Input Data Optimization for Pauliceia 2.0 Platform's Historical Geocoding Web Service

Diego de Sousa[1], Daniela Leal Musa[2], Nandamudi Vijaykumar[3], Rodrigo M. Mariano[4], Luciana Rebelo[5], Raphael Augusto O. Silva[6], Luanna Nascimento[7], Luís Antônio Coelho Ferla[7], Karla Donato Fook[8]

[1]Escola Nacional de Ciências Estatísticas (ENCE)
[2]Universidade Federal de São Paulo (UNIFESP) / ICT - São José dos Campos
[3]Instituto Nacional de Pesquisas Espaciais (INPE)
[4]Framework Digital
[5]Instituto Federal de São Paulo (IFSP)
[6]Universidade Virtual de São Paulo (UNIVESP)
[7]Universidade Federal de São Paulo (UNIFESP) / EFLCH - Guarulhos
[8]Instituto Tecnológico de Aeronáutica (ITA) / Divisão de Ciência da Computação (IEC)

diegosalazar.est@gmail.com, karla@ita.br

Abstract. *The Pauliceia 2.0 platform is an outcome of a project in which collaborators and volunteers are encouraged to share historical research about So Paulo from 1870 to 1940, a period of growth and modernization. The Geocoding Web Service is an essential component of the platform. Currently, the web service's data is processed and input essentially manually by HÍMACO and IT project teams. This process increases the time between data collection and data availability while also making data cleaning more difficult and error-prone. The current work aims to address this issue by developing a solution that provides a user-friendly data input interface while also automating or semi-automating the treatment and data input into the Geocoding Web Service for the HÍMACO team. This was accomplished by building a prototype and applying software engineering techniques. The HÍMACO team is currently evaluating the prototype.*

Resumo. *A plataforma Pauliceia 2.0 é resultado de um projeto em que colaboradores e voluntários são incentivados a compartilhar pesquisas históricas sobre São Paulo de 1870 a 1940, período de crescimento e modernização. O Serviço Web de Geocodificação é um componente essencial da plataforma. Atualmente, o tratamento e catalogação dos dados utilizados pelo serviço web de geocodificação são feitos, praticamente de forma manual pelas equipes do HÍMACO e da TI do projeto. Esse processo aumenta o tempo entre a coleta e a disponibilidade dos dados, ao mesmo tempo que torna a limpeza dos dados mais difícil e propensa a erros. Para resolver este problema, o presente trabalho visa desenvolver uma solução que automatize ou semiautomatize o tratamento e entrada de dados do Serviço Web de Geocodificação Histórica, ao mesmo tempo que disponibiliza uma interface amigável para entrada de dados utilizada pela equipe do HÍMACO. Utilizando técnicas de Engenharia de Software, foi criado um protótipo para este fim. Atualmente, o protótipo está em processo de avaliação pela equipe HÍMACO.*

1. Introduction

The Pauliceia 2.0 platform is a collaborative project contribution by providing historical maps of São Paulo spanning the period from 1870 to 1940 (Ferla et al., 2020). Layers for the platform are created through the vectorization process using data from registration books and maps. In this case, the majority of historical data sets identify previous physical places through textual addresses (Ferreira et al., 2018). The process by which textual data is converted into geographical information is known as geocoding. As a result, a Geocoding Web Service that transforms outdated textual addresses into geographic coordinates is an essential part of the Pauliceia platform (Ferreira et al., 2018).

During the initial phase of the Pauliceia Project, several collaborations have been developed with several institutions, namely Universidade Federal de São Paulo (UNIFESP) / EFLCH (School of Philosophy, Modern Languages, and Human Sciences) and ICT (Institute of Science & Technology), Instituto Nacional de Pesquisas Espaciais (INPE), Arquivo do Estado de São Paulo, and Emory University. At present, the Pauliceia 2.0 Platform is being utilized by researchers from several institutions, such as UNIFESP, Instituto Tecnológico de Aeronáutica (ITA), Emory University, and the University of North Carolina, as part of Phase 2. Consequently, new requirements arise, requiring updates to the Portal's functionalities as well as a spatial extension to accommodate additional study areas (Fook et al., 2021).

For the Geocoding Web Service, data input and availability processes are carried out independently. The first process is performed by HÍMACO (History, Maps, and Computers) team from EFLCH, and another one by TI team. As a result, the primary goal of this work is to bridge the gap between these two groups by developing a solution to ensure consistency and efficiency in the availability of Geocoding Web Service Pauliceia's data. The goal of the approach is to apply Software Engineering techniques to provide a computational solution to automate the HÍMACO team's workflow and input data treatment for historical Geocoding Web Service. As a result, the time between delivery of addresses and their availability on the platform's map is reduced, while data consistency in the cleaning process is improved.

The paper is organized as follows: Section 2 provides a theoretical foundation for this work. Section 3 depicts the work development and discusses the improvements made to this point, while Section 4 depicts the Final Considerations.

2. Theoretical Foundation

2.1 Pauliceia 2.0

As mentioned earlier, the central aim of Pauliceia 2.0 is to make a digital platform that encourages collaborative mapping of São Paulo's urban-industrial modernization history from 1870 to 1940 and is available to researchers (Ferla et al., 2020).

The Pauliceia 2.0 platform operates as an open-source, web-based system with a service-oriented design. A service-oriented architecture facilitates seamless data and functionality exchange among various systems, enhancing integration and interoperability across different technologies. Spatiotemporal vector data in Pauliceia is stored within a PostGIS database system, while raster data is stored in Geotiff files. The architectural framework has two sets of web services, as illustrated in Figure 1.

The initial set comprises geographical web services conforming to the standards of the Open Geospatial Consortium (OGC). These include the Web Map Service (WMS) for rendering map images, the Web Feature Service (WFS) for managing vector data, the Web Coverage Service (WCS) for handling coverage data, and the Catalog Service Web (CSW) for managing metadata related to spatiotemporal data, services, and associated objects (Longley et al., 2013). OGC's contributions have been instrumental in advancing geospatial data interoperability through the establishment of web service standards for visualizing, distributing, and processing geospatial data. The second set consists of the VGI (Volunteered Geographic Information) protocol and Geocoding Web Services (Sansigolo, 2017; Mariano et al., 2018).

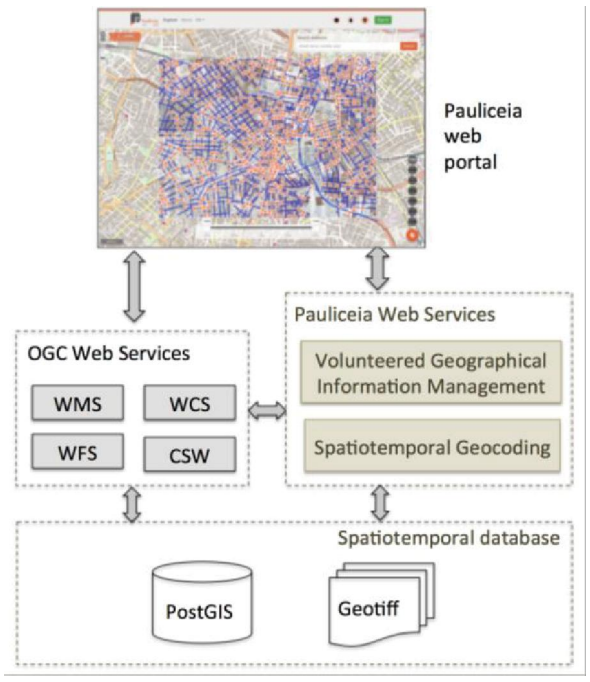

Figure 1: Pauliceia 2.0 Platform Architecture (Ferreira et al., 2018).

The authors emphasize that the presence of Geocoding Web Service is an essential feature of the architecture. According to Ferreira et al. (2018), numerous geocoders have been proposed for effectively processing contemporary addresses; however, these geocoders do not address the handling of historical data. An historical geocoder must work with spatiotemporal data sets, or spatial entities whose geometries and properties change through time. The primary difficulties associated with developing an address geocoding system for historical data mostly originate from the diverse range of changes in street and building names, geometry, and numeration systems during different time periods. Every spatial element, such as a street segment and a location with an address, has an associated period in the Pauliceia 2.0 database that specifies how long it is valid for.

2.2 Historical Geocoding Web Service

The Pauliceia 2.0 platform allows historians to share geographic data from the past that is the result of their research. Textual addresses are utilized by most historical data collections to denote past spatial locations. Thus, a geocoding web service was developed with the ability to transform textual historical addresses into corresponding geographic coordinates.

The development of the geocoding web service intended to enhance the capabilities of the OGC standard services, addressing specific and essential requirements of the Pauliceia 2.0 Project. Historians can geocode a single address or a group of

addresses using CSV files by using this web service. Every address must contain the street name, house number, and year. From the historical locations and street segments kept in the platform database, the service calculates the geographic coordinates related to the addresses.

3. Optimization of input data processing for the Historical Geocoding Web Service

The following subsections describe the steps performed during this work.

3.1 Requirements identification

This work primarily revolves around addressing the challenge of capturing both a client's needs and a project's requirements. The initial and significant phase of this work centers on functional requirements, non-functional requirements, and the requirements gathering process, including meetings and consultations with the stakeholders (Pressman and Maxim, 2021). The first part of this challenge involved applying techniques that would help achieve the goal of understanding the needs of the stakeholders.

Upon engaging with the HÍMACO team, a clear perspective emerged regarding their specific needs. After applying requirements identification techniques, such as Interviews, Workshops and Prototyping, three main requirements were identified: a streamlined means of inputting data directly into the platform without third-party assistance; a user-friendly method to visualize their previously collected data; and avoiding a laborious one-by-one approach. Considering that the requirements of a system are a description of what the system should do, after the stakeholders meeting, it was clear what the difficulties were that the system should resolve.

As a result of this task, there is a list of functional requirements, one of the artifacts obtained by applying requirements identification techniques with the HÍMACO researchers. Table 1 contains some, and the Use Case Diagram with application requirements is shown in Figure 2. Based on these requirements, a specification and a prototype were developed, following the visual identity of the Pauliceia Platform 2.0.

Table 1 - Partial list of requirements

R001 - The data must be validated before inclusion in the Geographical Database.
R002 - There should be a functionality that allows you to view the cataloged address in database.
R003 - The new data must be inserted individually, through a form
R004 - The new data must be inserted in a batch form, by importing files.
R005 - The information contained in a line of the Address cannot be

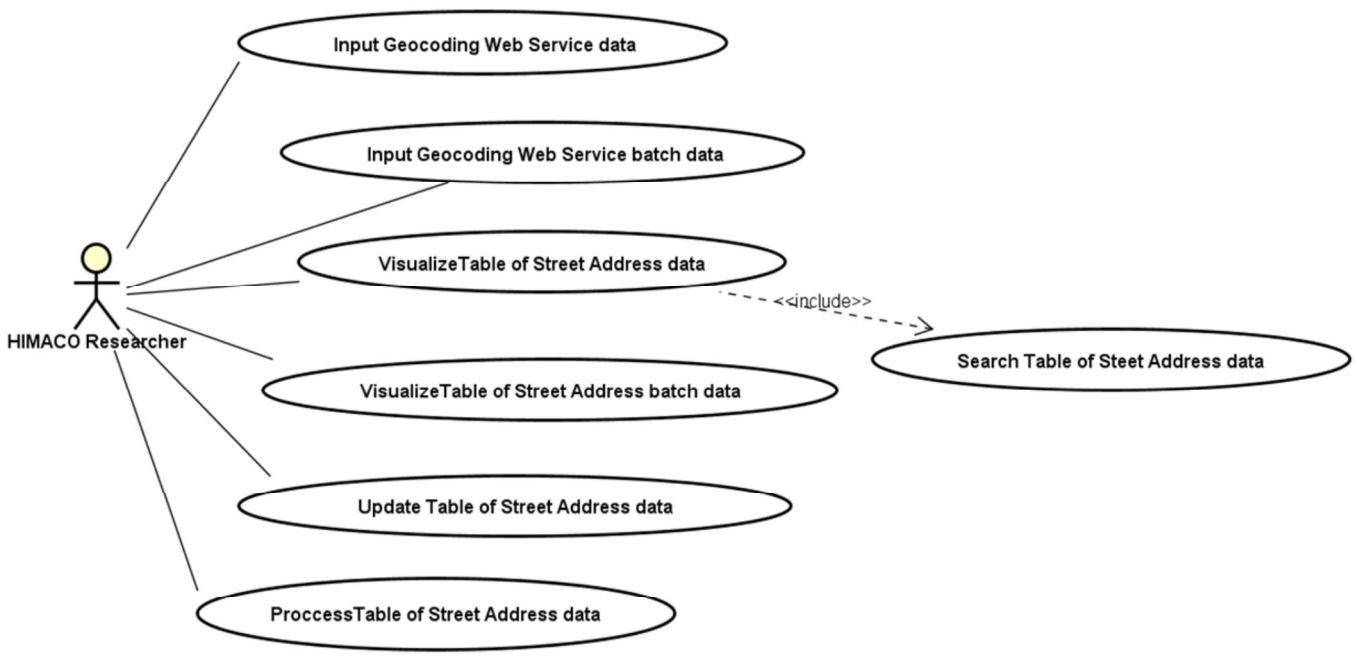

Figure 2: Application' Use Case Diagram. Source: Authors.

3.2 Prototype production

Subsequently, the focus of the project shifted to capturing the demands of a CRUD (Create, Read, Update, Delete) process, its user interface aspects, and also the importance of maintaining the site's identity in order to inherit the already-built user experience. The Pauliceia platform interface is presented in Figure 3.

Figure 3 - Pauliceia 2.0 Platform interface (Ferla et al., 2020)

Another step in completing this interface involved identifying the essential tools necessary for its development. A critical non-functional requirement that needed to be respected was the utilization of Python as the primary backend language. This choice was made because there was already an existing process in the file upload method on the platform that uses Python, as well as other existing aspects of Pauliceia 2.0. Furthermore, for the creation of an API solution supporting the backend of the forms interface, Django framework proved to be simple and versatile due to its built-in models and the way an interface could be easily developed with the use of its Views method (Dauzon et al., 2016).

The HÍMACO team did not provide this approach, but rather the team's developer did, and it provided a flexible way to meet two different requirements. One way to think about the data catalog is to send the data directly to the production database or its test

version, but another approach is to send them to a staging database, in case further processing is needed or implementation of new features is needed before consolidating the final database.

Figures 4 and 5 show the interface of the prototype obtained. The core idea was to allow users to log in to Pauliceia 2.0 and access an area for inserting new data into the platform database, leading them to the next page. As evident, this page shows the typical forms format, with variables with the placement for Address and Address Book catalog entries, among others. Positioned on the left, is a menu offering various other options, enabling users to navigate and access additional features.

Figure 4 - (a) Basic form for data input; (b) CSV file data uploader. Source: Authors.

One of the most challenging aspects of this project emerged in how the research team gathered the information. This is where JavaScript proved to be very beneficial. Due to the working method of the research team, which involved initially collecting information from books and manually recording all details, it became crucial to provide a mechanism for passing digital input to a later stage. Thus, the need to avoid the task of inputting each individual record one by one led to the development of a feature enabling the batch upload of data saved within a spreadsheet file.

Finally, the prototype gives the user access to the entire database (third button). This feature will need to be carefully improved in the future, when Pauliceia service will be much more populated with data than it is now. But, for now, it attends one of the main requirements for the initial project.

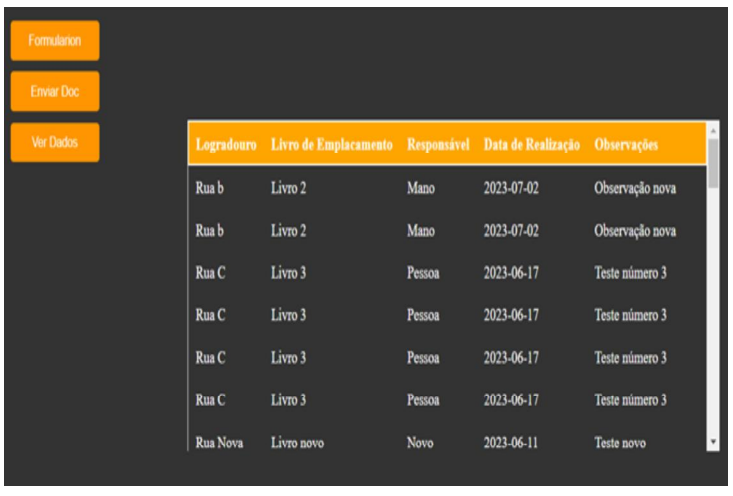

Figure 5 - Database data Visualization. Source: Authors.

The obtained prototype, as depicted in Figure 4, showcases an interface where

users have the option to input data directly into the forms, either one by one or many as a batch input. This approach enables immediate data validation, ensures proper formatting, and improves the data process significantly. For instance, dates can be automatically generated, reducing the need for manual entry. In possession of a mobile phone, users can seamlessly perform this task while concurrently consulting their address book, maintaining a smooth workflow.

An example of historical data that can be inserted in this way can be found in (Mariano et al, 2018). In scenarios requiring batch data uploads, the second button shown in Figure 4 serves this purpose. This feature allows users to upload a CSV file containing multiple sets of data to the database. Currently, respecting the database format is essential, although the long-term objective is to develop a natural language processing approach that enhances this feature's capabilities for the user.

The final feature, aligned with the primary requirements, facilitates researchers' access to previously provided data. By navigating to the form presented in Figure 5, users can review historical data, including addresses and other pertinent information that has been supplied to the platform.

3.3 Results and Discussion

After studying Software Engineering and Requirements Engineering techniques and concepts, it was possible to establish a baseline for evaluating the issue. This work's main challenge is to reduce the existing gap during input data processing for the Historical Geocoding Web Service. This process is required in order to add new study areas to the Pauliceia 2.0 Platform. The current workflow can be seen in Figure 6.

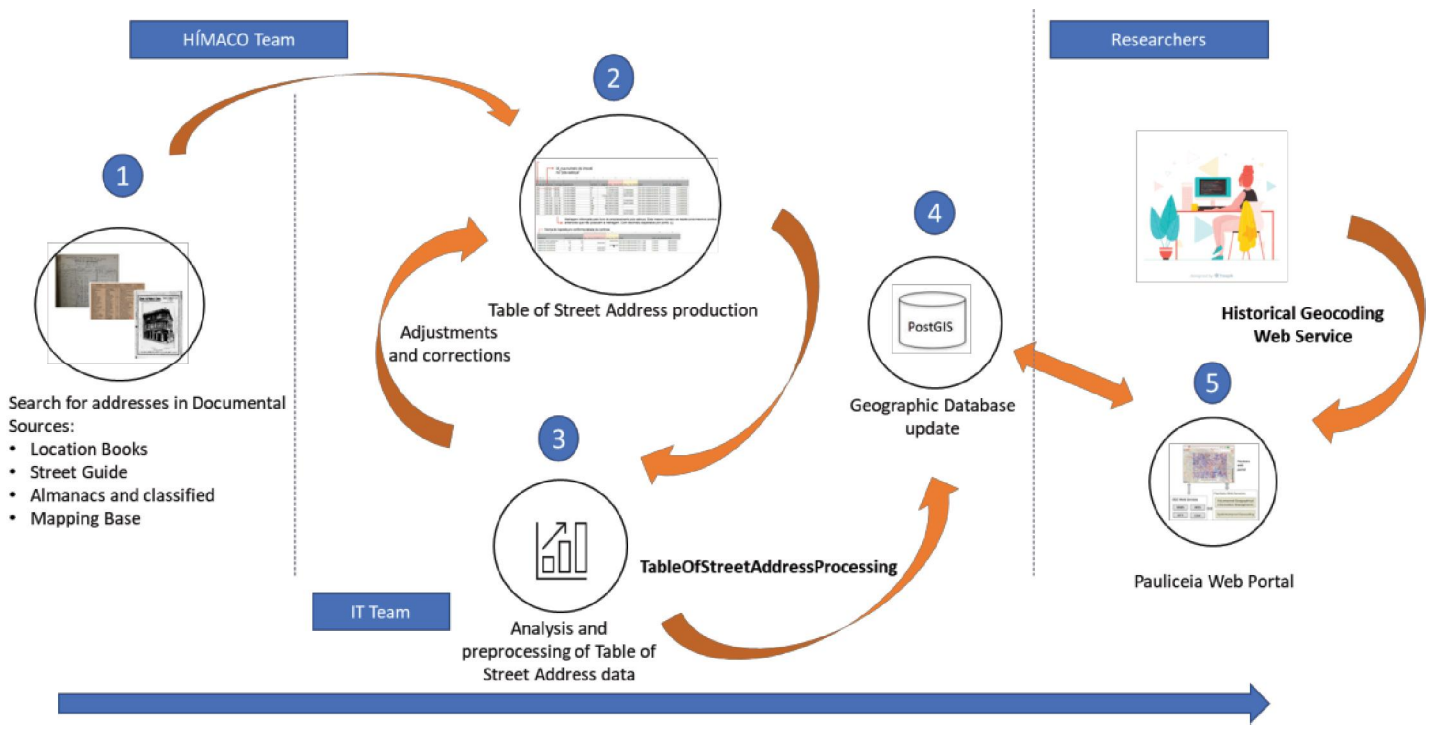

Figure 6: Current workflow. Source: Authors.

As shown in Figure 6, there are few steps in the workflow for the Pauliceia platform to have the input data for use of the Historical Geocoding Web Service. In the beginning, HÍMACO researchers used to gather addresses from documentary sources like location books, street guides, etc. Such information is summarized in the spreadsheet called Table of Street Address (steps 1 and 2). The Project IT Team receives the Table of

Street Address and assesses and preprocesses the data (steps 3 and 4). The spreadsheet is sent back to HÍMACO for any necessary revisions if any inconsistencies are found. The process is repeated until the Table of Street Address is ready to be sent into the algorithm that will catalog that data into the Pauliceia geographic database (steps 2 and 3). The Historical Geocoding Web Service will later on use this data. Typically, the carrying out of steps 2 and 3 causes delays in the delivery of data for the historical geocoding web service.

After identifying the requirements of the researchers of the Pauliceia platform, an application was specified in order to eliminate the detected delay. With the use of the developed application, the workflow has been optimized, as shown in Figure 7.

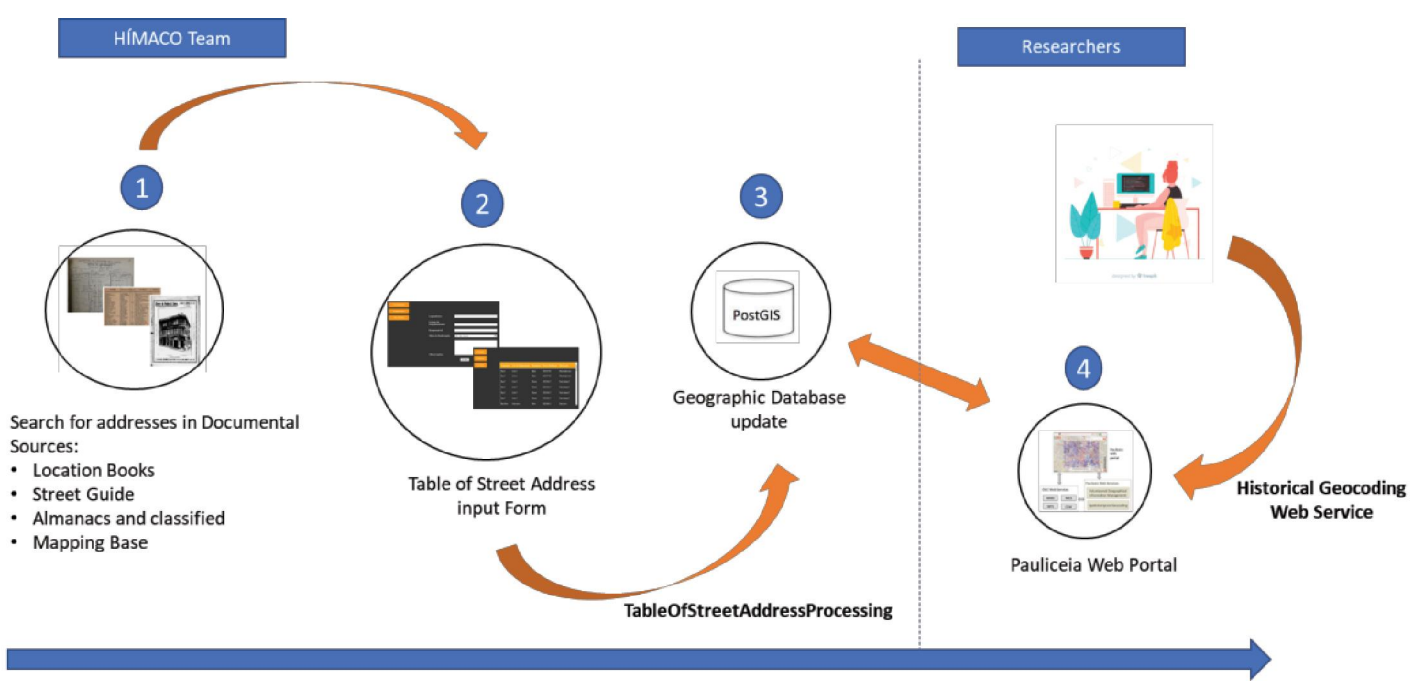

Figure 7: Optimized workflow. Source: Authors.

Steps 2 and 3 of the current workflow were combined into a single phase (step 2) in the workflow developed after specification and prototype construction, where the produced application ensures consistency and gets the data ready for insertion into the database. The elimination of delay is observed in the process.

Due to ongoing server migration from the INPE server to the UNIFESP server, the project has not yet been deployed to a platform. The project can therefore still be used locally and acts as a script that can be run while connected to the server.

4. Final Considerations

This work consisted of the study of Software Engineering techniques to improve the understanding of the demands of the HÍMACO team, who manage the Pauliceia platform. More especially on the processing of data used by the historical Geocoding Web Service.

From the use of Requirements Engineering techniques it was possible to assess, diagnose and propose a computational solution to optimize the workflow for including data used by the Historical Geocoding Web Service of the Pauliceia 2.0 platform. Thus, this work has achieved the objectives initially outlined, and it will significantly improve both historical and computational team work.

In order to further enhance the functionality and usability of the built application,

it is advisable to integrate it within the Pauliceia platform environment in future endeavors. As previously stated, the infeasibility of this task is attributed to the ongoing movement of the Pauliceia platform from the server hosted by the Instituto Nacional de Pesquisas Espaciais (INPE) to the server operated by the Universidade Federal de São Paulo (UNIFESP).

ACKNOWLEDGEMENTS

Our thanks to FAPESP/FAPESP eScience Program for funding Phases 1 and 2 (Scholarships: 2016/04846-0 and 2020/03700-7) of the Pauliceia project, and to CNPq for granting a scientific initiation scholarship.

REFERENCES

Fook, K.; Musa, D.; Vijaykumar, N.; Mariano, R.; Morais, G.; Silva, R.; Sansigolo, G.; Rebelo, L.; Silva, V.; Ferla, L.; Almeida, C.; Nascimento, L, Santos, M.; Torres, A.; Pereira, Â.; Atique, F.; Lesser, J.; Rogers, T.; Britt, A.; Laguardia, R.; Barbour, A.; Farias, O.; Marco, A.; Dickson, C. and Camargo, T. **Collaborative Historical Platform for Historians: Extended Functionalities in Pauliceia 2.0**. WEBIST, 17th International Conference on Web Information Systems and Technologies, 2021.

Ferla, L.; Ferreira, K.R.; Atique, F.; Britt, A.G.; Fook, K.D.; Lesser, J.; Miyasaka, C.; Musa, D.; Rogers, Thomas D.; Vijaykumar, N. **Pauliceia 2.0: mapeamento colaborativo da história de São Paulo, 1870-1940**. HISTÓRIA, CIÊNCIAS, SAÚDE-MANGUINHOS (IMPRESSO), v.27, p.1207 - 1223, 2020. Home page: [https://www.scielo.br/j/hcsm/a/LsTg5nrNLZXdd8mfdGSNr7C/?format=pdf&lang=pt] [doi:10.1590/s0104-59702020000500010]

Fook, K.; Musa, D.; Ferla, L.; Vijaykumar, N.; Ferreira, K.R.; Queiroz, G.R.; Miyasaka, C.R.; Atique, F.; Lesser, J.; Rogers, T.; Britt, A.; Laguardia, R.; Mariano, R.M.; Barbour, A.M.; Guarnier, O.; Santos, M.; Sansigolo, G.; Yamamoto, J.; Meireles, P.M.; Mazzarello, W.; Almeida, C.R.; Nunes, E.R.; Nascimento, L.; Silva, V.M.F.; Pricinato, B.; Taveira, D. Noronha, C. A. **Pauliceia 2.0: enriquecendo as Humanidades Digitais com Geocodificação e Informação Geográfica Voluntária**. RBHD, v. 1, p. 110-133, 2021.

Pressman, R. S.; Maxim, B. **Engenharia de Software**. 9a. ed., McGraw-Hill Bookman, 2021.

Somerville, I. **Engenharia de Software**. 10a. ed., São Paulo: Pearson Addison-Wesley, 2019.

Dauzon, S. Bendoraitis, A. and Ravindran, A. **Django: Web Development with Python** — Birmingham B3 2PB, UK: Packt Publishing Ltd., 2016 .

Ferreira, K.R.; Ferla, L.; Queiroz, G.R.; Vijaykumar, N.L.; Noronha, C.A.; Mariano, R.M.; Taveira, D.; Sansigolo, G.; Guarnieri, O.; Rogers, T.; Lesser, J.; Page, M.; Atique, F.; Musa. D; Santos, J.Y.; Morais, D.S.; Miyasaka, C.R.; Almeida, C.R; Nascimento, L.G.M; Diniz, J.A. and Santos, M.C. **A Platform for Collaborative Historical Research based on Volunteered Geographical Information**. Journal

of Information and Data Management, Vol. 9, No. 3, December 2018, Pages 291–304.

Sansigolo, G; *Web Service para geocodificação de endereços em banco de dados espaço-temporais*. Trabalho de Conclusão de curso de Tecnólogo em Análise e Desenvolvimento de Sistemas, FATEC, São José dos Campos, 2017.

Mariano, R.M; Ferreira, K.R.; Ferla, L.A.C. **VGI Protocol and Web Service for Historical Data Management**. Proceedings of XIX GeoInfo. Campina Grande, PB. 103-115. 2018.

Longley, Paul A.; Goodchild, Michael F.; Maguire, David J.; Rhind, David W. **Sistemas e Ciência da Informação Geográfica**. 3. Ed. Porto Alegre: Bookman, 2013.