A Microservice-Based Health Information System for Student-Run Clinics

Itamir de Morais Barroca Filho[1(✉)], Silvio Costa Sampaio[1], Gibeon Soares Aquino Junior[2], Rafael Fernandes de Queiroz[2], Dannylo Johnathan Bernardino Egidio[2], Ramon Santos Malaquias[1], and Larissa Gilliane Melo de Moura[1]

[1] Digtal Metropolis Institute, Federal University of Rio Grande do Norte, Natal, Brazil
{itamir.filho,silviocs,ramonstmalaquias,larissa.moura}@imd.ufrn.br
[2] Department of Informatics and Applied Mathematics, Federal University of Rio Grande do Norte, Natal, Brazil
gibeon@dimap.ufrn.br, rfqueiroz.91@gmail.com, dannylojohnathan@gmail.com

Abstract. Student-run clinics are units found in universities and colleges that provide health care to the internal and external public, integrating theoretical knowledge with situations involving treatment provided to real patients, thus reproducing the environment that the student will find in his or her professional life. In this context, this article presents an information system for the management of health services and electronic medical records in student-run clinics called SigSaude. SigSaude is a solution whose architecture is based on microservices, contemplating specialties such as dentistry, psychology, nutrition, and medicine. This system is being used by the student-run clinics of the Federal University of Rio Grande do Norte (UFRN), bringing benefits to the improvement of the quality of patient care and helping teachers and students in the evidence-based teaching-learning process. This system is being used by these clinics and the current results demonstrate the solution's high scalability.

Keywords: Student-run clinic · Microservice · API · Electronic medical records

1 Introduction

In the academic context of health, student-run clinics arise from the need to integrate teaching, research and extension activities with health practices. The activities in student-run clinics are usually integrated with the curriculum guidelines of the health professional education in which, in addition to the theoretical disciplines, a set of practical internship components is required. The primary objective of this requirement is to integrate the concepts learned in the classroom and the reality that students will find after they graduate. In this sense,

S. Misra et al. (Eds.): ICCSA 2019, LNCS 11623, pp. 3–16, 2019.
https://doi.org/10.1007/978-3-030-24308-1_1

activities in student-run clinics are assuredly articulated with the reality of the population served, thus reproducing the private practice and training professionals to be ready for this type of performance.

Therefore, student-run clinics consist of an environment associated to an educational institution, in which students perform clinical practices while being taught, under the guidance of a professional who assumes the role of teacher and/or supervisor [1,3].

Another essential feature of student-run clinics is their social role since they serve an economically or geographically disadvantaged population. Usually sheltered in the university environment, student-run clinics provide free treatments to the internal and external community, receiving patients sent from other student-run clinics and health units both in the municipality and macroregion. In fact, in many Brazilian towns, the assistance provided by the student-run clinics guarantee access to health services, which is insufficient and even unavailable in the public network, to a representative number of people.

The activities carried out in the context of a student-run clinic involve different types and natures of information, which need to include all the structures of health care. The maintenance and access to this information, in its turn, takes on even greater importance considering the complicated routine made up of activities that vary according to the academic context. However, there is not much reference in the literature related to health information management systems geared towards this specific student-run clinic environment, that is, of systems capable of establishing a dialogue between the health care practical activities and the curricular activities of the health university courses.

This need is urgent, since, as pointed out by Oliveira [15], we are undergoing social and political changes that affect both the academy and the health services, which has a direct impact on the needs and demands of this area. When it comes to services, there are problems related to the units' physical structure [5]; to the lack of professionalization of the health management; to the physical structure of the communication networks; to the low connectivity in some locations; to the use of diverse information systems; and to the existence of deficient or non-integrated data and records, which, most of the time, are stored in non-standardized paper files, making it difficult to organize and share them with the other individuals involved [8,10].

Seeking to collaborate with the knowledge in this area, this work describes a health information system - SigSaude - based on a microservices architecture, integrated with the administrative and academic management of the university's student-run clinics. The SigSaude aims to incorporate a complex set of services linked to employees with different functions and objectives, without losing its academic vocation.

The university studied was the Federal University of Rio Grande do Norte (UFRN), which offers about 14 health services that are provided through student-run clinics, contributing to the academic training of undergraduate and graduate students. Different types of services/specialties such as dentistry, nutrition, physiotherapy, nursing, psychology, and medicine are provided, which, altogether, totals about 6,000 monthly visits, according to the data supplied by the clinics themselves.

UFRN's student-run clinics, for the most part, do not use a computation system for managing their administrative data, and there is no automated control of the data related to patients and their care. In addition to resulting in a large amount of paper used, this causes a considerable difficulty to maintain and access information. Considering that many of these clinics are distant from each other, the geographic factor is also aggravating their current work model, making it difficult to perform medical referrals and to exchange information between services.

The difficulties described above and the circumstances in which student-run clinics are inserted point out the indispensability of the unification, standardization and easy availability of the patients' data for the services and employees involved in this process. Taking all of this into consideration, the SigSaude proposes the use of the concepts of distributed Electronic Medical Record (EMR) and microservices. The first one aims to aggregate the data that contributes to studies and decision-making [14], while the second one allows to address, in an elegantly and scalable way, the distribution of the EMR data. This article focuses on the concept of microservices architecture.

The ultimate goal of the system described in this work is to make it possible to operate student-run clinics more efficiently, resulting in better training of the students involved in the activities and better treatment for patients.

The remainder of this paper is organized as follows: Sect. 2 presents the microservices architectural model; in Sect. 3, the architecture of the proposed solution is detailed; in Sect. 4, some of the results already obtained in the development of the SigSaude are presented; Sect. 5 brings together some related works; and, finally, Sect. 6 presents the conclusions and directions for future works.

2 Microservices

Microservices are a trend in software architecture that emphasizes the design and development of highly sustainable and scalable software [7]. Dragoni et al. provide a clear distinction between microservices and microservices architecture. Microservices are a cohesive and independent process that interacts through messages, while a microservices architecture is a distributed application in which all of its modules are microservices.

Each microservice is built for a business capacity, runs its processes, and can communicate with other microservices through simple mechanisms such as services APIs [18]. This model aims to achieve an increase in implementation capabilities directly impacting the time and independence of implementation, modifiability, and quality of services [6,16]. Pahl and Jamshidi [16] add that the microservices model favors the decentralization of control and provides greater support to heterogeneity.

Each microservice must exhibit a communication interface so that other microservices in the application ecosystem can exchange data with it. These interfaces often use custom protocols and remote function calls (RFCs) [22]. Gateway APIs are used as intermediaries or points of entry of communication between the microservices [17]. These are usually connected to the internet and use protocols such as HTTP, open standards such as REST or SOAP, and interoperable data exchange technologies, such as XML and JSON [22].

In addition to enabling communication between microservices, the Gateway API also allows the existence of mechanisms for the registration, identification, and discovery of microservices and the control of clients request routes [13]. It is also possible, as an option, to use mechanisms for the management of the identity and data privacy through authentication services such as OAuth [13].

Some trade-offs should also be considered when considering adopting a microservice based architectural model. Among the most important one, Fowler [9] points out that although the microservices' modular structure facilitates the work of large teams, common remote calls in microservices are usually slow and have a considerable risk of failure. Although the independent implementation is a strong point for this architectural model, consistency maintenances in distributed systems are often difficult, which leads to considering that everyone should manage possible consistencies. Finally, the author adds that although microservices provide the freedom to use multiple technologies, frameworks, and data storage resources in their applications, it is essential to have the operational maturity to deal with this universe of various technologies.

The decision to adopt a microservices architecture for the SigSaude was taken given the need to develop an information system with built-in features that could work for many different specialties, and that contemplated the increase in implementation capacity (time and independence), modifiability and quality of services. As a result, the decentralization of control and the greater support for heterogeneity is expected, since different student-run clinics have different demands.

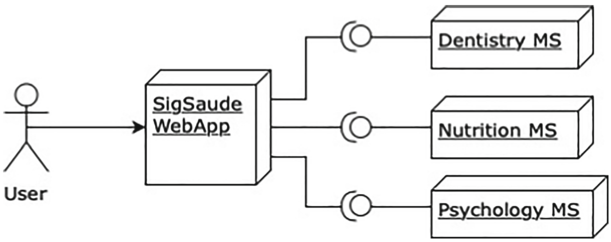

Fig. 1. SigSaude's architecture overview.

3 Solution's Architecture

The architecture of the proposed solution follows the principles of the microservices architecture model. As already mentioned, each student-run clinic has its specific set of services offered. Taking this into consideration, the proposed architecture will allow, for example, a psychology student-run clinic to use only the clinic module and other general management modules, without having to bear the costs of maintaining modules related to other services that are not even used and provided. In another example, the architecture will make it possible to manage instances of each microservice in a flexible way, thus allowing to meet the requirements of a more demanding service.

The structure of the proposed architecture is presented in Fig. 1. In this architecture, the services offered by the student-run clinics are represented by blocks (for example Dentistry MS, Nutrition MS, and Psychology MS, where MS indicates that it is a microservice), where each block represents a self-contained and independent microservice, thus fitting with the assertion [12] that each service must have a single purpose and be responsible for only one of the application's logical context. It is important to clarify that because of space issues, Fig. 1 represents a narrow view of the proposed architecture. Thus, the complete architecture integrates a much higher number of blocks or microservices.

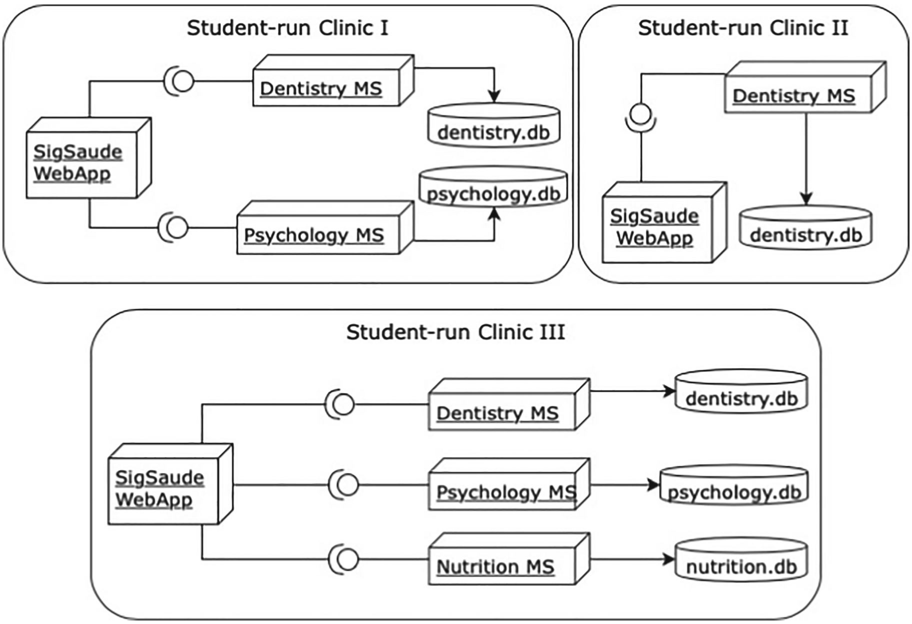

Fig. 2. Implementation architecture in student-run clinics.

The architecture's microservices are organized according to the services offered by each student-run clinic. For example, Dentistry MS represents the microservice of the architecture responsible for the logical processes related to Dentistry services, such as the removal of a patient's teeth. Similarly, the Nutrition MS and the Psychology MS are responsible for the specific rules of each specialty, for example, calculating a patient's body mass index or holding psychological counseling meetings.

In a microservices architecture, it is essential that the components communicate with each other using simple mechanisms [12]. Thus, in Fig. 1, these mechanisms are symbolized by the interfaces linked to the microservices, representing the APIs (Application Programming Interfaces) that provide communication to the microservices to which they are interconnected.

The requests made to the microservices APIs are carried out through the use of another component of the architecture, which is called SigSaude WebApp. For the architecture, this component is a web application responsible for building the interface presented to the user, as well as for intermediating requests to the other microservices. This way, the SigSaude WebApp works as a Gateway API, abstracting all the APIs from the other microservices, receiving requests and designating them to their proper destination. Thus, the SigSaude WebApp assumes the role of orchestrator, controlling the destination and traffic of requests to avoid the congestion of requests sent to the target microservices. Since it is in front of the microservices, the SigSaude WebApp performs authorization/authentication functions related to the requests originating from the users. However, future versions will have to outsource access control.

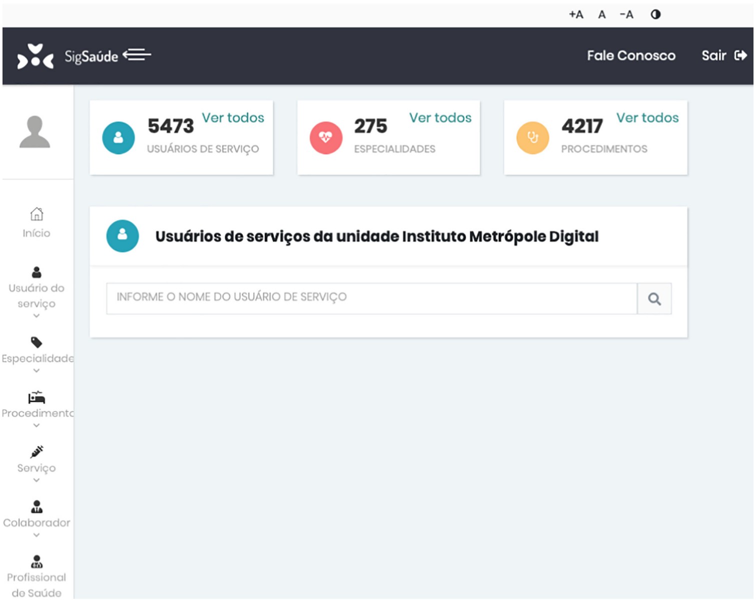

Fig. 3. SigSaude's dashboard.

Student-run clinics usually have several infrastructure problems, which make the deployment of large-scale systems challenging. Some student-run clinics are located in remote locations, where there is no internet with good bandwidth rates, making it difficult to carry out large traffic of data over the network. Another situation found is that some clinics do not have a computer infrastructure like servers with high processing power, thus making it impossible to implement a monolithic system, which requires a lot of storage and processing capacity.

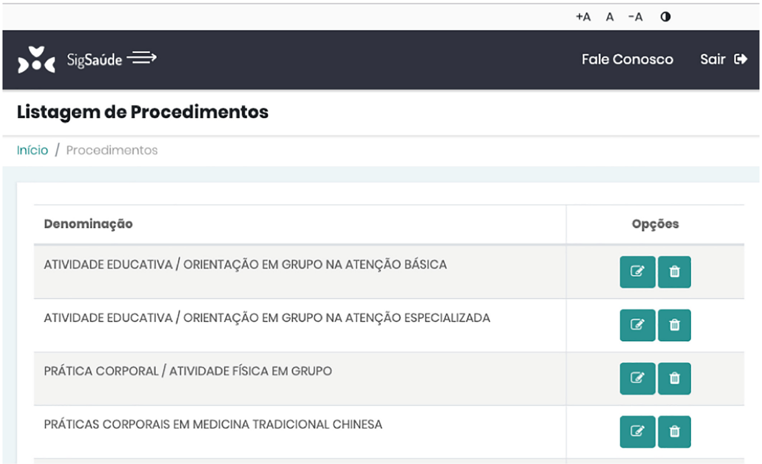

Fig. 4. List of procedures registered in SigSaude.

Again, it should be noted that the use of microservices based on the services offered by student-run clinics will allow the implementation of this health platform with only the microservices needed to attend the specific needs of a student-run clinic. This way, the resources necessary to implement the platform are reduced, since small microservices do not require servers with large storage and processing capacities, nor internet connections with high bandwidth rates, as the information that will travel will be only those that are inherent to the microservice related to the student-run clinic. This reduction in the implementation costs is an important differential of this proposal since most of the student-run clinics offer free care and do not have numerous financial resources.

In the different contexts of student-run clinics, the implementation of the health platform proposed is generally carried out as shown in Fig. 2. In this Figure, each group of microservices represents a context for the student-run clinic, with its own platform implementation, containing its own microservices and database. The SigSaude WebApp component is required in all of the implementation environments, allowing to configure the interface presented to the user according to the environment in which it is implanted.

The group represented by Student-run Clinic II, for example, portrays a simpler scenario for a student-run clinic, which offers only a health service

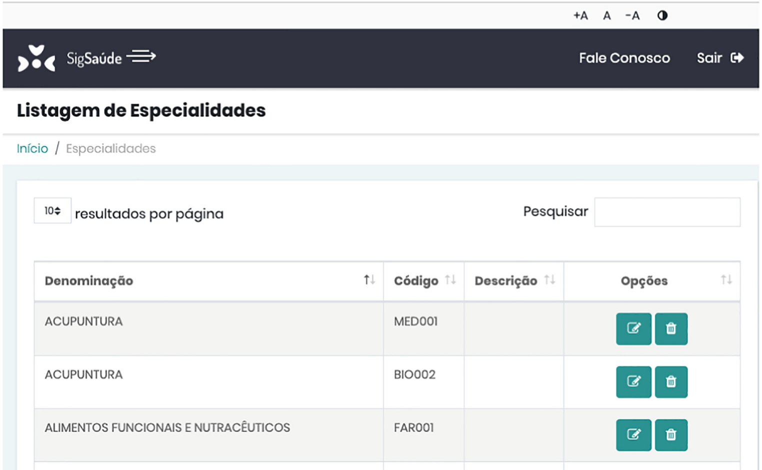

Fig. 5. List of specialties registered in SigSaude.

(dentistry). This way, only one microservice is made available for it, which performs the logic processing needed for the student-run clinic.

In its turn, the groups presented in Student-run Clinic I and Student-run Clinic III have a larger number of microservices available (dentistry, psychology and nutrition). In this case, these groups represent the contexts of student-run clinics that have greater infrastructure capabilities when compared to the Student-run Clinic II group, considering the number of health services offered by them.

For these student-run clinics scenarios, an architecture based on the monolithic model would be impracticable, since in each context of these student-run clinics, it would be necessary to implement a large system involving all of the services available, including those that are not needed by the clinic because it does not provide the corresponding health service. Each implementation environment will have an instance of the deployment of the architectural structure shown in Fig. 1.

4 Results

The SigSaude was implemented according to the architecture mentioned in the previous section and is already available in many of UFRN's student-run clinics, who are using it every day, feeding the platform with patient registration data, procedures, specialties, services and scheduling control. Up until the moment this article was written, the platform had 5473 patients registered, 275 specialties and 4217 procedures, as shown in the dashboard found in Fig. 3.

Table 1. Metrics assessed in the SigSaude WebApp's code.

Metric	Total	Mean	Std. Dev	Maximum
McCabe Cyclomatic Complexity (avg/max per...)	–	1.365	1.335	19
Number of Parameters (avg/max per method)	–	0.652	0.85	7
Nested Block Depth (avg/max per method)	–	1.106	0.573	4
Afferent Coupling (avg/max per packageFragment)	–	14.1	25.282	111
Efferent Coupling (avg/max per packageFragment)	–	8.55	9.351	30
Instability (avg/max per packageFragment)	–	0.625	0.354	1
Abstractness (avg/max per packageFragment)	–	0.124	0.24	0.9
Normalized Distance (avg/max per packageFragment)	–	0.344	0.318	1
Depth of Inheritance Tree (avg/max per type)	–	1.581	0.943	3
Weighted Methods per Class (avg/max per type)	1729	6.972	9.715	79
Number of Children (avg/max per type)	74	0.298	1.301	9
Number of Overridden Methods (avg/max per...)	22	0.089	0.381	3
Lack of Cohesion of Methods (avg/max per type)	–	0.255	0.382	1.6
Number of Attributes (avg/max per type)	577	2.327	3.452	34
Number of Static Attributes (avg/max per type)	18	0.073	0.478	6
Number of Methods (avg/max per type)	1185	4.778	7.408	74
Number of Static Methods (avg/max per type)	82	0.331	1.623	18
Specialization Index (avg/max per type)	–	0.034	0.177	2
Number of Classes (avg/max per packageFragment)	248	12.4	13.89	57
Number of Interfaces (avg/max per packageFragment)	32	1.6	5.86	27
Number of Packages	20	–	–	–
Total Lines of Code	10415	–	–	–
Method Lines of Code (avg/max per method)	3.809	3.006	4.909	52
Method Lines of Code (avg/max per method)	3.809	3.006	4.909	52

To validate the proposal, the values for the metrics illustrated in Table 1 were collected from the SigSaude WebApp source code and allow to evaluate important characteristics, such as the average of methods in each class, lines of code, number of attributes, method cohesion, number of parameters, etc. Since it is a WebApp, which behaves as an integrating resource for the associated microservices (Gateway API), its numbers are very expressive. Depending on their complexity and purpose, the other microservices may have reduced numbers.

As already reported, each clinic includes the microservices related to them. Figures 4 and 5, respectively, represent some of the procedures and specialties already registered by the users of the UFRN's student-run clinics.

Due to the need for the health systems to have reduced response times, tests were implemented using the J-Meter tool, which makes it possible to easily customize aspects such as workload and duration of testing [4,19].

Thus, requests stress situations were simulated to evaluate SigSaude's scalability. The tests were performed in two moments: the first one with 100 threads, and the second one with 1,000 threads. Each thread, in its turn, executed dozens of requests, totaling 1,700 requests in the first test and 17,000 requests in the second one. Of the requests made, 16 (0.94%) resulted in errors in the first test; for the second test, the number of errors totaled 5,674 (33,38%) because the random characteristic of the generation of requests ended up violating SigSaude's validation rules.

The first test case, which had 1,700 requests, resulted in response times within the acceptable standard - 684 requests were answered in less than 1.5 s, 304 requests were responded between 1.5 and 3 s, and 696 were answered in over 3 s, as shown in Fig. 6.

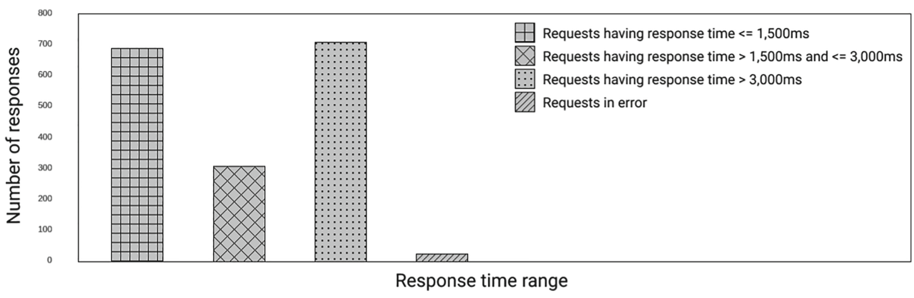

Fig. 6. Overview of the response times for 1,700 requests.

In the second test case, as shown in Fig. 7, 17,000 requests were sent, of which 792 had a response time of less than 1.5 s, 954 took between 1.5 and 3 s, and 9,580 had a response time of more than 3 s.

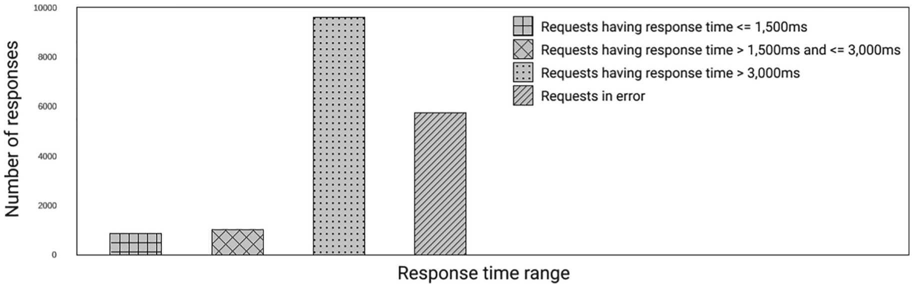

Fig. 7. Overview of the response times for 17,000 requests.

The test results give good indications that the implementation of the microservice based architectural model for the SigSaude performs well, meeting the scalability requirements initially imposed. However, more complex tests should be carried out in the future to confirm these results.

5 Related Works

Although the literature on health management systems is extensive, up until this article was written, it was not possible to find works focused on the specific context of student-run clinics. The papers found in this area focus on proposing hospital management systems, which present dimensions and characteristics that are often incompatible with the environment discussed in this article. Thus, in this section, we present some works that show the need for management systems in student-run clinics, as well as papers that corroborate our choice of using a microservices architecture.

Ali et al. [2] propose a platform based on a microservices model and a WoO (Web of Objects) approach to aid depressive disorders. The WoO approach provides a framework based on IoT services for the Web, virtualizing the objects involved in the application and interconnecting them with their respective physical resources in the infrastructure. The author uses the microservices architecture to reduce the complexity of the project through modular and independent microservices. Thus, each microservice has a specific function in the application, from training models using Machine Learning techniques for the recognition of specific situations of potential users, to services that analyze the user's data searching for characteristics that make it possible to identify symptoms of a depressive disorder.

The tendency of the health community to turn to a complex set of services that incorporate many people with different capabilities and objectives is evidenced in [11]. The work proposes a multi-agent system for health care based on a service-oriented architecture (SOA), which is particularly based on microservices. The microservices of the proposal are based on IoT services. An architecture based on microservices was used to meet the application's scalability requirements, reducing or avoiding overheads in the application messages. The system is equipped with microsensors that capture vital data from patients; this data is transferred to the system that will analyze it and that can make specific decisions, such as notify someone responsible for the patient of some alarming data.

The use of microservices in health has enabled a significant increase in the applications' productivity, since it provides high availability, thus preventing the services from being unavailable, as found in Williams et al. [20]'s work. Also, the segregation of services ends up generating individual security for each module - a critical requirement for hospital systems [20]. Yilong et al. [21] also present a proposal based on microservices to share medical data preserving the patients' privacy. In this work, the author emphasizes that the exchange of medical information is not trivial due to potential privacy intrusions. Thus, the work approaches de-identification techniques, represented as "black boxes," through the use of a microservices architecture, and guarantees the legitimacy of the user through rule-based control access (RBAC).

6 Conclusions and Future Work

This paper discusses the need for an integrated system for the management of health services for student-run clinics and presents, as a solution, the SigSaude. The SigSaude is an initiative from the Federal University of Rio Grande do Norte, which offers many health services such as student-run clinics. Although in it still in its first year, the solution already presents significant results concerning the integration of patient information, schedules, treatments, and patient follow-ups.

The main characteristic of the SigSaude discussed in this article is its design, which is totally microservices-oriented, mainly due to the limitations of the infrastructure for the implementation of large-scale platforms in UFRN's student-run clinics. This way, the clinics only use services that are needed for its context, thus avoiding the waste of resources and increasing the solution's scalability. It is worth noting that despite the integration, each clinic has a different context of use and a certain individuality regarding the services available, so the solution using microservices allows the preservation of the particularities of each clinic and its area of work in the health field.

This work also presents the results of the scalability tests that were performed on the solution. These stress tests were developed and applied to 100 and 1,000 threads simultaneously and the results were considered satisfactory. Taking into consideration the student-run clinics' high demand and the need for availability that the platform needs to meet, the test results suggest that the developed tool can be used by clinics. However, it should be noted that more complex tests should be carried out during the development of the solution.

As future work, there is an intention to apply functionalities for monitoring the patients' clinical evolution, as well as increase the platform's features, including functions like patient care and laboratory and health procedures follow-up. Another important point to be considered in the next phase of the SigSaude is the consolidation of the distributed Electronic Medical Record (EMR). This implies exploring the communication between microservices to draw up the patient's entire history. The EMR will make it possible to reduce bureaucratic processes and customize the patient's care. Other expected functionality is the inclusion of microservices related to research and analysis of health data from the SigSaude's distributed bases.

Acknowledgements. This research was partially supported by the SigSaude Project, funded by the Federal University of Rio Grande do Norte - UFRN. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

1. Albuquerque, V., Gomes, A., Rezende, C.H., Sampaio, M., Dias, O., Lugarinho, R.: A integração ensino-serviço no contexto dos processos de mudançãa na formação superior dos profissionais da saúde. REVISTA BRASILEIRA DE EDUCAÇÃO MÉDICA **32**(3), 356–362 (2008)
2. Ali, S., Kibria, M.G., Jarwar, M.A., Kumar, S., Chong, I.: Microservices model in WoO based IoT platform for depressive disorder assistance. In: 2017 International Conference on Information and Communication Technology Convergence (ICTC), pp. 864–866. IEEE (2017)
3. Amaral, A.E., Luca, L., Rodrigues, T., Leite, C., Lopes, F., Silva, M.: Serviços de psicologia em clínicas-escola: revisão de literatura. Bol. psicol. **62**136 (2012)
4. de Camargo, A., Salvadori, I., Mello, R.d.S., Siqueira, F.: An architecture to automate performance tests on microservices. In: Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services, pp. 422–429. ACM (2016)
5. Cavalheiro, M.T., Guimarães, A.: Formação para o sus e os desafios da integração ensino serviço. Caderno FNEPAS. Dezembro 2011, vol. 1 (2011)
6. Chen, L.: Microservices: architecting for continuous delivery and DevOps. In: IEEE International Conference on Software Architecture (ICSA) (2018)
7. Dragoni, N., et al.: Microservices: yesterday, today, and tomorrow. Present and Ulterior Software Engineering, pp. 195–216. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67425-4_12
8. Junior, E., Onofre de Lira, A.C.: Prontuário eletrônico e as perspectivas do e-health. Debates GVsaúde, Debates GVsaúde- Segundo Semestre de 2009 - Número 8 (2009)
9. Fowler, M.: Microservice trade-offs. Dosegljivo (2015). http://martinfowler.com/articles/microservice-trade-offs.html
10. Gonçalves, J., Batista, L., Carvalho, L., Oliveira, M., Moreira, K., Leite, M.: Prontuário eletrônico: uma ferramenta que pode contribuir para a integração das redes de atenção à saúde. Saúde em Debate, Rio de Janeiro, vol. 37, no. 96, pp. 43–50 (2013)
11. Hill, R., Shadija, D., Rezai, M.: Enabling community health care with microservices. In: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), pp. 1444–1450. IEEE (2017)
12. Lewis, J., Fowler, M.: Microservices: a definition of this new architectural term. Mars (2014)

13. Lu, D., Huang, D., Walenstein, A., Medhi, D.: A secure microservice framework for IoT. In: 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE), pp. 9–18. IEEE (2017)
14. Martins, C., Lima, S.M.: Vantagens e desvantagens do prontuário eletrônico para instituição de saúde. RAS, Abr-Jun 2014, vol. 16, no. 63 (2014)
15. Oliveira, M.: O papel dos profissionais de saude na formacao academica. Olho Magico **10**(2), 37–39 (2003)
16. Pahl, C., Jamshidi, P.: Microservices: a systematic mapping study. In: CLOSER (1), pp. 137–146 (2016)
17. Richardson, C.: Pattern: API gateway/backend for front-end. Google Scholar (2017)
18. Soldani, J., Tamburri, D.A., Van Den Heuvel, W.J.: The pains and gains of microservices: a systematic grey literature review. J. Syst. Softw. **146**, 215–232 (2018)
19. Ueda, T., Nakaike, T., Ohara, M.: Workload characterization for microservices. In: 2016 IEEE International Symposium on Workload Characterization (IISWC), pp. 1–10. IEEE (2016)
20. Williams, C.L., Sica, J.C., Killen, R.T., Balis, U.G.: The growing need for microservices in bioinformatics. J. Pathol. Inf. **7** (2016)
21. Yang, Y., Zu, Q., Liu, P., Ouyang, D., Li, X.: Microshare: privacy-preserved medical resource sharing through microservice architecture. arXiv preprint arXiv:1806.02134 (2018)
22. Yu, Y., Silveira, H., Sundaram, M.: A microservice based reference architecture model in the context of enterprise architecture. In: 2016 IEEE Conference on Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 1856–1860. IEEE (2016)