

Development of application for registration of veterinary procedures with Python language, frameworks and database

Alyson Bueno Francisco
 Professor Doctor in Geography
 Sao Paulo State University
 Bachelor in Information Technology
 Sao Paulo, Brazil
 Contact: alysonbueno@gmail.com

ABSTRACT

There are care entities with a lack of systems for registering information, with the need to develop a computer technology to facilitate data records. Faced with this challenge, an application was developed with front-end and back-end technologies for cadastral computerization. The development of the application had the back-end technologies with the Python language, Django framework and MySQL database; and front-end with the HTML, JavaScript and CSS markup language through the Bootstrap framework. The development carried out by the team was recorded via Git in a remote repository on GitHub, for monitoring, version control and implementations of other functionalities. The application that has been developed will contribute to the storage, retrieval and manipulation of essential data, providing agility in the registration, consultation of animal records and conferring, a significant gain in the time spent by users in the computerization of data.

Keywords

back-end and front-end development; computerization; database

1. INTRODUCTION

It is noticeable that in cities far from the large centers there is a technological discrepancy in relation to the computerization of public institutions or even private sectors of these municipalities. This stems from several factors, often associated with the lack of technological infrastructures, such as difficulties in accessing good quality internet and the absence of adequate equipment and software. Another important aspect arises when there is even contact with certain technologies, but there is no appropriate training for the people who make use of them.

Low investment in information and communication technology in small cities can limit access to modern and effective information tools and systems, impairing the time spent on data manipulation and strategic decision-making in various sectors.

The implementation of a computer application can be an opportunity for a care entity to have access to a tool that facilitates the management of information and data, in

addition to helping to increase the efficiency of operations and queries performed.

In addition to ensuring the registration of data to facilitate the work of veterinary procedures, professionals in the area need an accessible computer technology. In this sense, the interface generated for the software aims to ensure that users make the most of the technology with quality in use. The recent computational technologies need to be accessed in the various devices with ease of use and data registration.

Recent computer technologies require a team of developers to ensure favorable results, whose version control is a history of the changes made in the development. Given this, GitHub is one of the examples of a version control system, whose programming codes are stored and can be accessed and changed by members of the developer team. In the prototyping of the software it is necessary an integration of ideas and tasks for the team of developers to realize the project and ensure favorable use of application of the software.

In the development of an information system it is necessary an implementation of the database and the interface for favorable visualization of the elements. In addition, the systems need to be integrated into the network for data entry remotely, since the user community depends on the use of personal devices.

In the development of the application it is necessary to diagram the database with the selection of attributes to make the registers before handwritten in information for manipulation and query. In short, the registration of veterinary procedures needs a database with access facilitated by the interface to professionals with initial knowledge of computing. In addition, back-end development needs to be safe, because the loss of data, even if it does not directly involve finances, can hinder the accomplishment of the tasks of the veterinary branch.

Information systems can contribute to the access to technologies of assistance entities with a lack of resources for the acquisition of software with a commercial license. In this sense, research projects in universities in the area of

computing contribute to the generation of software directed to the target audience of care entities in regions with social inequalities and absence of technological diffusion.

From these aspects, the objective of the investigation was to develop software to register veterinary procedures, with integration of a database, and a version control to favor the collaboration of the developers.

In the veterinary care entity, with a shortage of financial and technological resources, the records of the procedures are performed manually on cards. The forms were separated to record the data of the domestic animals, the performance of surgical procedures, the donors of the domestic animals and the veterinary medicine professionals who perform the surgical procedures. In view of these aspects found in an interview conducted with the manager of the entity, it was found the need to develop a computerized system with database. In the specification and design of the system, it was decided to elaborate a system executable in browser in the Windows operating system, due to the existence of a desktop in the assistance entity of domestic animals.

From this objective was developed the prototyping and implementation of the software with the technologies of frameworks and version control.

2. THEORETICAL REVISION

During software development occurs the definition of the algorithm to solve a problem and the choice of programming language where the code to be understood by the computer will be written, with the definitions of the functions, commands, data types or other resources that the problem demands. The general use and the extensive library that the Python language makes available for the development of various types of sophisticated applications, have made it a popular tool and is often used as an introductory language for new programmers, because its interface is relatively simple (Perkovic, 2016).

Regarding the concepts related to fundamentals and web development, according to Maciel (2020, p. 279): "framework is a set of classes implemented in a given programming language, which serves to facilitate the creation of applications". Django is a robust framework with support for various types of applications, among which it also applies – but is not limited – to web development, and is often used by professionals in the area of Computing, and can be applied in a Database Management System associated with Python code.

In general, when it comes to the manipulation and access to data, it is necessary to implement a database that makes direct communication with the developed application, within the choices of each project. According to Dietel (2008, p. 446) "a database is an organized collection of data [...] Programs connect to a relational database through an

interface, that is, software that facilitates communication between a database management system and a program." In addition to storing information, the database also has, among its main functionalities, the retrieval and updating of data, and the ability to perform queries and generate reports and analyses from this information.

In the application of schemas and architectures for database manipulation, it is recommended to implement a DBMS, since its use assists and automates processes of integrity checking and access control, adding layers of security and avoiding information loss, among other applicable functionalities. For the implementation of data modeling in a relational database, it is essential to use the SQL language, responsible for querying, defining and manipulating data (Pichetti; Cortes; Paixao, 2020).

Regarding access to the database, the files are not read, written or accessed by the application program directly (PERKOVIC, 2016). They communicate by sending SQL commands to a DBMS for the sake of the application. Still according to the author, the Python standard library has a SQLite3 database module, but this language can also make use of other databases. In this sense, about other database engines, "to access them, you need to use an API (that is, a Python module) that offers classes and functions that allow Python applications to send SQL commands to the database engine" (Perkovic, 2016, p. 444).

Database management can occur through open license systems and integration with programming languages such as: SQLite, MySQL and PostgreSQL. In the implementation of the commands for access to the databases, the Python language requires the installation of the drivers of the chosen DBMS. The pip command performs the installation of the packages in the Python code. Due to the integration of the Python language with the Django project, there is a need to change the settings.py file of the django.conf package to include the information of the chosen DBMS, if it is different from the default database (Perkovic, 2016).

Regarding the Django framework, being the main one applied to the Python language in back-end development, its main purposes are: to create a database script to store product information, to create classes that manage database access, to create classes that implement business rules (constraints).

The Django framework divides an application into several parts through the architecture of MTV layers (model, template, view), with weak coupling. The model layer consists of the part of the application that communicates with the database management system through object-relational mapping. The template layer consists of the presentation codes, to ensure visualization by the user. The view layer embeds business logic to control access to application data (Maciel, 2020).

To ensure the visualization of the application in the browser, the Django framework has a mapping to define

the project in the URL protocol. The `urls.py` file is the default location for defining the project in Django, and you must have the identifier of the resource to be mapped. Django Template Language presents facilities for inheritance of programming codes favoring the integration with the styles for the visualization (cascade) and the HTML markup language.

To ensure front-end development with favorable results with CSS-style language and favorable integration with JavaScript, Bootstrap is the recommended framework. This framework, open license, presents editable templates and styles, with satisfactory results for various environments.

Because of its wide diffusion since the early days of the internet, HTML is commonly used in the construction and structuring of pages for web applications. As a complement, CSS is a mechanism that adds style to web pages, and enables the implementation of visual resources that facilitate and harmonize the contents to those who access these pages. In favor of the streamline and other features of interaction of the page, it is also indicated the inclusion of JavaScript in the construction of the web interface. These three resources are technologies, considered by many as fundamental for applications.

As a consequence, the use of the Bootstrap framework becomes almost indispensable, since it aggregates and makes available these three technologies (HTML, CSS and JavaScript) for the structuring of web interfaces. Its use streamlines this phase of web application development and minimizes errors in the integration of these resources (Santiago et al., 2020).

Because the development of the software is usually carried out by teams, there are numerous changes in the source code, requiring a method to control the versions of the developed program. The version control system (CVS) is a tool that allows the group of developers to have access to the source files of the changes in the prototype of the application, forming a history of continuous development, such as GitHub (Maschietto et al., 2020).

According to Ghimire (2020 apud Santiago et al., 2020), web products are recurrently divided into two other by-products: back-end and front-end. While the back-end involves concerns about logic programming mechanisms, such as the technologies mentioned above in this topic, the front-end is basically limited to the aesthetics of the program and the ways in which content is displayed to users.

3. METHODOLOGIES

In the development of the system was specified the use of the Python programming language due to favorable integration with the back-end steps by the Django framework, open source, and JavaScript in Bootstrap framework, open source. The Python language has the largest developer community with disclosures in open

source and accessible libraries. Other languages such as Java and PHP will be more complex with larger character volumes and difficulties of developing the system to run on Windows. As the hardware available in the assistance entity is a desktop, it opted for the use of the Python language of easy integration, Django back-end framework due to direct relationship with the main language for running the system, and Bootstrap as a framework for front-end development by open source with editable templates and styles.

In relation to the development of the design of computational applications, the models are defined according to the objective to solve the problem and ensure the stages of the life cycle of the program. As we saw in Filatro & Cavalcanti (2017), Desing Thinking can be applied as an innovation approach, a problem-solving methodology or a teaching-learning strategy. Although it is used, initially, thinking only of one of these applications, it is common that throughout the process we can see more elements of this approach.

In the prototyping phase, the elaboration stage of the database, the diagramming of the tables was performed, in MySQL Workbench, as shown in Figure 1.

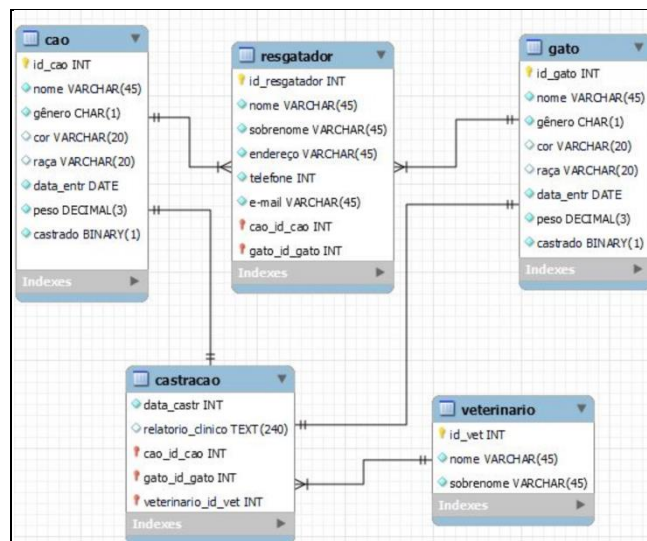


Figure 1. Entity-relational diagram

In the implementation stage, the programming codes were written in the PyCharm IDE for the execution of the software project. To link the database diagrammed in MySQL, the Django framework code, version 4.2, was imported.

Django is a framework that uses the MTV (model-template-view) software standard. Through of Django's object-relational mapping (ORM), the Animal table and its fields were created from the models.py file, where classes were written in Python code.

To ensure the modeling of the database fields it was necessary to create forms through the Django framework, whose codes were inserted in PyCharm, as shown in Figure 2.

```

From django import forms
from models import Animal

class AnimalModelForm(forms.ModelForm):
class Meta:
model=Animal
fields=['numreg', 'especie', 'genero', 'init_c',
'data_entr', 'porte', 'raca', 'cor', 'pelagem',
'faixa', 'imagem', 'obs']
widgets={
'especie':forms.RadioSelect(attrs={'class':
'form-check form-check-input',
'type':'radio'}),
'genero':forms.RadioSelect(attrs={'class':
'form-check form-check-input',
'type':'radio'}),

```

Figure 2. Code of file forms.py

In the front-end stage, by the model-view-control architecture, the codes were inserted in PyCharm from the CRUD function for URL creation, for later the software to be run in the browser.

With the use of the Bootstrap framework, for front-end, the HTML markup language was implemented to develop the visual presented to the user in the browser page.

4. RESULTS AND ANALYSIS

The development of an application for registration of veterinary procedures resulted in an integration of the architecture of the data model registered and registered, the visualization of the form for users and control of the data entered.

From the integration developed by the Django framework in back-end it was possible to implement the database with the insertion and registrations, in addition to the query.

While all these steps above have been done, templates have also been created, as they are the files that allow the immediate viewing of the results in the browser. This is because Django has a local server that can be accessed via terminal by the python command manage.py runserver.

To develop the visualization of the application was implemented the front-end language with Bootstrap

framework, whose JavaScript codes are inserted through tags.

The templates are responsible for organizing and arranging the way everything will appear to the user who accesses the application. As Bootstrap was installed in the Django project, it was possible to use it both as a markup tag and as an HTML language next to the structures of each page.

The figure 3 shows the codes inserted through the Bootstrap framework in the PyCharm language project.

```

{%bootstrap_messages%}
<divclass="grid">
<formaction="{% url 'cadastro_animal' %}"method="post"
class="form"autocomplete="off"
enctype="multipart/form-data">
{%csrf_token%}
{%bootstrap_formform%}
{%buttons%}
<buttontype="submit"class="btn btn-primary">
EnviarCadastro</button>
{%endbuttons%}
</form>
</div>

```

Figure 3. PyCharm codes for Bootstrap base

The storage of the registrations made by the interface of the web application can also be checked by accessing the local server dedicated to the MySQL database.

The JavaScript and CSS were loaded following Bootstrap's own pattern, with minor changes from the latter to adjust some point formatting in the written HTML. Thus, each page has been assembled and customized, when necessary, to show specific elements while maintaining structures in common between all pages, such as the bar navigation and the footer application.

The application for registration was executed in the browser as shown in figure 4.

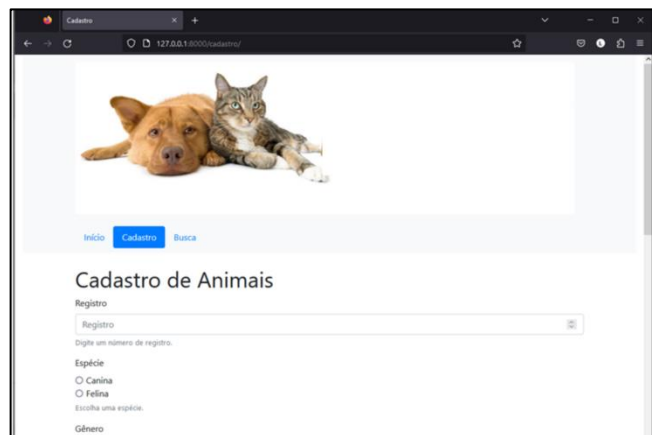


Figure 4. Layout with application for database forms

This application with data registration with the main proposal for registration still becomes introductory, since it has the need for development to update the data. Another important factor is the dependence on a browser installed on a desktop and preferably on an operating system. Given this, there is a need for advances for the development directed to mobile applications on mobiles due to more accessible use by the members of the assistance entity.

Due to the lack of resources for private providers, the possibility of hosting the application on the Heroku server was analyzed, with satisfactory results in developer communities.

5. CONCLUSIONS

The proposal aimed to solve the problem of absence of a system of registration of domestic animals and procedures for an entity with a lack of technological resources, this being a cadastral computerization with security for the data. Given this, the developers chose not to open to external users using mobile devices. The innovation of the proposal is present in the integration of open source languages and frameworks for a desktop system that depended on the old languages Visual Basic and Delphi.

The development of a computational technology to register veterinary procedures was one of the solutions applied through the Python language, Django and Bootstrap frameworks, by MySQL as a management system. In this sense, it was satisfactory to develop with popular languages to implement on a free access server, such as Heroku.

Despite the introductory phase of the project, which resulted in the implementation for running in desktop operating system browser, the system was satisfactory in terms of security and visualization.

The application developed through the MVC architecture needs an adaptation to be executable on a mobile device,

with the possibility of accessing data and recording information, since many veterinary procedures are remote and the information noted may be lost.

The cadastral computerization is an important topic in the face of demands for technology and access to updated data.

REFERENCES

- [1] Deitel, P. J. 2008. *Ajax, Rich Internet Applications e desenvolvimento web para programadores*. São Paulo: Pearson Prentice Hall.
- [2] Filatro, A. C., Cavalcanti, C. C. 2017. *Design thinking na educação presencial, a distância e corporativa*. Saraiva.
- [3] Maciel, F. M. B. 2020. *Python e Django: desenvolvimento web moderno e fácil*. Rio de Janeiro: Alta Books.
- [4] Maschietto, L. G., Rodrigues, T. N., Bianco, C. M. D., Moraes, D. M. P., Vettorazzo, A. S., Andrade, A. L. 2020. *Processos de desenvolvimento de software*. Porto Alegre: SAGAH.
- [5] Perkovic, L. 2016. *Introdução à computação usando Python: um foco no desenvolvimento de aplicações*. Rio de Janeiro: LTC.
- [6] Pichetti, R. F. V., Cortes, E. S., Paixão, V. S. M. 2020. *Banco de dados*. Porto Alegre: SAGAH
- [7] Santiago, C. P., Veras, N. L., Aragão, A. P., Carvalho, D. A., Amaral, L. A. Desenvolvimento de sistemas web orientado a reuso com Python, Django e Bootstrap. In: Teles, A. S., Calçada, C. D., Veras, N. L. 2020. *Minicursos da Escola Regional de Computação do Ceará, Maranhão e Piauí*. Porto Alegre: Sociedade Brasileira de Computação, p. 97-120