

## A collaborative application for incidence data recording and geographic distribution of myiasis

Stefano W. P. Pontes<sup>1</sup>, Tiago B. Borchardt<sup>1</sup>, Lívio M. Costa Júnior.<sup>2</sup>

<sup>1</sup>Department of Informatics

<sup>2</sup>Department of Pathology  
Federal University of Maranhão (UFMA)  
São Luiz – MA – Brazil - CEP 65080-805

{stefano.walker, tiago.bonini, livio.martins}@ufma.br

**Abstract.** *This paper presents rMiíase, a collaborative mobile application that allows reporting cases of myiasis. To support the application, a complete structure with a web server was developed to receive, provide and synchronize the registration data of the cases reported by the users. It also has a management system for administrators to control the data registered. The application encourages users to submit or request the collection of samples that will be reviewed by laboratories for purposes of identification of myiasis-causing species. With the data obtained by the application, will be possible to study the feasibility of implantation of myiasis control plan in the affected regions.*

### 1. Introduction

Myiasis is one of the most common parasitic diseases that affects warm-blooded animals. In this disease, flies lay their eggs on the injured skin of animals. When the larvae hatch from eggs, they invade the subcutaneous tissue from where feed, causing severe tissue damage and leading to death [Teixeira 2013]. Myiasis can occur in both rural and urban areas, infecting animals and humans. The regions most favorable to insects appearance that causes myiasis are tropical and subtropical regions [Hall and Wall 1995].

*Cochliomyia hominivorax*, the main species causing myiasis, causes financial losses to brazilian cattle herds that have been estimated at about 150 million dollars a year. Losses generated by myiasis include weight loss, a decrease in milk production, leather damage and animal mortality. The prevention and treatment of these affected animals, in addition to raising the cost of production, contribute to the presence of drug residues in milk and meat, bringing harm to human health [Borja 2003].

The *C. hominivorax* had its eradication and control in the United States and part of Central America since the 1980s. Prior to eradication, the annual cost of preventive and control measures was US\$ 120 million in 1960. In Panama reached US\$ 43 million per year [Oliveira et al. 1982].

In Brazil, in order to implement a control plan for myiasis, it is necessary to collect data for knowledge of the following indices: the number of disease occurrence in the South America regions, the time of the year which they is most abundant, the fly specie and the animal host that is most infested. Currently, such data are non-existent. About these records, Teixeira (2013) concludes that there is a small number of health professionals who identify the agents that cause the infestation of this disease. Therefore, the

occurrence rates in Brazil may be underestimated in relation to myiasis in animals include humans.

Based on this need, rMiíase (*report* myiasis) has been developed with the goal of being an accessible application for everybody record cases of myiasis. The user can register a case of myiasis by providing some basic data, such as host species, address, and location of the injured body. Then you can add the point in the map where the case is being recorded and capture an image of the injured body part by myiasis. In the end, the user is encouraged to collect a larvae sample and send it to one of the accredited laboratories, which will also appear on the application.

The user can see all the cases registered by others and the photo registered by them. Encouraged by the application, people are expected to send the samples to the accredited laboratory nearest to their locality, in order to the cases to be confirmed, obtaining information on the species of fly that is causing the pathology. As the main objective, the application is to provide a database that will serve to study the possibility of implementing a control program of myiasis.

The remainder of this article is organized as follows: Section 2 presents some related works. In Section 3, the system architecture of the application is described. In Section 4, some analysis is done about the application. Finally, in Section 5 the conclusions of the study are presented.

## 2. Related Work

There are some projects that aim to function as geographic data recording platforms, using maps and allowing users to add data to the application base. Some of these works will be presented below.

As a first example, we can cite the app Fruit Map (2016), in which users can search for fruit trees in any region, besides to registering the presence of new trees. The user indicates on the map the location of the trees, selecting the type and degree of availability of the fruit. This information is available for everyone to see through the device so that other users can know which fruits are closest.

As another example, there is the app Dengue SC (2016), that is intended to reinforce the control of *Aedes aegypti* mosquito. Through this channel, the population can denounce the suspected outbreaks of dengue fever. Once the complaint is registered, the city hall has access to the event through a web tool that allows to follow the requests received and proceed with the verification.

There is the app WildHelp [Clark 2006], available for iOS devices, which helps users report sick or injured wildlife to an appropriate agency. The user captures and sends an animal photo to the application which, based on GPS coordinates, sends the photos and information to a nearby environmental agency. The information that users provide can help researchers, rescuers, and wildlife managers to understand better how and where animals get hurt.

The US Centers for Disease Control and Prevention (2017) developed the Epi Info™ application for the vector of mosquitoes, among them mosquitoes, *Aedes aegypti*, *Culex* and *Anopheles*. It is an application aimed at entomologists and agents to combat

endemics. This application provides analysis of entomological indicators for planning and appropriate decision making on the actions of vector control measures.

These presented projects have several similarities with rMiíase, following the idea of reporting cases based on geographic location. However, the main difference between these tools and the rMiíase is the need to collect the samples from the registered cases to be analyzed in the laboratory, in order to know the species causing the myiasis. Because an eventual myiasis control plan needs to be individualized for certain species [Borja 2003]. Due to the complexity of the explanation, the other features, which allow differentiating the method from the other tools presented, will be described in the following sections as part of this article.

### 3. rMiíase Application

The client-server architecture, proposed by [Fielding 2002], was implemented to meet data synchronization requirements, such as registration and update data of user and cases. As shown in Figure 1, the architecture components are (1) mobile application, (2) Google Maps APIs, (3) GPS, (4) web server, (5) database and (6) web management system. Some of these components will be detailed in the following subsections.

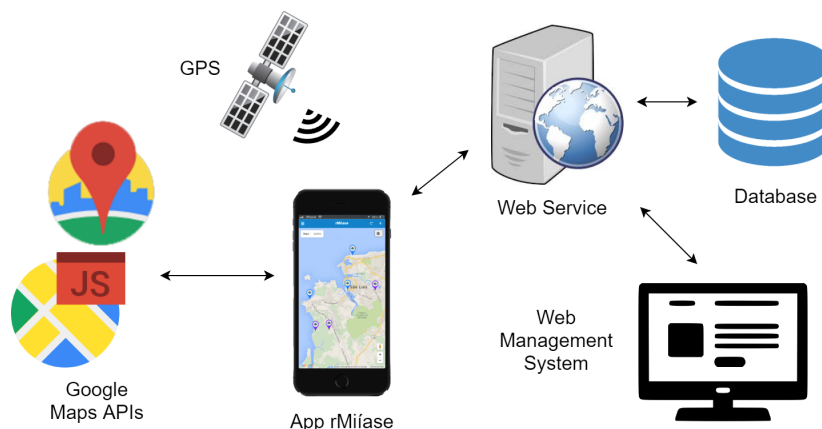


Figure 1. Application Architecture

#### 3.1. Mobile Application

A collaborative mobile tool was designed to solve the need to create a database that records the occurrence rate of myiasis. The basic features functionality of the application are as follows:

- Allow the user to register a new case of myiasis. The case data is:
  - Species of the affected animal (canine, bovine, goat, human, etc.), the address of the occurrence and the part of the body injured.
  - The geographical location of the case, which will be through marking the location on the map provided on the screen, suggesting the current location by the GPS.
  - Image of the injured body part, which can be captured or added from the device gallery.

- Indication of commitment to collect sample from the injured part of the animal to be sent to one of the accredited laboratories (optional).
  - Selection of the laboratory to which the user wishes to send the collected a sample (depending on whether the previous item was flagged in the affirmative).
- View on the map the cases registered by other users of the application, showing details, including the case image.
  - Visualization and editing the data of the registered cases by the user. The user cannot exclude a case that has already been analyzed and approved by the laboratories.
  - Registration and updating of the user's personal data (name, e-mail and telephone number).
  - Access to information about myiasis disease, the importance of collecting and sending the samples, and access the website of the application.

The initial version of rMíase is designed for Android and iOS mobile operating systems and is available on Play Store and App Store. rMíase is a hybrid application, i.e., it is partially native, because of accesses the functionalities of the hardware, and partially web, because uses HTML, CSS, and JavaScript as web technologies. This type of development has the great advantage that the same application works on multiple platforms and is not necessary to write a different code for each one [Budiu 2013].

Apache Cordova was used in development [Wargo 2015], which is a multiplatform open source web application framework. The development of applications using HTML, CSS, and Javascript web technologies is possible with Cordova. This tool provides a set of APIs for accessing native functions of the Operating System and device hardware using Javascript without writing native code.

The development IDE used was Intel XDK [Intel 2016]. The graphical interface was developed based on Framework7 (2015), that is a free and open source HTML framework for application interface development.

### **3.2. Google Maps APIs**

Google offers several types of Application Programming Interface (API). APIs allow an application to use the features provided by it, without getting involved in the details of the software implementation.

The Google Maps JavaScript API belongs to the category of web APIs and allows users to use maps and custom mapping information in their applications [Google 2016a]. Map customization styles are defined using the JSON<sup>1</sup> format, so the definition of the same map is supported in every web application. This customization format is dynamically processed in the application at runtime.

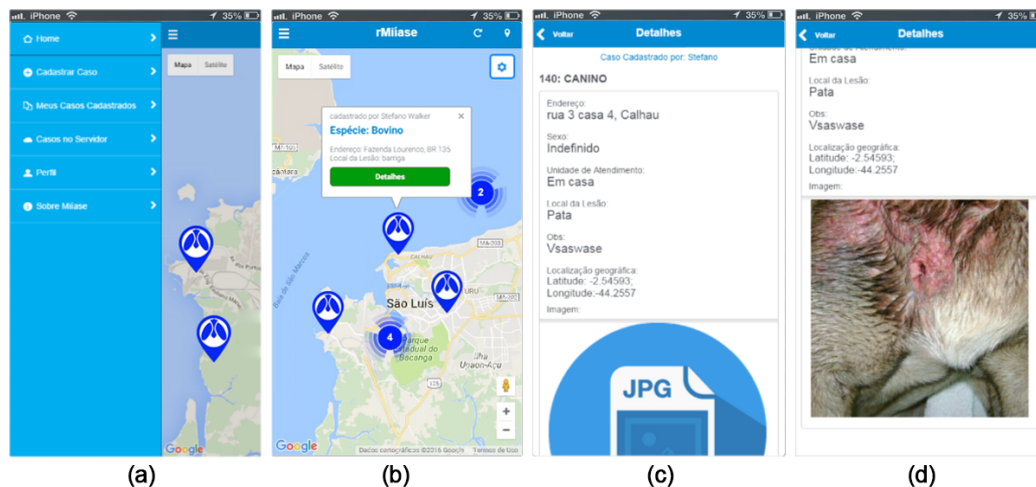
---

<sup>1</sup>JSON is an acronym for “JavaScript Object Notation”, it is a light format for computer data interchange. JSON is a subset of JavaScript object notation, but its use does not require JavaScript exclusively (FLANAGAN, 2006).

The mobile app also uses Google Geolocation (2016) to get the location of the user's device. The Google Geolocation API returns a location and a precision radius based on GPS information, cell towers, and Wi-Fi access points that the mobile client can detect [Google 2016b]. Communication is performed by HTTPS using POST method. Both the request and response have JSON formatting and the content type of both is *application/json*.

### 3.3. Application Front-end

First, the app loads the Google Maps interface (2016), displaying the cases registered in nearby locations. By clicking on the case marker, the user visualizes a small frame with some data of the case as species, the user that registered and address informed. To view more details of the case, the user must click on the green button "Details" (Figure 2 (b)). Figure 2 (c) shows the possibility for user to access details of a case registered by another user, initially having the image of this case not loaded. The user can see the image only if clicks on the image download button (Figure 2 (d)).



**Figure 2. Application: (a) Menu; (b) Initial screen showing the map and registered case markers; (c) Details screen of a case of another user; (d) Details screen after downloading the image.**

To register a new case, the user will double-click anywhere on the map, inserting a new case marker (Figure 3 (a)) which, when clicked, will start the "Register Case" function. Another alternative to access this function is through the menu. After that, the data entry screen will appear. (Figure 3 (b)).

The user is directed to the map where he should mark the locality where the case occurred after completing the form, if not previously marked. In the next step of the case registration, there is the option to insert an image through the camera or the image gallery (Figure 3 (c)). If the option "do you want to send a sample?" is marked as affirmative, before finalizing the register, the map will appear again for the user select the laboratory to send the collected sample (Figure 3 (d)). The application always suggests the nearest laboratory to location of the registered case. After saving the case, the data is stored in the device and the web server, and can be edited or deleted by the user, except when the case is approved after the sample be analyzed in the laboratory.

Figure 4 (a) shows the list of cases registered by the user, where he is allowed to see the details and also edit and delete the cases. The list of this user's cases is stored on the device and web server and is synchronized with every change made by the user or by the management system administrator. Figure 4 (c) shows the user registration screen, invoked during the registration of the first case.

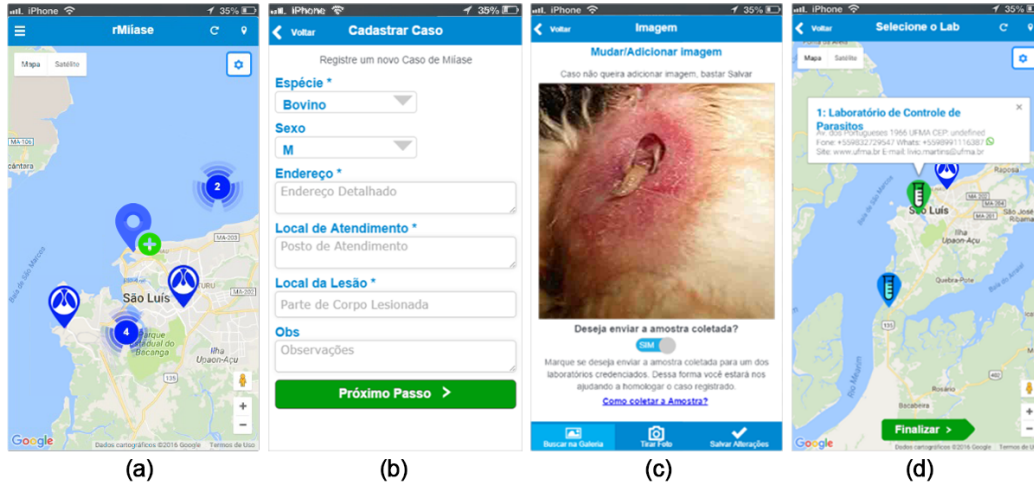


Figure 3. rMíase application: (a) Home screen with the addition of the case register marker. (b) Case registration form. (c) Image capture from the camera or gallery. (d) Selection of laboratories to send samples collected.

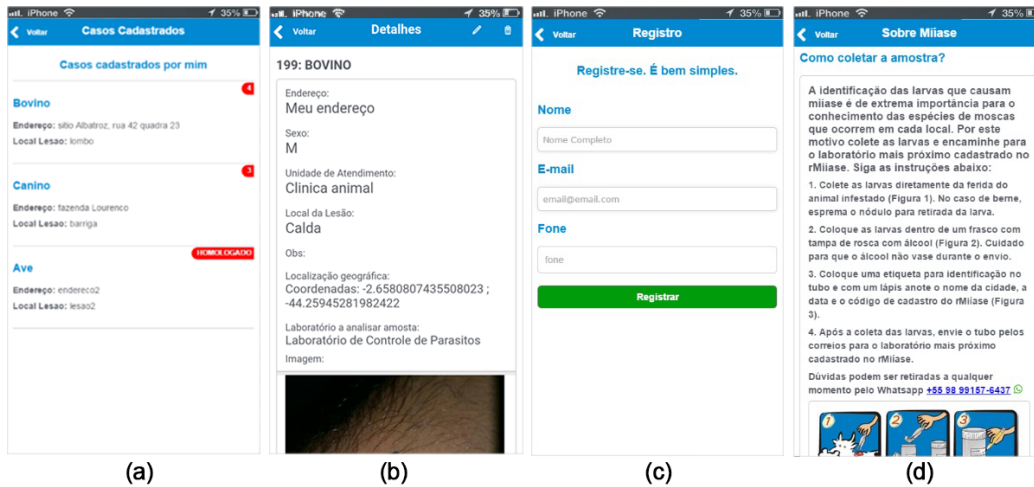


Figure 4. rMíase application: (a) List of cases registered by the user. (b) Detail screen of a case registered by the user. (c) User registration form. (d) Section about myiasis.

### 3.4. Application web server

The application server has been implemented in PHP language with a MySQL database and has the same modeling structure as the internal database of the mobile application.

The web server can be accessed by POST requests, which return data or error parameters. The possible types of requisitions are detailed below:

- *uploadUser*: Receives user registration data from the application to the web server.
- *uploadCase*: Receive the case data.
- *deleteCase*: Delete case on server.
- *seeHomologatedCase*: Checks if the case has already been approved by laboratories (called before *deleteCase*).
- *updateUser*: Receive user data and update on the server.
- *updateCase*: Receives the case data and updates on the server.
- *downloadPicture*: Receives the case id and returns the corresponding image.

All cases registered in the server appear on the map application in the form of markers (Figure 2 (b)). The application accesses the web server to retrieve the case data in JSON format and load the markers. The same process is performed to access the list of laboratories.

The web server API is responsible for data integration. The server provides the database information for the mobile application platform and for the management system via the GET request. This service may provide two types of information: the data (exception for the image) of the registered cases and the data of the registered laboratories. When the API is accessed, it queries the data in the database and mounts a JSON package to a web address, allowing the mobile application to capture that package to list the data at a given time. The request to access the API data is done through the Ajax<sup>2</sup> request, taking advantage of its asynchronous characteristic that allows the loading of isolated components.

Images of cases registered by other users are not loaded with the map markers. Otherwise, this process would require a lot of server data flow, and a smartphone would not support a large amount of data loaded into its RAM. Therefore, the images of the cases are only displayed if the user accesses the details of the case and click the specific button for image display. The download and upload of the images are also done through Ajax requests.

### 3.5. Web Management System

The app has a homepage<sup>3</sup>. This page contains information about the project, application download links, and contact information. The Web Management System has restricted access by login and password. The administrators of this web system can perform the following functions:

- edit or delete the cases registered by any user of the application;

---

<sup>2</sup>Asynchronous JavaScript and XML. Ajax basically uses JavaScript, XML and HTML dynamically. It allows, immediately, the validation of forms, suggestion of automatic completion and other exchange of data, without having to reload the entire web page.

<sup>3</sup>Available in <http://www.rmiase.ufma.br>

- edit, delete or block any user;
- manage the list of accredited laboratories;
- homologate the registered cases, when the sample sent by the user is analyzed by one of the laboratories;
- download the database into spreadsheets.

The management system was developed using Phreeze (2016), a framework for developing CRUDs<sup>4</sup> in PHP. Phreeze generates a complete generic system in the MVC (Model-View-Controller) and ORM (Object Relational Mapping). Figure 5 presents the web management system screen, in the case editing functionality.

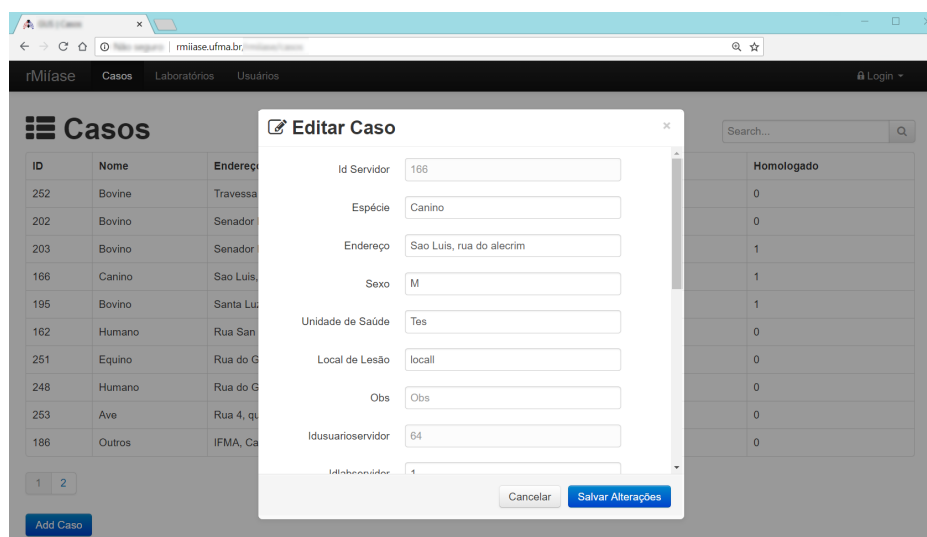


Figure 5. Web Management System: “Edit Case” functionality.

#### 4. Application Analyzes

The main tests of the application were made in the Android platform, in smartphone considered of intermediate level, composed of a processor with 1.2GHz of clock, 2GB of RAM, the screen of 5 inches and rear camera with 13Mpx.

For the Android platform, the app was produced with backward compatibility up to version 4.0 (API Android 14), reaching approximately, in this year, 94% of Android devices. For iOS, it reaches up to version 8.0 and has compatibility with 96.5% of iOS devices. The application, in the versions for Android and iOS, presented a good graphic interface layout with Framework7.

The images captured by the camera provided enough quality photos for the proposal. The storage size of the images obtained was in the range of 40 to 90 KB, with the approximate resolution of 500x500 pixels, after the application of compression algorithm. The small file size makes easy transfer data and leaves the quality of the photos at an acceptable level.

<sup>4</sup>CRUD is an acronym for Create, Read, Update, and Delete, the four basic operations used in relational databases.



Images in JPG format are stored in the gallery (in the default storage directory) and are also stored in the application's internal database in the format Base64<sup>5</sup>. The computational cost dedicated to Base64 file format conversion is small and produces a binary file with approximately 75% compression. In tests, the process of downloading registry data of 18 cases, spent 6.3KB of connection traffic to the server, an average of 0.35KB per case. All cases data of other users are not permanently stored on the user smartphone that visualizes.

The data synchronization scheme was implemented to address the situation of no connection to the internet. Synchronization routines are triggered in all register and edit functions, as well as in application initialization events (*OnStart()*), background return (*onResume()*) and connection to the internet (*onOnline()*).

The application dynamically loads cases. When the user accesses the map, the cases are viewed in a grouped way, according to the zoom of the map screen, and downloads the data registered by other users as the user navigates the map. Finally, rMíase has support for the Portuguese, Spanish and English languages, in order to reach users from the Americas.

## 5. Final Discussions

The rMíase has proven to be a good solution for registering cases of myiasis around the world, especially in South America. The application will provide reliable data for myiasis studies in this continent, along with the implementation of the web system for homologation/confirmation of cases, providing a great contribution to the area of public and animal health.

The tool has the advantage of requiring little data usage when connecting to the server. In addition, the rMíase works offline, allowing the manipulation of data of registered cases and their subsequent synchronization, once it is connected to the internet. The application does not require professionals to be scattered across regions by collecting samples, the system administrators can receive regional data directly from affected populations.

The application has advantages of maintaining source code, by using the web-based development, which allows the same source code to be used in the production of a cross-platform application. This results in a reduction in software update time and cost.

As future enhancements, the authors suggest adding a dynamic data visualization tool to the management system, providing disease statistics as graphs, filtering queries, and exporting reports to web system administrators.

The rMíase is registered<sup>6</sup> in the National Institute of Industrial Property (INPI) and is expected to begin to be used for official data collection in 2018.

## 6. Acknowledgment

The authors thanks the technical support provided by the Information Technology Nucleus (NTI) of UFMA and the financial support by FAPEMA, MAPA and CNPq for the execution of this work.

---

<sup>5</sup>Base64 is a method for encoding data for download on the Internet.

<sup>6</sup>Computer program registration number: BR5120160018321 (<http://www.inpi.gov.br/>)

## References

- Borja, G. E. M. (2003). Erradicação ou manejo integrado das miíases neotropicais das américas. *Pesquisa Veterinária Brasileira, SciELO Brasil*, vol. 23.
- Budiu, R. (2013). Mobile: Native apps, web apps, and hybrid apps. <https://www.nngroup.com/articles/mobile-native-apps/>. [Accessed on October 13, 2016].
- CIASC (2016). Dengue SC. <http://www.ciasc.sc.gov.br/302/>. [Accessed on November 8, 2016].
- Clark, S. (2006). New WildHelp app helps users report sick or injured animals. *Mercury News*. <https://www.mercurynews.com/2016/06/24/>. [Accessed on June 20, 2018].
- Fielding, R. (2002). Principle design of the modern web architecture. <http://dl.acm.org/citation.cfm?id=514185/>. [Accessed on August 13, 2016].
- Framework7 (2016). Framework7. <https://framework7.io/>. [Accessed on January 13, 2017].
- FuitMap (2016). FruitMap. <https://fruitmap.app/>. [Accessed on November 8, 2016].
- Google (2016a). The Google Maps API. <https://developers.google.com/maps/>. [Accessed on August 24, 2018].
- Google (2016b). The Google Maps Geolocation API. <https://developers.google.com/maps/documentation/geolocation/>. [Accessed on August 24, 2018].
- Hall, M. and Wall, R. (1995). Myiasis of humans and domestic animals. *Advances in parasitology*, 35:257–334.
- Intel (2016). Intel xdk. <http://xdk-software.intel.com>. [Accessed on December 13, 2017].
- Oliveira, C., Mova, G., and Mello, R. (1982). Flutuação populacional de *Cochliomyia hominivorax* no município de Itagual. *Brazilian journal of veterinary research*, Rio de Janeiro.
- Phreeze (2016). Phreeze Framework. <http://www.phreeze.com/>. [Accessed on December 27, 2017].
- Teixeira, D. G. (2013). Principais dípteros causadores de miíases. *Programa de Pós-graduação em ciência animal*.
- Wargo, J. M. (2015). *Apache Cordova 4 Programming*. New York, Pearson Education.