ISSN 1828-5961

Carla dos Santos Sá
Carla is a software developer who works for an IT Solutions company as Systems Analyst. She has worked at development projects of software development like SAAS (Software As A Service), web applications and a specific project involving GIS area. Carla holds a Bachelor Degree in Computer Science from University Center of Belo Horizonte (UNIBH).

Danilo Marques de Magalhães
Bachelor (2010), Master (2013) and PhD (beginning in 2017) in Geography at UFMG. He is Professor at University Center of Belo Horizonte – UNIBH where teaches GIS, Remote Sensing and Geoprocessing. He develops researches and works with applications of drone images, landscape metrics, and other spatial analysis models.

A Data Collection Application for Geosciences Professionals

Uma Aplicação de Coleta de Dados para Profissionais das Geociências

This article presents the process of elaborating a web platform that allows the georeferenced data collection in the Twitter API. This platform is intended for people who do not have computer language knowledge, including GIS users, since it simplifies the data collecting method from this social network. Moreover, it enables the realization of up to twelve types of queries that result in georeferenced data. As a case study, queries were made with the ENEM term and spatial analyzes were performed and they are presented in heat maps. The results show that there are no other applications with the same particularities of the present proposal and show possibilities to perform spatial analyzes through the data capture in social networks and GIS processing.

O trabalho apresenta o processo de elaboração de uma plataforma web que permite coletar dados georreferenciados na API do Twitter. Essa plataforma se destina ao público leigo em linguagem computacional, incluindo usuários de SIG, pois simplifica o método de coleta de dados dessa rede social. Além disso, possibilita a realização de até doze tipos de consultas que retornam dados georreferenciados. Como estudo de caso, foram realizadas consultas com o termo ENEM e realizadas análises espaciais que são apresentadas em mapas de calor. Os resultados mostram que não existem outras aplicações com as mesmas particularidades da presente proposta e evidenciam possibilidades de realização de análises espaciais por meio da captura de dados em redes sociais e processamento em SIG.

Keywords:
GIS, Geoprocessing, Twitter API, Web Application, Programming.
Palavras-chave:
SIG, Geoprocessamento, Twitter API, Aplicação Web, Programação.

ISSN 1828-5961

1. INTRODUCTION

Geographic Information Systems (GIS) are a set of computational tools for collecting, storing, retrieving, analyzing and viewing spatial data. GIS are composed of three main components: hardware, software and users. The hardware is responsible for the physical part and the access of the systems software, and can also make the communication with external devices, like mobile equipment used in data collection. The software is responsible for four basic actions: input and data verification; storage and data management; output and data presentation; transformation, analysis and data modeling. The users are the community that use and/or develop GIS (Burrough, Lloyd & McDonnell, 2015).

One of the tasks implemented in GIS software, is the geoprocessing which consists in the spatial analysis of geographic data to measure georeferenced relationships and properties (Machado, 2017). According to Xavier-da-Silva (2009), geoprocessing uses georeferenced databases applying computational methods that allows scans of territorial incidences, for the purpose of transforming data in knowledge to support decision-making. Geoprocessing can also be used in online applications. One of the advantages of this use is the facility to share information between the scientific community (Hofer, 2014).

According Machado (2017), GIS have been used in studies about disease distribution analysis, risk population mapping, resources allocation and interventions planning. Camboim & Sluter (2013, p. 1129) describe that one of the problems that users of geospatial data find, "is the difficulty to find relevant information for given use". According to the authors, this is an issue that National Spatial Data Infrastructures (NSDI) should strive to facilitate the access of such data to users.

Along with the difficulty to find relevant information, comes the difficulty to find professionals in the geosciences areas that use GIS in research and decision-making to use data collection tools. An example of this is the use of APIs (Application Programming Interfaces) that have as target users the computer science public.

In the last couple of years, companies like Facebook, Google and Microsoft have been using API in the creation of plug-ins of their applications. This allows that the developing community uses the API services and contribute to the growing of the applications (Pizetta, 2014). Twitter also offers the use of their API to platform developers. The services of API queries are used in data collection tasks for analysis of different nature. Some of those queries enable the gathering of georeferenced data, by means of obtaining the location of posts (Twitter, 2017).

The complexity level in the use of an API requires previous knowledge of tools and languages programming appropriate for the creation of a structure that can consume and treat the raw data. This complexity is a factor which can restrict the public who need to use this kind of service, like professionals of geosciences field, which use georeferenced data collection for spatial analysis for the purpose to generate useful knowledge to better targeting and informing decision making.

In this way, the research problem raised in the present work is: how to improve the experience of professionals in the geosciences field with little or no programming knowledge during the collection of data available in systems that use API?

The motivation for the approach was since data collection was one of the most important steps in geoprocessing, and due to the access difficulty or lack of specific knowledge of professionals in geosciences areas, it is intended to facilitate the access to a wider public, allowing that more scientific research can be done by using these means.

Due to this context, the general objective of the current work is the development of an application that makes inquiries to Twitter API to facilitate the data collection step and allow a better interaction of professionals in geosciences areas without specific programming knowledge. The specifics objectives are: Identify and specify the Twitter API services that collect georeferenced data; Allow search filtering only with georeferenced results in queries; Enable the user to download the collected data in structured formats for GIS software.

2. LITERATURE REVIEW

In this section, the existing applications are presented, which relate to the proposed application. A short description of its functionalities is performed and a comparison between them at the end.

2.1. APIGEE

Apigee is an APIs development and management platform. The platform offers an environment in which the companies can expose their requisition services.

The model is based on the requisition requested by the client-side developer, that when being processed returns the data in XML (eXtensible Markup Language) or JSON (JavaScript Object Notation) format (Apigee, 2017). It is presented the architecture of the requisitions model platform (Figure 01).

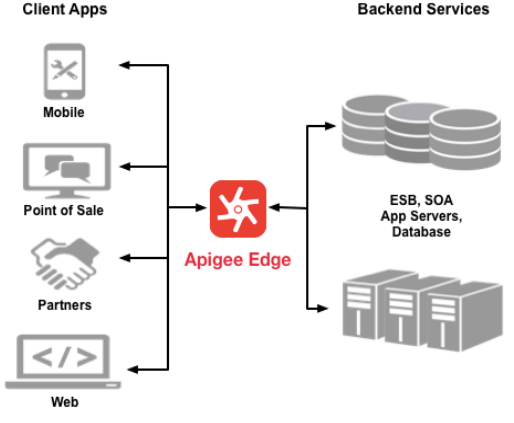

Figure 1 - Architecture of the requisitions model of Apigee. Source - Apigee, 2017.

ISSN 1828-5961

Some companies that use this platform are: Bing, Blogger, Facebook, Flickr, Foursquare, Google, Instagram, LinkedIn, Reddit and Twitter. The access to the services of the companies is carried out by means of a console application, in which it is possible obtain a list of the API services and perform the necessary requisitions.

One of the most frequent APIs used in data collection tasks is the Twitter API, which is also available in the Apigee platform, and all the Twitter services can be accessed at Apigee's own website. To obtain access to the API services, the user needs to make the authentication at the platform, by logging in to the application that the user wants to use. In the case of the Twitter API, the user connects to his personal account and gets access to the query as well as to other services and various functions of the social network.

2.2. SPATEXT

Spatext is an add-in implemented in Python language version 2.7 available at ArcGIS© software. Among its functions, nine tools exist which allow data collection at social medias like Twitter, YouTube, Wikimapia, Instagram, Foursquare and Panoramio. Spatext has an advantage over the social network APIs, to be run directly at a GIS interface, making the data integration process easier to support decision-making in urban and regional planning (Massa, Campagna, 2016; Campagna; Massa, 2014).

The operation flow of Spatext follows accordingly with the data collection, management and analysis functions. Visualization of Spatext architecture (Figure 2):

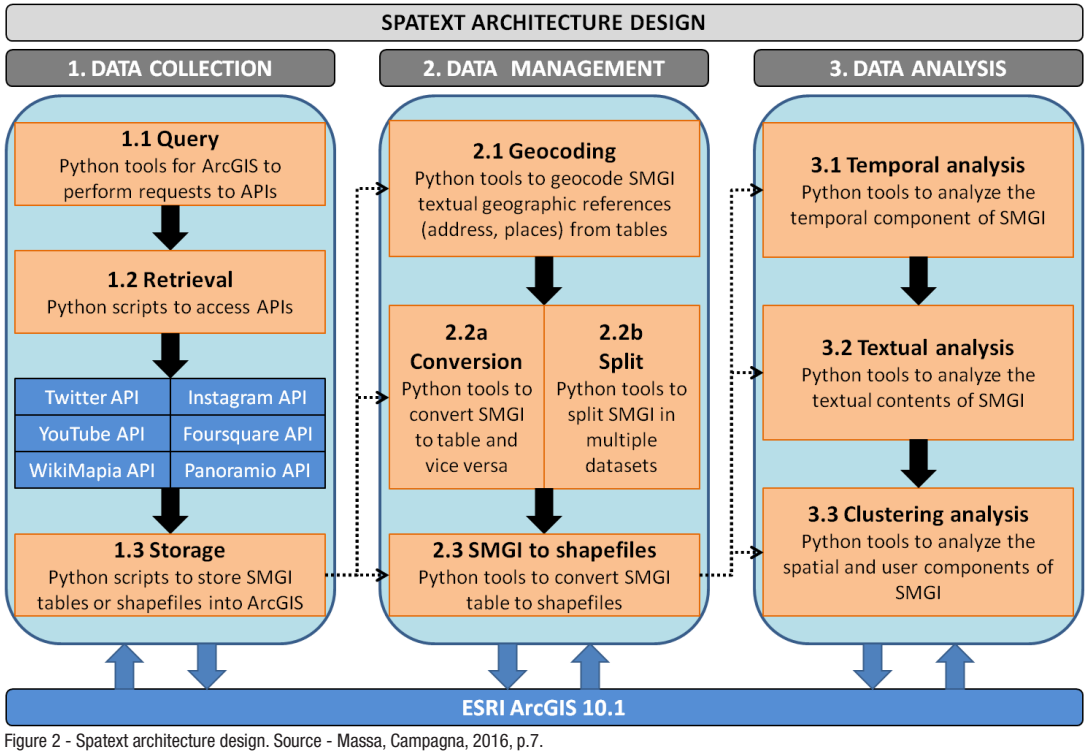

Figure 2 - Spatext architecture design. Source - Massa, Campagna, 2016, p.7.

2.3. COMPARISONS

Relating the main characteristics of these two applications described, it is possible to make the following analysis: both applications, Apigee and Spatext, perform queries to Twitter API and allow the search of georeferenced data. Apigee doesn't have a filter for this data. For the use of Apigee it isn't necessary to have any specific knowledge of programming language or training tools to use it. Due to the fact of Spatext running directly at ArcGIS interface, it is necessary that the user have a notion of the application operation and the knowledge of Python language is also necessary, as Spatext is based on scripts of this language. The result of Apigee queries are available only for visualization in JSON format, not allowing that the user download the queries returned.

The result of Spatext queries is stored directly at ArcGIS database, allowing that the user makes integrations with the projects of the application. The access of Apigee is open to the public, since the user has an application account to be used. For the current study, it wasn't possible to access directly Spatext, due the fact that the tool has been experimental only and have not been distributed publicly, being accessible only on scientific work that used the tool.

From the comparative, it is verified that the proposed application differs from the others by the following characteristics: the application offers a georeferenced data search filter; it isn't necessary that the user have specific knowledge of programming language and/or training of tool usage; the user can view the results at the application interface and also download the collected data; it is intended that the user can have free access to the application after being in full operation, ensuring that it is open for public use.

3. THEORETICAL BASIS

A data collection step in GIS can use varied information

ISSN 1828-5961

sources, in several application fields. In order to facilitate the data collection, some techniques can be used in the process. Marres & Weltevrede (2012) quote Web Scraping as a technique to automatize online data collection. They explain that the technique is widely used for data collection, analysis and visualization.

The following are detailed concepts of the data collection techniques studied and the main definitions necessary to understand the current work.

3.1. WEB SCRAPING

Web Scraping refers to a technique of information extraction. Such technique allows that the data of different locations can be collected together. The process allows that data can be collected from image pages, localization or search by keywords, to databases study (Marres, Weltevrede, 2012).

Scrapers basically transform unstructured data and store them in structured databases (Vargiu, Urru, 2012). An example of how to use Web Scraping, is the research of Daiya et al. (2017) in which a summarization system of Twitter news is created. According to them, the scraper component deals with extraction of all contents related to certain web search. The system allows that users search for words or specific phrases and after processing the search, returns the summarized tweets of certain searches. Meireles & Silva (2015) explain that the data collection using Web Scraping technique is limited to content available in HTML (HyperText Markup Language), XML and API exhibition. Because the application of the current work is based on API services, only such an approach will be detailed in the Theoretical Foundation.

3.1.1 APPLICATION PROGRAMMING INTERFACES - API

An API is a set of subroutine definitions, protocols and tools provided by application developers which allows that others can develop software which are linked to these applications (Macnamara, 2017).

Inácio Júnior (2007) explains that an API determines the software functionalities that can be used in other software. In other words, it defines the requisition address of the elements in order to allow that tasks are carried out and data is exchanged.

The author still quotes the following characteristics of an API:

- Information hiding: APIs are capable of "hiding" private modules, which restrict the access to internal logic. For being a principle of low coupling and high cohesion, it decreases the dependencies between the system parts.
- Separation specification/implementation: the API interface separates its visible methods, in order to allow different implementations by the designers.
- Interoperability: APIs can perform connections with distinct systems, furthermore, allow integration with different languages.
- Stability: the APIs development tends not to change constantly, keeping the compatibility with the systems and independence between providers and API clients.

3.1.1.1. TWITTER API

Twitter provides two access ways for the public API. The first way (REST - Representational State Transfer) is used to collect historical data, which usually are tweets before the date and time of collection. The second way (Streaming) allows data collection in real time without interruption, as long as users post contents (Macy, Mejova, Weber, 2015).

The default procedure to use both API ways is described as follows:

- Twitter authentication using Open Authentication (OAuth) mechanism.
- Creation of Twitter API call passing the appropriate parameters.
- Receiving and processing the response of the API.

The access keys for the API access are validated at the authentication step. After a developer account is created for an application, authentication tokens are generated, which should be used as configuration parameters. The access tokens are the parameters "Access Token" and "Access Token Secret". The access keys are the parameters "Consumer Key" and "Consumer Secret".

3.2. CROWDSOURCING

Crowdsourcing is described as a set of data collection techniques contributed by citizens without specific training, which enables the creation of databases. The combination between crowdsourcing and GIS results in crowdsourcing mapping, which allows the generation of large volumes of spatial data that can be used with diagnostic bases and included in both management and territorial planning actions (Borges, Davis Júnior, Jankowski, 2016).

Garcia-Molina et al. (2016) explains that crowdsourcing is the act of recruiting people to perform tasks that are considered too difficult for a computer to perform on its own. The authors quote sentimental analysis, visual classification and data comparison as some of the problems while working with crowdsourcing.

At the following sections, two processes are described to obtain information from crowdsourcing, according to Borges, Davis Junior and Jankowski (2016).

3.2.1 VOLUNTEERED GEOGRAPHIC INFORMATION - VGI

VGI is a concept directly linked to geographic data collection from citizen participation, with a clear purpose. Data collection when made by APIs does not need the participant users' knowledge. The first step to collect data by means of VGI, is the mobilization of the users by which they can contribute with their knowledge (Borges, Davis Júnior, Jankowski, 2016).

Campagna et al. (2015) describes VGI as the content generation by users that act as volunteered sensors. The use of this methodology is extremely useful in research about emergency management, environmental monitoring, spatial planning and crisis management.

Calafiore et al. (2016) presents some case studies that perform experiments with VGI systems, furthermore describes the proposed research in question. Among the case studies presented, the experiment of

ISSN 1828-5961

Borges, Jankowski and Davis Junior (2015) should be mentioned, in which the authors collected data from Twitter during the Brazilian Cup matches, and after analyses, they demonstrated the posts in groups by match, nationality and geotag. The proposal of Calafiore et al. (2016) is already based on a methodology that increases the participation in planning processes. The tool collects geographic information and concentrates on organization tasks and data analysis in order to support the elaboration of spatial changes.

3.2.2. SOCIAL MEDIA GEOGRAPHIC INFORMATION - SMGI

SMGI is a data collection process, so as VGI, but it doesn't have a specific objective on what is captured. Identifying the foundation and application of the information is a task for the researcher. The data can be used for mobility research purposes or geographic concentration in certain location points (Borges, Davis Júnior & Jankowski, 2016).

Campagna (2016) defines SMGI as geographic information contents implicit or explicit collected through social networks or mobile applications. These contents can be in text, images, videos or audios formats, referenced spatially.

The use of SMGI processes has some obstacles which limits its usage. Massa and Campagna (2016) present the lack of friendly tools and the complexity in treating a lot of data as main obstacles. Such problems are due to the growing volume of information and the specificity of the data structures, needing adjustments in the analysis methodologies and traditional development.

4.METHODOLOGY

With the aim of developing the application proposed in this work, the methodological flowchart is presented to explain the procedures adopted (Figure 3):

The process was carried out throughout the literature review procedure in books, scientific articles, publications and periodicals, theoretical background the concepts and GIS applications, geoprocessing, data collection using Web Scraping techniques,

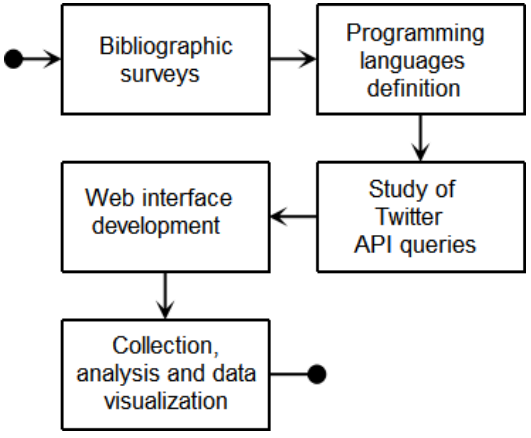

Figure 3 – Flowchart of the methodological procedures. Source – The authors, 2017.

APIs, crowdsourcing, VGI and SMGI. Besides the foundation, continuous improvement of the applied methods was sought, such as the analysis of work already done in the application area to improve the development of the present application.

To develop the proposed application, it was necessary to define the programming languages to be used, considering the environment in which the application will be made available to access. Furthermore, due to the focus of the application on georeferenced data collection, a study of the Twitter API services was performed with the purpose of filtering such services. The web interface development was created after the definition of the services that would be used in the application.

At the end of its development, data collection and spatial analysis were performed on the collected data, to demonstrate its application in a case study, which will be detailed in the Results section. The analysis procedure of spatial concentration points was performed, which were the tweets related to the "Exame Nacional do Ensino Médio" (ENEM) of 2017. The data was collected on the night of the first exam day, November 5, 2017.

5 DEVELOPMENT

The development step consists of the implementation of an application which by means of a web interface makes it possible that users have access to Twitter API services that returns georeferenced data.

In order to achieve the objectives of the current work, the following subsections detail the development processes performed.

5.1 TECHNOLOGIES AND PROGRAMMING LANGUAGES

The application of this work was developed for web platforms. This choice of the environment was based on the use of available services by means of the internet and by the access facility it provides, since it is not necessary that the application user performs any kind of installation to use it, just having internet access.

Delimiting the application to web platforms, Java language was chosen, Java EE (Enterprise Edition) standard, due to the greater knowledge and familiarity of the language by one of the authors. Java EE has an integration with many web resources like HTML, JavaScript and CSS (Cascading Style Sheets), besides that, it enables a relationship between other languages like Node.js and PHP (Oracle, 2017). (Figure 4).

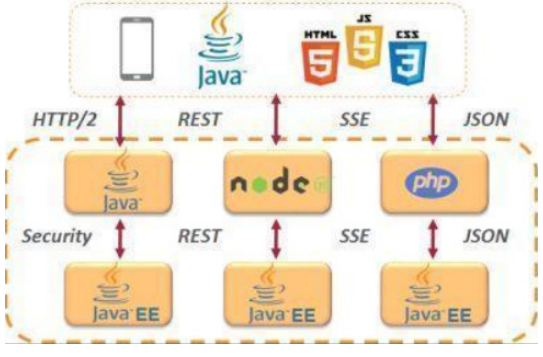

Figure 4 – A Standardized Development Model for all Java EE Developers. Source - Oracle, 2017.

ISSN 1828-5961

The application is based on MVC architecture using Spring Web MVC (model-view-controller) framework which integrated to Java EE allows that all parts of the application communicates.

MVC architecture has a mechanism that from calls in the view layer (usually known as screens, which the user can interact), the requisitions are received and managed by controller layer (where all the possible treatments of the requisitions are that the system can receive) to direct to a due process in the service layer with the capacity to relate with the model layer (the entities of system itself) (Figure 5).

For the development of the web interface, HTML was used, responsible for the skeleton page; CSS style language, responsible for formatting HTML contents, like text color, size and style; JavaScript language, is responsible for making the HTML page more interactive and convenient for the user, it may also perform validations, event actions and perform requests to the control layer without the need to update the requested page (Jepson, Macdonald, Stark, 2012); Bootstrap framework, which enables a better HTML and CSS structuring, in addition to providing table working libraries, plugins and handle application responsiveness. (Bootstrap, 2017).

The development of the whole application was performed using Eclipse IDE for Java EE developers, Neon version integrated to Java 8. This IDE offers some facilities as development tools of Java EE and JavaScript, Maven integration (project building tool which enables integrate external libraries, packages and other projects) and integration to the versioning control system GIT (Eclipse, 2017; Maven, 2017).

5.2. TWITTER API SERVICES

As already described in Theoretical Foundation subsection 3.1.1.1, Twitter API has two ways of data collection. The current work application only uses the first way, due to the focus on historical data collection. The possible requisitions available on the REST API include the following methods:

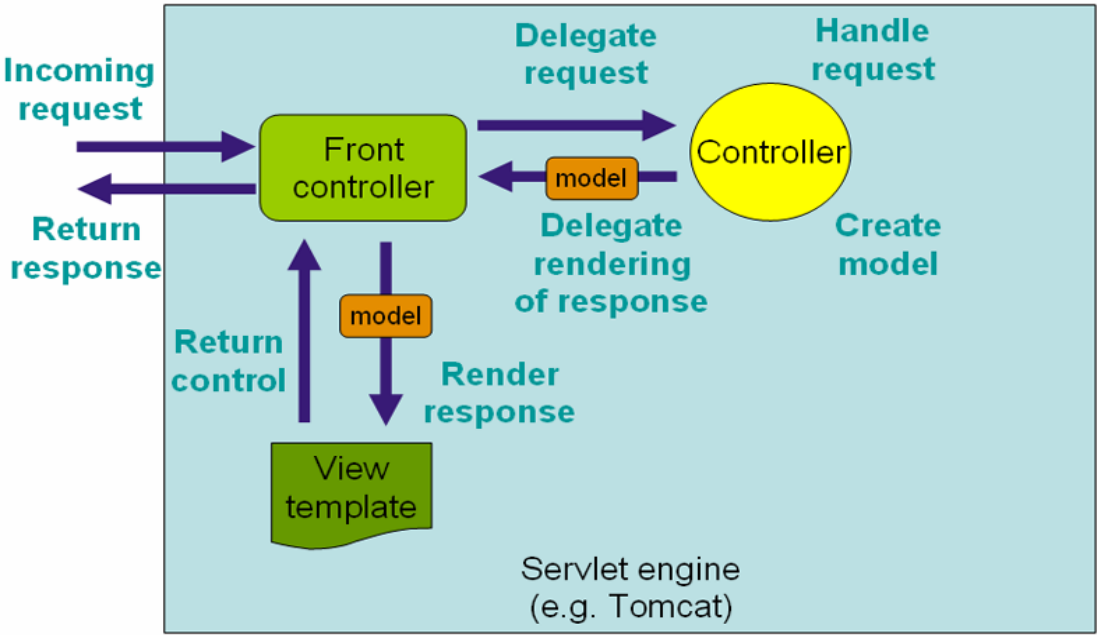

Figure 5 – The request processing workflow in Spring Web MVC (high level). Source - Spring, 2017.

- GET: returns a resource presentation.
- POST: creates a new resource.
- DELETE: deletes a resource (Alam, Cartledge & Nelson, 2014).

According to API documentation, sixty-nine GET requisitions exist; fifty POST requisitions and two DELETE requisitions available (Twitter, 2017).

With the aim of enabling that the application of the current work is directed to data collection, just the GET type requisitions were treated. Among the sixty-nine requisitions available, a survey of the quantity of georeferenced returned data was performed, reaching the result of only twelve queries, listed and detailed below:

- GET geo/id/:place_id: Returns all the information about a known place.
- GET geo/reverse_geocode: Given a latitude and a longitude, searches for up to 20 places that can be used as a “place_id” when updating a status.
- GET geo/search: Given a latitude and a longitude pair, an IP address, or a name, this request will return a list of all the valid places that can be used as the “place_id” when updating a status.
- GET geo/similar_places: Locates places near the given coordinates which are similar in name.
- GET search/tweets: Returns a collection of relevant Tweets matching a specified query.
- GET statuses/home_timeline: Returns a collection

ISSN 1828-5961

of the most recent Tweets and retweets posted by the authenticating user and the users they follow.

- GET statuses/lookup: Returns fully-hydrated Tweet objects for up to 100 Tweets per request, as specified by comma-separated values passed to the "id" parameter.

- GET statuses/mentions_timeline: Returns the 20 most recent mentions (Tweets containing a users's @ screen_name) for the authenticating user.

- GET statuses/retweets/:id: Returns a collection of the 100 most recent retweets of the Tweet specified by the "id" parameter.

- GET statuses/retweets_of_me: Returns the most recent Tweets from the notarized user that has been re-tweeted by others.

- GET statuses/show/:id: Returns a single Tweet, specified by the "id" parameter.

- GET statuses/user_timeline: Returns a collection of the most recent Tweets posted by the user indicated by the "screen_name" or "user_id" parameters.

From the previous listing the proposed application was built.

5.3 REQUISITIONS STRUCTURE CREATION

From the selection of the services to be consulted, the application development step was started, creating the structures to perform the requisition process.

Twitter API documentation indicates some resources to facilitate the implementation of public services available, being some maintained by the own company and others by third parties. In the case of Java applications, the HBC libraries are indicated, used in the API Streaming; twitter-kit-android, used in Android applications; and Twitter4J for Java and Android applications, maintained by Yusuke Yamamoto (Twitter, 2017).

Therefore, it was decided to use Twitter4J library. This one has project integration using the Maven tool, which allows easy inclusion dependencies of the library. Using the library allows access to different services and entities required for the implementation of API queries (Twitter4j, 2017).

After the integration of the library into the project, the developing structures of the application's query was started. The application layers are based on the MVC architecture using the Spring Web MVC framework, as described in Development subsection 5.1.

The model layer entities were created, according to the queries and based on the entities of the integrated library itself. The control layer is responsible for directing the requisitions and performs the required treatments. In this way, the requisitions were mapped in the control layer and a service layer was created to implement the requisitions, performing a direct communication to the API.

Lastly, the view layer was created, in which the user interacts directly with the system. In this layer there are screens that make it possible for the user to choose which service he wants to use and fill in the appropriate parameters to perform the query (Figure 6).

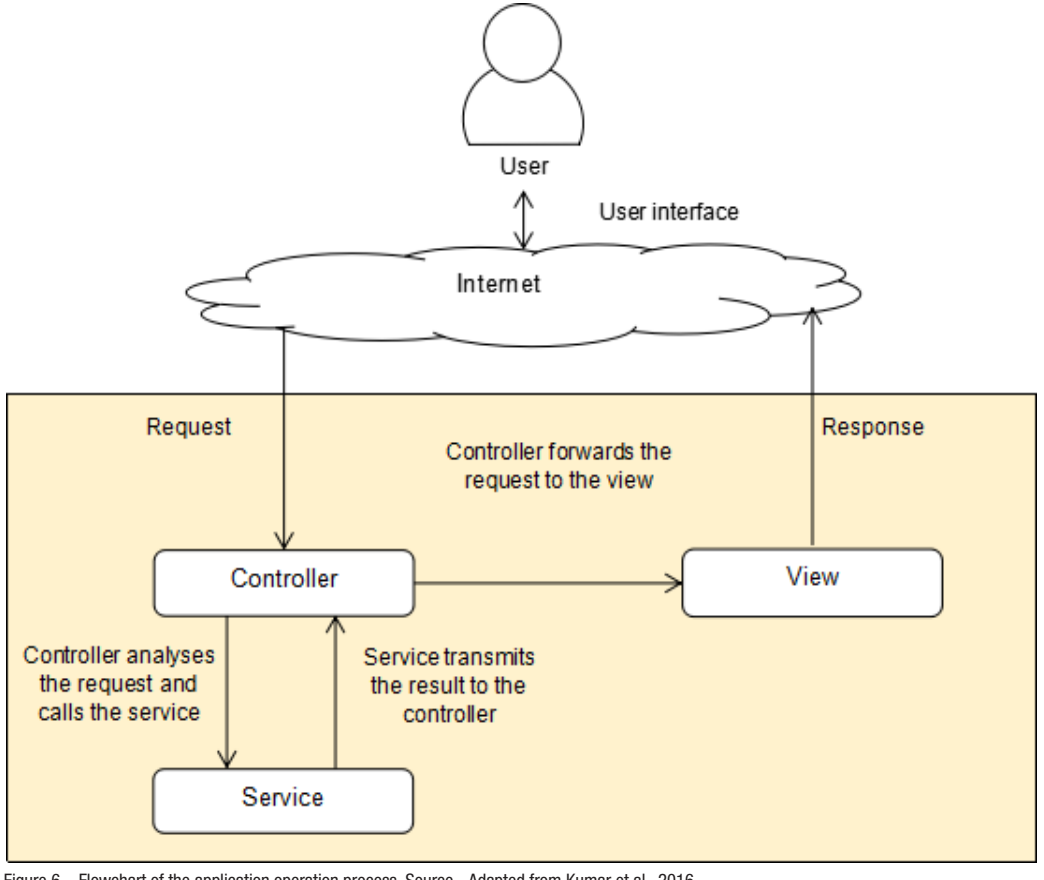

Figure 6 – Flowchart of the application operation process. Source - Adapted from Kumar et al., 2016.

ISSN 1828-5961

5.4 WEB INTERFACE CREATION

After fulfilling the structure requisitions, the web interface was started. The interface has a home page where the user is requested to log in using the Twitter account to have access to the application. After login in, the user is redirected to the home page where the available services are presented and a short description according to Twitter API documentation. All pages have a sidebar menu which is categorized by query type, them being GEO, SEARCH and STATUSES.

Every query has a specific page which is redirected when selected. The requisition pages have a table with the requisition parameters; fill out fields and a short parameter's description. A mandatory field flag is used in the mandatory parameters of each requisition.

For query services SEARCH and STATUSES, below the parameters table there is one georeferenced results filter, that when marked as "true", returns only results which have an invalid "place" object. In order to perform the filter results, there is a "max_retries" parameter that limits the number of attempts to recover the number of tweets requested in the "count" parameter, taking into account the request limit of Twitter API, in which each query has a distinct limit by logged user.

After filling out the parameter fields, the user submits the requisition and can receive two possible answers: a message with some error that can be occurred during the requisition processing or redirecting to the response viewing page. When the limit of attempts is reached, if the query has found a result, it is returned to viewing and a warning message is showed that the limit has been reached. If no results are found, the user stays on the same page and a message is showed for limit reached. In the response view page, the requisition response is displayed in JSON format, containing all attributes related to the returned object. If there are georeferenced results, it is possible to view the response in GeoJSON format.

The user is allowed to download the JSON and GeoJSON visualizations and the Shapefile (used by ArcGIS software) and KML (used by Google Earth) formats, being that the last three will be available only if there are georeferenced results. To download JSON and GeoJSON formats, the file generation is made directly in the application. Already for Shapefile and KML formats, Ogre web service is used, which enables file generation in several spatial formats based on a GeoJSON file (Ogre, 2017).

6 RESULTS

As described in the Methodology section, the current section presents the data collection steps and spatial analysis on collected data. Among the twelve queries approached in the research, "search/tweets" can be considered as the more relevant to be evaluated, since through it, a greater level of information can be retrieved using location filters.

6.1 "SEARCH/TWEETS" QUERY

Selecting the "search/tweets" query, the user can retrieve a maximum limit of one hundred tweets according to the established filter. This filter contemplates the following parameters:

- query: term to be searched, can be a word or phrase.
- geocode: location specified by latitude, longitude and radius in miles or kilometers.
- lang: tweet language.
- result_type: result type which is preferred to obtain, can be "recent" (latest results), "popular" (most popular results) or "mixed" (both recent and popular results).
- count: number of tweets that intend to obtain, being 100 the maximum.
- until: restriction of tweets older than the specified date.
- since_id: restriction of tweets with "id" greater than the specified id.
- max_id: restriction of tweets with "id" less than the specified id.
- max_retries: maximum number of attempts to retrieve the number of tweets specified in the "count" parameter.

Besides the previously detailed parameters, the user can request that only georeferenced results can be returned. Only the "query" parameter is mandatory fill out.

A behavior to be highlighted in relation to the query, is due to the fact that the API allows retrieving only tweets from a retroactive week to the date of collection. In the case of the user having to obtain tweets in a longer range, it is necessary to perform other queries from previous dates using the "until" parameter.

The following section details a data collection process using the "search/tweets" query.

6.2 DATA COLLECTION

The data collection proposal for the application tests is to recover tweets related to the 2017 ENEM in the Brazilian states and analyze the term's comprehensiveness spatially. The collection was done on the night of the first exam day, November 5, 2017.

One of the query filters is "geocode", which allows the creation of a location area to perform the query. In this way, one collection of each one of the twenty-seven Brazilian states was made, including the Federal District. The filter values applied to the query are listed below:

- query: enem.
- geocode: centroid latitude and longitude of a given state and its extension radius (in miles).
- lang: Portuguese
- result_type: mixed.
- count: 100.
- max_retries: 100.

Georeferenced filter results were marked as true in all queries.

It is important to emphasize that queries of this service have a restriction concerning the time interval. The maximum limit of retrieving tweets is a retroactive week, which means that only the interval between October 29 and November 5, 2017 was considered in the collection of the current work.

ISSN 1828-5961

For each collection performed, a directory was created containing the JSON, GeoJSON, Shapefile and KML files. Most of collections retrieved the maximum limit of 100 tweets, while others retrieved at least one tweet or an average of 30 tweets, totaling 1348 tweets. Some of the queries may have retrieved the same tweet, because the informed radius for a given state can include another neighbor.

6.3 SPATIAL ANALYSIS OF THE COLLECTED DATA

The case study of the current work provides a spatial analysis of the tweets collected using the proposed application. The analysis aims to generate a visualization of the searched term's comprehensiveness relating the tweets location, number of occurrences by locality and population of given locality.

For this, the QGIS software version 2.18 was chosen since it is a multiplatform open-source GIS. It offers several visualization and data analysis functionalities, among others (QGIS, 2017).

Initially, all directory Shapefiles were loaded onto the QGIS. The software considers that each file loaded is a layer on the map that is built. Therefore, all layers were merged to create a unique layer containing all tweets of all states, resulting in a layer with 1348 tweets.

Because during the data collection some tweets may have been returned in more than one query, it was necessary to perform database normalization. The normalization process consists in filter tweets using the "status_id" parameter, which is unique to the tweet. Therefore, a query using this filter on the database was performed, generating a new layer, totaling 1195 tweets to be analyzed.

Firstly, the tweets with ENEM term occurrence were displayed on the map. Each point on the map has the sum of the tweets found for that location. From this data, a heatmap was created using the Kernel Density tool to show the main concentration areas of tweets with “ENEM” term around the country. It can be noticed that the largest cities in the country, such as Brasilia, São Paulo, Rio de Janeiro, Recife, Vitória and Porto Alegre, were the ones that had the most results, since they concentrate a larger population as well (Figure 7).

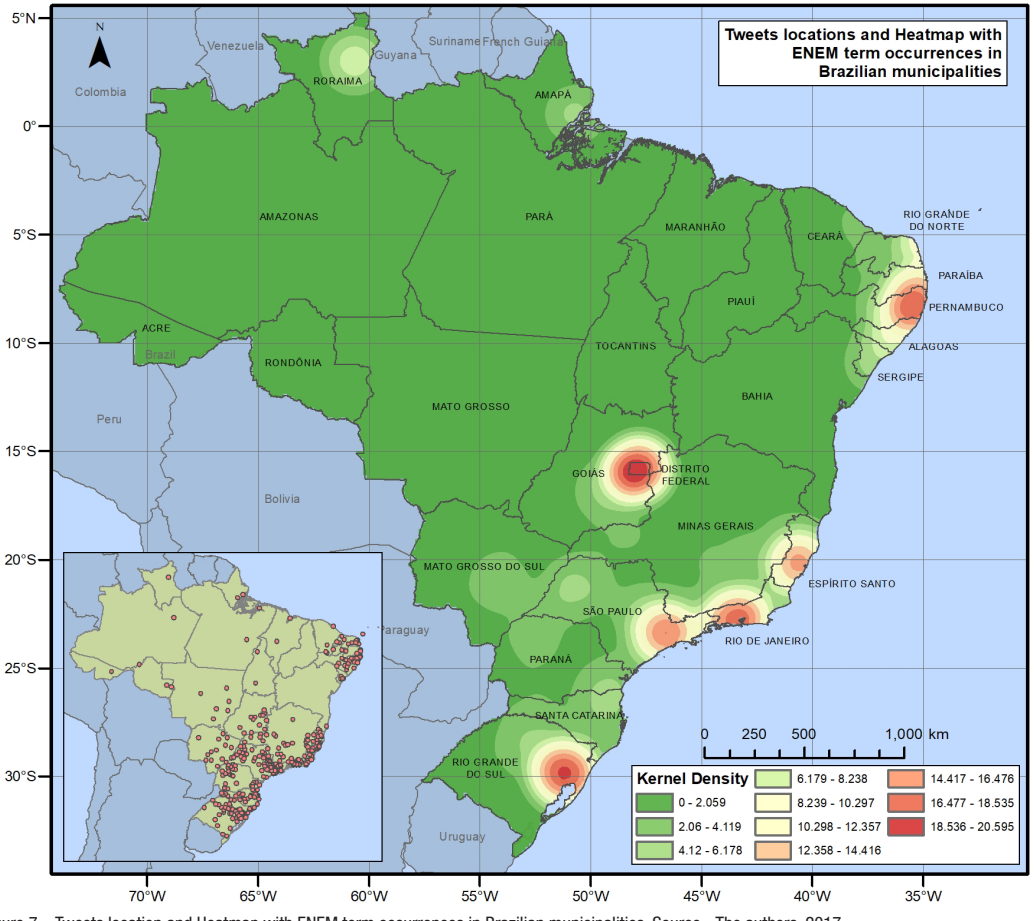

Figure 7 – Tweets location and Heatmap with ENEM term occurrences in Brazilian municipalities. Source - The authors, 2017.

For this type of analysis, it is expected that the more populous cities will also present greater results. Therefore, a second analysis was performed considering the percentage of people by locate that posted on twitter, correlating the number of tweets with the total population of the city.

Therefore, the integration between the generated layer and a layer containing the municipalities population was performed, using a database with the population estimates for 2017 published in the Official Diary of the Union and available at IBGE (Brazilian Institute of Geography and Statistics) (IBGE, 2017).

First, the basis was converted to dBASE (.dbf) format and then loaded onto QGIS. This layer was united to the tweets layer using the "NOME DO MUNICÍPIO" attribute on the population basis and "place_name" on the tweets basis.

ISSN 1828-5961

After joining the layers, it was noticed that there were tweets without a connection to the population. That happened because there were tweets with state location, not of the municipality. Therefore, the tweets were removed from the layer to keep the data homogeneity on municipality scale.

With the purpose of quantifying the tweet occurrences in each municipality, a "count" field in the layer table was created to store the number of tweets in one given location identified by "place_id" attribute. From this new field, a new field was created containing the percentage of occurrences in connection to the population.

The "perc" field was created from the calculation according to Eq. 1:

$$\frac{(count \times 100)}{est}$$

where: count = number of tweets, est = population estimate from given location.

Thus, a tweets basis containing localization information and percentage of occurrences in connection to the municipality's population, was consolidated.

6.4 DATA ANALYSIS VISUALIZATION

For the data visualization, the tweets layer was converted to points corresponding to the centroid of each polygon that represents a location. From the generated points, the heatmap visualization was created by means of the Kernel density application model, which consists of a statistical method in which a density matrix map is created from the points in vectors (Diniz, Palhares & Ribeiro, 2017). Using "perc" attribute as weight density, the method draws a circular vicinity around each occurring location of the attribute. The generated map represents the spatial concentration of the performed query (Figure 8).

Areas in greener shades demonstrate locations with fewer occurrences of "enem" term in connection to the population, while areas with warmer colors demonstrate a higher number in connection to the population.

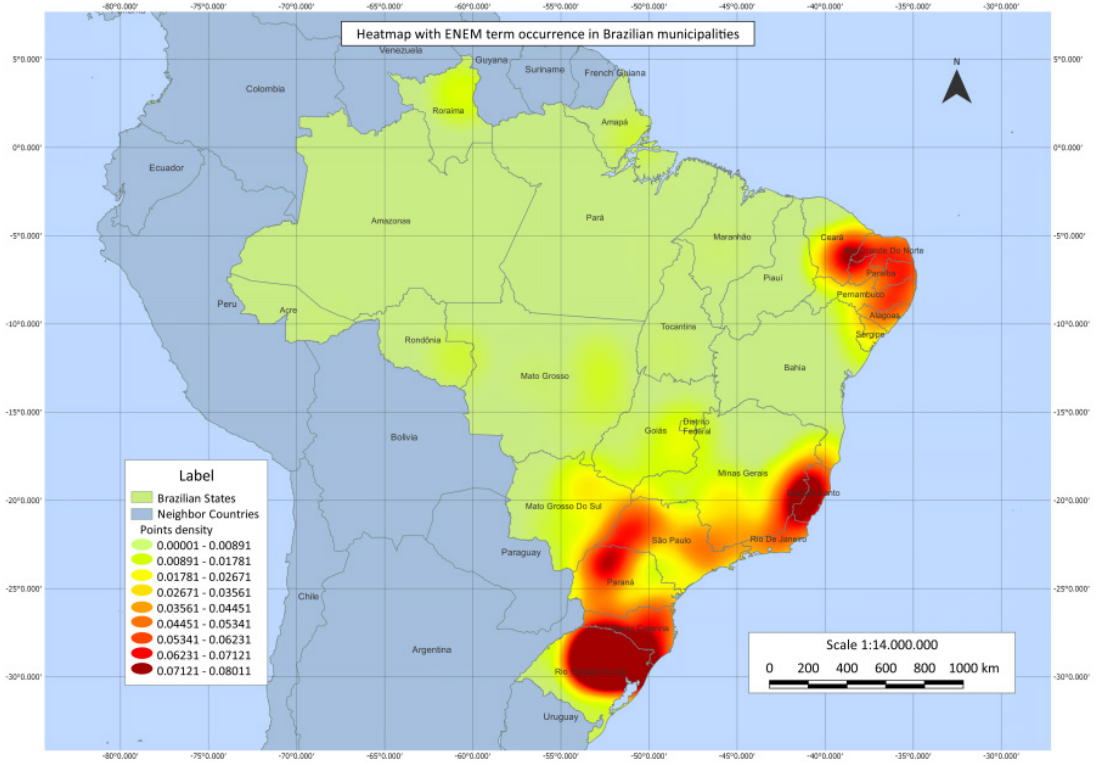

Figure 8 – Heatmap with ENEM term occurrences in Brazilian municipalities. Source - The authors, 2017.

From the map, it is possible conclude that the locations with higher percentages were in the states of Rio Grande do Sul, Paraná and on the boundary between Minas Gerais and Espírito Santo and between Ceará and Rio Grande do Norte. Relating the map visualization and the tweets table, the municipalities Estrela Velha (RS), Iguatu (PR), Taboleiro Grande (RN), Anta Gorda (RS), Paim Filho (RS), Ibarama (RS), Riqueza (SC), Selbach (RS) and Dom Cavati (MG) correspond to the locations with higher point densities, in descending order by percentage.

This spatial analysis model is widely used in the geosciences and its interpretation gives clues of spatial behavior in a given phenomenal. In this study, the "enem" term was used as example due to its relevance at the moment the research was developed. However, other queries can be made on the platform in order to contribute to conduct diagnostics that can support territorial planning, like in in public health cases, violence, cultural aspects identification, among others. To search other query terms, the user only needs to use the query filling out the "query" parameter with

the term or expression that the user wants to search.

7 CONCLUSION

The differential of the application developed is that it allows the interaction of users from several areas without required programming language knowledge. Furthermore, it has a greater focus on geosciences professionals, for allowing that the queries can be filtered to recover only georeferenced data and enables the results to be downloaded in specific formats used in GIS software, like ArcGIS and Google Earth, allowing a better integration between the data collection steps and data analysis.

With the data integration into a GIS, allowing the analysis and visualization of the data, it is possible to verify that the generated formats by the application are in accordance with the reality of the geosciences areas professionals.

The application proposed in the current work was initially developed according to the proposed objectives. However, some adjustments must be made so its operation is put into production. Considering the user's experience, it is necessary that the feedback of the queries is improved to provide real time information of the queries progress on the application. Until this moment, the user submits the query request and doesn't receive any information until the query is finished and its result returned.

With the application development, the main difficulty found was due to the Twitter API usage. Such difficulty was the API's requisitions limit, since the filter of the georeferenced results consists of a constant query interaction, until data according to the filter is found, which results in a high number of requisitions, compared to a query without this filter. This high number of requisitions it happens due to a Twitter API limitation, which establishes a maximum number of results according to the request, to decrease the processing and traffic of the data.

The main contributions of this work are:

- Proposal of a new solution which integrates the computer and geosciences areas, narrowing the gap between those areas;
- Easy access to data which are available from APIs with language restrictions to the computing public;
- New research possibilities are made from georeferenced data collected on Twitter using the application.

7.1 FUTURE WORK

For future work, considering the user experience, it is intended to perform tests with geosciences areas professionals to verify if the application usability meets the necessities of the specific public.

It is also intended to extend the public of the application in addition to the geosciences areas professionals. For this, it will be necessary to develop the other queries provided on the Twitter API and a study of data formats accepted in other specific software, as in the case of data mining.

REFERENCES:

Alam, S., Cartledge, C. L. & Nelson, M. L. (2014) Support for Various HTTP Methods on the Web. *Technical Report arXiv*:1405.2330, 1(1), 1-16.

Apigee (2017). *What is Apigee Edge?* Retrieved September 20, 2017, from http://docs.apigee.com/api-services/content/what-apigee-edge

Bootstrap (2017). *Ponto de partida.* Retrieved September 21, 2017, from http://getbootstrap.com.br/getting-started

Borges, J. L. de C., Davis Junior, C. & Jankowski, P. (2016). A Study on the Use of Crowdsourcing Information for Urban Decision-Making. *Revista Brasileira de Cartografia*, 68(4), 695–703.

Borges, J. L. de C., Davis Junior, C. & Jankowski, P. (2015). Crowdsourced information from Tweets during the WorldCup in Brazil: A theme search. In: *Proceedings of the International Conference on Changing Cities II: Spatial, Design, Landscape & Socio-economic Dimensions*. pp. 1511 – 1519. Porto Heli: International Conference on Changing Cities II: Spatial, Design, Landscape & Socio-economic Dimensions

Burrough, P. A., Lloyd, C. & McDonnell, R. A. (2015) *Principles of Geographical Information Systems*. 3th Edition. New York: Oxford.

Camboim, S. P. & Sluter, C. R. (2013). Uso de ontologias para busca de dados geoespaciais: uma ferramenta semântica para a Infraestrutura Nacional de Dados Espaciais. *Revista Brasileira de Cartografia*, 65(6), 1127–1142.

Calafiore, A., Borges, J., Moura, A. C., & Boella, G. (2016). Integrating VGI system in a Participatory Design Framework. In *Proceedings of 9th International Conference on Innovation in Urban and Regional Planning*. 441–446. Torino: International Conference on Innovation in Urban and Regional Planning.

Campagna, M. (2016). Social Media Geographic Information: Why social is special when it goes spatial? In: Capineri, C, Haklay, M, Huang, H, Antoniou, V, Kettunen, J, Ostermann, F and Purves, R. (Eds.) *European Handbook of Crowdsourced Geographic Information*, 45-54, London: Ubiquity Press.

Campagna, M., Massa, P. (2014). Social Media Geographic Information: Current Developments and Opportunities in Urban and Regional Planning. In *Proceedings of 19th International Conference on Urban Planning, Regional Development and Information Society*. 631–640.

Campagna, M., Floris, R., Massa, P., Girsheva, A., & Ivanov, K. (2015). The Role of Social Media Geographic Information (SMGI) in Spatial Planning. In: Geertman, S. et al. (Ed.) *Planning Support Systems and Smart Cities*. Cham: Springer International Publishing, 42-60.

Daiya, R., Khandekar, C., Parekh, R., Kellar, K. (2017). TweetSum: Automated News Summarization of Twitter Trends. *International Journal of Computer Applications*, 165(8), 5-8.

Diniz, A. M. A., Palhares, R. H., & Ribeiro, L. L. O impacto da realização da Copa das Confederações da FIFA de 2013 e da Copa do Mundo da FIFA de 2014 na criminalidade em Belo Horizonte. *Revista Franco-Brasileira de Geografia,* 32(1), 2-15.

Eclipse (2017). *Eclipse IDE for Java EE Developers*. Retrieved September 21, 2017, from https://eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/oxygenr

Garcia-Molina, H., Joglekar, M., & Marcus, A., Paramensawaran, A., Verroios, V. (2016). Challenges in Data Crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 28(4), 901–911.

Hofer, B. (2017). Uses of online geoprocessing technology in analyses and case studies: a systematic analysis of literature. *International Journal of Digital Earth*, 8(11), 901–917

IBGE - Instituto Brasileiro de Geografia e Estatística (2017). *Estimativas de População.* Retrieved November 6, 2017, from https://www.ibge.gov.br/estatisticas-novoportal/sociais/populacao/9103-estimativas-de-populacao.html?&t=downloads

Inácio Júnior, V. R. (2007). *Um Framework para Desenvolvimento de Interfaces Multimodais em Aplicações de Computação Ubíqua.* (Master Dissertation) University of São Paulo, Brazil.

Jepson, B., Macdonald, B, & Stark, J. (2012). *Building Android Apps with HTML, CSS, and JavaScript: Making Native Apps with Standards-Based Web Tools.* 2nd Edition. California: O'Reilly Media Inc.

Kumar, V., Kumar, A., Sharma, A. K., & Singh, D. (2016). Implementation of MVC (Model-View-Controller) design architecture to develop web based Institutional repositories: A tool for Information and knowledge sharing. *Indian Research Journal of Extension Education,* 16 (3), 1–9.

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2010). *Geographic Information Systems and Science*. Third Edition. Hoboken, NJ: Wiley.

Machado, J. (2017). *Um método para análise e visualização de dados georreferenciados relacionados ao trânsito de veículos.* (Master's thesis). University of Vale do Rio dos Sinos, Brazil.

Macnamara, J. (2017). *Evaluating Public Communication: Exploring New Models, Standards, and Best Practice*. 1st Edition. New York: Routledge.

Macy, M. W., Mejova, Y., & Weber, I. (2015). *Twitter: A Digital Socioscope.*

1st Edition. New York: Cambridge University Press.

Marres, N., & Weltevrede, E. (2013). Scraping the Social? Issues in real-time social research. *Journal of Cultural Economy,* 6(3), 313–335.

Massa, P., & Campagna, M., (2016). Integrating Authoritative and Volunteered Geographic Information for spatial planning. In: Capineri, C, Haklay, M, Huang, H, Antoniou, V, Kettunen, J, Ostermann, F and Purves, R. (Eds.) *European Handbook of Crowdsourced Geographic Information* (pp. 401–418). London: Ubiquity Press.

Maven (2017). *Welcome to Apache Maven*. Retrieved September 21, 2017, from https://maven.apache.org/

Meireles, F., & Silva, D. (2015). Ciência Política na era do Big Data: automação na coleta de dados digitais. *Revista Política Hoje*, 24(2), 87–102.

Ogre (2017). *ogr2ogr web client*. Retrieved October 1, 2017, from http://ogre.adc4gis.com

Oracle (2017). *Java™ EE at a Glance.* Retrieved September 21, 2017, from http://www.oracle.com/technetwork/java/javaee/overview/index.html

Pizetta, D. C. (2014). *Biblioteca, API e IDE para desenvolvimento de projetos de metodologias de Ressonância Magnética.* (Master Dissertation). University of São Paulo, Brazil.

QGIS (2017). *QGIS - The Leading Open Source Desktop GIS*. Retrieved November 6, 2017, from http://www.qgis.org/pt_BR/site/about/index.html

Spring (2017). *Web MVC framework.* Retrieved September 21, 2017, from https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html

Twitter. *Twitter Developer Documentation.* Retrieved August 30, 2017, from https://dev.twitter.com/docs

Twitter4j. *Main.* Retrieved September 22, 2017, from http://twitter4j.org/en/index.html

Urru, M., & Vargiu, E. Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artificial Intelligence Research*, 2(1), 44–54.

Xavier-da-Silva, J. O que é geoprocessamento. *Revista do Crea-RJ*, 79(1), 42–44.

ISSN 1828-5961

Uma Aplicação de Coleta de Dados para Profissionais das Geociências

1. INTRODUÇÃO

Sistemas de Informação Geográfica (SIG) são um conjunto de ferramentas computacionais de coleta, armazenamento, recuperação, análise e visualização de dados espaciais. Os SIG são compostos de três componentes principais: hardware, software e usuários. O hardware é responsável pela parte física e acesso aos softwares do sistema, podendo também realizar comunicação com dispositivos externos, como aparelhos móveis utilizados em coletas de dados. O software é responsável por quatro ações básicas: entrada e verificação de dados; armazenamento e gerenciamento de dados; saída e apresentação de dados; transformação, análise e modelagem de dados. Os usuários são a comunidade que utiliza e/ou desenvolve SIG (Burrough, Lloyd & McDonnell, 2015).

Uma das tarefas implementadas em softwares de SIG, é o geoprocessamento, que consiste na análise espacial de dados geográficos a fim de mensurar propriedades e relacionamentos georreferenciados (Machado, 2017). Segundo Xavier-da-Silva (2009), o geoprocessamento utiliza bases de dados georreferenciadas aplicando métodos computacionais que permitem varreduras de incidências territoriais, com a finalidade de transformar dados em conhecimentos para apoio à tomada de decisões.

O geoprocessamento pode ser utilizado também em aplicações online. Uma das vantagens dessa utilização é a facilidade de compartilhamento de informações entre a comunidade científica (Hofer, 2014).

Segundo Machado (2017), os SIG vêm sendo utilizados em estudos de análise de distribuição de doenças, mapeamento de populações em risco, alocação de recursos e planejamento de intervenções. Camboim e Sluter (2013, p. 1129) descrevem que um dos problemas que os usuários de dados geoespaciais encontram, "é a dificuldade de encontrar as informações relevantes para determinado uso". Segundo os autores, isso é uma questão que as Infraestruturas Nacionais de Dados Espaciais (INDE) devem se esforçar em facilitar o acesso desses dados aos usuários.

Juntamente com a dificuldade de encontrar informações relevantes, vem também a dificuldade de profissionais de áreas das geociências que utilizam SIG em pesquisas e tomadas de decisão para utilizarem ferramentas de coletas de dados. Um exemplo disso é o uso de Interfaces de Programação de Aplicações (Application Programming Interfaces - API) que têm como público-alvo o público de Ciência da Computação.

Nos últimos anos, empresas como Facebook, Google e Microsoft têm utilizado API na criação de plug-ins de seus aplicativos. Isso permite que a comunidade de desenvolvimento utilize os serviços da API e contribua com o crescimento das aplicações (Pizetta, 2014). O Twitter também disponibiliza o uso de sua API para desenvolvedores em sua plataforma.

ISSN 1828-5961

Os serviços de consultas da API são utilizados em tarefas de coleta de dados para análises de diversas naturezas. Algumas das consultas possibilitam a obtenção de dados georreferenciados, por meio da obtenção de localização em postagens (Twitter, 2017).

O nível de complexidade no uso de uma API requer conhecimento prévio de ferramentas e linguagens de programação apropriadas para a criação de uma estrutura que possa consumir e tratar dados. Essa complexidade é um fator que pode restringir o público que necessita utilizar esse tipo de serviço, como os profissionais de áreas das geociências, que utilizam a coleta de dados georreferenciados para análises espaciais a fim de gerar conhecimento útil e auxiliar na tomada de decisões.

Dessa forma, o problema de pesquisa levantado no presente trabalho é: como melhorar a experiência de profissionais de áreas das geociências com pouco ou nenhum conhecimento de programação durante a coleta de dados disponibilizados em sistemas que utilizam API?

A motivação da abordagem se deu pelo fato da dificuldade de acesso ou falta de conhecimento específicos dos profissionais de áreas das geociências restringir o acesso desses usuários na etapa de coleta de dados, que é uma das fases mais importantes do geoprocessamento. Assim, pretende-se facilitar o acesso a um público mais abrangente, permitindo que mais pesquisas científicas sejam realizadas utilizando-se desses meios.

A partir deste contexto, o objetivo geral do presente trabalho é a criação de uma aplicação que realize consultas à API do Twitter a fim de facilitar a etapa de coleta de dados e possibilitar uma melhor interação dos profissionais de áreas das geociências sem conhecimentos específicos de programação. Os objetivos específicos são: Identificar e especificar os serviços da API do Twitter que coletam dados georreferenciados; permitir filtro de buscas apenas com resultados georreferenciados nas consultas; Possibilitar que o usuário realize download dos dados coletados em formatos estruturados para softwares de SIG.

2. REVISÃO BIBLIOGRÁFICA

Nesta seção, são apresentadas as aplicações existentes que se relacionam com a aplicação proposta. É realizada uma breve descrição de suas funcionalidades e por fim um comparativo entre elas.

2.1. APIGEE

O Apigee é uma plataforma de desenvolvimento e gerenciamento de APIs. A plataforma oferece um ambiente no qual as empresas podem expôr seus serviços de requisição.

O modelo se baseia na requisição solicitada pelo desenvolvedor client-side, que ao ser processada retorna um dado em formato XML (eXtensible Markup Language) ou JSON (JavaScript Object Notation) (Apigee, 2017). A Figura 1 demonstra a arquitetura do modelo de requisições da plataforma.

Figura 1 – Arquitetura do modelo de requisições do Apigee. Fonte - Apigee, 2017.

Algumas das empresas que utilizam essa plataforma são: Bing, Blogger, Facebook, Flickr, Foursquare, Google, Instagram, LinkedIn, Reddit e Twitter. O acesso aos serviços das empresas é realizado por meio de uma aplicação console, na qual é possível obter a listagem dos serviços da API e realizar as requisições necessárias.

Uma das APIs frequentemente utilizada em tarefas de coleta de dados é a do Twitter, que também é disponibilizada pela plataforma do Apigee e todos os serviços podem ser acessados no próprio website da Apigee. Para se obter acesso aos serviços da API, o usuário necessita realizar a autenticação na plataforma, por meio do login da aplicação que se deseja utilizar. No caso da API do Twitter, o usuário se conecta em sua conta pessoal e passa a ter acesso aos serviços de consulta às diversas funcionalidades da rede social.

2.2. SPATEXT

Spatext é um add-in implementado em linguagem Python versão 2.7 disponível no software ArcGIS©. Dentre suas funcionalidades, existem nove ferramentas que permitem coleta de dados em mídias sociais como Twitter, YouTube, Wikimapia, Instagram, Foursquare e Panoramio. O Spatext possui uma vantagem em relação às APIs de redes sociais ao ser executado diretamente em uma interface de SIG, tornando mais fácil o processo de integração de dados a fim de apoiar tomadas de decisão em planejamento urbano e regional (Massa, Campagna, 2016; Campagna; Massa, 2014).

O fluxo de funcionamento do Spatext segue de acordo com as funções de coleta de dados, gerenciamento de dados e análise de dados (Figura 2).

Figura 2 – Design da arquitetura do Spatext. Fonte - Massa, Campagna, 2016, p.7.

2.3. COMPARATIVO

Relacionando as principais características das duas aplicações descritas, é possível fazer a seguinte análise: ambas as aplicações, Apigee e Spatext, realizam consultas a API do Twitter e permitem a busca de dados georreferenciados. O Apigee não possui um filtro desses dados. Para a utilização do Apigee não é necessário nenhum conhecimento específico de linguagem de programação ou treinamento da ferramenta para utilizá-lo. Devido ao fato do Spatext ser executado diretamente na interface do ArcGIS, é necessário que o usuário tenha uma noção do funcionamento da aplicação e também é necessário conhecimento da linguagem Python, já que o Spatext é baseado em scripts desta linguagem. O retorno das consultas do Apigee são disponibilizados somente para visualização em formato JSON, não permitindo que o usuário realize download das consultas retornadas. O resultado das consultas do Spatext é armazenado

ISSN 1828-5961

diretamente no banco de dados do ArcGIS, permitindo que o usuário realize integrações com os projetos da aplicação. O acesso ao Apigee é aberto ao público, desde que o usuário tenha conta na aplicação a ser utilizada. Para o presente estudo, não foi possível acessar diretamente o Spatext, devido ao fato da ferramenta ter sido apenas experimental e não ter sido distribuída publicamente, sendo acessível apenas em trabalhos científicos que utilizaram a ferramenta.

A partir do comparativo, constata-se que a aplicação proposta se diferencia das demais pelas seguintes características: a aplicação oferece um filtro de busca de dados georreferenciados; não é necessário que o usuário tenha conhecimento específico de linguagem de programação e/ou treinamento de uso da ferramenta; o usuário pode visualizar os resultados na interface da aplicação e também realizar download dos dados coletados; é pretendido que o usuário tenha livre acesso à aplicação após estar em pleno funcionamento, garantindo o uso aberto ao público.

3. FUNDAMENTAÇÃO TEÓRICA

A etapa de coleta de dados em SIG pode utilizar variadas fontes de informações, nos diversos campos de aplicação. A fim de facilitar a coleta de dados, algumas técnicas podem ser utilizadas no processo. Marres e Weltevrede (2012) citam o Web Scraping como uma técnica de automatizar a coleta de dados online. Elas explicam que a técnica é amplamente utilizada para fins de coleta, análise e visualização de dados.

A seguir são detalhados os conceitos das técnicas de coleta de dados estudadas e as principais definições necessárias para a compreensão do presente trabalho.

3.1. WEB SCRAPING

Web Scraping se refere à uma técnica de extração de informação. Tal técnica permite que dados de diferentes locais possam ser coletados juntos. O processo permite que sejam coletados dados de páginas de imagens, de localização ou busca por palavras-chave, para fins de base de dados de estudos (Marres, Weltevrede, 2012). Scrapers basicamente transformam dados desestruturados e os armazenam em base de dados estruturadas (Vargiu, Urru, 2012). Um exemplo de uso do Web Scraping, é a pesquisa de Daiya et al. (2017) em que é feita a criação de um sistema de sumarização de notícias do Twitter. Segundo eles, o componente do scraper lida com a extração de todos os conteúdos relacionados a determinada busca na web.

O sistema permite que os usuários busquem por palavras ou frases específicas e após o processamento da pesquisa, são retornados os tweets sumarizados da determinada busca. Meireles e Silva (2015) explicam que a coleta de dados utilizando a técnica de Web Scraping se limita a conteúdos disponíveis em exibição HTML (HyperText Markup Language), XML e API. Devido ao fato da aplicação do presente trabalho ser baseada em serviços de API, será detalhado apenas tal abordagem na Fundamentação Teórica.

3.1.1. INTERFACES DE PROGRAMAÇÃO DE APLICAÇÕES (APPLICATION PROGRAMMING INTERFACES - API)

Uma API é um conjunto de definições de sub-rotinas, protocolos e ferramentas fornecido por desenvolvedores de aplicações que permitem que outros desenvolvam softwares que se liguem a essas aplicações (Macnamara, 2017).

Inácio Júnior (2007) explica que uma API determina as funcionalidades de um software que podem ser utilizadas em outros softwares. Ou seja, ela define os endereços de requisição dos elementos a fim de permitir que sejam executadas tarefas e troca de dados.

O autor ainda cita as seguintes características de uma API:

- Ocultamento de informação: APIs têm a capacidade de "esconder" módulos privados, o que restringe o acesso à lógica interna. Por ser um princípio de baixo acoplamento e alta coesão, isso diminui as dependências entre as partes do sistema.
- Separação especificação/implementação: a interface da API separa seus métodos visíveis, de forma a possibilitar diferentes implementações por parte dos projetistas.
- Interoperabilidade: APIs podem realizar conexões com sistemas distintos, além de permitir integrações com diferentes linguagens.
- Estabilidade: o desenvolvimento de APIs tende a não mudar constantemente, mantendo a compatibilidade com sistemas e independência entre provedores e clientes da API.

3.1.1.1. TWITTER API

O Twitter fornece duas formas de acesso à sua API pública. A primeira forma (REST - Representational State Transfer) é utilizada para coletar dados históricos, que geralmente são tweets anteriores à data e hora da coleta. A segunda forma (Streaming) possibilita a coleta de dados em tempo real sem interrupção, a medida em que os usuários postam conteúdos (Macy, Mejova, Weber, 2015).

O procedimento padrão para se utilizar ambas as formas da API são descritas a seguir:

- Autenticação do Twitter utilizando o mecanismo Open Authentication (OAuth).
- Criação da chamada da API do Twitter passando os devidos parâmetros.
- Recebimento e processamento da resposta da API.

Na etapa de autenticação, são validadas as chaves de acesso para acesso a API. A partir de uma conta de desenvolvedor criada para a aplicação, são gerados tokens de autenticação, que devem ser utilizados como parâmetros de configuração.

Os tokens de acesso são os parâmetros "Access Token" e "Access Token Secret". As chaves de acesso são os parâmetros "Consumer Key" e "Consumer Secret".

3.2. CROWDSOURCING

Crowdsourcing é descrito como um conjunto de técnicas de coleta de dados contribuídos por cidadãos sem treinamento específico, que permite que bases de

dados sejam criadas. A combinação de crowdsourcing e SIG resulta no crowdsourcing mapping, que permite a geração de grandes volumes de dados espaciais que podem ser utilizados com bases de diagnósticos e incluídos em ações de gestão e planejamento territorial (Borges, Davis Júnior & Jankowski, 2016).

Garcia-Molina et al. (2016) explicam que crowdsourcing é o ato de recrutar pessoas para executar tarefas que podem ser consideradas muito difíceis para um computador executar por conta própria. Os autores citam análise de sentimento, classificação visual e comparação de dados como alguns dos problemas trabalhados utilizando-se de crowdsourcing.

Nas subseções a seguir, são descritos dois processos para se obter informações a partir do crowdsourcing, segundo Borges, Davis Junior e Jankowski (2016).

3.2.1. INFORMAÇÃO GEOGRÁFICA VOLUNTÁRIA (VOLUNTEERED GEOGRAPHIC INFORMATION - VGI)

VGI é um conceito diretamente ligado a coleta de dados geográficos a partir da participação de cidadãos, com um propósito claro. A coleta dos dados quando realizada por meio de APIs não necessita do conhecimento dos usuários participantes. O primeiro ponto para se coletar dados por meio de VGI é a mobilização dos usuários para que eles possam contribuir com seus conhecimentos (Borges, Davis Júnior & Jankowski, 2016).

Campagna et al. (2015) descrevem VGI como a geração de conteúdo por usuários que agem como sensores voluntários. O uso dessa metodologia é extremamente útil em pesquisas de gestão de emergências, monitoramento de ambientes, planejamento espacial e gestão de crise.

Calafiore et al. (2016) apresentam alguns estudos de caso que realizam experimentos com sistemas de VGI, além de descreverem a proposta da pesquisa em questão. Dentre os estudos de caso apresentados, pode-se citar o experimento de Borges, Jankowski e Davis Junior (2015), no qual os autores coletaram dados do Twitter durante os jogos da Copa do Brasil, e após as análises demonstraram as postagens em grupos por jogo, nacionalidade e geotag. Já a proposta de Calafiore et al. (2016) é baseada em uma metodologia que aumenta a participação em processos de planejamento. A ferramenta coleta informações geográficas e se concentra em tarefas de organização e análise de dados a fim de auxiliar na elaboração de mudanças espaciais.

3.2.2. MÍDIA SOCIAL DE INFORMAÇÃO GEOGRÁFICA (SOCIAL MEDIA GEOGRAPHIC INFORMATION - SMGI)

SMGI é um processo de coleta de dados, assim como o VGI, porém não há um objetivo específico do que é capturado. Identificar o fundamento e aplicação da informação é uma tarefa do pesquisador em si. Os dados podem ser utilizados para fins de pesquisas de mobilidade ou concentração geográfica em determinados pontos de localização (Borges, Davis Júnior & Jankowski, 2016).

Campagna (2016) define SMGI como conteúdos com informações geográficas implícitas ou explícitas coletadas através de redes sociais ou aplicativos móveis. Esses conteúdos podem ser em formato de textos, imagens, vídeos ou áudios referenciados espacialmente.

O uso do processo de SMGI tem alguns obstáculos que limitam seu uso. Campagna e Massa (2016) apresentam a falta de ferramentas amigáveis e a complexidade em tratar grande quantidade de dados como principais obstáculos. Tais problemas se devem ao crescente volume de informação e a especificidade das estruturas de dados, necessitando ajustes nas metodologias de análise e desenvolvimento tradicionais.

4. METODOLOGIA

Com o intuito de desenvolver a aplicação proposta neste trabalho, seguiu-se o fluxograma dos procedimentos metodológicos segundo a sequência (Figura 3):

Figura 3 - Fluxograma dos procedimentos metodológicos. Fonte - Os autores, 2017.

Realizou-se ao longo de todo processo o procedimento de revisão bibliográfica em livros, artigos científicos, publicações e periódicos, buscando embasamento teórico acerca dos conceitos e aplicações de SIG, geoprocessamento, coleta de dados utilizando técnicas de Web Scraping, APIs, crowdsourcing, VGI e SMGI. Além do embasamento, buscou-se a melhoria contínua dos métodos aplicados, como a análise de trabalhos já realizados na área de aplicação a fim de aperfeiçoar o desenvolvimento da presente aplicação.

Para o desenvolvimento da aplicação proposta, foi necessário definir as linguagens de programação a serem utilizadas, considerando o ambiente no qual a aplicação será disponibilizada para acesso. Além disso, devido ao foco da aplicação ser em coleta de dados georreferenciados, um estudo dos serviços da API do Twitter foi realizado com o propósito de filtrar tais serviços. Após a definição dos serviços a serem utilizados na aplicação, foi realizado o desenvolvimento da interface web.

Finalizado seu desenvolvimento, foram realizadas coletas de dados e análises espaciais sobre os dados coletados, a fim de demonstrar sua aplicação em um estudo de caso, que será detalhado na seção de Resultados. Foi realizado o procedimento de análise da concentração espacial de pontos, que foram os tweets relacionados ao Exame Nacional do Ensino Médio (ENEM) de 2017. Os dados foram coletados na noite do primeiro dia de provas do exame, 5 de novembro de 2017.

5. DESENVOLVIMENTO

A etapa de Desenvolvimento consiste na implementação de uma aplicação que por meio de uma interface web possibilita que os usuários tenham acesso aos serviços da API do Twitter que retornam dados georreferenciados.

A fim de atingir os objetivos do presente trabalho, as subseções a seguir detalham os processos de desenvolvimento realizados.

5.1. TECNOLOGIAS E LINGUAGENS DE PROGRAMAÇÃO

A aplicação deste trabalho foi desenvolvida para uma plataforma web. A escolha do ambiente foi baseada no fato de se utilizar serviços disponibilizados no

meio da internet e também pela facilidade de acesso que esse meio provê, já que não é necessário que o usuário da aplicação realize nenhum tipo de instalação para seu uso, apenas que tenha acesso à internet.

Delimitando a aplicação à plataforma web, foi escolhida a linguagem Java, padrão Java EE (Enterprise Edition) devido a um maior conhecimento e familiaridade com a linguagem por parte de um dos autores. O Java EE possui integração com diversos recursos web como HTML, JavaScript e CSS (Cascading Style Sheets), além de possibilitar um relacionamento entre outras :linguagens como Node.js e PHP (Oracle, 2017). Modelo de aplicação do padrão Java EE (Figura 4).

Figura 4 – Um modelo de desenvolvimento padronizado para todos os desenvolvedores de Java EE. Fonte - Oracle, 2017.

A aplicação é baseada na arquitetura MVC utilizando o framework Spring Web MVC (model-view-controller) que integrado ao Java EE permite que todas as partes da aplicação se comuniquem.

A arquitetura MVC possui um mecanismo que a partir de chamadas na camada view (usualmente conhecidas como telas, as quais o usuário consegue interagir), as requisições são recebidas e gerenciadas pela camada controller (camada onde se encontram todos os possíveis tratamentos das requisições que o sistema pode receber) a fim de direcionar para o devido processo em uma camada service com capacidade de se relacionar com a camada model (as entidades do sistema em si) (Figura 5).

Figura 5 – O processo de processamento de requisição no Spring Web MVC (alto nível). Fonte - Spring, 2017.

Para o desenvolvimento da interface web foram utilizadas a Linguagem de Marcação de Hipertexto (HTML), responsável pelo esqueleto da página; a linguagem de estilo CSS, responsável pela formatação dos conteúdos do HTML, como cor do texto, tamanho e estilo; a linguagem de script JavaScript, responsável por tornar a página HTML mais interativa e conveniente ao usuário, podendo também realizar validações, ações de eventos e realizar requisições à camada de controle sem a necessidade de atualizar a página na qual foi solicitada (Jepson, Macdonald, Stark, 2012); o framework Bootstrap, que permite uma melhor estruturação do HTML e CSS, além de disponibilizar bibliotecas de funcionamento de tabelas, plugins e tratar a responsividade da aplicação (Bootstrap, 2017).

Todo o desenvolvimento da aplicação foi realizado utilizando a Eclipse IDE para desenvolvedores Java EE, versão Neon integrada ao Java 8. Essa IDE oferece algumas facilidades como ferramentas de desenvolvimento Java EE e JavaScript, integração com o Maven (ferramenta de construção de projetos que permite integrar bibliotecas externas, pacotes e outros projetos) e integração ao sistema de controle de versionamento GIT (Eclipse, 2017; Maven, 2017).

5.2. SERVIÇOS DA API DO TWITTER

Conforme já descrito na subseção 4.1.1.1 da Fundamentação Teórica, a API do Twitter possui duas formas de coleta. A aplicação do presente trabalho utiliza apenas a primeira forma de coleta, devido ao foco na coleta de dados históricos. As possíveis requisições disponibilizadas pela REST API incluem os seguintes tipos de métodos:

- GET: retorna uma representação de um recurso.
- POST: cria um novo recurso.
- DELETE: deleta um recurso (Alam, Cartledge, Nelson, 2014).

De acordo com a documentação da API, existem sessenta e nove requisições GET, cinquenta requisições POST e duas requisições DELETE disponíveis (Twitter, 2017).

Com o intuito de permitir que a aplicação do presente trabalho seja direcionada para coleta de dados, foram tratadas apenas as requisições do tipo GET. Dentre as sessenta e nove requisições disponíveis, foi realizado um levantamento de quantas retornam dados georreferenciados, chegando ao resultado de apenas doze consultas, listadas e detalhadas abaixo:

- GET geo/id/:place_id: Retorna todas as informações conhecidas sobre um dado local.
- GET geo/reverse_geocode: Dada uma latitude e longitude, busca por até 20 locais que podem ser utilizados como parâmetro “place_id” em atualização de status.
- GET geo/search: Dado um par de latitude e longitude, endereço IP ou um nome, a requisição retorna uma lista de todos os possíveis locais que podem ser utilizados como parâmetro “place_id” em atualização de status.
- GET geo/similar_places: Localiza locais com nomes semelhantes e próximos às coordenadas fornecidas.
- GET search/tweets: Retorna uma coleção de tweets relevantes de acordo com uma busca específica.
- GET statuses/home_timeline: Retorna uma coleção dos tweets e retweets mais recentes postados pelo usuário autenticador.
- GET statuses/lookup: Retorna até 100 tweets por requisição, de acordo com os valores passados no parâmetro “id”, separados por vírgula.
- GET statuses/mentions_timeline: Retorna as 20 menções (tweets contendo o @screen_name do usuário) mais recentes para o usuário autenticador.
- GET statuses/retweets/:id: Retorna uma coleção dos 100 retweets mais recentes do tweet especificado pelo parâmetro “id”.
- GET statuses/retweets_of_me: Retorna os tweets mais recentes de autoria do usuário autenticador que foram retuitados por outros.
- GET statuses/show/:id: Retorna um único tweet, especificado pelo parâmetro “id”.
- GET statuses/user_timeline: Retorna coleção dos tweets mais recentes postados pelo usuário indicado pelos parâmetros “screen_name” ou “user_id”.

A partir da listagem de consultas acima foi construída a aplicação proposta.

5.3. CRIAÇÃO DA ESTRUTURA DE REQUISIÇÕES

A partir da seleção dos serviços a serem consultados, iniciou-se a etapa de desenvolvimento criando as estruturas para realizar o processo de requisições.

A documentação da API do Twitter indica alguns recursos

ISSN 1828-5961

para facilitar a implementação dos serviços públicos disponibilizados, sendo alguns mantidos pela própria empresa e outros de terceiros. No caso de aplicações Java são indicadas as bibliotecas hbc, utilizada na API Streaming; twitter-kit-android, utilizada para aplicações Android; e Twitter4J para aplicações Java e Android, mantida por Yusuke Yamamoto (Twitter, 2017).

Sendo assim, optou-se pela utilização da biblioteca Twitter4J. Esta possui integração de projeto utilizando a ferramenta Maven, que permite fácil inclusão das dependências da biblioteca. A utilização da biblioteca permite o acesso aos diferentes serviços e entidades necessários para a implementação das consultas da API (Twitter4j, 2017).

Após a integração da biblioteca ao projeto, iniciou-se o desenvolvimento das estruturas de consulta da aplicação. As camadas da aplicação são baseadas na arquitetura MVC utilizando o framework Spring Web MVC, conforme descrito na subseção 6.1 do Desenvolvimento.

Foram criadas as entidades na camada de modelo, de acordo com as consultas e baseando-se nas próprias entidades da biblioteca integrada.

A camada de controle é a responsável por direcionar as requisições e realizar os tratamentos necessários. Dessa forma, as requisições foram mapeadas na camada de controle e uma camada de serviço foi criada para implementar as requisições realizando uma comunicação direta com a API.

Por último, foi criada a camada de visualização, na qual o usuário interage diretamente com o sistema. Nessa camada estão as telas que possibilitam que o usuário escolha qual serviço deseja utilizar e preencha os devidos parâmetros para realizar a consulta. Fluxograma do processo (Figura 6):

Figura 6 – Processo de funcionamento da aplicação. Fonte - Adaptado de KUMAR et al., 2016.

5.4. CRIAÇÃO DA INTERFACE WEB

Depois de criar a estrutura de requisições, iniciou-se também a interface web. A interface possui uma página de entrada onde é requisitado ao usuário que o mesmo realize login utilizando sua conta do Twitter para ter acesso à aplicação. Após o login, o usuário é redirecionado à tela inicial onde são apresentados os serviços disponíveis e uma breve descrição de acordo com a documentação da API. Todas as telas possuem um menu lateral que é categorizado pelo tipo da consulta, sendo elas GEO, SEARCH e STATUSES.

Cada uma das consultas possui sua tela específica que é redirecionada ao ser selecionada. As telas de requisições contêm uma tabela com os parâmetros de requisição, campo de preenchimento de valor e breve descrição do parâmetro. Um sinalizador de campo obrigatório é utilizado nos parâmetros obrigatórios de cada requisição.

Para os serviços das consultas SEARCH e STATUSES, abaixo da tabela de parâmetros existe um filtro de resultados georreferenciados, que quando marcado como “true”, retorna apenas resultados que possuem um objeto “place” não nulo. A fim de se realizar o filtro de resultados, existe um parâmetro “max_retries” que limita o número de tentativas para se recuperar a quantidade de tweets requisitados no parâmetro “count”, levando em consideração o limite de requisições da API do Twitter, em que cada consulta tem um limite distinto por usuário conectado por login.

Após preencher os campos de parâmetros, o usuário submete a requisição e pode receber duas possíveis respostas: uma mensagem com algum erro que possa ter ocorrido durante o processamento da requisição, ou um redirecionamento para a tela de visualização da resposta. Quando o limite de tentativas é atingido, caso a consulta tenha encontrado algum resultado, este é retornado para visualização e é exibida uma mensagem de aviso de limite atingido.

Caso não encontre resultados na consulta, o usuário permanece na tela da mesma e é exibida uma mensagem de erro de limite atingido. Na tela de visualização da consulta é exibida a resposta da requisição em formato JSON, contendo todos os atributos relacionados ao objeto de retorno. Caso haja resultados georreferenciados, é possível visualizar a resposta em formato GeoJSON. É permitido que o usuário realize download das visualizações JSON, GeoJSON e formatos Shapefile (utilizado pelo software ArcGIS) e KML (utilizado pelo Google Earth), sendo que os três últimos só estarão disponíveis caso haja resultados georreferenciados.

Para o download nos formatos JSON e GeoJSON, a geração do arquivo é feita diretamente pela aplicação. Já para os formatos Shapefile e KML, é utilizado o web service Ogre, que permite a geração de arquivos em diversos formatos espaciais com base em um arquivo GeoJSON (Ogre, 2017).

6. RESULTADOS

Conforme descrito na seção de Metodologia, esta seção apresenta as etapas de coleta de dados e análises espaciais sobre os dados coletados. Dentre as doze consultas abordadas na atual pesquisa, pode-se considerar a “search/tweets” como a mais relevante a ser avaliada, visto que é por meio dela que se obtém o maior nível de informações coletadas aplicando-se filtros de localização.

6.1. CONSULTA “SEARCH/TWEETS”

Ao selecionar a consulta “search/tweets” o usuário tem a possibilidade de recuperar um limite máximo de cem tweets de acordo com o filtro estabelecido. Este filtro contempla os seguintes parâmetros:

- query: termo a ser pesquisado, podendo ser uma palavra ou frase.
- geocode: localização especificada por latitude, longitude e raio em milhas ou quilômetros.
- lang: idioma do tweet.
- result_type: tipo de resultado que se prefere obter, podendo ser “recent” (resultados mais recentes), “popular” (resultados mais populares” ou “mixed” (resultados tanto recentes quanto populares).
- count: número de tweets que se pretende obter, sendo o máximo 100.
- until: restrição de tweets anteriores à data especificada.
- since_id: restrição de tweets com “id” superior ao id

ISSN 1828-5961

especificado.

● max_id: restrição de tweets com "id" inferior ao id especificado.

● max_retries: número máximo de tentativas para se recuperar a quantidade de tweets especificada no parâmetro "count".

Além dos parâmetros detalhados acima, o usuário pode solicitar que sejam retornados apenas resultados georreferenciados. Apenas o parâmetro "query" é de preenchimento obrigatório.

Um comportamento a ser destacado com relação à consulta, é o fato de que a API permite recuperar apenas tweets de uma semana retroativa a data da coleta. Caso o usuário tenha necessidade de se obter tweets de um intervalo maior, se faz necessário realizar outras consultas de datas anteriores utilizando o parâmetro "until".

A subseção a seguir detalha um processo de coleta de dados utilizando a consulta "search/tweets".

6.2. COLETA DE DADOS

A proposta da coleta de dados para os testes da aplicação, é recuperar os tweets relacionados ao ENEM de 2017 nos estados brasileiros e analisar espacialmente a abrangência do termo. A coleta foi realizada na noite do primeiro dia de provas do exame, 05/11/2017.

Um dos filtros da consulta é o "geocode", que permite criar uma área de localização para se realizar a consulta. Dessa forma, foi realizada uma coleta para cada um dos vinte e sete estados brasileiros, incluindo o Distrito Federal. São listados abaixo os valores dos filtros aplicados na consulta:

● query: enem.

● geocode: latitude e longitude do centróide do determinado estado e o raio de extensão (em milhas) deste.

● lang: Portuguese

● result_type: mixed.

● count: 100.

● max_retries: 100.

Em todas as consultas foi marcado como "true" o filtro de resultados georreferenciados.

É importante ressaltar que as consultas desse serviço possuem uma restrição com relação ao intervalo de tempo. O limite máximo de recuperação de tweets é de uma semana retroativa, o que significa que apenas o intervalo de 29/10 à 05/11/2017 foi considerado na coleta do presente trabalho.

Para cada coleta realizada, foi criado um diretório contendo os arquivos JSON, GeoJSON, Shapefile e KML. A maioria das consultas recuperou o limite máximo de 100 tweets, enquanto outras consultas recuperaram pelo menos um tweet ou uma média de 30 tweets, totalizando 1348 tweets. Algumas das consultas podem ter recuperado o mesmo tweet, devido ao fato do raio informado para o determinado estado poder abranger outro vizinho.

6.3. ANÁLISE ESPACIAL DOS DADOS COLETADOS

O estudo de caso do presente trabalho prevê uma análise espacial dos tweets coletados utilizando a aplicação proposta. A análise tem como objetivo gerar uma visualização da abrangência do termo pesquisado relacionando a localização dos tweets, número de ocorrências por localidade e a população da determinada localidade.

Para isso, foi escolhido o software QGIS versão 2.18 devido ao fato de ser um SIG de código aberto multiplataforma. Ele oferece diversas funcionalidades de visualização e análise de dados, entre outras (QGIS, 2017).

Inicialmente, todos os arquivos Shapefile do diretório criado foram carregados no QGIS. O software considera que cada arquivo carregado é uma camada do mapa que está sendo construído. Sendo assim, todas as camadas foram mescladas para ser criada uma camada única contendo os tweets de todos os estados, resultando numa camada com 1348 tweets.

Considerando o fato de que na coleta de dados alguns tweets podem ter sido retornados em mais de uma consulta, foi necessário realizar a normalização da base de dados. O processo de normalização consiste no filtro dos tweets utilizando o parâmetro "status_id" que é único do tweet. Assim, foi feita uma consulta utilizando esse filtro na base de dados, gerando uma nova camada, totalizando 1195 tweets a serem analisados.

Primeiramente, são apresentadas no mapa as localizações dos tweets coletados pela aplicação. Cada ponto apresentado possui a soma de todos os tweets encontrados para aquela localidade. A partir desse dado, foi criado um mapa de concentração de tweets por meio da ferramenta de Densidade de Kernel para mostrar as principais áreas de concentração das publicações com o termo "ENEM" pelo país. Percebe-se que as maiores cidades do país, como Brasília, São Paulo, Rio de Janeiro, Recife, Vitória e Porto Alegre foram as que mais apresentaram retornos, pois concentram maior população também (Figura 7).

Figure 7 – Localização e Mapa de calor de tweets sobre ENEM nos municípios brasileiros. Fonte – Os autores, 2017.

Para esse tipo de análise é esperado que as cidades mais populosas também apresentarão maiores resultados. Portanto, uma segunda análise foi gerada considerando o percentual de pessoas que realizaram que postaram no twitter por localidade, correlacionado o número de tweets com o total da população da cidade. Portanto, foi realizada a integração da camada gerada com uma camada contendo a população dos municípios, utilizando uma base de dados com as estimativas de população para o ano de 2017 publicada no Diário Oficial da União e também disponibilizada pelo IBGE (IBGE, 2017).

Primeiramente, a base foi convertida para o formato dBASE (.dbf) e depois carregada para o QGIS. Essa camada foi unida a camada dos tweets utilizando o atributo "NOME DO MUNICÍPIO" na base da população e "place_name" na base de tweets.

Após a união das camadas, notou-se que haviam tweets sem relação de população. O fato se deu por serem tweets com localização do estado, e não do município. Sendo assim, os tweets foram removidos da camada a fim de se manter a homogeneidade dos dados em escala municipal.

Com o intuito de se quantificar a ocorrência de tweets em cada município, foi criado um campo

ISSN 1828-5961

"count" na tabela da camada para armazenar a quantidade de tweets em um dado local identificado no atributo "place_id". A partir desse novo campo, foi criado um novo campo contendo o percentual de ocorrência em relação à população.

O campo "perc" foi criado a partir do cálculo conforme a Eq. 1:

$$\frac{(count \times 100)}{est}$$

onde: count = quantidade de tweets, est = estimativa da população de dado local.

Sendo assim, consolidou-se uma base de tweets contendo informações de localização e percentual de ocorrências em relação à população municipal.

6.4. VISUALIZAÇÃO DA ANÁLISE DOS DADOS

Para a visualização dos dados, a camada de tweets foi convertida para pontos correspondentes ao centróide de cada polígono que representa uma localidade. A partir dos pontos gerados, foi criada a visualização de mapa de calor por meio da aplicação do modelo de densidade de Kernel, que consiste em um método estatístico no qual é criado um mapa de matriz de densidade a partir dos pontos em vetores (Diniz, Palhares, Ribeiro, 2017). Utilizando o atributo "perc" como ponderador de densidade, o método desenha uma vizinhança circular ao redor de cada local de ocorrência do atributo. O mapa gerado representa a concentração espacial da consulta realizada: (Figura 8):

Figura 8 – Mapa de calor com ocorrências de tweets sobre o ENEM em municípios brasileiros. Fonte – Os autores, 2017.

As áreas em tons mais verdes demonstram locais com uma quantidade menor de ocorrências do termo "enem" em relação à população, enquanto as áreas com cores mais quentes demonstram uma quantidade maior em relação à população.

A partir do mapa é possível concluir que os locais com os maiores percentuais foram os estados do Rio Grande do Sul, Paraná e nas divisas de Minas Gerais com Espírito Santo e Ceará com Rio Grande do Norte. Relacionando a visualização do mapa com a tabela de tweets, os municípios Estrela Velha (RS), Iguatu (PR), Taboleiro Grande (RN), Anta Gorda (RS), Paim Filho (RS), Ibarama (RS), Riqueza (SC), Selbach (RS) e Dom Cavati (MG) correspondem aos locais com maiores densidades de pontos, em ordem decrescente do percentual.

Esse modelo de análise espacial é amplamente utilizado nas geociências e sua interpretação dá indícios de comportamento espacial de determinado fenômeno. Neste estudo, o termo "enem" foi utilizado como exemplo em função de sua relevância no momento em que se desenvolvia essa pesquisa. Entretanto, outras consultas podem ser feitas na plataforma a fim de contribuir para realização de diagnósticos que podem auxiliar no planejamento territorial, como em casos de saúde pública, violência, identificação de aspectos culturais, dentre outros. Para a busca de outros temas de consulta, o usuário precisa apenas utilizar a consulta preenchendo o parâmetro "query" com o termo ou expressão que se deseja consultar.

7. CONCLUSÃO

O diferencial da aplicação desenvolvida é que ela permite a interação de usuários de diversas áreas sem exigir conhecimento de programação. Além disso, ela possui maior foco nos profissionais das geociências, por permitir que as consultas da aplicação sejam filtradas para recuperarem apenas dados georreferenciados e ainda possibilitar o download dos resultados em formatos específicos utilizados em softwares de SIG, como ArcGIS e Google Earth, permitindo uma melhor integração entre as etapas de coleta e análise de dados.

Com a integração dos dados em um SIG, permitindo a análise e visualização dos dados, é possível verificar que os formatos gerados pela aplicação também estão de acordo com a realidade dos profissionais de áreas das geociências.

A proposta de aplicação do presente trabalho foi desenvolvida de acordo com os objetivos propostos inicialmente. Contudo, alguns ajustes devem ser realizados para que seu funcionamento seja colocado em produção. Levando em consideração a experiência do usuário, é necessário que o feedback das consultas seja aprimorado para fornecer informações em tempo real do andamento das consultas realizadas na aplicação. Até o presente momento, o usuário submete a requisição da consulta e não recebe nenhuma informação até que a consulta seja finalizada e seu resultado retornado.

Com o desenvolvimento da aplicação, a principal dificuldade encontrada foi decorrente do uso da API disponibilizada pelo Twitter. Tal dificuldade foi o limite de requisições da API, visto que o filtro de resultados georreferenciados consiste em uma iteração constante da consulta, até que sejam encontrados dados de acordo com o filtro, o que resulta num maior número de requisições comparando-se com uma consulta sem esse filtro. Esse alto número de requisições ocorre devido a uma limitação da API do Twitter, que estabelece um número máximo de resultados de acordo com a requisição, a fim de reduzir o processamento e tráfego dos dados.

As principais contribuições deste trabalho são:

- Possibilidade de estudo e desenvolvimento de uma solução computacional utilizando os conhecimentos e técnicas aprendidos no decorrer do curso;
- Proposta de uma nova solução que integra as áreas da computação e geociências, estreitando o gap entre as duas áreas;
- Facilidade de acesso aos dados que são disponibilizados a partir de APIs com linguagem restrita ao meio da computação;
- Possibilidade de novas pesquisas serem realizadas a partir dos dados georreferenciados coletados no Twitter utilizando a aplicação.

7.1 TRABALHOS FUTUROS

Como trabalhos futuros, levando em consideração a experiência do usuário, é pretendido realizar testes com os profissionais de áreas das geociências a fim de averiguar se a usabilidade da aplicação atende às necessidades do público específico.

Também pretende-se estender o público da

ISSN 1828-5961

aplicação para além dos profissionais de áreas das geociências. Para isso, se fará necessário o desenvolvimento das demais consultas disponibilizadas pela API do Twitter e um estudo dos formatos de dados aceitos em outros softwares específicos, como no caso de mineração de dados.