A Brazilian Public Health Geospatial Surveillance Platform: a development case study

Gabriely Rangel Pereira

Capstone project presented to the
Institute of Mathematics and Statistics
of the University of São Paulo
for the degree of
Bachelor of Computer Science

Program: Computer Science
Advisor: Prof. Dr. Paulo Meirelles
Coadvisor: Prof. Dr. Fabio Kon

During this work, the author was supported by USP and FAPESP (nº 19/11900-9)

São Paulo
November, 2019

Abstract

Gabriely Rangel Pereira. **A Brazilian Public Health Geospatial Surveillance Platform: a development case study**. Capstone project (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2019.

In the context of health, there is a growth in stored and generated data from health facilities. However, it can be a difficult task to analyze these data because of its size and complexity. In this project, we worked in collaboration with the São Paulo Municipal Health Secretariat (SMS-SP), in a government-academia collaboration to develop a data visualization platform. This platform is a surveillance dashboard for large scale data processing that enables analysis via advanced techniques for data visualization. This project is based on public data from the Brazilian National Health System (SUS). We consolidated the platform software architecture to enable the integration with the Hospital Information System datasets (SIH-SUS) from any region of Brazil. The platform final architecture follows the *Model-View-Controller* (MVC) software design pattern. With this approach, we brought benefits over the initial prototype version, and overcome the microservices disadvantages. The flexibility brought by the chosen architecture decreases the code complexity and brings modularity to the system. Moreover, the platform indicates that this kind of platform can support the development of public health policies to the Brazilian population.

Keywords: Data Visualzation. Software Architecture. Web Application. Smart Cities. Public Health Management.

Resumo

Gabriely Rangel Pereira. **A Brazilian Public Health Geospatial Surveillance Platform: a development case study**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

No contexto de saúde, existe um crescimento na quantidade de dados gerados e armazenados por estabelecimentos de saúde. Entretanto, pode-se tornar uma tarefa difícil a análise desses dados por conta de seu tamanho e complexidade. Neste projeto, trabalhamos em conjunto com a Secretaria Municipal de Saúde de São Paulo (SMS-SP), isto é, através de uma colaboração governo-academia, para desenvolver uma platforma de vigilância de processamento de dados em larga escala que possibilite análises via técnicas avançadas de visualização de dados. Este projeto é baseado na análise de dados públicos do Sistema Unificado de Saúde (SUS). Nosso objetivo foi consolidar a arquitetura de software dessa platforma para habilitar a integração com base de dados do Sistema de Informações Hospitalares (SIH-SUS) de qualquer região do Brasil. A arquitetura final segue o padrão de software *Model-View-Controller* (MVC). Com isso, pretendemos trazer mais benefícios que a versão prototipada inicialmente, e superar as desvantagens da versão em microserviços. A flexibilidade trazida pela arquitetura escolhida reduz a complexidade do código e traz modularidade ao sistema. Além disso, a platforma sinaliza que este tipo de ferramenta pode auxiliar no desenvolvimento de políticas eficientes de saúde para a população brasileira.

Palavras-chave: Visualização de Dados. Arquitetura de software. Plataforma Web. Cidades Inteligentes. Gestão Pública em Saúde.

List of Figures

Contents

Chapter 1

Introdution

With the growth in the urban center population, mainly in developing countries, a mismatch in the management of their resources to meet citizens' needs has come up. The Brazilian city of São Paulo is a good example of this, it is the most populous municipality of the country (Instituto Brasileiro de Geografia e Estatistica - IBGE, 2019). On the other hand, the constant advancement in computing technologies may provide tools to support information analysis in the cities and contribute to local public management. This use of technology is what defines the so-called smart cities.

Smart cities aim at improving urban efficiency and quality of life, contributing to the city's sustainability, mobility, health, accessibility, and so on. The Institute of Mathematics and Statistics of the University of São Paulo (IME-USP) hosts the National Institute of Science and Technology (INCT) for the Future Internet and Smart Cities – the InterSCity project [1] – to investigate strategies and create applications to promote the technological innovation in the country. The project in question in this paper is one of the projects supported by InterSCity.

In the context of health, there is a growth in generated and stored data from health facilities. A large body of data can provide relevant information that can be used by public managers to benefit the creation of public policies. The World Health Organization (WHO) advises countries to develop responsive and resilient health systems that create evidence-based policies as one of its results.[2] So, the collection and processing of health data become fundamental for obtaining the evidences to create evidence-based policies.

[1] http://interscity.org
[2] https://www.who.int/healthsystems/about/progress-challenges/en/

1.1 Motivation

It can be a difficult task to analyze a large body of data because of its size and complexity. In this project, we worked in collaboration with the São Paulo Municipal Health Secretariat (SMS-SP) to develop a platform for large scale data processing. It enables the analysis of health data via advanced techniques for data visualization. This platform intends to be a useful tool which, in term, may help the elaboration and management of public health policies.

The purpose of the platform is to analyze the Hospital Information System datasets (SIH-SUS), provided by the Brazilian National Health System (SUS). These datasets contains hospitalization information such as the patient diagnosis, the hospital location, the geo-anonymized (see Section 4.4) patient location, and so on. So, the platform also deals with many georeferenced data that would be visualized on a dynamic map.

There were a previous project, developed by IME-USP undergraduate students, that aimed to resolve the same situation using the São Paulo SIH-SUS dataset. However, this previous project resulted in a unsatisfactory performance and a non generalizable system, i.e. the developed system was attached to the São Paulo SIH-SUS dataset, not allowing the use of SIH-SUS datasets from other regions of Brazil.

The platform we propose aims to deal with any SIH-SUS dataset. The platform also faces the performance problem by refactoring the previous system and implementing a new software architecture called *Model-View-Controller* (MVC).

To achieve this goal, we studied the possibility to use *Microservices* as a solution to the problems presented above. Nevertheless, the use of the *Microservices* technologies would not be a good solution to our case because we could not implement and maintain these technologies (Section 4.1 describes this better). Finally, we found a better to implement and maintain way to face the problems using the MVC software pattern. It brought performance benefits over the previous project, and also enabled the possibility to use other SIH-SUS datasets beyond the São Paulo dataset.

1.2 Methods

As mentioned previously, this project had the collaboration of SMS-SP. The collaboration between government and academia is often challenging, and its main cause is the poor project management (Anthopoulos *et al.*, 2015). According to Wen *et al.* (2019), when in collaboration, government and academia should strive to increase the chances of success. Successful government-academia collaborations need to mitigate interinstitutional

conflicts. The adoption of practices from agile methods and FLOSS ecosystems impact the people involved in the project notably (Wen *et al.*, 2019).

The contact between IME-USP and SMS-SP exists since the development of the previous platform. Since then, there are two SMP-SP agents keeping in touch with us. They receive the notifications of advances and difficulties we are handling, and we receive their feedback. In our case, the use of practices from agile methods improved the communication between us and SMS-SP agents. We are used to communicate through email, but we also had some important presential meetings along the year.

The use of the agile methods, such as: the presential meetings with SMS-SP, the development iterations, the use of Kanban, and so on, helped us to detail the platform features and limits. For example, through these meetings we knew the available resources at SMS-SP to implement and maintain the platform in development. It helped us to define the better architecture to fit their needs and their available resources.

1.3 Organization

The capstone project is organized as follows. In Chapter 2, we discuss the clustering method as a well know big data analysis technique used to facilitate the comprehension and visualization of large datasets, such as the São Paulo SIH-SUS dataset. Chapter 3 describes the hospitalization dataset we are dealing with (the Hospital Information System; SIH-SUS), the Brazilian National Health System (SUS), the dataset creation, its structure, and how we structured this dataset on the platform database. Chapter 4 presents the aspects of the previous platform, the *Microservices* pros and cons in our case, the final platform, its usage, its features, and the used design principles along the platform development. Finally, we discuss the lessons learned, implications and future works to the platform in Chapter 5.

Chapter 2

Clustering

We are in an era full of data. Every passing day people, companies, and government entities are collecting amounts of data to analyze and retrieve some meaningful information from them. Big Data Analytics in healthcare, for example, is evolving into a promising field for providing insight from very large datasets (W. Raghupathi and V. Raghupathi, 2014). However, the large number of data can also become an obstacle to do data analysis because of its size (Tsai *et al.*, 2015).

Classifying or grouping them into a set of categories is a way to solve this problem. Cluster analysis is the organization of a collection of patterns into clusters based on similarity (Jain, Murty, *et al.*, 1999).

In terms of Big Data, it is very confusing or impossible to get something useful from the entire data. Basically, instead of dealing with this, the method simplifies them into clusters according to the similarity among these data. That is one of the most primitive activities of a human being (Anderberg, 1973). To understand a new object or event, we describe its features and find similarities among other known objects or events. So, we classify them based on the similarity or dissimilarity according to some standards.

Besides the use of cluster analysis to reduce an extensive body of data to a short description, there is a wide variety of other applications. Many other fields of study are using this analysis, such as Life sciences; Medicine; Social sciences; Earth sciences; Engineering sciences; and Policy sciences (Anderberg, 1973). Among all applications developed by these fields, we can mention the following cases, respectively: to construct taxonomies; for making diagnoses in the treatment of patients; to find behavior patterns; to analyze land and rock; to identify handwritten characters, and; to identify credit risk.

The variety of algorithms for grouping data into clusters is enormous. Each field of

study developed its own without notice of other fields developing similar methods. We need to use some algorithm to get the clusters because the quantity of possible classifications of a dataset into groups is too large. According to Anderberg (1973), the number of possible results when classifying a dataset containing n items into m clusters is a sum of n Stirling partition numbers, where each Stirling number represents the possibilities to group n items into a specific number of clusters, where $1 \leq m \leq n$. All the options we would investigate is the sum below:

$$\sum_{m=1}^{n} S(n, m) = \sum_{m=1}^{n} \frac{1}{m!} \sum_{k=0}^{m} (-1)^{m-k} (mk) k^n.$$

For instance, the possibilities to sort 20 elements into *eight* distinct clusters is the following:

$$S(20, 8) = 15.170.932.662.679.$$

It would take an extraordinary amount of time to analyze so many partitions, and lots of them would be uninteresting because we are considering all the possibilities of arrangements.

2.1 Cluster Analysis Algorithms

Clustering is the classification of patterns into groups. It is considered an intrinsic or unsupervised classification (Jain and Dubes, 1988). Figure 2.1 shows a tree of classification methods, suggested by Lance and Williams (1967). We describe each level of the tree below.

Exclusive and non-exclusive: The exclusive classification is a partition of the dataset into the groups, i.e., every data item belong to only one group. On the other hand, the non-exclusive or overlapping classification permits the item to belongs to more than one group. A fuzzy clustering method is one kind of non-exclusive classification. It assigns degrees of membership in several clusters to each data item.

Intrinsic and extrinsic: Intrinsic, or unsupervised, when only the proximity measure between items is used to obtain the classification. Outward or supervised, when classification is done using the proximity measure and the category label too. The intrinsic classification choses the groups utilizing the similarity (or dissimilarity) among the data items. The extrinsic classification choses the groups discriminating the items according to the given labels.

Hierarchical and partitional: A hierarchical classification is a nested sequence of partitions which can be represented in a *dendrogram*. On the other hand, a partitional classification produces a single separation of the data instead of a clustering structure,

such as the *dendrogram* provided by the hierarchical classification.

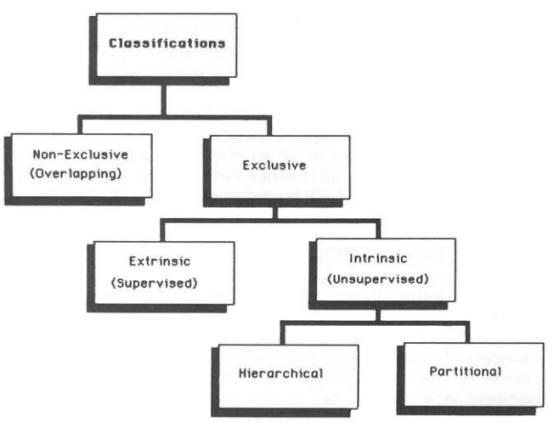

Figure 2.1: *Classification methods (extracted from (Jain and Dubes, 1988)).*

2.1.1 The main algorithms

There are two main categories of clustering algorithms, the hierarchical and the partitional methods. As discussed previously, hierarchical methods produce a nested series of partitions, whereas partitional methods produce only one.

Hierarchical

Most of the hierarchical algorithms are similar to the single-link (Sneath and Sokal, 1973) and the complete-link (King, 1967) algorithms. Both of them follow the agglomerative way of grouping items. In other words, considering a dataset containing n items, it starts with n distinct groups, one for each item, and these groups are merged until a stopping condition is met.

The stopping condition is based on the similarity between clusters. The single-link distance considers the *minimum* of the distances between all pairs of patterns from the two clusters. The complete-links consider the *maximum* of these distances, though.

Besides this difference, both algorithms work following these steps (Jain, Murty, *et al.*, 1999):

1. Set a group for each item, build a list containing the distances between all the possible pairs of items, and sort this in ascending order.

2. Step through the sorted list. For each distinct similarity value, form a graph connecting the pair of items closer than this value. Stop if all of the items are connected to the graph.

3. The result of this algorithm can be represented in a *dendrogram* like the one shown in Figure 2.2. Every similarity level can provide a different grouping. The dashed line at Figure 2.2 produces the *three* clusters *A+B+C*, *D+E*, and *F+G*, for example.

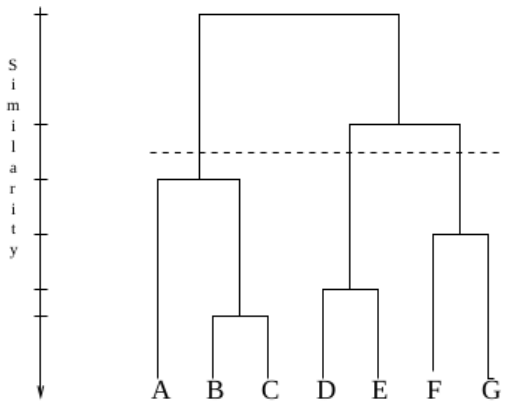

Figure 2.2: *Dendrogram obtained using a hierarchical algorithm (extracted from (Jain, Murty, et al., 1999)).*

Partitional

Partitional algorithms obtain a single partition of the data considering a specific number k of clusters. This technique produces the clusters using a criterion function; usually, a *square error method* or a *graph-theoretic divisive clustering algorithm.*

The *k-means* is the most common algorithm using a squared error criterion (MacQueen, 1967). It has linear-complexity $O(n)$, where n is the number of items of the dataset. This algorithm is described as the following steps:

1. Choose randomly k clusters centroids as seeds. They can be k different items from the body of data or random points inside the feature space of the dataset.

2. Attribute each item to the closest cluster.

3. Recalculate the centroid of the cluster considering all items inside the cluster and calculating the midpoint among the items.

4. If it converges, stop. We can consider it converges when we produce the same clusters as the previous result. If it does not converge, return to step 2.

The main graph-theoretic divisive clustering algorithm is based on the *minimal spanning tree* (MST) of the data (Zahn, 1971).

1. Construct the *minimal spanning tree* of the data.

2. Delete the largest edges of the created MST, where each edge represents the distance measure (e.g., the Euclidean distance).
3. Stop when a specific criterion is met.

2.2 Clusterization at the platform

The developed platform provides a spatial visualization through clusters (see Figure 4.5 and Section 4.5) using georeferenced data. Once the data are displayed in a geographic map, so the similarity measure is the geographic distance where the coordinates is defined in terms of latitude and longitude.

The algorithm implemented at the platform uses the *maximum* distance between the clusters. It is a kind of a complete-link algorithm (described in the Subsection 2.1.1) called hierarchical greedy clustering. In the platform implementation, we are using the *Leaflet* library (Leaflet.markercluster[1]), a free software project that provides this algorithm.

The *hierarchical greedy clustering* algorithm, described by (Agafonkin, 2016), intends to resolve the grouping of the data for each zoom level. These steps show how it works:

1. Choose an item, represented by geographical coordinates.
2. Find the items around the chosen item within a certain radius.
3. Make a cluster with those items. The cluster is represented by its centroid (the coordinates of the first chosen item).
4. Choose an item that is not part of any cluster.
5. Repeat until every item is visited.

This approach becomes too expensive when we need to process the entire dataset for each zoom level, following those steps. The algorithm avoids this problem by reusing calculations. For instance, consider we need to follow those steps for 18 different zoom levels. Instead of recalculating every cluster, we can reuse the calculation of the last zoom level to calculate the next one.

The Figure 2.3 show how it works. So, we start from the deeper zoom level (*z18* in the figure case), where we extract more clusters. The next level (*z17*) reuses the clusters from the last zoom level (*z18*), considering them as the original items from the dataset. This process is repeated for each zoom level, and it is much faster than the simple solution and can be fast enough to handle millions of points on the map.

[1] https://github.com/Leaflet/Leaflet.markercluster

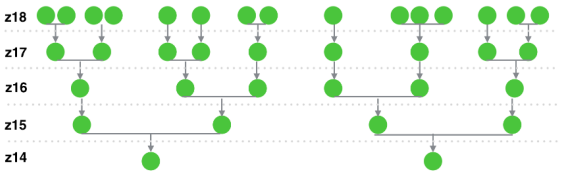

Figure 2.3: *Hierarchical greedy clustering algorithm at different zoom levels (extracted from (Agafonkin, 2016)).*

Besides, there are two expensive operations in this algorithm: Find an item that is not part of any cluster and find every cluster on the current screen. But it can be improved using a method proposed by Agafonkin (2016) called spatial index. So, instead of a loop into the dataset seeking a point every time, we can preprocess the database into a particular data structure and then use it to find any item. In this way, we can index any item from the dataset at any zoom level.

With the clustering computational approach, the platform provides for the public health professionals the ability for performing analyzes and understanding patterns. It can help the task of finding hospitalization concentrations on the map for a specific diagnosis, for example. This kind of analyzes is not possible without the aid of software that uses data visualization and clustering techniques. The last part of Chapter 4 (specifically in Section 4.5) presents the usage of clusters to show spatial data in practice.

Chapter 3

Dataset

The Brazilian Unified Health System (*Sistema Único de Saúde* – SUS) enables the process of gathering large datasets into different health information systems, such as: National Notification System (SINAN); Mortality Information System (SIM); Information System on Live Births (SINASC); Hospital Information System (SIH-SUS); Outpatient Information System (SIA-SUS); and so on (Brasil. Ministério da Saúde. Secretaria de Vigilância em Saúde, 2005). The state finances the SUS services at the federal, state, and municipal levels. It provides essential, universal, and free access services to the entire population of the country (Paim *et al.*, 2011).

According to the 2013 National Health Survey (*Pesquisa Nacional de Saúde* – PNS), performed by the Brazilian Institute of Geography and Statistics (*Instituto Brasileiro de Geografia e Estatística* – IBGE), the majority of the population uses SUS services. This survey indicated that more than 80% of the Brazilian population is dependent on SUS (Stopa *et al.*, 2017). The SUS datasets become vital to the awareness of the health situation in Brazil. Moreover, they can support the management of public health services.

The SUS Information Technology Department (*Departamento de Informática do SUS* – DATASUS) organizes the data and provides free access to them. DATASUS aims to integrate Health Information to the Brazilian Ministry of Health, according to a Federal Government decree on April 16th, 1991:

> ***Art. 12.*** *It is the Information Technology Department of SUS that specifies, develops, implements and operates information systems for the core activities of SUS, following the sectoral organ guidelines.* (Brazil, 1991)

From then on, DATASUS developed about 200 different systems (such as SINAN, SIM, SINASC, among others cited previously) to provide software solutions to improve the

computerization of SUS data (Brasil. Ministério da Saúde. Secretaria Executiva, 2002). The generated data are widely available through the DATASUS site[1].

3.1 Hospital Information System Dataset (SIH-SUS)

The Hospital Information System (SIH-SUS) is one of the systems developed by DATASUS. This system is composed of many filled forms from the country health facilities and it becomes available on the dataset within a month after the form filling. These forms follows the Hospital Admission Authorization (AIH) form model and they are composed of, among other things, diagnosis code for hospital admission (International Classification of Diseases - ICD-10), age group, gender, amount paid, duration time, patient location, and health facility location.

Through the SIH-SUS dataset, we have access to hospitalization data that represents about 70% of all hospitalizations of the country (Brasil. Ministério da Saúde. Secretaria de Vigilância em Saúde, 2005). So, these data can be a good representation of the country real situation.

However, there are constraints on the accuracy and reliability of the data. Fields can suffer variations producing biased estimates. There are cases such as unreliable patient location, false diagnosis, and duplicated hospitalization data.

In spite of its limitations, this dataset is essential for the awareness of the SUS healthcare assistance and for the prioritization of precautionary measures. Its data provides meaningful information to public health managers. For example, through these data it is possible to find which region lacks hospitals, or even which hospital lacks a specific specialty service (like pediatrics).

The SIH-SUS dataset is composed of data from all over Brazil, however it may differ subtly from region to region. The form composition can be different containing distinct field names or more fields than the default AIH form. For example, the São Paulo SIH-SUS dataset contains the patient latitude and longitude coordination (inserted by São Paulo Municipal Health Secretariat), but it is not required by the AIH form. Next section explain how it works at São Paulo.

[1] http://datasus.saude.gov.br/

3.2 The São Paulo SIH-SUS dataset

The great São Paulo population estimate is about 20.935.000 people (Demographia, 2019). So, nearly 10 percent of the Brazilian population (210.536.000, based on IBGE's population projection[2]) lives in great São Paulo. As well as other megacities, great São Paulo is facing challenges such as crime, traffic congestion, urban sprawl, and also public health problems.

São Paulo city has one of the best public health care system in Brazil. Its health information management is advanced compared to most other Brazilian cities. In this context, for example, São Paulo city has specific departments in its Municipal Health Secretariat (SMS-SP), such as the Epidemiology and Information Coordination (*Coordenação de Epidemiologia e Informação* – CEInfo) created in 1989 (before DATASUS) (Prefeitura do Município de São Paulo, 2016).

The CEInfo had an essential role in the platform development. They preprocessed the SIH-SUS dataset removing the corrupted and blank data, anonymizing the entire dataset, and inserting the geolocation (in terms of latitude and longitude) of the census sector centroid of the patient location (i.e. geo-anonymizing the patient location, described at the end of the Section 4.4).

3.2.1 Dataset structure

Each hospitalization data at the SIH-SUS dataset contains information about the patient, the health facility where the patient was admitted, its localization, the patient diagnosis, and so on. This data is inserted into the dataset through an AIH form (described above in Section 3.1). Figure 3.1 represents part of this form.

Since 1995, the SIH-SUS datasets retrieved 285.745.862 hospitalization data according to the DATASUS TabNet system [3]. The preprocessed SIH-SUS dataset, made available to us by CEInfo, covers 554.202 hospitalization data of São Paulo city patients from December 2014 to December 2015. This dataset is large and complex enough to be a good example for the platform usage.

3.2.2 Dataset representation at the platform

The developed platform represents the SIH-SUS dataset in the way shown in Figure 3.2.

[2] https://www.ibge.gov.br/apps/populacao/projecao/index.html, retrieved on October 3rd 2019.
[3] http://www2.datasus.gov.br/DATASUS/index.php?area=0202&id=11633

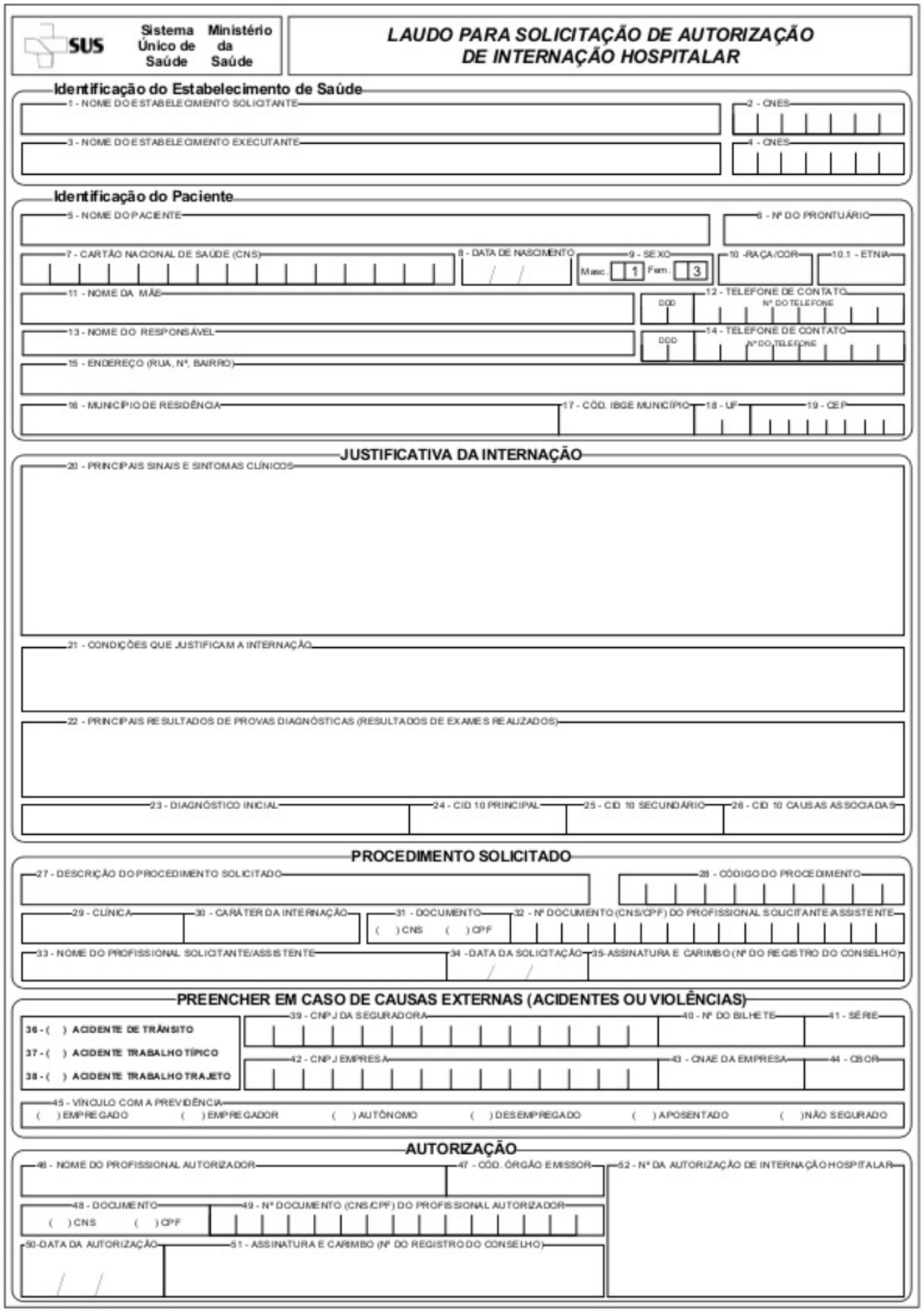

Figure 3.1: *First part of the form used to feed the SIH-SUS dataset.*

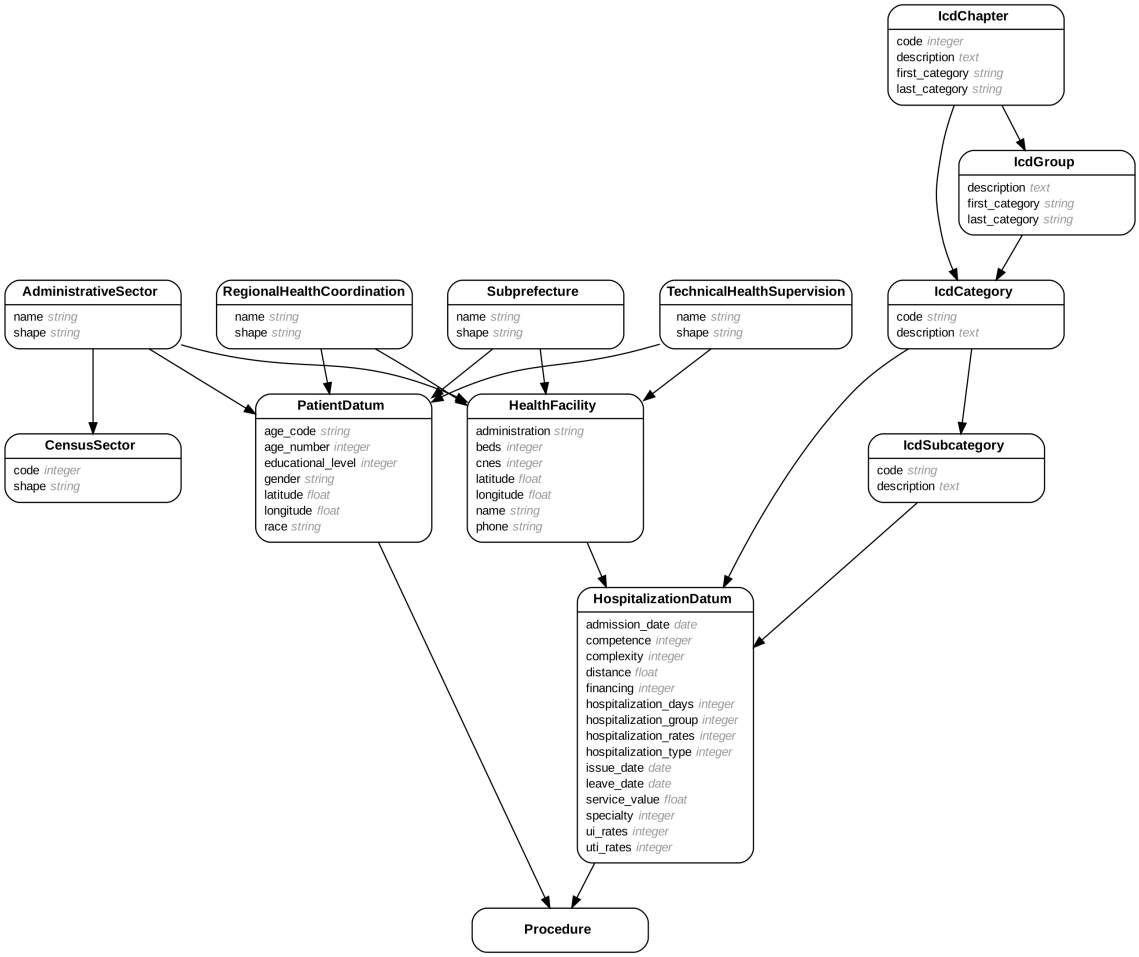

Figure 3.2: *The platform database model to receive the SIH-SUS dataset.*

We divided this model into three main sections: Diagnosis, Location, and Procedure. The first one (Diagnosis) is responsible for cataloging the ICD-10 Categories, Subcategories, Groups, and Chapters. The Location section saves the name and the geographic data structure of the municipality subdivisions, such as Subprefectures, Administrative Sectors, Census Sectors, etc. The Procedure is the core section of the model. It contains the patient anonymized data (age, gender, educational level, etc.), health facility data (number of beds, phone number, etc.), and hospitalization data (admission date, leave date, hospitalization complexity, etc.).

That is how the platform database is structured to receive the SIH-SUS dataset. This representation makes the *Procedure* model (the bottom model in Figure 3.2) the center model, and it is the purpose because of its importance. From the procedure, we can found any information about the patient hospitalization. We make the database queries from the procedures at *Advanced Search* page (more details in Section 4.5), for example.

Chapter 4

The Platform

The current chapter presents the developed platform, its architecture, the design principles followed during the development, the main features, and some usage examples. The platform focuses on promoting better analysis for georeferenced hospitalization data through a data visualization dashboard. The implemented architecture aims to deal with any SIH-SUS dataset, making possible the analysis of public hospitalizations from any region of Brazil.

A good data visualization dashboard should be clear presenting the key information, should be concise summarizing all the data that matters, and must be interactive allowing an easy investigation. The dashboard should also allow the data importation and exportation in different forms, such as PDF and CSV formats. We are dealing with a georeferenced dataset so it is important to the platform to have he ability to visualize the data spatially. The developed platform meet these requirements to deal with the SIH-SUS large dataset.

Below, we present the previous similar work, its results and its problems. After this, we will discuss the possibility of using a *Microservices* architecture and its pros and cons. In the end, we will show the final platform, its features and usage examples.

4.1 The previous solution

There was a previous version of this platform to solve the same situation (enable the visualization of a large georeferenced heath dataset). It followed the monolithic architecture, i.e., it is a software application containing interconnected components designed without modularity. This version started to be developed in early 2017 as a health dashboard application for São Paulo hospitalization georeferenced data. It enables the user to filter the data, search for specifics hospitalizations and visualize this data on the São Paulo city

map, just like shows Figure 4.1.

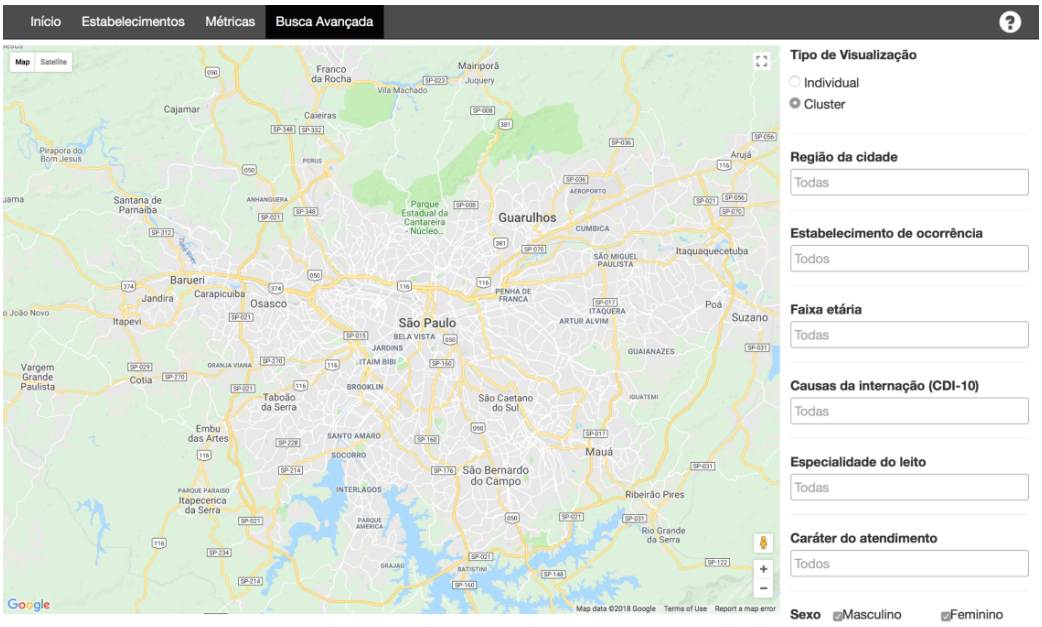

Figure 4.1: *The previous platform developed for health data visualization.*

During the development, the number of added features was increasing, and so the size of the codebase. But it brought to the platform some problems like:

- A significant complexity caused directly by the large codebase;
- Great difficulty and time consuming for a developer understand the code;
- Big challenge to adopt new technologies and programming languages;
- Difficulty to fully test the entire system, due to its complexity, and this lack of testability can release bugs to production;
- Disruption of the whole system if there is one failure.

4.2 The microservices solution

Migrating to microservices can be a good solution to solve the specific problems presented in the last section. Microservices architecture is a service-oriented architecture composed of loosely coupled elements that have bounded contexts (Cockcroft, 2014). It splits the code base into relatively small services for different purposes. Among other benefits, described later in Subsection 4.2.1, this solution would add modularity to the platform, an essential feature when developing large applications.

Furthermore, a service-oriented architecture can also facilitate the creation of different

applications for visualization of any spatial data (see Figure 4.2). It allows the use of different geo-referenced data, such as SIH, SINASC, SIM, and other datasets besides the ones cited at Chapter 3, like safety and mobility data (Pinheiro, 2018).

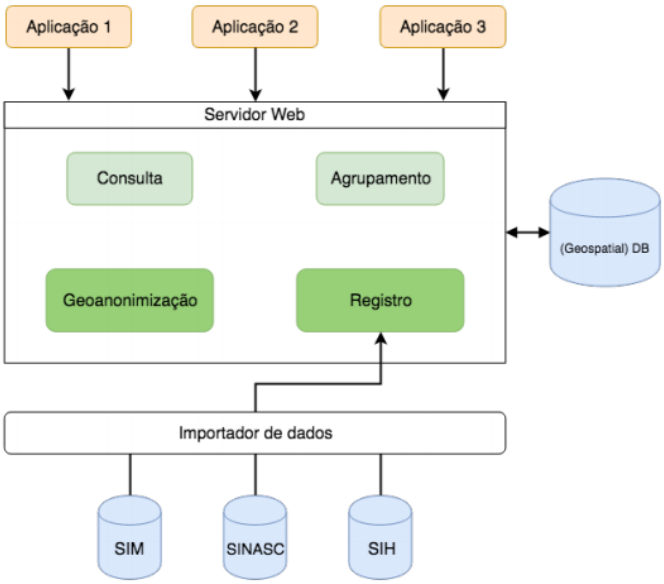

Figure 4.2: *Architecture of GeoMonitor da Saúde platform (extracted from (Pinheiro, 2018)).*

The structure of *Geomonitor da Saúde*, proposed by Pinheiro (2018), contains three main components: the applications, the service, and the datasets. The applications offer different ways to display data from the datasets. In this case, the platform would be just one of these applications. The service is composed of four modules: *Registro*, to receive the geo-located data from the datasets; *Geoanonimização*, to anonymize the patient location; *Agrupamento*, to process data into clusters; and *Consulta*, to deliver this data to the applications. The datasets presented at the bottom of Figure 4.2 could be any dataset containing georeferenced information.

There are many factors when evaluating if this architecture is suitable for our platform (Taibi *et al.*, 2017). The two next sections discuss the advantages and disadvantages concerning the use of microservices in our case.

4.2.1 Advantages to migrating to microservices

Moving to a microservices-based approach makes application development faster and easier to manage (Richardson, 2016). It happens because of its modularity. According to Richardson (2019), the independence of its services promotes many features to a platform.

The architecture modularity decomposes the application into a set of services. It contributes to the understanding of the code and also combats the complexity problem. This migration also promotes deployment and scalability independence. Each service can be deployed independently on the hardware that covers its requirements. This approach would make continuous deployment possible.

The independence between the components of the system promotes the autonomy of teams, delegates placed responsibilities, removes the need for synchronization between them, and fosters the development of parallelization. The developers are free to choose the technology to be employed for a new service. It also facilitates the refactoring of an existing service to new technology or language. Moreover, it enables the use of different programming languages and technologies to various components.

Finally, its relative small services improve the test coverage, because of the division of the code base into modules easier to cover. And also, the failure of one component does not disrupt the whole platform.

However, microservices are not silver bullets, like every other technology (Brooks, 1975). So there are also drawbacks and issues, and we describe them below.

4.2.2 Disadvantages to migrating to microservices

The goal of microservices is to sufficiently decompose the application to facilitate agile application development and deployment (Richardson, 2016). This goal can benefit many different systems, but the decision of re-architecting a platform must be based on real facts and actual issues from the current application.

The migration to microservices results in some aspects to the future system, but not always our needs or available resources meet these points. First, the decomposition of the application into different services aims to facilitate the understanding of one service for a team, because each team usually handles with few services or only one. If there is just one team for the maintenance of the entire application, this division could increase the difficulty to understand and contribute to the system evolution. Moreover, each component could use different technologies, and it becomes arduous to only one team deal with this kind of complexity. In our case, we are developing this platform to be used and maintained by public entities. They could not sustain multiple teams to manage the platform like companies usually do.

Second, microservices are focused on solving industry problems, and their solutions may not be significant or even possible to us. Each service of the architecture carries its specific deployment, resource, scaling, and monitoring requirements (Richardson, 2016).

In this context, the limited available resources at public entities might be an infrastructure bottleneck and must impact important topics like deployment and scalability.

Section 4.4 shows how we get around these points with the final architecture. But first, we expose below the design principles followed during the development.

4.3 Design Principles

These design principles are used to build a feasible, robust, and concrete application architecture. They contribute to the software extensibility and maintainability. We considered the benefits of using the design principles below as the main practices to create resilient software.

The platform needs to contain **modularity**, at some level. It facilitates the comprehension of the code and increases its cohesion due to its division in logical components. It also improves communication between parts of the code. Moreover, it enables the full test coverage because it is easier to test small modules than the entire code base.

The **reuse of open source software projects** can improve faster code development, save costs, and enhance code reliability. However, it is important to be aware of the package quality and to the use of well known open source projects with an active developer community.

We are also following the ***Don't Repeat Yourself* principle** (DRY). Dave Thomas, the author of *The Pragmatic Programmer*, said that the idea behind DRY is the need for every piece of knowledge in the development of something to have a single representation. Furthermore, a piece of knowledge can be either the build system, the database schema, the tests, or even the documentation.

Finally, we believe the **internationalization** of the application is a relevant point. It is used to provide multi-language support. With that, we can make it easier to customize and extend the platform for other languages. We consider other countries could want to know this platform and base on this one to develop a similar solution for them.

According to these design principles we developed the platform, described in the section below.

4.4 The final solution

This work developed an architecture for a platform that enables the visualization of any SIH-SUS dataset. The final architecture intends to bring benefits over the mono-

lithic version, and overcome the microservices disadvantages, described in Subsection 4.2.2.

To suit these conditions, the software design pattern **Model–View–Controller** (MVC) were selected as the core of the platform. This architectural pattern, first described by Krasner and Pope (1988), is composed of three objects: *Model, View*, and *Controler*. The communication between them complies with the diagram shown in Figure 4.3.

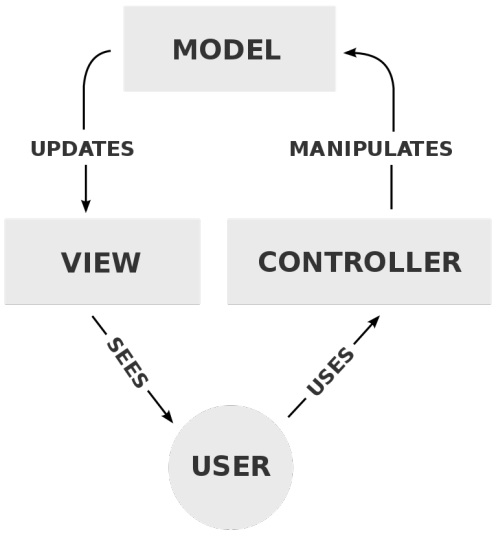

Figure 4.3: *Diagram of interactions within the MVC pattern.*

Model components are those parts of the system application that simulate the application domain. In our case, the models represent the SIH-SUS dataset inside the platform. It is described in Subsection 3.2.2. *Views* deal with everything graphical. They display aspects of their models. As detailed in the end of the Chapter 3, the hospitalization *Procedure* model is the main model in the platform, and it updates its view called *Advanced Search* page (Section 4.5 describe this and other views). *Controllers* contain the interface between their associated models and views and the input devices. They send messages to the model and provide the interface between the model with its related views (Krasner and Pope, 1988).

According to Gamma *et al.*, 1995, usually known as the "Gang of Four" book:

> *Before MVC, user interface designs tended to lump these objects* (models, views, and controllers) *together. MVC decouples them to increase flexibility and reuse.*

This flexibility decreases the code complexity and brings modularity to the system. So, the platform obtains, on some level, benefits such as code reuse, high cohesion (due to MVC logical grouping), joint development (because of the code modularity and independence),

and others. It will be better discussed in next chapter (Chapter 5).

The end of the Section 3.1 describes that the SIH-SUS datasets may differ subtly from region to region, so we also need a service to receive the different SIH datasets and adapt the application according to them. Figure 4.4 shows the effect of the simple **database generalizer**. It imports the SIH-SUS dataset through a CVS table containing some predefined columns. By doing this, we can generalize any SIH-SUS dataset to make possible its visualization through the platform.

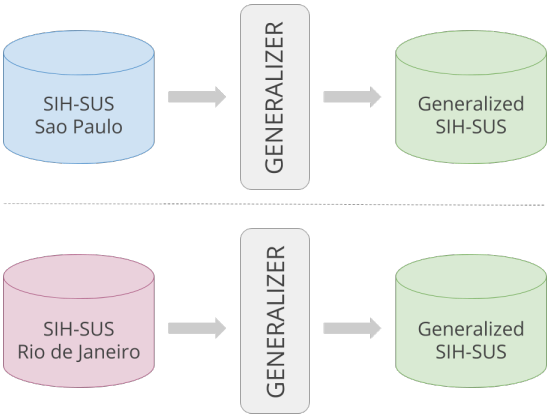

Figure 4.4: *Diagram of the database generalizer.*

Moreover, the platform provides a module for **location anonymization**. Anonymization is an essential procedure to protect individual privacy, including health data. The location anonymization aims to generalize the patient location and keep the data reliability. The solution used by SMS-SP is the transition of the patient location to its census sector (or census tract) centroid. This handwork solution is too slow and expensive for SMS-SP, so we developed a module to automatically convert the patient location to the census sector centroid, in terms of latitude and longitude.

Through this architecture, we developed some essential features to the platform. They are described in the section below.

4.5 Features

From the imported SIH-SUS dataset into the platform, it is possible to view the hospitalizations and health facilities locations on the map. In this project, we are using data from São Paulo city, specifically. But it is possible to import SIH-SUS datasets from other regions of the country.

The platform is a web application, and it is divided into different pages for different purposes. The main page is called **Advanced Search** (see Figure 4.5), and it contains

the view of the *Procedure* model (as described in the end of Subsection 3.2.2). It contains the patients' geo-anonymized locations, the locations of the health facilities they were admitted, and the hospitalization concentration by region.

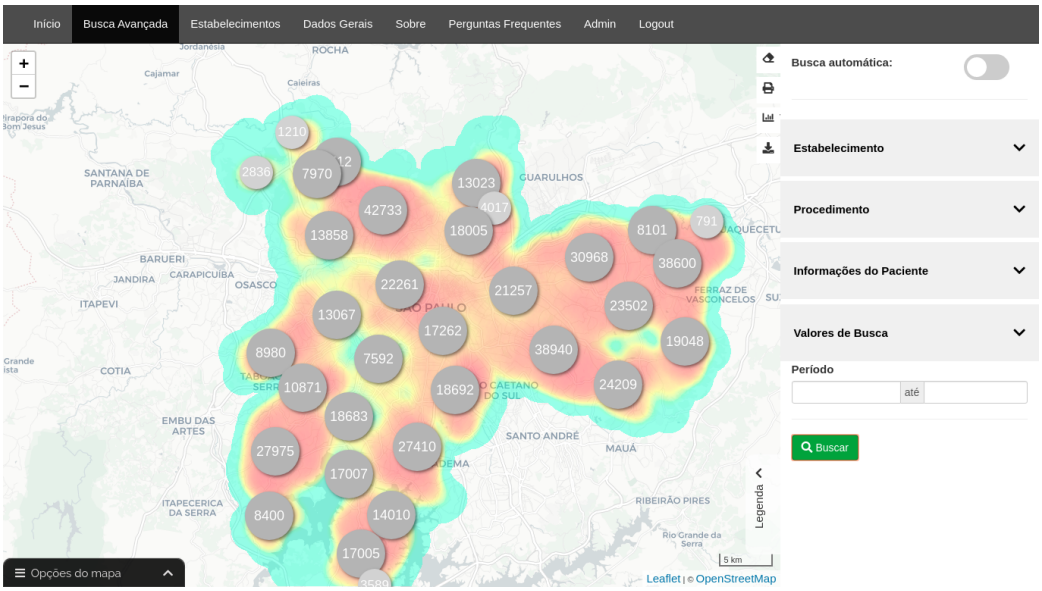

Figure 4.5: *Advanced Search page displaying all São Paulo hospitalizations in 2015.*

Inside this page, we can find the right side menu with three sections containing fields that work just like filters. These three sections are: Health facility (*Estabelecimento*), Procedure (*Procedimento*), and Patient data (*Informações do paciente*). These fields enable the search of hospitalization information by the health facilities, such as its administration or its name, by the hospitalization data, including the diagnosis or the costs, and by patient data, such as their age or gender. We can search by these fields and receive the hospitalization data that satisfies the query.

This engine complies with the need to access the information in a large dataset. The data is visualized in a map through a heatmap and through clusters representations. The colorful heatmap represents the intensity of the hospitalizations in each region. On the other hand, the clusters represent the quantity of these hospitalizations, where the darker gray cluster contains more than 10k grouped data, the lighter gray contains less than 10k, and the small orange cluster (Figure 4.8 shows it) contains the hospitalizations in the same census sector.

The second page is called **Health Facilities**. This page contains more information on health facilities, such as its location at the map, its specialties, the number of beds, and so on. Figure 4.6 shows an example containing the information of a hospital selected previously (*Hospital Santo Antonio*).

The **General Data** page is composed of a large variety of charts. Each possible attribute

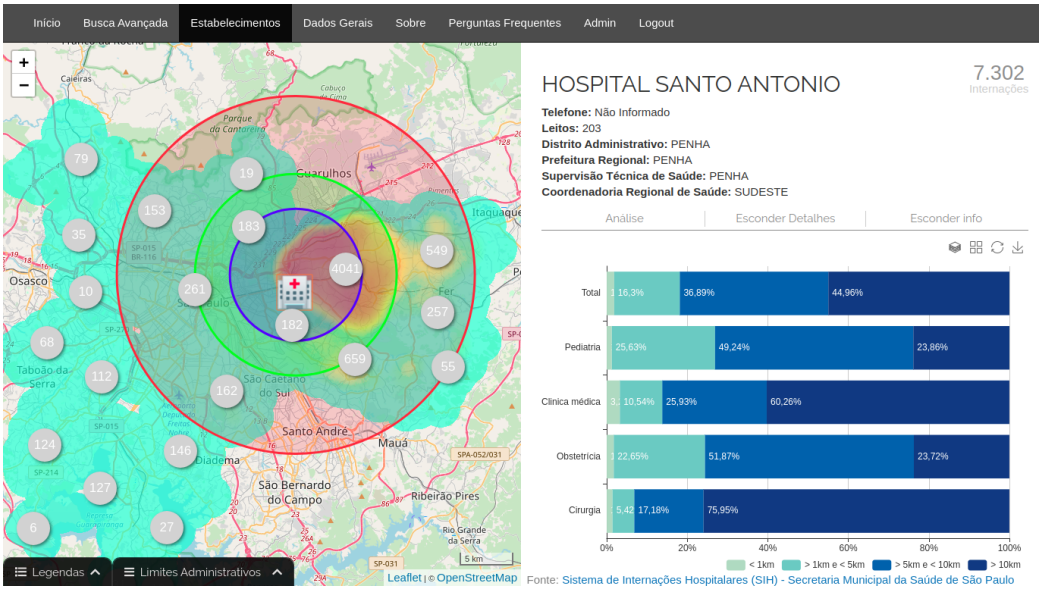

Figure 4.6: *Health Facilities page displaying Hospital Santo Antonio hospitalizations.*

of the database can be displayed in a chart on this page. There are some examples in Figure 4.7.

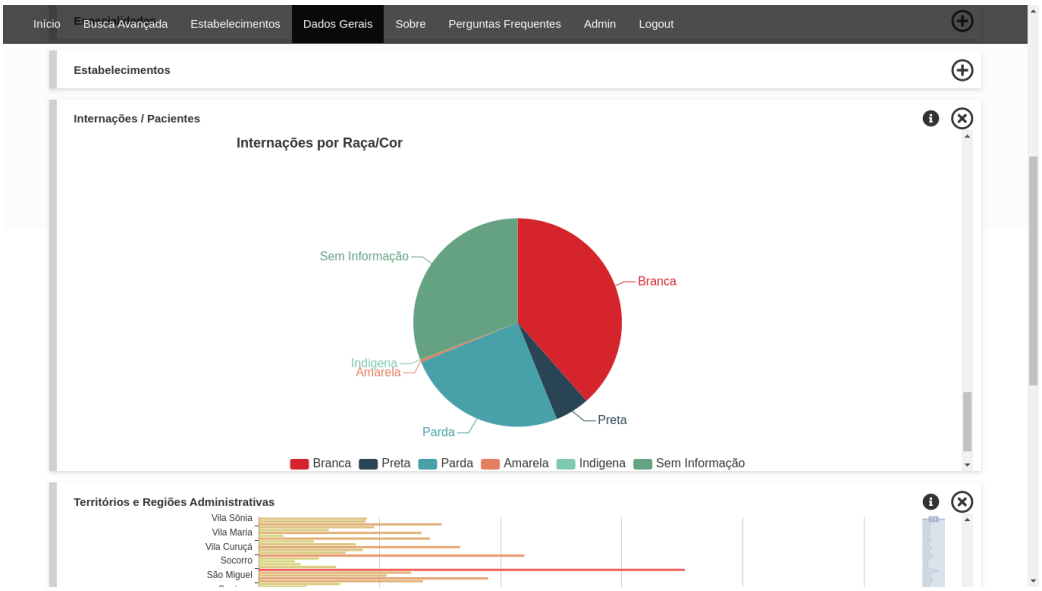

Figure 4.7: *Part of the General Data page.*

These features were designed together with the consultation of SMS-SP, so it intends to be a great tool for health public management purposes.

4.6 Usage examples

To illustrate the platform usage, here we show how these features can be used to become an efficient tool for public health management. Through this platform, the user can evaluate the health situation of specific areas, the hospitalizations distribution on the map over time, the area, the diagnosis, and over many other variables. The user could also be aware of the distance traveled by the patient to be hospitalized. This information can be retrieved via the Health Facilities page.

For example, Figure 4.6 shows on the map the location distribution of patients attended by *Hospital Santo Antonio*. Moreover, it shows the distance traveled by the patient by each hospitalization specialty on the right-side chart. The last chart bar deep blue partition indicates that most of the surgical procedures corresponds to patients that traveled more than 10km to the medical procedure.

The use of the Advanced Search page to analyze the distribution of a specific diagnosis is another good example. Figure 4.8 shows the hospitalizations due to dengue, in 2015. The user could retrieve this information over different periods and analyze the evolution of this scenario.

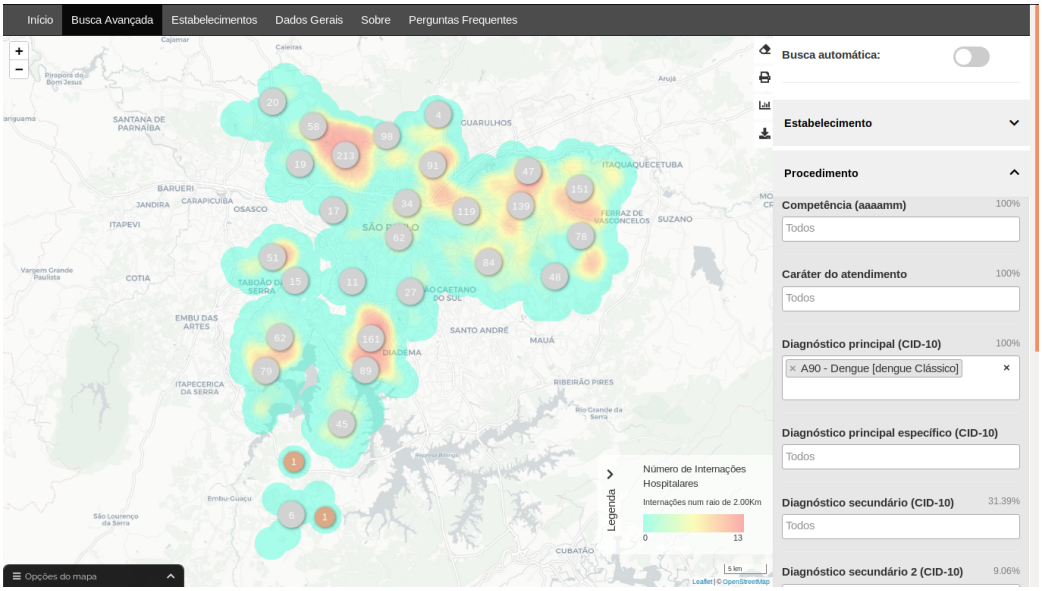

Figure 4.8: *The platform displaying the hospitalizations due to dengue, in 2015.*

There are a large number of possibilities and combinations of information provided by this platform. The Advanced Search page and General Data page offers many different variables to be combined and retrieved, such as patient diagnosis, age, race, gender, region, and so on. As a result, this rich platform could produce good evidences to establish effective health policies.

Chapter 5

Final remarks

We can observe the increasing use of technology tools in governments in the last years. It is happening due to the digital transformation of the public sector and the innovation of tools used to improve their services, such as Geographic Information Systems (GIS), public services to citizens, and the interoperability of the services. Smart city projects realize this same movement to provide innovative applications and promote the technological evolution in governments. However, it is still a recent initiative, and it can be limited by technical evolution, financial resources, and organizational philosophies.

Despite this initiative, developing platforms for governments or public entities can be a challenging task. The limited available resources at public entities might be an infrastructure bottleneck to implement some technologies. Also, public entities may not sustain multiple teams to manage the platform like companies usually do, so it limits the use of some technologies. The development of the platform dealt with this situation.

The platform architecture choice faced this limitations and opted to use the Model-View-Controller software design pattern as the core of the platform instead of the Microservices architecture. I happened because the microservices implementation requirements did not correspond with the available resources.

In this project, we developed a platform for data visualization of a large body of georeferenced hospitalization data. Moreover, it aimed at being able to receive and execute SIH-SUS datasets containing hospitalization data from any region of Brazil. Mainly, this project can retrieve crucial detailed information on hospitalizations spatial distribution, and contribute to the creation of public health policies.

This project dealt only with SIH-SUS datasets, but there are many different georeferenced health datasets at SUS, managed by DATASUS (see Chapter 3). Moreover, there are

other available georeferenced datasets about mobility, education, security, and so on. As future work, it is possible to modify the current application to use these other datasets. Some parts of the software, like the database model, should be different, but the most significant part of the frontend application could be reused.

References

[Agafonkin 2016] Vladimir Agafonkin. *Clustering millions of points on a map with Supercluster*. Mar. 2016. URL: https://blog.mapbox.com/clustering-millions-of-points-ona-map-with-supercluster-272046ec5c97 (cit. on pp. 9, 10).

[Anderberg 1973] Michael R. Anderberg. *Cluster Analysis for Applications*. Ed. by Michael R. Anderberg. Probability and Mathematical Statistics: A Series of Monographs and Textbooks. Academic Press, 1973. DOI: https://doi.org/10.1016/B978-0-12-057650-0.50007-7. URL: http://www.sciencedirect.com/science/article/pii/B9780120576500500077 (cit. on pp. 5, 6).

[Anthopoulos *et al.* 2015] Leonidas Anthopoulos, Christopher Reddick, Irene Giannakidou, and Nikolaos Mavridis. "Why e-government projects fail? An analysis of the Healthcare.gov website". In: *Government Information Quarterly* (Aug. 2015). DOI: 10.1016/j.giq.2015.07.003 (cit. on p. 2).

[Brasil. Ministério da Saúde. Secretaria Executiva 2002] Brasil. Ministério da Saúde. Secretaria Executiva. *DATASUS Trajetória 1991-2002*. Brazilia, DF, BR, 2002 (cit. on p. 12).

[Brasil. Ministério da Saúde. Secretaria de Vigilância em Saúde 2005] Brasil. Ministério da Saúde. Secretaria de Vigilância em Saúde. *Guide to Epidemiological Surveillance*. Brazilia, DF, BR: Ministério da Saúde, 2005. ISBN: 85-334-1047-6 (cit. on pp. 11, 12).

[Brazil 1991] Brazil. *Federal Official Gazette of Brazil (DOU) no 100/1991*. Apr. 1991. URL: http://www.planalto.gov.br/ccivil_03/decreto/1990-1994/D0100.htm (cit. on p. 11).

[Brooks 1975] Frederick Brooks. *The Mythical Man-Month*. Addison-Wesley, 1975. ISBN: 0-201-00650-2 (cit. on p. 20).

[Cockcroft 2014] Adrian Cockcroft. *Fast Delivery*. nginx.conf. Oct. 2014. URL: https://www.nginx.com/blog/microservices-at-netflix-architectural-best-practices/#videos (visited on 05/12/2019) (cit. on p. 18).

[Demographia 2019] Demographia. *Demographia World Urban Areas (15th Annual Edition: April 2019)*. Apr. 2019. URL: http://www.demographia.com/db-worldua.pdf (cit. on p. 13).

[Gamma *et al.* 1995] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN: 0-201-63361-2 (cit. on p. 22).

[Instituto Brasileiro de Geografia e Estatistica - IBGE 2019] Instituto Brasileiro de Geografia e Estatistica - IBGE. *Cidades Panorama São Paulo*. 2019. URL: https://cidades.ibge.gov.br/brasil/sp/sao-paulo/panorama (cit. on p. 1).

[Jain and Dubes 1988] A. K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN: 0-13-022278-X (cit. on pp. 6, 7).

[Jain, Murty, *et al.* 1999] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data Clustering: A Review". In: *ACM Comput. Surv.* 31.3 (Sept. 1999), pp. 264–323. ISSN: 0360-0300. DOI: 10.1145/331499.331504. URL: http://doi.acm.org/10.1145/331499.331504 (cit. on pp. 5, 7, 8).

[King 1967] Benjamin King. "Step-Wise Clustering Procedures". In: *Journal of the American Statistical Association* 62.317 (1967), pp. 86–101. DOI: 10.1080/01621459.1967.10482890. eprint: https://amstat.tandfonline.com/doi/pdf/10.1080/01621459.1967.10482890. URL: https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1967.10482890 (cit. on p. 7).

[Krasner and Pope 1988] G. Krasner and Stephen Pope. "A cookbook for using the model - view controller user interface paradigm in Smalltalk - 80". In: *Journal of Object-oriented Programming - JOOP* 1 (Jan. 1988) (cit. on p. 22).

[Lance and Williams 1967] G. N. Lance and W. T. Williams. "A general theory of classificatory sorting strategies: II. Clustering systems". In: *The Computer Journal* 10.3 (Jan. 1967), pp. 271–277. ISSN: 0010-4620. DOI: 10.1093/comjnl/10.3.271. eprint:

http://oup.prod.sis.lan/comjnl/article-pdf/10/3/271/1333425/100271.pdf. URL: https://doi.org/10.1093/comjnl/10.3.271 (cit. on p. 6).

[MacQueen 1967] J. MacQueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, CA, USA: University of California Press, 1967, pp. 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992 (cit. on p. 8).

[Paim *et al.* 2011] Jairnilson Paim, Claudia Travassos, Celia Almeida, Ligia Bahia, and James Macinko. "The Brazilian health system: History, advances, and challenges". In: *The Lancet* 377 (May 2011), pp. 1778–97. DOI: 10.1016/S0140-6736(11)60054-8 (cit. on p. 11).

[Pinheiro 2018] Eduardo Pinheiro. "Um serviço para o desenvolvimento de aplicações de visualização de dados de saúde georreferenciados". In: (Nov. 2018) (cit. on p. 19).

[Prefeitura do Município de São Paulo 2016] Prefeitura do Município de São Paulo. *Epidemiologia e Informação na Secretaria Municipal de Saúde de São Paulo no período de 1989 a 2001: elementos para escrita de uma história*. São Paulo, SP, BR, Sept. 2016 (cit. on p. 13).

[Richardson 2016] C. Richardson. *Microservices - From Design to Deployment*. 2016 (cit. on pp. 19, 20).

[Richardson 2019] C. Richardson. *Microservices Patterns: With Examples in Java*. Manning Publications, 2019. ISBN: 9781617294549. URL: https://books.google.com.br/books?id=UeK1swEACAAJ (cit. on p. 19).

[W. Raghupathi and V. Raghupathi 2014] W. Raghupathi and V. Raghupathi. "Big data analytics in healthcare: promise and potential". In: (Feb. 2014) (cit. on p. 5).

[Sneath and Sokal 1973] Peter H. A. Sneath and Robert R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. San Francisco, CA, USA: W. H. Freeman and Co., 1973. ISBN: 0716706970 (cit. on p. 7).

[Stopa *et al.* 2017] Sheila Rizzato Stopa *et al.* "Use of and access to health services in Brazil, 2013 National Health Survey". en. In: 51 (2017). ISSN: 0034-8910. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0034-89102017000200308&nrm=iso (cit. on p. 11).

[Taibi *et al.* 2017] Davide Taibi, Valentina Lenarduzzi, and Claus Pahl. "Processes, Motivations and Issues for Migrating to Microservices Architectures: An Empirical Investigation". In: *IEEE Cloud Computing* 4 (Oct. 2017). doi: 10.1109/MCC.2017.4250931 (cit. on p. 19).

[Tsai *et al.* 2015] Chun-Wei Tsai, Chin-Feng Lai, Han-Chieh Chao, and Athanasios V. Vasilakos. "Big data analytics: a survey". In: *Journal of Big Data* 2.21 (Oct. 2015). doi: 10.1186/s40537-015-0030-3 (cit. on p. 5).

[Wen *et al.* 2019] Melissa Wen *et al.* "Leading successful government-academia collaborations using FLOSS and agile values". In: (Oct. 2019) (cit. on pp. 2, 3).

[Zahn 1971] C. T. Zahn. "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters". In: *IEEE Transactions on Computers* C-20.1 (Jan. 1971), pp. 68–86. doi: 10.1109/T-C.1971.223083 (cit. on p. 8).