FallReportAPI: Integration of Digital Health Systems for Fall Detection in Hospital Systems

Nadiana Kelly N. Mendes[1,*], Francisco Victor S. Pinheiro[1], Victória T. Oliveira[1], Francisco Laurindo C. Junior[1], Rafael L. Gomes[2] and Rossana M. C. Andrade[1]

[1]*Federal University of Ceará, Fortaleza, Brazil*
[2]*State University of Ceará, Fortaleza, Brazil*

Abstract

The Internet of Things (IoT) enables devices with processing and communication capabilities to connect to the internet, creating numerous opportunities in both academic and industrial sectors. In healthcare, the adoption of this technology has grown significantly, with applications ranging from emergency services to long-term interventions, driving the development of new systems. One example is Health Risk, a system that identifies falls in patients using wearable devices and machine learning techniques, achieving an accuracy rate of 94.4%. This work focuses on developing an API to integrate two health systems: Health Risk, a mobile solution based on IoT for fall detection, and Smart Hospital, a hospital management system developed for a university hospital. To facilitate this integration, a literature review on the Internet of Things was conducted, aiming to deepen the understanding of development standards and explore successful examples of IoT-based healthcare systems. The developed API, named FallReportAPI, was implemented in Java, using Spring Boot to create the routes, RabbitMQ for message exchange, and PostgreSQL as the database. The integration between the systems was tested through functional API tests and performance tests, ensuring proper operation. With this API, communication between the systems is immediate, allowing notifications of falls detected by Health Risk to be efficiently transmitted to Smart Hospital. This integration aims to reduce response times for medical care in fall situations that require urgent intervention.

Keywords

Systems Integration, Internet of Health Things, Application Programming Interface

1. Introduction

The Internet of Things (IoT) enables objects with computational and communication capabilities to connect to other devices via the internet, as long as they share these same capabilities. This connectivity between common objects opens up a range of possibilities and opportunities, both in academia and industry [1]. With the constant evolution of technology in society, it becomes evident how it can be applied to solve social problems.

One of the main sectors that can benefit from this technology is healthcare. The growing application of the Internet of Things in healthcare has the potential to support patients in various situations, from emergency treatment in hospitals to long-term interventions. In this context, IoT in healthcare stands out in the technological landscape, promoting the development of new systems that contribute to advances in the field [2].

Within this framework, the Group of Computer Networks, Software Engineering, and Systems (GREat) developed the Health Risk fall detection system, inspired by IoT. This system can play a vital role in healthcare, especially in monitoring patients and the elderly who require constant attention. The system assists in providing help in cases of falls, using data from wearable devices as input for an IoT system that applies Machine Learning models supported by cloud computing. The model used

ICAIW 2024: Workshops at the 7th International Conference on Applied Informatics 2024, October 24–26, 2024, Viña del Mar, Chile
*Corresponding author.
nadianakelly@alu.ufc.br (N. K. N. Mendes); victor.pinheiro.ce@alu.ufc.br (F. V. S. Pinheiro); victoriat.oliveira@alu.ufc.br (V. T. Oliveira); laurindocostajr@alu.ufc.br (F. L. C. Junior); rafaellgom@gmail.com (R. L. Gomes); rossana@ufc.br (R. M. C. Andrade)
0009-0000-9814-4624 (N. K. N. Mendes); 0000-0002-5693-0499 (F. V. S. Pinheiro); 0000-0002-1400-522X (V. T. Oliveira); 0000-0001-6924-0668 (F. L. C. Junior); 0000-0001-7922-0695 (R. L. Gomes); 0000-0002-0186-2994 (R. M. C. Andrade)

by Health Risk achieves an accuracy of 94.4%, correctly distinguishing between a fall and events that do not represent falls [3]. To use the application, an Android device and a compatible smartwatch are required.

Another project by GREat is the Smart Hospital, a system developed to serve the hospital units of Ceará. This system was designed to solve various problems faced by the state's healthcare units, standing out as a technological innovation among other existing hospital software. The project is the result of a partnership between the State University of Ceará and industry, represented by HUAWEI [4]. Aiming to improve the integration between the fall detection system and other technologies, the idea of creating an API (Application Programming Interface) emerged. An API is a means of communication between systems that facilitates integration, allowing one system to provide information and services to another without needing to know the underlying software implementation details [5]. The API developed in this work integrates the Health Risk fall detection system with the Smart Hospital system, contributing to improved hospital care.

The objective of this work is the development of the FallReportAPI, which integrates the fall detection system, Health Risk, with the hospital system Smart Hospital, aiming to contribute to the hospital aspect of healthcare. It is expected to positively impact the reduction of response times in healthcare facilities using Smart Hospital during fall incidents, with a primary focus on the elderly population.

The structure of this article is organized as follows: In addition to this introductory section, Section 2 presents the Theoretical Foundation, providing the fundamental concepts necessary for understanding the work. Section 3 covers related works. Section 4 describes the Development of the FallReportAPI, detailing the process of creation and implementation of the proposed API. In Section 5, the Discussion of Results is presented, where the tests conducted and the main results are discussed. Finally, Section 6 presents the Conclusion, focusing on the main contributions of the work and directions for future research.

2. Theoretical background

Weiser, in his article "The Computer for the 21st Century", explores the future of technologies, anticipating a scenario that closely resembles what we know today as the Internet of Things. He suggested that devices could connect in an ubiquitous and seamless manner, becoming practically invisible to the user, which would allow everyday tasks to be carried out without the need to worry about the installation, configuration, or maintenance of computational resources [6].

The Internet of Things (IoT) can be seen as an extension of the current Internet, enabling everyday objects equipped with computing and communication capabilities to connect to the global network. This connectivity allows for the remote control of these devices and their use as service providers. The introduction of these new functionalities into the objects we use daily opens up a range of opportunities for both academic research and industrial applications [1].

Currently, IoT applications range from telemetry, which involves data collection in various configurations, to direct interaction with objects and communication between them. This interaction can occur both between the objects themselves and between objects and people, either intentionally or discreetly [7]. The literature highlights the wide range of IoT applications in healthcare, bringing innovations that simplify procedures, save time and resources, customize treatments, and provide essential data for disease prevention and management [8].

The Internet of Medical Things includes all IoT-based healthcare solutions, such as connected hospital systems, vital signs monitoring devices (such as blood oxygen levels and pulse), and systems that use sensor data to improve patients' quality of life. The combination of rapid technological advancement with the development of sensors and actuators offers promising opportunities for the creation of intelligent systems in healthcare [9] [10].

With the evolution of intelligent systems and advanced algorithms, it is possible to collect and analyze large volumes of critical data in real-time, which has the potential to revolutionize research, management, and intensive care. The integration of medical devices with IoT aims to improve the quality and efficiency of healthcare services, especially for the elderly, patients with chronic diseases, and those who require continuous monitoring [11].

The aging population is an unprecedented global challenge. In 2019, about 703 million people worldwide were 65 years or older, representing 9% of the global population. By 2050, this number is expected to double, reaching 1.5 billion, which will correspond to approximately 16% of the world's population [12] [13].

Data revealed that elderly fall victims have a higher likelihood of being classified with colors that indicate a greater degree of priority, such as orange and red, which mean very urgent. Falls, even those from standing height, can result in serious and potentially fatal injuries [14].

There are already projects focused on elderly health, such as a technological platform that provides support to elderly individuals, home caregivers, and healthcare professionals [15]. Other examples include an application developed to help healthcare professionals manage the care of elderly individuals with special needs [16], and a taxonomy created to support the development of software focused on elderly health, organizing the essential characteristics and information for these applications [17].

Health Risk began as one of the projects of the Computer Networks, Software Engineering, and Systems Group (GREat) [1]. The application is a mobile system for fall detection, running on both a smartwatch and a smartphone. Health Risk was developed in response to the increasing need to monitor elderly individuals. It was created to intelligently monitor those requiring special care, such as elderly people who often spend most of their time alone and need special attention. The system monitors the user's movement in real-time, detecting falls through the analysis of sensor data and issuing help alerts when necessary. The application was developed for the Android platform and utilizes the following sensors: accelerometer and gyroscope. When a fall is detected and there is no response or confirmation of well-being, the smartwatch connects to the smartphone, which then sends a fall alert to a caregiver contact, selected by the user and previously registered. To identify potential falls, data collected from wearable devices are used as input for an IoT system that employs Machine Learning models. In the case of Health Risk, cloud computing is also utilized. The algorithm used in Health Risk for fall detection has an accuracy of 94.4%, with a low false negative rate of 4.3% [3]. Images of the Health Risk application can be seen in Figures 1 and 2.

3. Related Work

3.1. A framework for the elderly first aid system by integrating vision-based fall detection and BIM-based indoor rescue routing

This study presents an integrated framework for a first aid system designed for elderly individuals, capable of automatically detecting falls in indoor environments, monitoring the real-time status of the elderly, and formulating the rescue route with the support of the BIM environment.

For fall detection, the proposed method involves identifying personal objects in each video frame, followed by tracking these objects over time. Then, skeleton detection is performed to recognize the body structure of each individual. Finally, a convolutional neural network (CNN) is employed to determine whether the elderly person is falling or not.

Upon detecting a fall, the corresponding internal rescue route is generated, considering the principle of the shortest path and the building's attributes. Decision-making for internal rescue routing is carried out within the BIM environment, using the Revit API to facilitate access to information and inferences.

The experimental results showed high accuracy in detecting falls, reaching a precision of 94.1%. Furthermore, the proposed method allows the evaluation of accessibility requirements for different exit points, considering the selection of various first aid transport equipment. By proposing a holistic framework for a first aid system aimed at elderly individuals, this study also demonstrated how the implementation of computer vision and BIM techniques can effectively address the problem of falls among the elderly, particularly in environments like nursing homes [18].

[1]https://www.great.ufc.br/

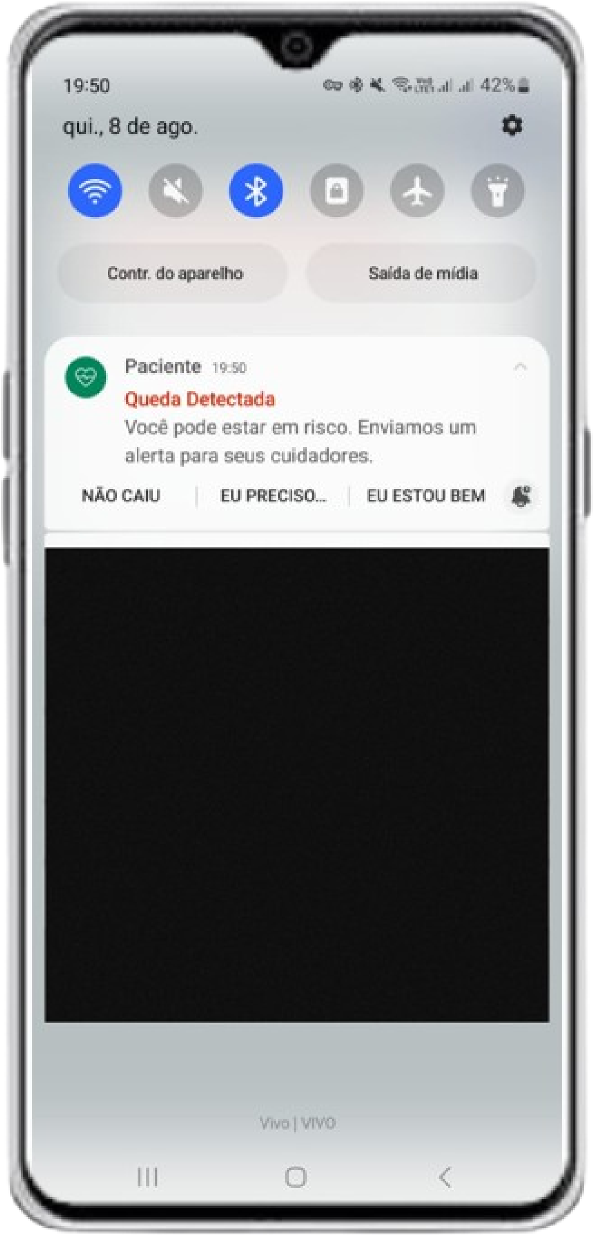
Figure 1: Health Risk app home screen. Source: Created by the author.

3.2. Smart healthcare in smart cities: wireless patient monitoring system using IoT

This study presents the proposal for an intelligent ambulance system using emerging technologies such as 5G and the Internet of Things (IoT). The central objective is to provide support to patients who require assistance during ambulance transport. The system integrates health monitoring sensors, enabling the transmission of vital data to the hospital, which speeds up diagnosis and treatment. Additionally, the patient can request an ambulance and send emergency details to nearby contacts. This approach aims to optimize response time and provide faster and more efficient medical assistance.

The research demonstrated the effectiveness of remote monitoring of patients' health parameters through Android devices and IoT. The application enabled the rapid transmission of vital information to the hospital, accelerating the medical care process. The implementation of the proposed system resulted in a shorter response time compared to the traditional system, which still predominantly relies on phone calls. The system's ability to remain in real-time contact with the patient for continuous monitoring was highlighted as a significant advantage. The installation of the system in ambulances

Figure 2: Notification of possible crash of the Health Risk application. Source: Created by the author.



facilitated the transmission of emergency information to the medical center and allowed the patient to track the ambulance's route. An essential feature of the system is the ability to send emergency messages to nearby contacts, including medical information and the patient's location. The possibility of expanding the application with security mechanisms to protect health data was also suggested. These results highlight the system's potential to improve medical care and save lives [19].

3.3. COVID-SAFE: An IoT-Based System for Automated Health Monitoring and Surveillance in Post-Pandemic Life

This study proposes an IoT-based solution to address public health challenges during pandemics, such as COVID-19. It presents a framework that integrates low-cost IoT devices, a mobile application, and machine learning algorithms for real-time data analysis. This solution enables remote health monitoring for users, including parameters like body temperature and respiratory rate, while also providing guidance on necessary physical distancing to reduce the virus's spread. The experiments conducted demonstrate the system's effectiveness in accurately measuring the distance between devices and predicting the risk of infection spread. Furthermore, energy and bandwidth analysis reveals varying demands in different data transmission scenarios, allowing the system to adapt to the specific needs of various environments. Compared to other existing solutions, this system offers a comprehensive and effective approach for continuous health monitoring and mitigating the risk of virus exposure [20].

3.4. LungNet: A hybrid deep-CNN model for lung cancer diagnosis using CT and wearable sensor-based medical IoT data

This study aims to enhance lung cancer diagnosis by using a hybrid approach that integrates Convolutional Neural Networks (CNN) with data from the Internet of Medical Things (IoMT). The neural network, called LungNet, processes computed tomography (CT) images to classify the stage of cancer with high accuracy. Additionally, data collected by wearable MIoT devices, which monitor various physiological signals, are incorporated into the diagnostic process, providing a more comprehensive view of the patient's condition. This integration results in a more reliable and convenient diagnosis, while also facilitating remote access to diagnostic services. The paper details the hybrid architecture, the combination of MIoT data with image features, and proposes an ubiquitous lung cancer staging classification service.

LungNet is a system that combines CNNs with IoMT devices, which form a MBAN (Mobile Body Area Network), providing additional data to enhance the accuracy of the classifier. This improves both the reliability and accuracy of predictions. After the initial classification, detected nodules are segmented and further classified based on their size. The performance of LungNet is evaluated using metrics such as accuracy, sensitivity, and specificity. The system comprises four main stages: data acquisition, data processing, sub-classification of cancer stage based on nodule size, and diagnosis and decision-making. Data are collected by MIoT devices like the OMRON HeartGuide and sent to a server where LungNet is trained, with implementation and training carried out in MATLAB and TensorFlow.

LungNet demonstrated robust results in classifying lung cancer nodules, outperforming other networks by achieving an overall accuracy of 96.81% and a sensitivity of 97.02%. After applying elimination criteria, the false positive rate was reduced to an average of 3.35%. Furthermore, compared to transfer learning methods, LungNet showed superior effectiveness. A distinctive feature of LungNet is its ability to classify nodules into five classes and four subclasses, positioning itself as a unique approach in the field. Despite limitations, such as dependency on MBAN data and the need for augmented images, LungNet shows promise for the development of widely accessible and reliable automatic lung cancer diagnostic systems [21].

As presented, all the analyzed works feature systems implemented in the healthcare domain and utilize APIs for integration between healthcare systems or for the functioning of the necessary resources in their respective applications. The integrated emergency system for elderly people with fall detection and BIM routing does not use IoT but employs the API for various purposes related to the integration and manipulation of BIM model data. On the other hand, the smart medical monitoring system for smart cities uses IoT to optimize patient monitoring and ambulance tracking, with the API ensuring efficient communication between health devices, users, and medical centers. The COVID-SAFE system leverages IoT to collect and transmit health data, connecting a smartphone application to the server via an API. The hybrid CNN model for lung cancer diagnosis integrates data from wearable sensors with computed tomography images, using the API to normalize and store this data in a relational database, enhancing the diagnosis. Finally, this thesis also employs IoT to collect and transmit fall data, with the API sending patient and fall information to a hospital system.

4. API Development

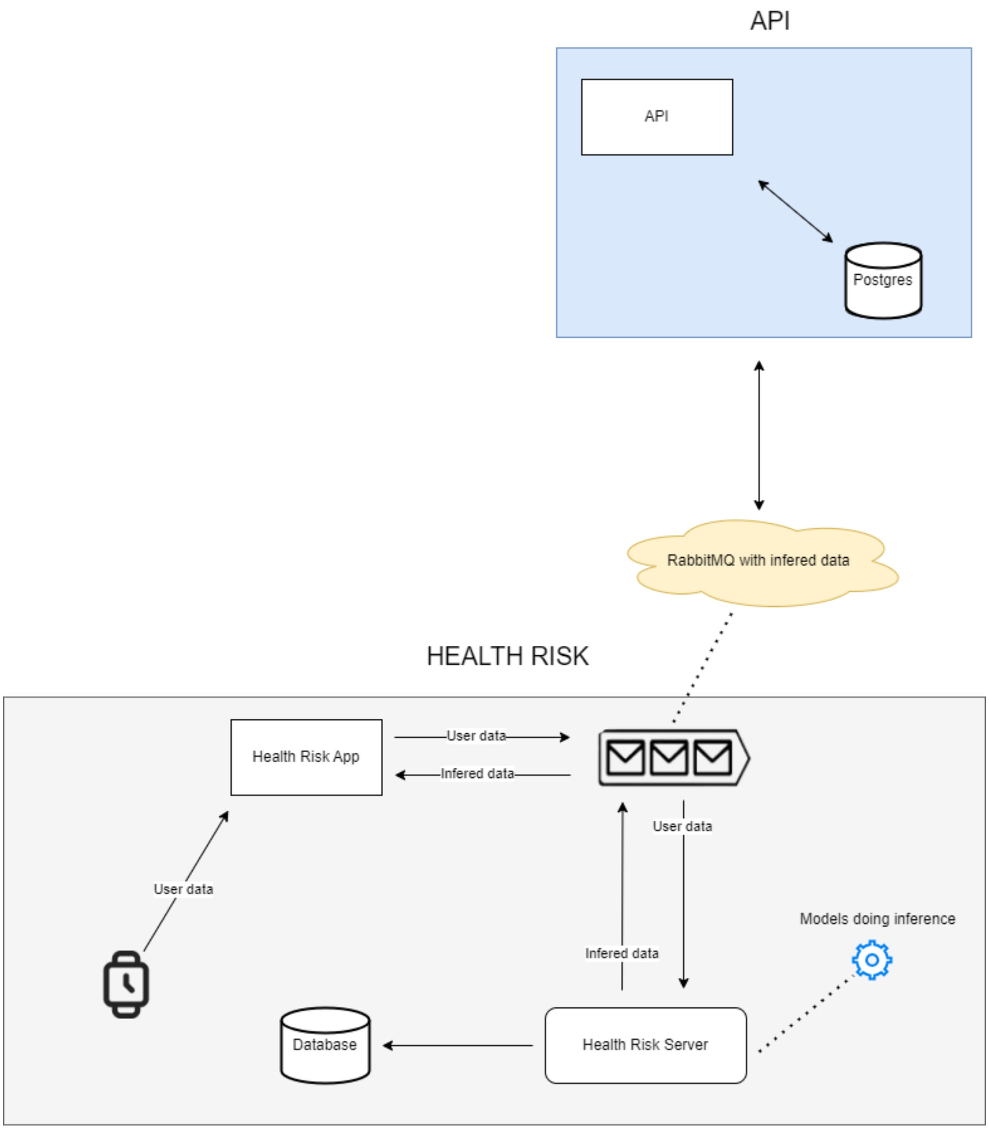
At the start of the project, a study was conducted on the source code of the Health Risk system, which was already completed, while the Smart Hospital project was still under development. The focus was on understanding the patient-related structures and how the frontend communicated with the backend.

Necessary integration points for building the fall report API were identified, along with the essential data for consistent integration. Based on this analysis, a requirements document was created specifying the data to be transmitted to the API and consumed by the Smart Hospital system.

The main systems in this project were at different stages of development. To handle these two applications at different stages—one fully completed, Health Risk, and the other still in development, Smart Hospital—it was necessary to choose the best approach to start developing the fall report API. After examining the Health Risk code and considering that this application is responsible for sending the main data to Smart Hospital, the priority was given to integration, focusing initially on communication with Health Risk. Once this communication phase was completed, the necessary modifications were made to the Smart Hospital project, which was adjusted in its more advanced version to receive the data transmitted by the API.

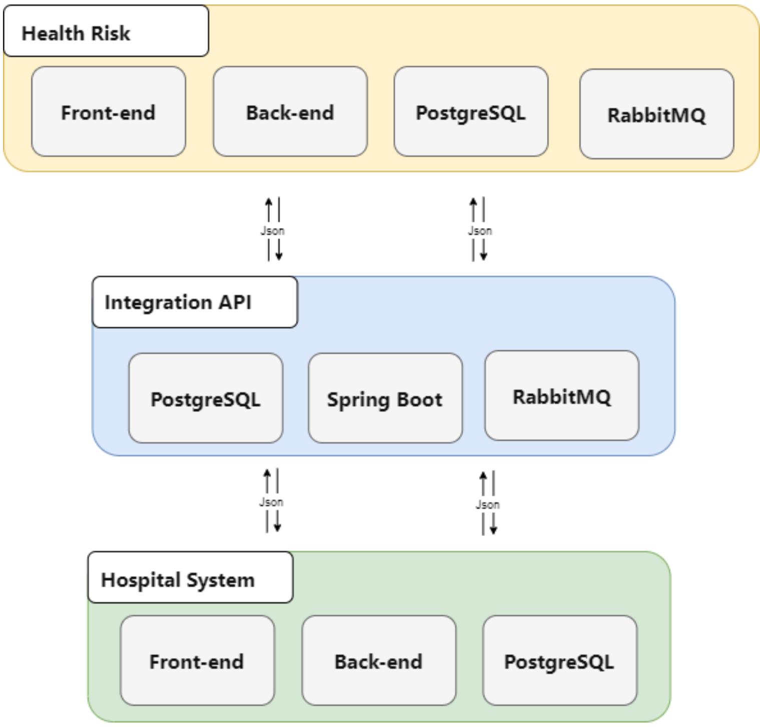
To solidify the understanding of the API's communication with Health Risk and to gain a deeper knowledge of the Health Risk code, a figure was created to represent the communication between Health Risk and the API, as well as a figure illustrating the architecture of the integration between Health Risk and Smart Hospital. Figure 3 presents the integration between the Health Risk application and the API, while Figure 4 outlines the architecture that will serve as the basis for the API's implementation.

Figure 3: Integration diagram. Source: Created by the author.



After understanding how the Health Risk system functions, from the patient data structure and the data sent by the smartwatch to the training model, the next step was to proceed with the actual construction of the API. The chosen programming language for development was Java, using the Spring Boot framework. Java was selected for this project because it is a mature programming language with extensive documentation, ensuring stability for projects and supporting a long lifecycle. The combination of Java and Spring Boot allows for the rapid and efficient development of APIs, thanks to its simplified configuration and the variety of tools available. Applications built with Spring Boot can be easily scaled to meet growing demands, and the organized and modular code typical of Spring Boot applications facilitates maintenance and understanding by other developers.

Figure 4: Systems integration architecture. Source: Created by the author.



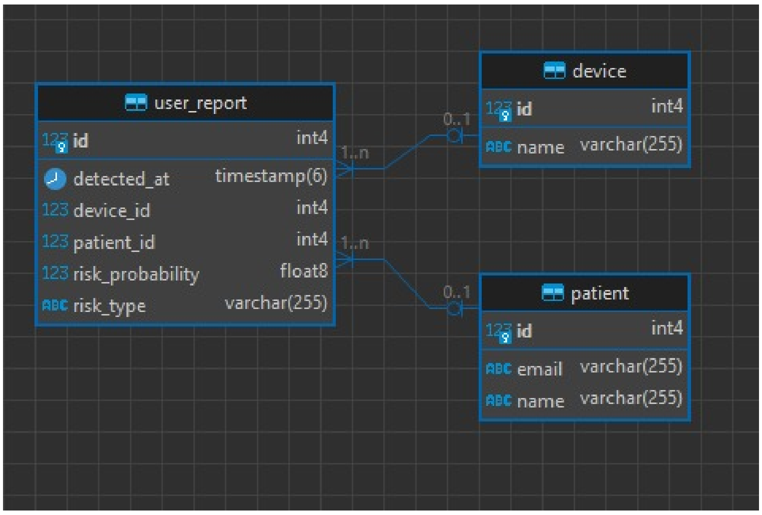
Considering Figures 3 and 4, it became clear that the integration should occur through the messaging system that Health Risk was already using, which is RabbitMQ, specifically utilizing the MQTT protocol via RabbitMQ. As a development strategy, the decision was made to deploy the developed API alongside a RabbitMQ consumer service, as this would be the best option to encapsulate the logic for fetching data from the database within a single service. This approach eliminates the need to create data retrieval logic in the Smart Hospital backend.

For the construction of the fall report API, the report message indicating a possible patient fall was considered. This message, containing the necessary data for the report, was formatted in JSON and was broken down into three related models that include this information: patient data, device data, fall probability, timestamp of the possible fall, and the type of risk.

The message was divided into three entities: Device, Patient, and User Report, with the perspective of system expansion. This way, each of these entities can be supplemented with more information in the future. Each entity has a dedicated service layer. The developed entities and their organization are detailed in Figure 5. The fall report API has only three endpoints, which are used to retrieve reports for a patient within the last 24 hours based on the patient's email, retrieve all reports from the database, and retrieve all reports from the database within the last 24 hours.

Focusing on the communication between the API and the Smart Hospital system, since the fall report API was developed separately from the Smart Hospital's backend, no modifications were needed in the backend. The only changes were made directly to the Smart Hospital's frontend. Essentially, the fall report API routes were added to the centralized routing file. A screen dedicated to displaying fall reports was created in the Smart Hospital's frontend. The purpose of this new service in the frontend

Figure 5: API database table. Source: Created by the author.

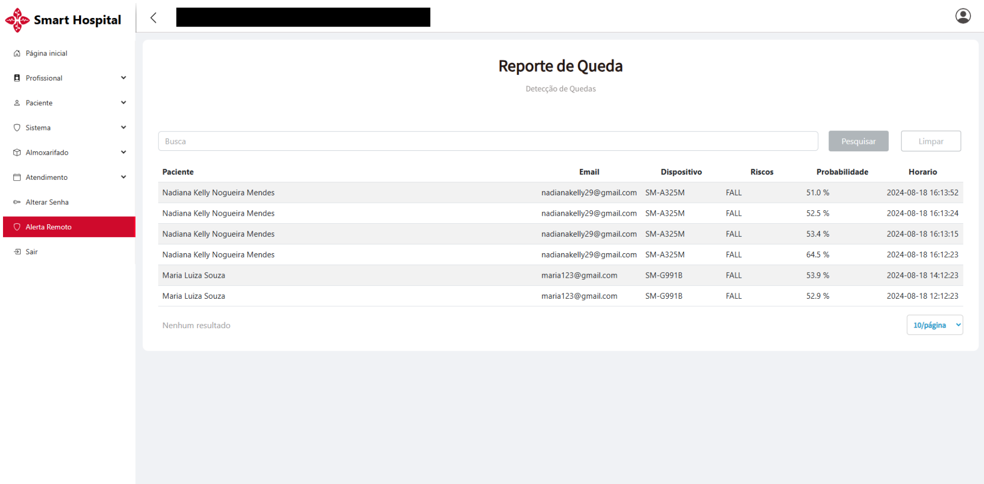


is to provide healthcare professionals with information on possible falls of remote patients, who are mostly users of Health Risk.

The system is designed to receive information only about falls that occurred within the last 24 hours. This limitation is based on the observation that most fall situations that qualify as emergencies require attention within this timeframe. Additionally, it is possible to search for fall information for a specific patient based on their email. It is important to note that, in this work, the patient's email was used as the unique search identifier because it is the only unique data available in Health Risk and is also present in the Smart Hospital's patient database. However, future project versions are planned to include more commonly used unique identifiers in the hospital environment, such as CPF (Brazilian individual taxpayer registry) or SUS (Brazilian Unified Health System) card.

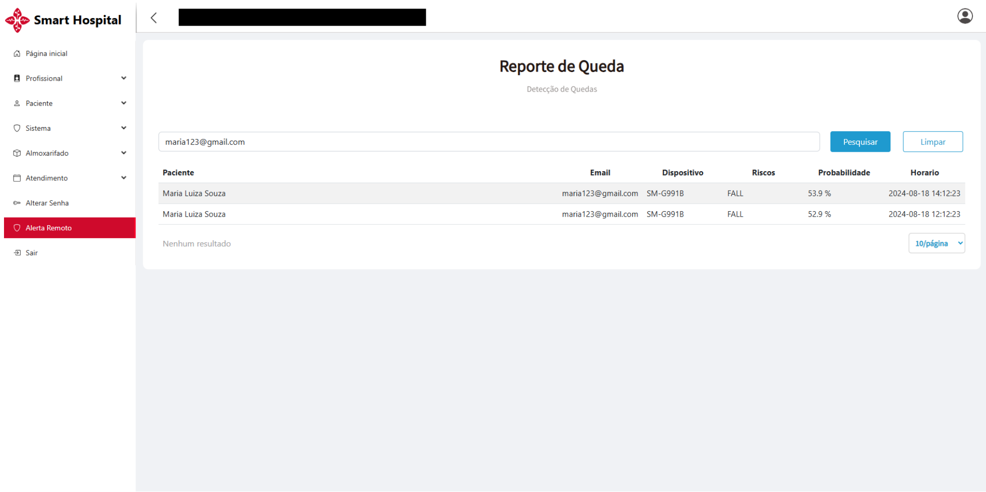
Figures 6 and 7 represent the screen created in Smart Hospital to display the fall reports of Health Risk users. The first screen shows all reports within the 24-hour interval, while the second screen displays the fall reports of a specific patient identified by their email. Figure 8 represents the complete implementation of the fall report API, illustrating the communication with Health Risk and the transfer of data to Smart Hospital.

Figure 6: Smart Hospital screen that receives fall reports. Source: Created by the author.



It is important to note that, to populate data in the Smart Hospital system's database, an account was created in the Health Risk application, as represented by Figure 1, and real data was used with the help of a Samsung smartwatch, specifically the Galaxy Watch4 model. Additionally, to quickly add more

Figure 7: Smart Hospital screen that filters patient email reports. Source: Created by the author.



users, some were registered directly in the database using the DBeaver system. The implementation code for this API is available exclusively in a private project repository due to the confidentiality of the projects involved.

API testing involves evaluating the quality and functionality of programming interfaces to ensure they operate as expected and meet specified requirements. This process includes simulating various interactions with the API and comparing the obtained results with the expected outcomes. The primary goal of API testing is to ensure that the programming interfaces deliver accurate and efficient results. To achieve this, multiple requests are made to the API, simulating different usage scenarios, and the responses are analyzed for errors or unexpected behaviors. API testing is a crucial step in API management, as it ensures quality and reliability before the API is made available to end users [22].

There are several methods to verify if an API is functioning correctly, each addressing different aspects of its functionality. In this work, functional tests were chosen to ensure the expected results are achieved. The main focus was on verifying the endpoints, ensuring that requests are processed correctly and that the returned responses align with expectations.

For the documentation of the API tests that integrate the Smart Hospital and Health Risk systems, a detailed document was created covering every stage of the process, including preparation, execution, and results of the tests. The executed API tests revealed that the integration of the Health Risk and Smart Hospital systems was successfully accomplished. No issues were found in the communication between the systems via the API. Additionally, the API performed the necessary communication between the systems as expected by the requirements elicitation. Figures 9 and 10 depict the tests conducted to assess the API's behavior according to the required specifications.

To verify the operation of Smart Hospital with the data provided by the API, the same functional tests executed for the API were performed. After creating the screen to receive data via the API, an analysis of the obtained data was conducted based on scenarios relevant to the proposed requirements. As shown in Figures 6 and 7, the executed tests yielded satisfactory results, adhering to the acceptance criteria specified in the project's requirements document.

The tests primarily focused on functional requirements, including the validation of data returns for all users and the retrieval of specific information based on the user's email, both concerning results from the last 24 hours. The testing strategy adopted involved conducting functional tests using manual approaches. For manual testing, the Insomnia tool was used on a DELL i5 device running Windows 11, while performance testing of the API was conducted using automated tests with Postman, also on the same device.

Figure 8: Integration of Health Risk and Smart Hospital systems. Source: Created by the author.

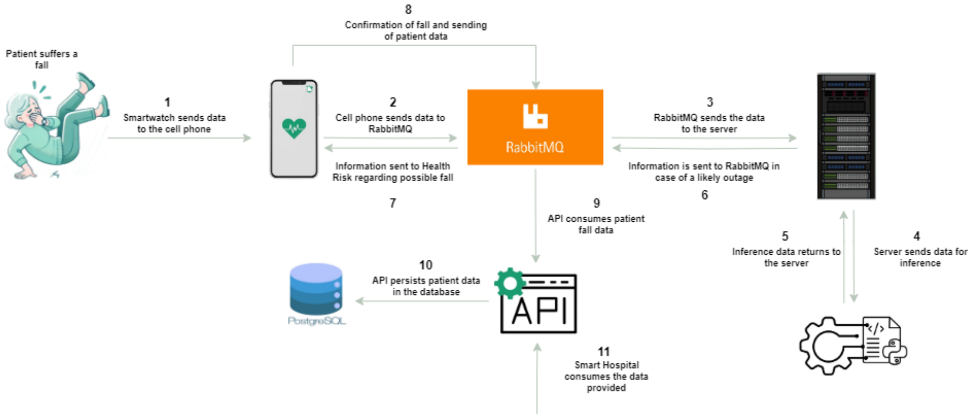


Figure 9: Return test of all users from the last 24 hours. Source: Created by the author.

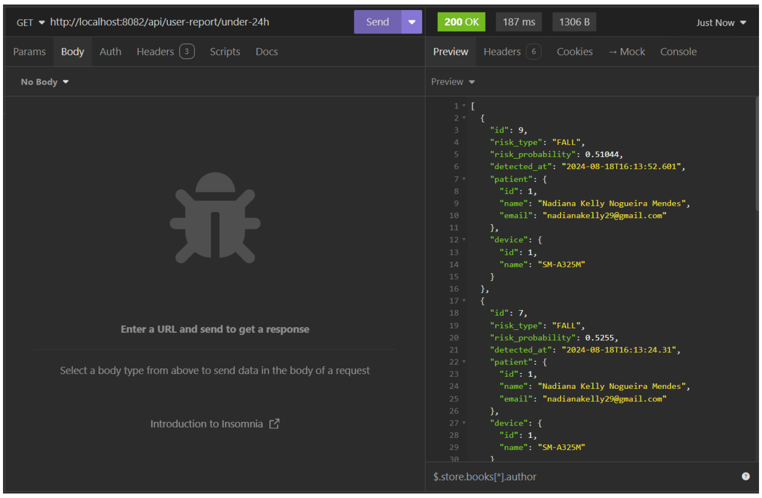
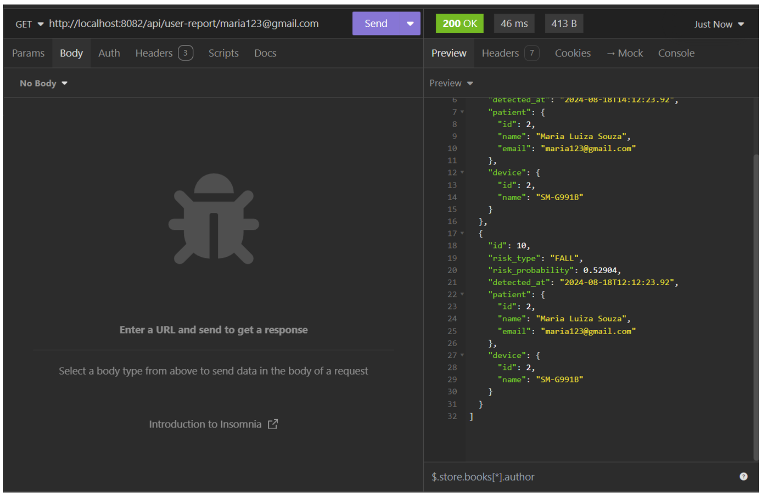


Figure 10: Testing feedback from a specific user via email. Source: Created by the author.



5. Discussion of results

Falls among the elderly are considered urgent situations, especially if there is resulting edema or pain, as it is necessary to investigate the possibility of trauma [23]. An example of how falls can also be considered emergency situations with life-threatening risks is that injuries caused by falls are among the most common treatments in trauma categories. In Campina Grande, for instance, 3,271 patients over the age of 60 were treated due to falls. Fractures in the lower limbs caused by falls have a high mortality rate among elderly patients, as they often result in prolonged periods of immobility [24].

Using this analysis of the work presented, along with the information that falls among the elderly are categorized as urgent and emergency situations, it is necessary to calculate the average expected response time of healthcare units for these cases. Taking into account the definition of ISO/IEC 25010, which defines performance efficiency as performance in relation to the amount of resources used under specified conditions [25], we can compare the response time difference between the scenario where fall information is transmitted to the hospital via API and the response time of the scenario where fall information is provided to the hospital by telephone.

Ambulance services are evaluated by the time elapsed between receiving an emergency call and the arrival of a vehicle at the patient's location. In this system, all calls are classified into four categories: life-threatening situations, emergency situations, urgent situations, and less urgent situations. Focusing on the emergency and urgent categories, which can encompass fall scenarios, particularly among the elderly, all ambulance services are required to respond to emergency category calls within an average of 18 minutes. For urgent category calls, ambulance services should respond to 90% of calls within 120 minutes [26]. Recent data reveal that the average wait time for the Mobile Emergency Care Service (SAMU) for serious patients is much higher than expected. According to recent surveys from some Brazilian states, the average wait time for SAMU service ranges from 15 to 38 minutes [27].

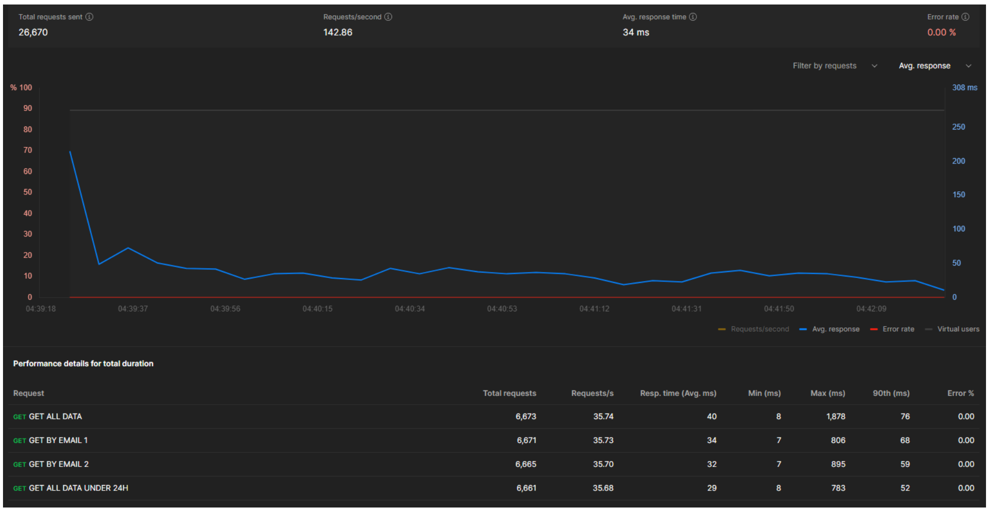
According to the most recent data from 2022, the average SAMU response time varies at different stages of the process. Initially, the Medical Care and Regulation Technicians (TARMs) handle the initial requests, with 59% of cases being attended to in 1 to 2 minutes, 25% in less than 1 minute, 14% between 2 and 3 minutes, and 2% taking more than 3 minutes. After the initial response, the call is transferred to a Medical Regulator, who classifies the situation and decides on the type of assistance. This regulatory response is completed in less than 2 minutes in 46% of cases, between 2 and 4 minutes in 22% of cases, and takes more than 4 minutes in 32% of occurrences. After regulation, the time for dispatching the vehicles by the Occurrence Regulator (RO) is less than 1 minute in 79% of cases, between 1 and 2 minutes in 9% of cases, and more than 2 minutes in 12% of cases. The response team starts to travel to the occurrence in less than 2 minutes in 67% of cases, between 2 and 4 minutes in 16%, and more than 4 minutes in 17%. The time for the ambulance to reach the occurrence site is under 5 minutes in 14% of cases, between 5 and 10 minutes in 29%, between 10 and 20 minutes in 35%, and over 20 minutes in 22% of cases. Considering the total time from receiving the call to the ambulance's arrival on-site, 11% of responses are completed in less than 5 minutes, 27% in less than 10 minutes, 36% between 10 and 20 minutes, and 26% take more than 20 minutes [28]. The representation of this service flow is shown in the figure 11.

Figure 11: SAMU service flow. Source: Created by the author.

It is important to highlight the distinction between SAMU and other available ambulances for care. In summary, SAMU provides pre-hospital care with a specialized medical team to handle emergency cases, while ambulances are responsible for transporting patients to hospitals or healthcare units [29].

In order to obtain a measure that demonstrates the good performance of the API in sending information about the user who suffered a fall to the desired system, in this case, the Smart Hospital, a performance test of the API was conducted using the Postman tool, which has performance testing functionality. The performance test conducted on the API analyzed all API routes. To obtain the API's performance test, a test was configured with 100 virtual users, all sending requests to the API over a period of 3 minutes. The analysis of the API's performance presented in Figure 12 shows that the API has consistent performance, with average response times ranging from 29 ms to 40 ms. The overall average response time is 33.75 ms, calculated as the average of the average response times of the endpoints (40 ms, 34 ms, 32 ms, and 29 ms). The overall minimum response time is 7 ms, while the overall maximum response time is 1,878 ms.

Figure 12: Performance test result with 100 virtual users



The testing of the API faced several significant limitations. Firstly, it was not possible to evaluate the API in real scenarios involving its full integration with the Smart Hospital system, as it is still under development during this study. Additionally, the lack of tests conducted on the SAMU response flow represents a limitation, as it prevented the assessment of the API's performance in a real emergency environment. These constraints limit the scope of the results obtained and indicate the need for future studies to evaluate the API's effectiveness in a real-world context.

Focusing on the TARM service, which captures the same information provided by the API, it is noted that the registered response time varies from 1 to 3 minutes per call. Responding to a person who has fallen is classified as an urgent or emergency situation, depending on the patient's condition after the incident. Since SAMU is responsible for emergencies, the API's communication within the SAMU response flow, particularly during the TARM phase, has the potential to optimize the efficiency of the process in healthcare units that utilize the Smart Hospital.

The FallReportAPI, which connects the Health Risk system to the Smart Hospital, provides immediate updates on the status of monitored users, with an emphasis on falls that require assistance. With this integration, the expectation is that communication between the patient and the healthcare unit will become more agile.

In summary, the FallReportAPI enables a quick update of information regarding falls of patients linked to the Health Risk. Compared to the traditional communication method via telephone, which results in a waiting time of 1 to 3 minutes for a caregiver to contact after a fall, the use of the API promises a significant reduction in this time. Thus, it is expected that the FallReportAPI will positively impact the efficiency of TARM responses, demonstrating that its performance in expediting service exceeds traditional methods, such as phone calls to SAMU.

6. Conclusion

This work aimed to develop an API for integrating two healthcare systems: a mobile fall detection system based on the Internet of Things and a hospital management system. The project began with a literature review focusing on systems based on the Internet of Medical Things (IoMT). This was the first step of the project. Subsequently, a detailed study of the applications involved in the integration was conducted, specifically the Health Risk and Smart Hospital systems. This study resulted in the creation of a requirements document, highlighting the information to be transmitted between the systems to ensure they could collaborate in fall situations.

The FallReportAPI was developed using Java as the programming language, with the Spring Boot framework and Hibernate for communication with the SQL database, and PostgreSQL for storing the obtained data. The FallReportAPI underwent API testing and performance testing, yielding satisfactory results that met the requirements established in this work, focusing on efficient and coherent integration between the Health Risk and Smart Hospital systems.

As future work, it is considered to implement geolocation, allowing the API to collect and transmit the exact location of a user who has suffered a fall, which would increase the accuracy of emergency services. Additionally, exploring integration with hospital care flows is essential, ensuring that the fall detection system can be seamlessly incorporated into emergency and urgent care processes, enabling hospital teams to dispatch care effectively based on the API's data. Another future goal would be to expand the API beyond fall detection to monitor and transmit information about other health emergencies, such as heart attacks, epileptic seizures, or critical blood pressure elevations, using data from various connected health devices.

Acknowledgments

We recognize that the development of FallReportAPI was made possible through a collaborative effort. To CNPq for the productivity grant from Rossana M. C. Andrade 1D (306362/2021-0), from Rafael Lopes Gomes DT-2 (303877/2021-9) and Huawei for the partnership in this project.

References

[1] B. P. Santos, et al., Internet das coisas: da teoria à prática, in: Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, volume 31, 2016, p. 16.

[2] R. O. da Silva, J. L. S. de Oliveira, A internet das coisas (iot) com enfoque na saúde, Tecnologias Em Projeção 8 (2017) 77–85.

[3] I. Araújo, M. B. Pereira, W. Silva, I. Linhares, V. Marx, A. M. Andrade, R. M. Andrade, M. F. de Castro, Machine learning and cloud enabled fall detection system using data from wearable devices: Deployment and evaluation, in: Anais do XXII Simpósio Brasileiro de Computação Aplicada à Saúde, SBC, 2022, pp. 413–424.

[4] A. F. Ferreira Costa, V. Tomé Oliveira, V. L. Leitão Dantas, I. De Sousa Santos, R. M. De Castro Andrade, R. Lopes Gomes, Impacts of using pdca in the requirements specification process, in: Proceedings of the XXII Brazilian Symposium on Software Quality, 2023, pp. 244–253.

[5] E. A. G. Junior, R. D. Rocha, R. de Souza Maciel, Desenvolvimento de api rest com spring boot, RIOS - Revista Científica do Centro Universitário do Rio São Francisco 15 (2021) 499.

[6] M. Weiser, The computer for the 21st century, ACM SIGMOBILE mobile computing and communications review 3 (1999) 3–11.

[7] M. Faccioni Filho, Internet das coisas, Unisul Virtual (2016).

[8] C. M. Rosa, P. A. R. d. Souza, J. M. d. Silva, Inovação em saúde e internet das coisas (iot): Um panorama do desenvolvimento científico e tecnológico, Perspectivas em Ciência da Informação 25 (2020) 164–181.

[9] J. J. Rodrigues, D. B. D. R. Segundo, H. A. Junqueira, M. H. Sabino, R. M. Prince, J. Al-Muhtadi, V. H. C. De Albuquerque, Enabling technologies for the internet of health things, Ieee Access 6 (2018) 13129–13141.

[10] Y. Yuehong, Y. Zeng, X. Chen, Y. Fan, The internet of things in healthcare: An overview, Journal of Industrial Information Integration 1 (2016) 3–13.

[11] C. Bhatt, N. Dey, A. S. Ashour, Internet of Things and Big Data Technologies for Next Generation Healthcare, volume 23 of *Studies in Big Data*, Springer, 2017. Includes case studies of smart healthcare applications, such as telemedicine and health management applications.

[12] Y. Chen, A. Bouferguene, Y. Shen, M. Al-Hussein, Difference analysis of regional population ageing from temporal and spatial perspectives: A case study in china, Regional Studies 53 (2019) 849–860.

[13] U. Nations, World population ageing, Department of Economic and Social Affairs (2015) 1–164.

[14] C. I. Martins, K. R. da Silva, M. C. S. Camargos, Fatores sociodemográficos e clínicos associados ao atendimento de idosos vítimas de quedas em um pronto-socorro, Revista de Administração Hospitalar e Inovação em Saúde (2022).

[15] R. L. A. Almeida, J. d. O. V. Paiva, T. N. Gouveia, H. L. C. T. Barroso, J. B. F. Neto, I. d. S. Santos, A. L. d. P. Evangelista, L. O. M. d. Andrade, I. C. d. H. C. Barreto, R. M. C. Andrade, Fictitious personas for interdisciplinary team alignment in the requirements elicitation activities, in: Proceedings of the XVIII Brazilian Symposium on Software Quality (SBQS '19), ACM, 2019, pp. 276–285.

[16] I. Barreto, L. Andrade, A. Evangelista, I. Santos, R. Andrade, A. Oliveira, J. F. Neto, V. Frota, Relato de concepção de aplicação para gestão da saúde da pessoa idosa dependente, in: Anais do XX Simpósio Brasileiro de Computação Aplicada à Saúde, SBC, 2020, pp. 422–427.

[17] I. L. de Araújo, E. C. Junior, P. A. d. S. Duarte, I. de Sousa Santos, P. A. M. de Oliveira, C. M. O. Mendes, R. M. de Castro Andrade, Towards a taxonomy for the development of older adults healthcare applications., in: HICSS, 2020, pp. 1–10.

[18] Y. Chen, Y. Zhang, B. Xiao, H. Li, A framework for the elderly first aid system by integrating vision-based fall detection and bim-based indoor rescue routing, Advanced Engineering Informatics 54 (2022) 101766.

[19] M. Poongodi, A. Sharma, M. Hamdi, M. Maode, N. Chilamkurti, Smart healthcare in smart cities: wireless patient monitoring system using iot, The Journal of Supercomputing (2021) 1–26.

[20] S. S. Vedaei, A. Fotovvat, M. R. Mohebbian, G. M. E. Rahman, K. A. Wahid, P. Babyn, H. R. Marateb, M. Mansourian, R. Sami, Covid-safe: An iot-based system for automated health monitoring and surveillance in post-pandemic life, IEEE Access 8 (2020) 188538–188551. doi:10.1109/ACCESS.2020.3030194.

[21] N. Faruqui, M. A. Yousuf, M. Whaiduzzaman, A. Azad, A. Barros, M. A. Moni, Lungnet: A hybrid deep-cnn model for lung cancer diagnosis using ct and wearable sensor-based medical iot data, Computers in Biology and Medicine 139 (2021) 104961.

[22] Astera, The ultimate guide to api testing, 2024. URL: https://www.astera.com/pt/type/blog/the-ultimate-guide-to-api-testing/, acesso em: 07 ago. 2024.

[23] Secretaria da Saúde do Estado do Ceará, Quedas em idosos: entenda riscos, prevenção e saiba quando levar a uma emergência, 2023. URL: https://www.saude.ce.gov.br/2023/08/29/quedas-em-idosos-entenda-riscos-prevencao-e-saiba-quando-levar-a-uma-emergencia/, acesso em: 09 ago. 2024.

[24] Governo do Estado da Paraíba, Lesões causadas por quedas são atendimentos mais comuns no trauma, 2023. URL: https://paraiba.pb.gov.br/noticias/lesoes-causadas-por-quedas-sao-atendimentos-mais-comuns-no-trauma-cg, acesso em: 09 ago. 2024.

[25] ISO, Iso/iec 25010: Systems and software engineering - systems and software quality requirements and evaluation (square) - system and software quality models, 2011. Acesso em: 9 ago. 2024.

[26] Nuffield Trust, Ambulance response times, 2023. URL: https://www.nuffieldtrust.org.uk/resource/ambulance-response-times, acesso em: 9 ago. 2024.
[27] G1, Samu demora mais do que o tempo recomendado para atender casos graves em 5 capitais, 2019. URL: https://g1.globo.com/ciencia-e-saude/noticia/2019/12/26/samu-demora-mais-do-que-o-tempo-recomendado-para-atender-casos-graves-em-5-capitais.ghtml, acesso em: 9 ago. 2024.
[28] H. L. A. Forastieri Filho, C. M. F. de Araujo, A. d. S. M. Junior, H. L. C. Forastieri, et al., Tempo resposta no samu–192 e suas implicações, Cadernos UniFOA 17 (2022) 173–183.
[29] G. R. de Siqueira, Diferença entre samu e ambulância: entenda as principais distinções, 2023. URL: https://brasilemergenciasmedicas.com.br/qual-a-diferenca-entre-samu-e-ambulancia/, acesso em: 7 set. 2024.