olap query performance in column oriented databases.

Tang, D. et al. (2004). Design choices when architecting visualizations. *Information Visualization*, 3(2):65–79.

Turner, T. (2002). What is metadata. *Kaleidoscope*, 10(7):1–3.

Vaz, J. C. et al. (2010). Dados governamentais abertos e seus impactos sobre os conceitos e práticas de transparência no brasil. *Cadernos ppg-au/ufba*, 9(1).

Viegas, F. B. et al. (2007). Manyeyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128.