UNISINOS

Programa de Pós-Graduação em

Computação Aplicada

Mestrado/Doutorado Acadêmico

Arthur Tassinari Cabral

HealthTranslator: A model for integration between IoT devices and healthcare systems

São Leopoldo, 2023

Arthur Tassinari Cabral

HEALTHTRANSLATOR: A MODEL FOR INTEGRATION BETWEEN IOT DEVICES AND HEALTHCARE SYSTEMS

Dissertação apresentada como requisito para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Advisor:
Prof. Dr. Rodrigo da Rosa Righi

São Leopoldo
2023

C117h Cabral, Arthur Tassinari
Healthtranslator : a model for integration between IoT devices and healthcare systems / por Arthur Tassinari Cabral. – 2023.
45 f. : il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2023.
Advisor: Prof. Dr. Rodrigo da Rosa Righi.

1. IoT. 2. Data enrichment. 3. Iteroperability. 4. FHIR. 5. Periodicity. I. Título.

CDU 004.738.5

Catalogação na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

ABSTRACT

Nowadays, a great number of articles in the health field use IoT devices in the proposed models, techniques, and approaches. The manufacturers of these devices present different ways to provide the vital signs captured by the equipment. Thus, third-party applications and solutions that require the use of vital signs need to adapt to the different approaches proposed by each manufacturer, which culminates in an increase in the complexity of the development of these applications. In addition, it is noted that the current state-of-the-art does not present, for the theme, an integration solution that analyzes vital signs aiming to describe the user's health status, establishing a pre-diagnosis of the health status. Given this scenario, this dissertation has the objective of promoting an integration model between IoT devices and health systems, facilitating contact between the parties and ensuring interoperability. This article promotes HealthTranslator, a model that collects the data captured by different IoT devices and provides the collected data in a file with a unique format and type, promoting the use of the data enrichment technique through the analysis of vital signs, aiming precisely to describe the user's health status. The model consists of promoting 4 microservices, with the objectives of communicating with the user, consulting vital signs collected by APIs from device companies that collect vital signs, defining the periodicity of reading vital signs based on the values of these signs, and generating the final files with a description of the user's state of health. The output object uses the FHIR standardization, used globally for healthcare solutions. HealthTranslator also features an intelligent collection of users' vital signs, as it determines a periodicity for the collection based on the individual characteristics of each user's vital signs. This model has been tested to assess its accuracy based on functional tests and unit tests. These tests presented a good work of the model, generating the description of health and performing the other steps correctly. With the proposed model, numerous healthcare applications around the world will benefit, as they will have a model that facilitates the capture of vital signs, reducing the complexity of implementing the applications themselves and related solutions.

Keywords: IoT. Data enrichment. Interoperability. Periodicity. FHIR.

CONTENTS

1 INTRODUCTION

The Internet of Things (IoT) refers to a network of ubiquitously connected smart devices that are deployed to perform a great number of tasks, including health monitoring (QADRI et al., 2020). IoT can also be understood as "*a network of massively interconnected devices (digital and mechanical machinery, items, and so on) with unique IDs*" (NANDA; PANDA; DASH, 2023). IoT has a great capacity to enhance healthcare globally (MUDAWI, 2022). The Internet of Things can be used *as a medical adherence likewise as a home monitoring tool*, and in this way, *IoT Is Transforming Human Life* (DAS et al., 2021). Currently, many articles in computer science, applied to the medical area, use IoT devices in the approaches and techniques presented. In general, in these approaches, the IoT is part of a medical device layer that communicates with other layers of the approach, which can be, for example, fog computing.

According to Nanda, Panda e Dash (2023), "*IoT devices, in a nutshell, can sense and gather data and send it via the Internet.*" In addition, according to Mukati et al. (2023), "*IoT powered technologies and sensors can be found almost everywhere to gather, monitor and greatly enhance regular healthcare lives and redefine how healthcare facilities and systems enhance their lives*". In this way, relating to the theme of IoT in the area of vital signs collection, it is important to mention that some companies provide Web APIs, as it can be noticed in Xiaomi (2022), based on the use of cloud computing, to work with the vital signs collected by their devices, and from this, it is possible to implement, for example, a model that presents a direct connection between the Web API of a particular company and a particular server dedicated to the model. Device APIs are also provided by some companies, allowing the development of device applications that use a specific device API for acquiring data from the device. These two categories of APIs are provided, for instance, by Fitbit (2022), and Garmin (2022), presenting examples of applications that use their APIs. Aiming to provide quality communication between different IoT devices and other software layers or applications, with their specific characteristics, it is relevant to implement an API that ensures interoperability between these technologies.

1.1 Motivation

In general, IoT healthcare systems have three layers: body area sensor network; Internet-connected gateways; and cloud and big data support. Commonly, gateways *act as a hub between a sensor layer and cloud services* (RAHMANI et al., 2018). The works present in the computer science literature that implement IoT systems for the mentioned area, in general, present the gateway layer, responsible for transferring the data captured by the IoT sensors to the other layers or applications of the system, without having it as one of the objectives of the work. Then, many works briefly mention the use of this layer. Furthermore, it is noticeable that the works that have the promotion of the gateway layer as one of their objectives do not address, or briefly address, the theme of the periodicity of vital signs reading according to the user's symptoms presented (for example, feverish body temperature), using this information to calculate a better periodicity of vital signs reading, according to the vital signs of each user.

Notwithstanding the existence of the scenario described, it is noted that the current landscape of IoT healthcare systems reveals a notable and significant gap in developing models that integrate wearable health devices with other applications to pre-diagnose users and document their health status using vital signs data, using the data enrichment technique. For example, Jaleel et al. (2022) apply this technique in their work, but with the objective of ensuring the improvement of data considering optimization, security, syntax, semantics, and protocol aspects. In addition, Santana e Otani (2021) also uses this technique, but it is done to perform conversion of the text into audio content. While IoT frameworks have shown promise in monitoring and collecting vital signs through body area sensor networks and transmitting the data through gateway layers to cloud services, there is still a lack of emphasis on utilizing this wealth of information to its full potential. By establishing robust connections between wearable health devices and sophisticated algorithms capable of analyzing and interpreting vital signs data, it becomes feasible to offer users more accurate and timely pre-diagnostic insights, empowering them to take proactive measures to maintain their well-being.

Considering the presented scenario, among the computational models that communicate IoT devices that collect vital signs with applications that make use of these vital signs, we note the scarcity of works that use the data enrichment technique to perform an analysis of the read vital signs. It is important to mention that a module or algorithm that aims to determine a specific periodicity for collecting the vital signs considering the individual characteristics of the vital signs of each user is also a gap presented by the literature. In addition, no articles were located that implement a gateway that aims to make the data collected on the devices available in a single file that uses the FHIR health file standard, a standard that will be discussed in Chapter 2. Therefore, the motivation for this work arises exactly from the comprehension of the existence of the two mentioned gaps. More details on these gaps are provided in Chapter 3.

1.2 Research Question

The research question that the present document has the objective to answer is the following: *"How to communicate multiple wearable devices with healthcare applications, promoting the implementation of a smart reading of the vital signs captured by the devices?"* This question considers "multiple wearable devices" as globally commercialized smart wristbands and watches from the companies Amazfit, Fitbit, Garmin, Samsung, and XIAOMI. As "healthcare applications", the question considers commercial, personal, and academic healthcare projects, such as software, frameworks, and APIs, that require working with vital signs collected by devices such as those mentioned. The "smart reading" mentioned by the research question refers to the aforementioned periodicity of reading according to the individual characteristics of the users.

1.3 Objectives

As previously mentioned, the current computer science literature presents a small number of works that have the objective of fostering an interoperability layer between technologies such as the one mentioned above. Many works that present approaches that aim to improve the performance of the medical area, either in the prevention or treatment of a single disease or many diseases, briefly discuss the communication between IoT devices and the other layers of the approach being proposed. The papers do not use data enrichment, performing a vital signs analysis. In addition, the works do not focus on presenting a way to calculate the correct periodicity for collecting the vital signs that consider the conditions and symptoms of the user. The works also do not use the FHIR standardization, a standard for healthcare data exchange used on a worldwide scale (HL7, 2023), further discussed. Given this scenario, the present research has the objective of promoting an model that ensures the interoperability of different IoT devices with other software layers and applications. This API will work focused on IoT devices that work with the collection of vital signs, accomplishing the portability of these devices with different technologies. Therefore, as specific objectives, the HealthTranslator aims to implement a smart reading of the vital signs of the users, considering the values of the vitals for calculating the correct periodicity of reading; the use of the data enrichment technique, promoting analysis of the vital signs; the use of the FHIR standardization as the outcomes format.

1.4 Research Development Stages

The development of this research was organized into 6 stages. As a starting point, a study of the theories related to the research topic was conducted to form the theoretical foundation. Once this is done, the stage of researching works related to the research topic begins. Then, step 3 is performed, which consists of the analysis of related works aiming at identifying the gaps present in the state-of-the-art of the document's theme. After this, in step 4, the model is proposed and developed in order to meet the proposed objective and answer the research question. Then, in the fifth step, the model is implemented, and in the sixth step, the tests. Steps 4, 5, and 6 can be performed more than once, depending on the results of the tests.

1.5 Document Structure

This document has a structure composed of five sections. Once this introduction is done, the background of the theme of the present paper is presented. Then, the related works to its theme are discussed, considering the use of a comparative table between such works. In the next section, the model proposed by this document is shown, specifying how the proposed interoperability between the technologies mentioned will be performed. Next, a section is provided that addresses how the model's tests will be conducted. Finally, the final considerations section is presented, which concludes this paper.

2 BACKGROUND

This chapter covers the major concepts that are related to the provision and collection of vital signs based on the state-of-the-art in the field. Section 2.1 presents the concepts regarding IoT devices that collect vital signs and the ones considered by the present document. Section 2.2 focuses on concepts dealing with the standardization of vital signs files in healthcare. Section 2.3 presents concepts related to the periodicity of vital sign readings. Finally, in section 2.4, partial considerations are made about the topics discussed in the chapter.

2.1 IoT Devices in Healthcare

This document considers, for the proposed model, the use of globally marketed IoT devices and sensors that capture the following vital signs: body temperature, heart rate, oxygen saturation, and respiration rate. This paper covers devices from the following companies, considering the coverage of their WebAPIs: Amazfit, Fitbit, Garmin, Samsung, and Xiaomi. These companies have been chosen because they are, as mentioned before, globally marketed brands, so, more people will benefit from the model here proposed. As cited by Business Connect (2023), Fitbit and Garmin are among the best sellers of 2023. Also, WebAPIs have been chosen because they provide more data than other APIs (DeviceAPIs, for example).

The Amazfit devices utilize the Zepp OS operating system (ZEPPOS, 2022). The Device App API, the App Services API, and the Side Service API are the APIs for Amazfit devices. The APIs make it possible to develop applications running on devices and smartphones, but the official documentation does not specify whether it is feasible to build applications for the use of these APIs on personal computers or servers. It is possible to use the Device App API to work with the body temperature, the heart rate, the heart rate variability, and the oxygen saturation.

Fitbit fosters two public Web APIs, the Device API and the Web API, allowing the use of the vital signs data collected by the Fitbit devices for new approaches (FITBIT, 2022). Among the 5 vital signs considered by the present paper, the Device API works with the heart rate and the heart rate variability, and the Web API works with these two ones, the body temperature, the oxygen saturation, and the breathing rate. It is relevant to observe that Fitbit also provides three more APIs named Companion API and Settings API, where the first one is used to *Get detailed technical information about the Fitbit Companion APIs* and the other one is used for configuring application settings on the devices.

Garmin provides 3 APIs, Health API, Standard SDK, and Companion SDK, to make it possible to work with the vital signs collected (GARMIN, 2022). The Health API is a cloud solution that provides access by external approaches, working with heart rate, heart rate variability, and breathing rate, among the vital signs covered by this document. The Standard SDK allows users to *access all health and fitness activity data directly from their mobile app for Android and iOS without the need for web service integration.*, and the Companion SDK too, with the difference that the second one is indicated to *access to real-time health metrics and sensor streams for episodic sampling but incorporating the all-day metrics logged by the Garmin wearable*, and the first one is indicated to *control device features or ensure the data logged by Garmin wearables is only sent to the user's application and platform*. Both Standard SDK and Companion SDK work with the heart rate, heart rate variability, and respiration rate.

The use of vital signs captured by Samsung devices is possible by using Health Connect, an Android API and platform, for sharing data for other smartphone applications (ANDROID, 2022). A variety of solutions can be built that communicate Samsung devices and smartphones running the Android operating system with the use of this API. The heart rate, the heart rate variability, the body temperature, the oxygen saturation, and the respiratory rate are provided by Health Connect.

In order to collect the data captured by Xiaomi devices, it is possible to utilize Xiaomi Health Cloud Open, a cloud solution (XIAOMI, 2022). By using this solution, the implementation of smartphone applications that operate with the data from Xiaomi devices is a possibility. Among the vital signs that this cloud solution provides, the following ones can be cited: heart rate, and heart rate variability. The documentation is unclear about providing the other vital signs considered by the present document.

2.2 Standardization of Vital Signs Files in Healthcare

Healthcare computing presents solutions, for sharing health data between different applications and systems, ensuring the interoperability between them, in the form of standardization of vital sign files. Several organizations aimed to propose ways for standardizing healthcare files. An organization responsible for providing standards for it is the HL7, Health Level Seven International. This organization *is a not-for-profit, ANSI-accredited standards-developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery, and evaluation of health services* (HL7, 2023). These standards are supported by more than 1600 members globally, which includes corporate members representing healthcare providers, government stakeholders, and pharmaceutical companies, for instance.

An example of a healthcare file standard provided by HL7, featuring a pre-defined structure and syntactic rules, is HL7 CDA (Clinical Document Architecture) (PLASTIRAS; O'SULLIVAN, 2018). This standard *"is a document markup standard that specifies the structure and semantics of "clinical documents" for the purpose of exchange between healthcare providers and patients"* (HL7, 2023). The HL7 CDA is able to support the exchange of clinical documents, and it is possible to reuse this standard in various applications. In the literature, for example, it was proposed a mobile patient health record (PHR) application *that allows tethering with an electronic health record (EHR) using HL7 CDA* (SARIPALLE; RUNYAN; RUSSELL, 2019), which highlights the presence of this standard in the computer science.

Another solution HL7 provides to facilitate interoperability of systems and services is FHIR, an acronym for "Fast Healthcare Interoperability Resources", which is "*a standard for health care data exchange*", that is "*intended to facilitate the exchange of healthcare information between healthcare providers, patients, caregivers, payers, researchers, and anyone else involved in the healthcare ecosystem*"(HL7, 2023). This standard is composed of a content model in the form of resources and a specification for the sharing of these resources, using the real-time RESTful interface structure, documents, and messages. Vorisek et al. (2022) explain that FHIR *"has been successfully implemented in clinical, public health, and epidemiological research at the stages of recruitment and consent management, data capture, and standardization as well as analysis of patient data"*. It is important to mention that FHIR supports the terminologies widely used in healthcare "Systematized Nomenclature of Medicine Clinical Terms" (SNOMED CT) and Logical Observation Identifiers, Names, and Codes (LOINC). The SNOMED CT "*is the most comprehensive, multilingual clinical healthcare terminology in the world*" (SNOMEDINTERNATIONAL, 2023), and LOINC "*is a common language (set of identifiers, names, and codes) for identifying health measurements, observations, and documents"* and it is used globally, specifically in 196 countries (LOINC, 2023). It is also relevant to highlight that, according to Ayaz et al. (2021), at that moment, FHIR was present in articles in at least 13 countries in the world, and that, in 2018, *"six large technology companies, including Microsoft, IBM, Amazon, and Google, pledged to remove barriers for health care interoperability and signed a letter that explicitly mentions FHIR as an emerging standard for the exchange of health data."*, and FHIR is currently used by government agencies in a range of countries (GRIFFIN et al., 2022).

2.3 Periodicity of Vital Sign Readings

The periodicity of reading the users' vital signs should be calculated in an intelligent way that considers the recently collected data, and the recent history of collection. By knowing a user's vital signs, it is possible to select the correct frequency of the next vital signs readings for them. For this, the first point to be highlighted is the values considered normal for the vital signs covered in this document. According to Rohmetra et al. (2021), for an average adult, the values treated as being normal are as follows:

- Body temperature: 36,2-37,1ºC (97-99ºF).
- Respiratory rate: 12–20 breaths per minute at rest.
- Pulse rate: 60–100 beats per minute at rest.
- Oxygen saturation: 95–100%.

Some rules can be elaborated based on the data related to the ranges of values understood as normal for an average adult, as presented in the list above. For example, considering that the

Table 1: Periodicity rules

Vital signs conditions	Periodicity
All normal	Every 6 hours.
Only body temperature abnormal	Every 30 minutes. If the body temperature is now normal, then the periodicity becomes every 1 hour. If the body temperature remains normal, then the periodicity will again be every 6 hours.
Other abnormal vital signs, regardless of body temperature.	Every 15 minutes. If the body temperature is now normal, then the periodicity becomes every 1 hour. If the body temperature remains normal, then the periodicity will again be every 6 hours.

Source: Authorial

user is presenting values understood to be normal for the vital signs covered by this document, a possible periodicity for reading is to follow what the Australian Commission on Safety and Quality in Health Care (2022) supports, for the majority of patients: *"vital signs should be measured at least every six hours."* In this case, a periodicity of vital sign readings used in a hospital environment would be applied to the home environment.

Specifically for the user's temperature, it can be implemented a rule just for this vital sign: Once it is measured and shows no abnormalities, it can be checked *X* number of times during the day, but, according to Ajay (2020), it is important to *"check it at around the same time each day"*, because the body temperature *"fluctuates hour by hour"*. If the user has a fever, Cleveland Clinic (2020) suggests that the temperature can be taken once or twice an hour. In this case, the other vital signs considered by this document can be taken, since some diseases can present abnormalities in other vital signs.

As can be observed, rules for the periodicity of vital signs readings can be implemented both individually and collectively, grouping 2 or more vital signs. A possibility is to establish a reading of all vital signs in a grouped way for when the values are normal, and specific readings for when one or more values are abnormal, considering the diseases that the combinations of vital signs with abnormality can indicate.

This work considers the periodicity rules presented by Table 1 that shows the rules for collecting vital signs. These rules have been established based on the Australian Commission on Safety and Quality in Health Care (2022), Cleveland Clinic (2020), Rohmetra et al. (2021), and in technical interviews with a healthcare employee of the Regina Hospital of Novo Hamburgo (2023) and a healthcare employee of Mãe de Deus Hospital (2023). The interviews with the employee of Regina Hospital of Novo Hamburgo were conducted in January, June, and July 2023, and the interview with the employee of Mãe de Deus Hospital was conducted in January 2023.

3 RELATED WORK

This chapter has the objective of presenting and discussing the articles present in the literature that address approaches that work with the interoperability of multiple IoT devices. The analysis of the related papers can be observed in Table 2, which provides a comparison among the selected papers. The comparison considers the reference for the work, its year, ways for performing data enrichment, the supported devices by the approach, and how the work addresses scalability. This section is also responsible for presenting the research methodology that is employed to compose this paper, highlighting the databases that were considered.

3.1 Research Methodology

This section discusses the research methodology used for selecting the related works. For the selection of the works, only papers that presented an approach, aiming at communication between healthcare IoT devices and other applications or technologies could be elected as papers related to the present document. It was considered the following databases: ACM, IEEE, SpringerLink, and ScienceDirect. The filter considers a year interval between 2021 and 2023, only research articles that proposed new scientific models and approaches related to the theme of this work, the research string *("Internet of Things" OR "IoT" OR "Sensor") AND ("Healthcare" OR "Health") AND ("Conversion" OR "Translation") AND ("Interoperability" OR "Portability")*, and the fact that the content of the abstract has words that follow the rule *("Internet of Things" OR "IoT" OR "Sensor" OR "Conversion" OR "Translation" OR "Interoperability" OR "Portability") AND ("Healthcare" OR "Health")* and does not have words that follow the rule *("Survey" OR "literature review" OR "blockchain" OR "privacy" OR "cryptography" OR "artificial intelligence" OR "machine learning" OR "security")*. Then, 15 articles have been selected for reading and evaluating in order to verify if they fit with the proposal of the present article. Finally, nine articles have been selected.

3.2 State-of-the-art

The related works were analyzed considering how they perform data enrichment, the supported IoT devices, and how they handle scalability. Berta et al. (2021) propose a research that *explores and critically analyzes HL7 FHIR to design and prototype an interoperable mobile PHR that conforms to the HL7 PHR Functional Model and allows bi-directional communication with OpenEMR*. This research covers IoT devices in general as the supported devices, and presents use cases in medical, fabrics, and driving areas. To ensure scalability, this work uses RESTful API, NOSQL databases, and cloud principles. Table 1 shows more details on how this research works with scalability.

Jaleel et al. (2022) investigate *how to develop a nonvendor-locked framework, which exploits state-of-the-art data management technologies, and targets effective and efficient development in this article.* With a focus on the concept of measurement, this work abstracts *an architecture that could be applied in a variety of domains and contexts.* IoT devices in general, presenting a use case in the MIOT area, are used in this work. For scalability ends, this work uses broker and converter plugins, described in Table 1.

Hussain e Park (2021) present a *cyber-physical cardiac monitoring system for stroke management, consisting of a wearable ECG sensor, data storage and data analysis in a big data platform, and health advisory services using data analytics and medical ontology*, where the ontology has a fundamental participation on, among other points, data semantics. This work considers ECG sensor device as the device covered by its approach. To ensure scalability, it uses the message broke ActiveMQ, ElasticSearch search engine, Hadoop ecosystem platform, and MariaDB database.

Bourgais, Giustozzi e Vercouter (2021) propose *to use Stream Reasoning associated with an ontological representation of the medical context of a patient.* According to the authors, *this permits to combine in real-time static knowledge stored in an ontology and dynamic information provided by smart sensors.* This work presents the use of the data enrichment technique since it performs the definition of constraints and situations based on health data read. This work considers the use of medical IoT devices in general.

It is important to note that the state-of-the-art, in the theme of the use of ontologies, also considers applying filters on the data and for data enrichment, which was used by Wu et al. (2021), in a work that presents *an edge-based hybrid network system architecture*, that *consists of hybrid routers and an IoT gateway.* This approach works with IoT devices in general, and to ensure scalability, uses the MongoDB database, and some specific rules for organizing data, which are described in Table 1.

Mudawi (2022) proposes an *edge-based hybrid network system architecture*, which *consists of hybrid routers and an IoT gateway. The router supports two wireless protocols, BLE and long-range (LoRa), and is equipped with a solar energy harvester to extend the router's lifetime.* This system covers medical IoT devices, uses fog nodes, and proposes a specific algorithm for using them in order to ensure scalability.

Rehman et al. (2021) present *a multi-layered architecture of an IoT-based cancer observation system that can be utilized as a platform to remotely diagnose and monitor cancer patients.* In this work, an implementation framework is also presented, *along with a prototype design of a Patient Side Unit (PSU) represented by a wearable wrist band.* The sensors covered are *Temperature sensor, pulse sensor, enzymatic sensor and immunosensor*. This work is organized with a three-layers architecture, named "Patient", dealing with the IoT sensors, "Network", addressing the connection between the IoT sensors and the central switch of a hospital, and "Application", which is composed of the mentioned switch and the applications liked to it.

Santana e Otani (2021) present *a prototype of a low-cost mobile Galvanic Skin Response*

(GSR) sensor and two case studies exploring its use, namely: (1) a cloud-based support application designed to help speakers during talks when a stress event is identified; (2) a quantitative User eXperience (UX) analysis during the gameplay of a popular tower defense multiplayer mobile game (Clash Royale), comparing arousal events of winning and losing matches. This approach performs the conversion of text into audio content, which represents the presence of the data enrichment technique. This work presents a mobile galvanic skin response sensor as the IoT device covered.

Aloi et al. (2021) propose *a novel simulation-driven platform named E-ALPHA (Edge-based Assisted Living Platform for Home cAre) which supports both Edge and Cloud Computing paradigm to develop innovative AAL services in scenarios of different scales. E-ALPHA flexibly combines Edge, Cloud or Edge/Cloud deployments, supports different communication protocols, and fosters the interoperability with other IoT platforms.* This approach covers IoT devices in general, presenting two use cases in health area, and uses MW2MW Integration (Middleware-to-Middleware Integration) for works with scalability.

It is noted that, regarding the devices supported by the papers that compose the current state-of-the-art, one part of the verified papers mentions the use of IoT devices in general, and the other part mentions specifically IoT devices that work with health data. This is because one part of the papers deals with approaches for different contexts, not limited only to the health context, while the other part deals specifically with the health context. Taking into account the articles specifically concerned with devices that collect health data, the use of devices to capture the following vital signs, among other examples, was observed (HUSSAIN; PARK, 2021), (MUDAWI, 2022), and (REHMAN et al., 2021):

- Heart rate.
- Temperature.
- Blood pressure.
- Oxygen saturation.
- Blood measurements (Blood circulation, glucose, lactate, calcium, potassium).
- Immunity data.

Another relevant topic addressed by the current state-of-the-art is scalability. Some of the related works, to work on this topic, use project organization strategies. For instance, Berta et al. (2021) provides the use of RESTful API, NOSQL databases and cloud principles. In addition, the authors implement the use of Node.js in the Web Server, as it "[...] *avoids the multithreading burden by employing a nonblocking single-thread pattern and is able to efficiently serve multiple concurrent clients by operating asynchronously, employing the event-loop mechanism, which is particularly suited for microservices architecture.*", They also worked with abstractions of entities for ensuring scalability.

3.3 Analysis and Research Opportunities

Based on the readings of articles related to the theme of the present document, the following gap has been identified: The absence of an integration model between IoT devices and healthcare applications that performs the deduction of observations about users' health status based on the read vital signs. Therefore, it is possible to affirm that the current state of the art presents the lack of a model for healthcare such as the aforementioned that works with data enrichment, establishing a pre-diagnosis on the health of the user based on the vitals. Table 1 presents an analysis of the related works.

Table 1 shows a wide range of supported devices across the articles, including general IoT devices, ECG sensors, medical IoT devices, and specific sensors like temperature, pulse, enzymatic, and immunosensors. While this diversity is advantageous in exploring different health-related use cases, Table 1 shows a wide range of devices supported in the articles, including general IoT devices, ECG sensors, medical IoT devices and specific sensors such as temperature, pulse, enzyme and immunosensors. Although this diversity is advantageous to explore different health-related use cases, it is noted that not all papers present a health focus, which denotes a possible area of computer science that can be explored, which is the promotion of data integration models focusing on health devices.

To address the aforementioned gap in the current state of the art, the development of an innovative and cohesive integration model between IoT devices and healthcare applications becomes imperative. Such a model should strive to seamlessly integrate data from various IoT devices, consulting WebAPIs for getting vital signs. By employing the data enrichment technique, this integrated system can intelligently process and analyze the collected information to deduce valuable insights about users' health statuses. Through this approach, a pre-diagnosis mechanism can be established, enabling timely and accurate assessments of individuals' well-being. The integration of IoT and healthcare in this manner holds great promise for enhancing personalized healthcare delivery.

Table 2: Comparison Among the Related Works

Article	Year	Data Enrichment	Supported devices	Scalability
Atmosphere, an Open Source Measurement-Oriented Data Framework for IoT (BERTA et al., 2021)	2021	Not covered	IoT devices in general, presenting use cases in medical, fabrics, and driving	Use of RESTful API, NOSQL databases and cloud principles. Use of Node.js in the Web Server, as it "[...] *avoids the multithreading burden by employing a nonblocking single-thread pattern and is able to efficiently serve multiple concurrent clients by operating asynchronously, employing the event-loop mechanism, which is particularly suited for microservices architecture.*" Use of Abstractions of entities
Autonomic interoperability manager: A service-oriented architecture for full-stack interoperability in the Internet-of-Things (JALEEL et al., 2022)	2022	Improvement of data considering optimization, security, and syntax, semantics, and protocol aspects	IoT devices in general, presenting an use case in MIOT area	Use of broker plugins and converter plugins. Use of a "*[...] request queue and a pool of worker threads to handle multiple requests.*" Use of a conventional time-out-based invalidation for devices that are absent. Use of cache of result devices and request queue of devices. Optimization in data conversion by avoiding redundant computations
Big-ECG: Cardiographic Predictive Cyber-Physical System for Stroke Management (HUSSAIN; PARK, 2021)	2021	Ontology	ECG sensor device	Use of ActiveMQ, ElasticSearch, Hadoop ecosystem, MariaDB
Detecting Situations with Stream Reasoning on Health Data Obtained with IoT (BOURGAIS; GIUSTOZZI; VERCOUTER, 2021)	2021	Use an ontology for data enrichment. Definition of constraints and situations with basis in health data read	Medical IoT devices in general	Not covered
Edge-Based Hybrid System Implementation for Long-Range Safety and Healthcare IoT Applications (WU et al., 2021)	2021	An initial filtering function that "[...] *removes any unwanted wireless data, such as duplicate data from the same network router or IoT devices.*"	IoT devices in general	Use of MongoDB database. There are two levels of storage data: Level 1 and level 2. The level 1 data contain all the raw data received from the wireless sensor nodes, which is stored in the original format for future data recovery purposes, and the level 2 data storage contains all the processed data.
Integration of IoT and Fog Computing in Healthcare Based the Smart Intensive Units (MUDAWI, 2022)	2022	Not covered	Medical IoT devices.	Use of fog nodes and an algorithm for using them
IoT Powered Cancer Observation System (REHMAN et al., 2021)	2021	Not covered	Temperature sensor, pulse sensor, enzymatic sensor and immunosensor	Three-layers architecture: Perception, Network, and Application
Measuring Quantitative Situated User Experience with a Mobile Galvanic Skin Response Sensor (SANTANA; OTANI, 2021)	2021	Conversion of text into audio content	Mobile Galvanic Skin Response sensor	Not covered
Simulation-Driven Platform for Edge-Based AAL Systems (ALOI et al., 2021)	2021	Not covered	IoT devices in general, presenting two use cases in health area	MW2MW Integration

Source: Authorial.

4 HEALTHTRANSLATOR

This chapter is responsible for describing the HealthTranslator model, which is a healthcare-oriented API focused on reading vital signs collected by different IoT devices and providing the data in a single format using the HL7 FHIR healthcare file standard, ensuring interoperability between different systems, and doing a pre-diagnosis of the health status of the user based on the vitals. With this model, the players, like Medical applications, academic projects, and personal studies that need to work with data collected by WebAPIs would benefit from not having to adapt to the different ways of connecting to WebAPIs offered by the responsible companies, and would rely on an approach that pre-diagnoses the health status mentioned. So, HealthTranslator acts between the WebAPIs of the companies that develop vital signs collection devices and the software and applications that need to use the vital signs that these devices have collected, communicating between the two parties. Section 4.1 presents the design decisions for the realization of the model proposed by this paper. In Section 4.2, the architecture of HealthTranslator is presented. Section 4.3 defines the IoT devices and applications supported by HealthTranslator. Section 4.4 shows the operation and evaluation metrics of the proposed model. Section 4.5 describes the evaluation methodology. Section 4.6 addresses the preliminary implementation and evaluation. Finally, Section 4.7 performs partial considerations on the model.

4.1 Project Decisions

Different applications in computer science can operate with vital signs collected by multiple IoT devices by fitting only the file format provided by the HealthTranslator, considering its proposal. Hence, applications would not have the need to adapt to the different file types and formats provided by the IoT device companies, as well as to the several APIs these companies provide for data collection. HealthTranslator analyzes the consulted vital signs data and writes an observation of the health of the users based on their vitals. In addition, the HealthTranslator API performs a smart reading of the data collected by the devices, establishing vital sign reading periodicities aligned with healthcare guidelines. The reading periodicities are established considering the vital signs values. Considering these points, the HealthTranslator API aims to answer the following sentences:

1. What technologies are needed to implement the conversion of different vital signs data into a single format and type?

2. What format and what type of file should the API output have?

3. How can a smart vital signs reading be performed in a way that considers the individual characteristics of each user's vital signs?

4. Which IoT devices can be read by an API that provides the vital signs data in a single

format?

5. Which systems can benefit from an API that provides vital signs data in a single format?

6. How can a model analyzes and writes observations of the health of the wearable users based on the vital signs data read?

To develop an API that aims to convert files of different types and formats into a file with a single format and type, it is necessary to understand what technologies and techniques can be used for this reason and why. Following this reasoning, it is important to understand in what format the output file needs to be and of what type it needs to be, aiming at the best adaptation for multiple systems and approaches. Considering these points, it is also needed to understand which IoT devices allow the reading of vitals to other applications, how they do this, and also what data is made available. The systems that will benefit from the model proposed by this paper also need to be known for clarifying the impact that HealthTranslator has on computer science, as well as highlighting the importance of the proposed model. In addition, it is also necessary to establish rules for the periodicity of reading vital signs, so that it is appropriate to the individual needs of the users, improving the effectiveness of the API.

To the development of the model, and to answer the questions listed, 5 project decisions were made. The following list presents them:

1. The IoT devices covered by HealthTranslator must be wearable devices, like smart bracelets and smartwatches, from Amazfit, Fitbit, Garmin, Samsung, and Xiaomi.

2. The vital signs considered by the proposed model are: Body temperature, heart rate, heart rate variability, oxygen saturation, and respiration rate.

3. The output file must be in HL7 FHIR standard.

4. The output file must be of XML or JSON type.

5. The rules for the periodicity of vital sign readings need to consider the cases where all vital signs are normal and also the cases where there are variations, with appropriate reading patterns for the combinations of values, as needed.

6. The output file must have an observation about the health of the users based on the vital signs data read.

The established project decisions aim to meet the objective of this work, to answer the research question, to answer the questions listed in this section, and to ensure the efficiency and effectiveness of the proposed model. The choice for IoT devices from the mentioned companies was done because they offer solutions for collecting vital signs by third-party implementations, because of their great influence in the current device market, and because they have different selling prices in the market, being accessible to people with different rents, and then, more IoT device users are benefited. The first version of HealthTranslator consider to work with the data collected from the WEB APIs of the companies, since all of the companies use WEB APIs and not all use Device APIs or something similar. The cited vital signs were selected due to the fact that these are vital signs that most devices covered by this document, in common, capture and provide. For the output file, the HL7 FHIR was chosen due to the worldwide recognition of this file standardization in healthcare, and XML and JSON types were elected in order to provide the output file to systems that demand both higher security and faster processing speed. So, with these format and type selected, more systems and approaches are benefited. Moreover, with data reading being performed in a smart way, the service to users is more assertive, and there is greater accuracy in the model's use of resources, as unnecessary vital sign readings are avoided.

4.2 Architecture

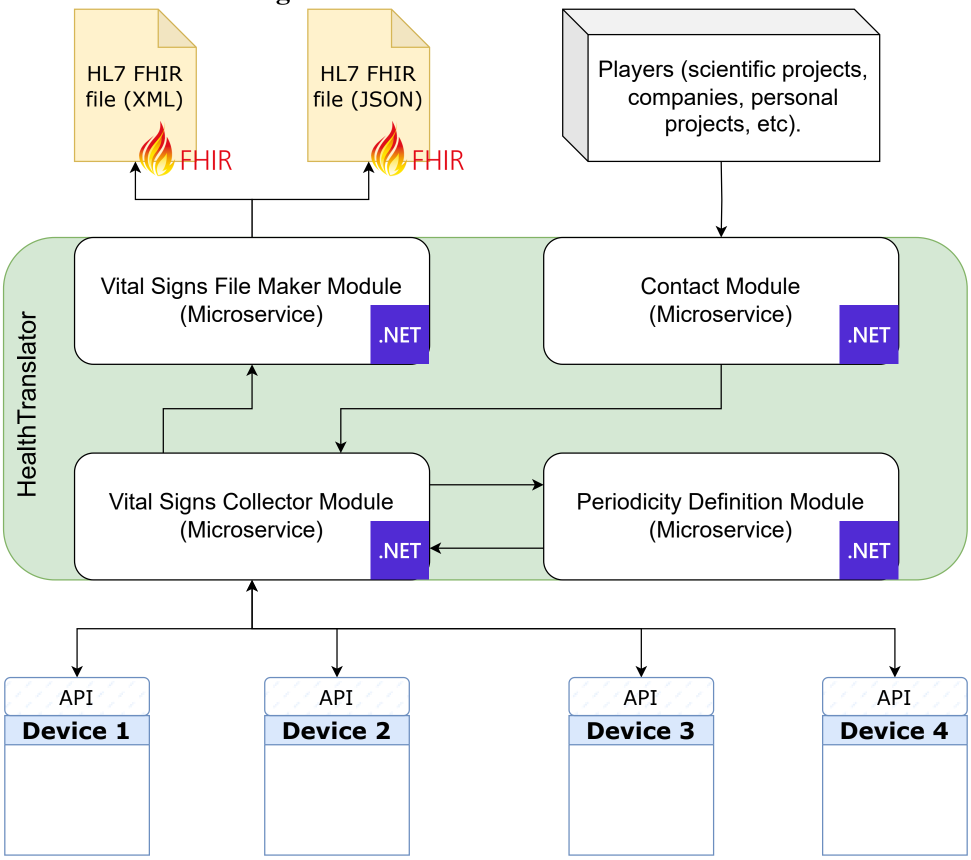
The HealthTranslator can be understood as a middleware, responsible for facilitating the collection of vital signs from IoT devices to other systems, ensuring the interoperability between them. The interface, therefore, acts on the Edge Controller. The model is composed of four microservices: The Contact Module, The Periodicity Definition Module, The Vital Signs Collector Module, and the Vital Signs File Maker Module. Figure 1 presents the architecture of the model proposed.

All microservices have been developed using the .NET framework, aiming at the facilities for the development of microservices that the framework offers, and using the set of Domain Driven Design (DDD) principles for the organization of the code, privileging the existence of a business layer, making the system modeling based on the reality of the business (MICROSOFT, 2022). The microservices communicate with each other by sending JSON files, establishing HTTP connections. Each microservice has access to its own database tables, which was implemented using SQL Server because it is a relational database, since this category of database is used for cases where it is necessary for data to be managed in a consistent and secure way (MICROSOFT, 2023), and strong compatibility with the .NET framework. The generation of output files relied on the use of FHIR standardization and JSON format, as it is a highly used format in computing, which facilitates the integration of HealthTranslator with other applications.

4.3 Operation

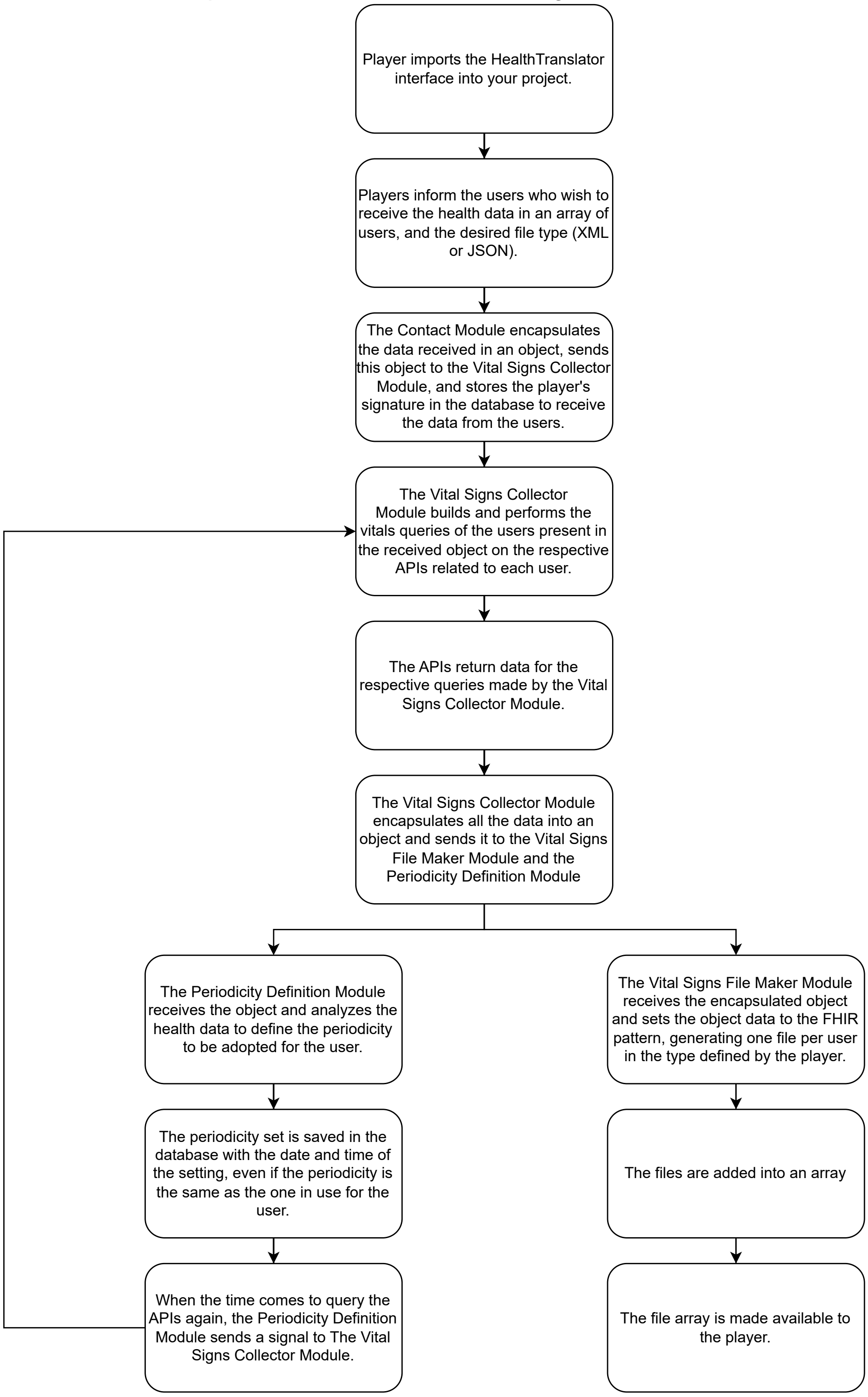
HealthTranslator begins its operation, that can be fully observed in Figure 4, by receiving a solicitation, from a player, for consulting the vital signs collected by one or more IoT devices, specifying the type of file desired, between XML and JSON. The player therefore subscribes to receive the health data of certain patients, which represents a Publish/subscribe architecture. Then, a signal is sent to the Vital Signs Collector Module, the module that consults the APIs of

Figure 1: HealthTranslator Architecture



Source: Authorial

Figure 2: HealthTranslator Working Flowchart



Source: Authorial

the companies' devices considered by this document for the collection of health data captured by such devices. The standard periodicity for checking vital signs is every six hours, considering what the Australian Commission on Safety and Quality in Health Care (2022) defends. The Vital Signs Collector Module consults the APIs, and then, this module sends the data to two other modules. The first one is the Vital Signs File Maker Module, which is responsible for analyzing the vitals, writing an observation about the vitals, and finally creating a file with the FHIR pattern for these data, being this file of XML or JSON type, as requested by the application user of the API. It is important to mention that here, with this analyzing work, the data enrichment technique is performed. It is also relevant to mention that, for focusing on the main scientific contribution of the present dissertation, the first version of HealthTranslator provides just one endpoint to the player, where they can subscribes for receiving the files of specific users and finally receive these files. The other module is the Periodicity Definition Module, which is responsible for determining the periodicity of API consultation by the Vital Signs Collector Module. It is important to mention that if the player wants to stop the subscription for receiving the vital signs data of a specific patient, the player can use the stop function provided by the HealthTranslator, and then, the Contact Module will register the cancellation in the database.

The Vital Signs Collector Module, as previously mentioned, consults each of the different APIs of the IoT devices for the collection of vital signs. This module will query the APIs after receiving an authorization signal for this action, sent by the Periodicity Definition Module. When it is the moment to take the reading, the Periodicity Definition Module, which will be discussed later, sends a signal to the Vital Signs Collector Module, which will consult the APIs. In general, the APIs use the REST architecture for making the data available, and the JSON format for it. Therefore, a JSON object will be instantiated for each data read from the API, being the first representation of the data after reading. The object will be composed of properties to cover the reading of the following vital signs: body temperature, heart rate, oxygen saturation, and respiration rate. The values that will compose this object considers, for body temperature, oxygen saturation, and respiration rate, the respective values obtained when querying the APIs, and for the heart rate, an array of the values obtained throughout the periodicity being used by the Vital Signs Collection Module. Thus, if considered a periodicity of one reading every six hours, the array of values refers to the heart rate recorded within this time interval. This object will be sent to the Periodicity Definition Module and Vital Signs File Maker Module. Although the HealthTranslator definition considers the use of Web APIs to consult vital signs, the implementation of the model, in order to focus on its main scientific objective, which is the translation process, consults the data in a dataset, containing the data of one user that has a Fitbit device and a Garmin device.

The Periodicity Definition Module, as mentioned, considers, by standard, a consultation periodicity of every six hours. This module, upon receiving the vital signs data sent by the Vital Signs Collector Module, checks if the values are within the normality of vital signs, considering the values understood as normal for human health, to determine the periodicity of vital signs consultation to be used by the Vital Signs Collector Module. This determination is based on the guidelines used by scientific articles related to health, by vital signs collection rules used by hospitals, and by governmental bodies and health agencies around the world. An example of the periodicity used is to check vital signs twice an hour when the user has a fever. In cases like this, the periodicity of checking every 30 minutes is determined by the Periodicity Definition Module. If a new periodicity established is different from the one being used, the recording of this value and the date and time of the recording is made in the database connected to the module. If there is no change in periodicity from what is already in use, a new record, containing the same periodicity being used and a new date and time will be added. After defining a new periodicity, when it is the moment of taking a new read in the APIs, a signal is sent to the Vital Signs Collector Module. The work considers the periodicity shown in Table 1 in subsection 2.3.

The Vital Signs File Maker Module, the final module operated by HealthTranslator, receives the data sent by the Vital Signs Collector Module and inserts it in a file that has the FHIR standardization. The file generated by the module can be of XML or JSON type, according to the request made by the user application. The files are released to the user application as it requests them, considering that this request is received in the Contact Module. Vital Signs File Make Module is responsible for applying the data enrichment technique, the principal scientific contribution of the present dissertation, in an analysis of the vital signs, producing observations on them. The Figure 3 presents an example of FHIR Observation file that contains an observation saying that the vitals of the user are normal. This file is the principal practical representation of the application of the data enrichment technique that this work proposes. Without the technique, the player would receive only the raw data requested, receiving, for example, in the case of a request for a user's body temperature, only this data and the date of collection, as shown in figure x. The Table 3 shows the observations produced by this module. These observations have been written based on (ROHMETRA et al., 2021), the characteristics of the vital signs themselves (for example, "Low oxygen saturation" when the oxygen saturation is below 95), and the technical interviews with a healthcare employee of Regina Hospital of Novo Hamburgo. The Algorithm 1 presents the code responsible for generating the observation. The first version uses just the JSON format for focusing on the main scientific objective of this dissertation.

The HealthTranslator is projected to support the devices from the companies mentioned in section 2.1, those being: Amazfit, Fitbit, Garmin, Samsung, and Xiaomi. Future versions of the API proposed by this paper will have a graphical interface that will allow users to configure HealthTranslator to read APIs not considered by it.

4.3.1 Use case

An example of a use case that could be mentioned is monitoring many patients who have a specific illness. For example, suppose that a group of 20 patients has the flu, and a player is testing the effects of a medication on this group. It is also assumed that these patients have vital

Algorithm 1 Algorithm implemented for writing an observation on the vitals of the user

Input: *VitalSignsToBeProcessed*

```
while VitalSignsToBeProcessed do
    if BodyTemperature < 36.2 then
        Observation ← Observation + "Low temperature."
    end if
    if BodyTemperature > 37.1 then
        Observation ← Observation + "Fever."
    end if
    if HeartRate < 60 then
        Observation ← Observation + "Bradycardia."
    end if
    if HeartRate > 100 then
        Observation ← Observation + "Tachycardia."
    end if
    if OxygenSaturation < 95 then
        Observation ← Observation + "Low oxygen saturation."
    end if
    if RespiratoryRate < 12 then
        Observation ← Observation + "Low respiratory rate."
    end if
    if RespiratoryRate > 20 then
        Observation ← Observation + "High respiratory rate."
    end if
    if RespiratoryRate < 12 and HeartRate < 60 then
        Observation ← Observation + "Possible hemorrhage."
    end if
    if BodyTemperature > 37.1 and HeartRate > 100 then
        Observation ← Observation + "Possible generalized infection."
    end if
    if All vitals are normal then
        Observation ← "Normal vital signs."
    end if
    FHIRFiles ← "GenerateFHIRFiles()"
end while
     return FHIRFiles
```

Figure 3: FHIR Observation File Generated by the Vital Signs File Maker Module


```
{
    "resourceType": "Observation",
    "id": "Observation",
    "meta": {
        "profile": ["http://hl7.org/fhir/StructureDefinition/Observation"]
    },
    "status": "final",
    "code": {
        "coding": [{
                "system": "http://loinc.org",
                "code": "48767-8",
                "display": "Annotation comment [Interpretation] Narrative"
            }
        ],
        "text": "Observation Note"
    },
    "subject": {
        "reference": "Patient/0f5974d2-264b-408d-8602-188babd139a0",
        "display": null
    },
    "valueString": "Normal vital signs."
}
```


Source: Autoral

signs collection devices of different brands. For this monitoring, instead of the player adapting to the different Web APIs provided by the companies that manufacture the patients' devices, it would just adapt to HealthTranslator, and over time it would receive the files with the vital signs, already having a pre-diagnosis of how the patient is feeling, which could facilitate the reading and interpretation of the vital signs.

Table 3: Observations produced by the Vital Signs File Maker Module

Vital signs	**Value**	**Observation**
Body temperature	$x < 36.2$	Low temperature
Body temperature	$x > 37.1$	Fever
Heart rate	$x < 60$	Bradycardia
Heart rate	$x > 100$	Tachycardia
Oxygen saturation	$x < 95$	Low oxygen saturation
Respiratory rate	$x < 12$	Low respiratory rate
Respiratory rate	$x > 20$	High respiratory rate
Heart rate (x) and respiratory rate (y)	$x < 60$ and $y < 12$	Possible hemorrhage
Heart rate (x) and body temperature (y)	$x > 100$ and $y > 37.1$	Possible generalized infection
All	All normal	Normal vital signs

Source: Authorial

Figure 4: Example of body temperature collected on Fitbit WebAPI


```
{
    "tempCore": [
        {
            "dateTime": "2020-06-18T10:00:00",
            "value": 37.5,
        },
        {
            "dateTime": "2020-06-18T12:10:00",
            "value": 38.1,
        },

    ]
}
```


Source: FITBIT. Get Temperature (Core) Summary by Date. Available in: <https://dev.fitbit.com/build/reference/web-api/temperature/get-temperature-core-summary-by-date> Accessed in 30 Jul 2023

5 EVALUATION METHODOLOGY

The present chapter has the objective of presenting the evaluation methodology employed to evaluate the model proposed by this document. The metrics defined evaluate the application's performance and efficiency with respect to resource utilization. The following sections detail the evaluation application methodology. It is relevant to emphasize that, among the related works, no standard was identified for the making of model tests such as the one being addressed by this work, so, for the tests, a new methodology was developed. The dataset used was created considering the vital signs covered by this dissertation, collected by the covered devices. For these tests, a computer with a 11th Gen Intel(R) Core(TM) i7-11390H @ 3.40GHz 2.92 GHz processor, and 16,0 GB of RAM Memory has been used.

5.1 Initial Prototype

The initial prototype of the HealthTranslator model will feature all the architecture described throughout the article. The initial version of the application will support only the devices belonging to the companies listed in the article, communicating with the WEB APIs of these devices. It is important to highlight that, according to what was mentioned in the architecture section, to focus on the main scientific objective of this paper, the implementation of the initial prototype uses sheets with the vitals as the data source, simulating the data collected by the WEB APIs. In addition, for the same reason, the first version uses just the JSON format, provides just one endpoint to the player, where they can subscribe for receiving the files of specific users and finally receive these files, and consider the querying for the 4 vitals covered by this dissertation. The initial prototype considers what is presented in Figure 1 with the considerations here addressed.

5.2 Load Test Data

The initial test load will be established by the use of a dataset. This dataset was made considering the vital signs covered by the present work. In this set, we considered the data of a fictitious user with wearables from two different companies. In the dataset, this user, in the interval of 23 hours, presented moments when their vital signs were normal and when they were with values different from normality, aiming to test the generation of observations for different cases.

5.3 Evaluation Scenarios

The test coverage considered the implementation of unit tests, in the case of the Vital Signs File Maker Module, and functional tests, which are execution tests of the models, for all modules. In the functional tests, the individual execution of each module was considered, and in the unit tests applied in the aforementioned module, the construction of scenarios aimed at thoroughly evaluating the generation of observations on vital signs was considered. All this was done with the main purpose of evaluating the accuracy of the HealthTranslator, especially of the Vital Signs File Maker Module, the module that contains the principal scientific contribution of this work. It is important to mention that "accuracy" and "precision", in this work, mean the ability to correctly establish the pre-diagnosis according to the user's data (for example, if the heartbeat is less than 60 BPM and the breathing rate is less than 12 breaths per minute, the model must be accurate in pre-diagnosing the user and establish that they have a possible hemorrhage). Since it is desired to test the precision of this module, unit tests are perfect for that, because of their characteristic of testing the functionality of a unit, as the name suggests. It is important to mention that, according to Bai, Smith e Stolee (2021), "*Unit testing is widely practiced in industry, and the ACM suggests that software testing should be integrated into Computer Science and Software Engineering curricula*". The test cases implemented for testing the Vital Signs File Maker Module were the following ones:

- Scenario 1: Make observation with normal values should set observation to "Normal vital signs".
- Scenario 2: Make observation with low body temperature should set observation to "Low temperature".
- Scenario 3: Make observation with high body temperature should set observation to "Fever".
- Scenario 4: Make observation with low respiratory rate and low heart rate should set observation to "Possible hemorrhage".
- Scenario 5: Make observation with low heart rate should set observation to "Bradycardia".
- Scenario 6: Make observation with high heart rate should set observation to "Tachycardia".
- Scenario 7: Make observation with low oxygen saturation should set observation to "Low oxygen saturation".
- Scenario 8: Make observation with low respiratory rate should set observation to "Low respiratory rate".
- Scenario 9: Make observation with high respiratory rate should set observation to "High respiratory rate".
- Scenario 10: Make observation with multiple conditions should set observation in accordance with the characteristics of these conditions.

- Scenario 11: Make observation with multiple users and WebAPIs should set observation for each user.

6 RESULTS

This chapter has the objective of discussing the results that the HealthTranslator model presented after the execution of tests. As mentioned, the tests performed aimed to validate the accuracy of the model, with emphasis on the Vital Signs File Maker Module, where the main scientific contribution of the work is present. This chapter, then, discusses the results of the application of the functional and unit tests and what can be performed with a future implementation of all the features presented by this dissertation.

6.1 Individual Tests of the Modules

The HealthTranslator model performed satisfactorily in terms of its accuracy, respecting all the criteria established for each module. The functional and unit tests showed that, in terms of accuracy, the HealthTranslator is suitable for communicating between WebAPIs of mobile devices that collect vital signs with software and other technologies that need to use such vital signs. The following subsections address the results of the individual tests for each module.

6.1.1 Individual Tests of The Contact Module

The Contact Module appropriately received subscription requests from new users and recorded these requests in the database. Subsequently, a signal was duly passed to the Vital Signs Collector Module, which started the data collection process. Although the initial version of the HealthTranslator does not work with the generation of XML files, the Contact Module was able to receive the request for this file format, as expected. The system's ability to handle diverse data formats and seamlessly integrate with other modules showcases its potential for future enhancements and compatibility with evolving technologies.

The execution time of the functional test of this module was less than 3 seconds considering a scenario with 2 wearable users and 2 WebAPIs, reflecting the efficient performance of the system for this case. Nevertheless, as HealthTranslator moves towards actual integration with WebAPIs, it is crucial to perform scalability tests to ensure a higher level of precision regarding response times. The scalability tests will help assess the system's capacity to work with a higher volume of interactions with WebAPIs, allowing for optimization and adjustment of its performance to improve the user experience.

6.1.2 Individual Tests of The Vital Signs Collector Module

The Vital Signs Collector Module consulted with success the vital signs dataset that represented the WebAPIs. Once this module receives a signal sent by the Contact Module or the Periodicity Calculator Module, it consults the dataset and then provides the data for the Periodicity Calculator Module and the Vital Signs File Maker Module. It was noted that using the "Newtonsoft.Json" library was important for precise work converting the data received to the correct object format, performing this task quickly. The data have been correctly saved in the database after the process.

6.1.3 Individual Tests of The Periodicity Calculator Module

The Periodicity Calculator module interacted properly with the Vital Signs Collector Module, successfully calculating the consultation period based on the vital signs, according to the predefined idea. Using the polling technique, the module efficiently retrieved the information from the database, ensuring accurate timing for subsequent queries. It is important to mention that the processing time for individual requests remained consistently below 3 seconds considering a scenario with 2 wearable users and 2 WebAPIs. So, the functional tests showed that the system correctly considered the parameterizations communicated throughout this dissertation for collecting the vitals.

However, in a future scenario, unit tests will have to be performed to ensure the effectiveness of this calculation for more diverse scenarios, as well as scalability tests when HealthTranslator is tested in a scenario closer to the real one. The tests that were conducted on HealthTranslator focused on verifying the effectiveness of the generation of observations on the patient's health status, an action performed in the Vital Signs File Maker Module, so some scenarios with specific combinations of vital signs will need to be considered in future tests, considering tests closer to a real scenario.

6.1.4 Individual Tests of The Vital Signs File Maker Module

Finally, the Vital Signs File Maker Module was tested with unit tests and with functional testing. It was noted that the module correctly generated the observations according to the user's vital signs, which indicates the success of the application of the data enrichment technique. Six objects have been generated in FHIR format, using the "Patient" and "Observation" resources, as expected. The processing time of this module was, in general, like the others, considering a scenario with 2 wearable users and 2 WebAPIs, less than 3 seconds.

The FHIR standardization was correctly used in the observation files, which ensures better compatibility of HealthTranslator with other healthcare applications. The standardization relied on the use of "Patient" and "Observation" resources, representing the user's data, vital signs, and the observation of the health status. In addition, in the future, scalability tests should be conducted in this module, for the same reasons presented in the previous subsections.

6.2 Discussion

The tests performed on HealthTranslator show that the model fulfills the main objective of writing user pre-diagnoses based on the reading and interpretation of vital signs, delivering these pre-diagnoses to the player together with the vital signs. The collection periodicity was properly calculated, considering the reading of the vital signs of each user and adapting the periodicity to the characteristics of the vital signs read.

Using unit tests in order to test the Vital Signs File Maker Module is of paramount importance in ensuring the reliability and precision of the system. Vital signs play a critical role in assessing a user's health status, and any errors in their measurement could lead to potentially dangerous consequences. Unit tests help validate the functionality of the module by validating each individual component and its behavior, thereby identifying bugs or discrepancies early in the development process. Then, the use of this kind of test was very important in this manner.

It is understood that, with the development of all the features established by this dissertation, going beyond what the initial prototype implements, it will be necessary to implement tests to cover more aspects of HealthTranslator, such as performance and scalability tests in a scenario that considers more users, as well as tests that consider the use of WebAPIs, rather than a dataset. The present scenarios addressed by the dissertation aimed, as mentioned, to evaluate the accuracy of the model to favor the validation of the new idea being inserted in computer science, which is the use of the data enrichment technique to establish pre-diagnoses about the user's health status based on the reading of vital signs.

6.3 How are the project decision questions answered?

This subsection is intended to provide an overview of how this dissertation answers the questions of project decisions. Regarding the technologies needed to implement the conversion of different vital signs into a singular format and type, this dissertation presented that the use of a microservices-based architecture and the use of the FHIR standardization can perform this action. The data processing with the use of the mentioned standardization can gather the data in a singular format, which can facilitate the integration of HealthTranslator with healthcare applications. In addition, the use of JSON format contributes to answering this question as well, besides contributing to the mentioned integration. The use of the FHIR standardization and JSON type also answers the question of what format and type of file should the API output have.

The question of how can a smart vital signs reading be performed in a way that considers the individual characteristics of each user's vital signs, this dissertation considers a periodicity calculation based on the vitals of the users, stipulating the rules for consulting the WebAPIs with basis in parameters defined by the science and parameters used in a hospital. At this form, the users of whom the players want to consult the vital signs will have contemplated according to their respective health states.

The IoT devices that can be read by an API that provides vital signs in a unique format are IoT devices whose companies provide APIs that are compatible with external applications. For example, Garmin's API is compatible with external applications, so devices covered by Garmin's WebAPI can be covered by a model such as HealthTranslator. Also, regarding the systems that can benefit from a model that provides vital signs data in a unique format, as addressed throughout this dissertation, players like Medical applications, academic projects, and personal studies that need to work with data collected by WebAPIs can benefit.

Finally, about how a model can analyze and write observations about the health of wearable device users based on the vital signs read, this action can be done based on the intelligent reading of vital signs, considering the values of these signs and making the readings as needed, and taking into account the parameterizations that science and hospitals use to diagnose users, defining, for example, when they may be presenting an internal infection or not based on the values of vital signs. With the intelligent reading of vital signs and subsequent analysis, observations can be written considering a periodicity adapted for each user, using file standardizations widely used by science, such as the FHIR standardization.

7 CONCLUSION

The present dissertation presents an integration model that ensures interoperability between IoT devices from the companies Amazfit, Fitbit, Garmin, Samsung, and XIAOMI, that use WebAPIs, and information systems. This work implements the use of the data enrichment technique for writing a description of the status health of the user based on their vital signs. Using the FHIR standard and the XML and JSON types, the HealthTranslator has the important feature of the mentioned interoperability, since, given the mentioned standard and file types, a large number of health applications around the world will be able to have easier communication with several IoT devices, benefiting from the proposed model. In this way, HealthTranslator answers the research question.

7.1 Contributions

The HealthTranslator model introduces in computer science the technique of data enrichment in the context of health in communication models between devices that collect vital signs and technologies that make use of these signals. Differently from the related work, this dissertation performs the generation of observations on vital signs, which can help players establish studies on the health of patients being observed using the HealthTranslator. Allied to this point, the use of the FHIR standardization contributes to the facilitation of communication between the proposed model and the players, given the wide use of this standardization on a global scale. So, this dissertation is ready to help people on a large scale.

The present dissertation identified that, with basis on papers related to the theme of this document, the current literature presents an absence of a method that uses the data enrichment technique in order to help the players conduct studies about the patients being monitored, which represents a gap in the computer science. This dissertation concludes this gap in science. In addition, this work smartly determines the periodicity of vital sign readings according to the values that have already been obtained, seeking, in this way, to establish an individual reading periodicity for each IoT device user. Moreover, the research that was conducted for this paper indicates the FHIR health file standardization which can facilitate the integration with the players, according to what was mentioned before, and this point has been also addressed by the present document. This paper promotes two main contributions to computer science: The first contribution is to provide a communication model between IoT devices and health systems and technologies that uses the data enrichment technique to provide observations on the vitals, considering the characteristics of the data collected from each user; and the second one is facilitating the collection of vital signs considering an environment with different data collection devices, providing a file in a unique format and type. The first contribution was realized with the development of the Vital Signs File Maker Module, and the second one with the connection to the APIs of the devices, with the use of the FHIR standardization for the output file, and with the XML and JSON type for this file, which can be chosen by the user of the model.

7.2 Technology Transfer

The HealthTranslator has the potential to support academic research, medical implementations, and personal projects in the collection of users' vital signs, since, in addition to delivering the vital signs requested by the player, it performs a pre-diagnosis on the users' health status. In addition, the model proposed here performs an intelligent reading of users' vital signs, considering their health status, which configures the HealthTranslator as a model that collects data according to the individual needs of each user, increasing the reliability of the model. Furthermore, by using the FHIR standardization, the proposed model can easily work together with many applications on a global scale. All these factors contribute to HealthTranslator being seen as a model capable of supporting the scientific and commercial field on a large scale, improving people's quality of life.

7.3 Future Work

The following list presents the future works related to HealthTranslator:

- Implement a specific function where the user can manage the subscriptions for users.
- Provide the final files in other standards besides FHIR.
- Provide the final files in other types besides XML and JSON.
- Make HealthTranslator compatible with devices from other companies.

In future work, a specific function where the user can manage the subscriptions for users can be implemented, separating the subscription feature from the data-receiving feature. Moreover, another future work would be to make the files available in other standards besides FHIR, aiming to increase the number of players that would benefit from HealthTranslator. In addition, an important advance for the HealthTranslator would be to make it compatible with devices from other companies, and to do this, a graphical interface could be developed where the player could configure the device to be considered by the model, using parameters made available in this graphical interface. In this way, the HealthTranslator would behave dynamically in terms of its ability to use WebAPIs, and would not be restricted to just the WebAPIs considered in this work. Finally, another improvement, also present in the availability of files, concerns providing them in other types besides XML and JSON, also aiming to expand the number of players benefited.

REFERENCES

AJAY, S. **Checking your temperature**: here's what you need to know. Available at: <https://healthcare.utah.edu/healthfeed/postings/2020/04/temperature-screening.php>. Accessed in: 14 November 2022.

ALOI, G.; FORTINO, G.; GRAVINA, R.; PACE, P.; SAVAGLIO, C. Simulation-Driven Platform for Edge-Based AAL Systems. **IEEE Journal on Selected Areas in Communications**, [S.l.], v. 39, n. 2, p. 446–462, Feb 2021.

ANDROID. **Documentation**. Available at: <https://developer.android.com/guide/health-and-fitness/health-connect>. Accessed in: 13 November 2022.

AYAZ, M.; PASHA, M. F.; ALZAHRANI, M. Y.; BUDIARTO, R.; STIAWAN, D. The Fast Health Interoperability Resources (FHIR) Standard: systematic literature review of implementations, applications, challenges and opportunities. **JMIR Med Inform**, [S.l.], v. 9, n. 7, p. e21929, Jul 2021.

BAI, G. R.; SMITH, J.; STOLEE, K. T. How Students Unit Test: perceptions, practices, and pitfalls. In: ACM CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION V. 1, 26., 2021, New York, NY, USA. **Proceedings...** Association for Computing Machinery, 2021. p. 248–254. (ITiCSE '21).

BERTA, R.; KOBEISSI, A.; BELLOTTI, F.; DE GLORIA, A. Atmosphere, an Open Source Measurement-Oriented Data Framework for IoT. **IEEE Transactions on Industrial Informatics**, [S.l.], v. 17, n. 3, p. 1927–1936, March 2021.

BOURGAIS, M.; GIUSTOZZI, F.; VERCOUTER, L. Detecting Situations with Stream Reasoning on Health Data Obtained with IoT. **Procedia Computer Science**, [S.l.], v. 192, p. 507–516, 2021. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021.

CLEVELAND CLINIC, T. **Thermometers**: how to take your temperature. Available at: <https://my.clevelandclinic.org/health/articles/9959-thermometers-how-to-take-your-temperature>. Accessed in: 14 November 2022.

DAS, A.; NAYEEM, Z.; FAYSAL, A. S.; HIMU, F. H.; SIAM, T. R. Health Monitoring IoT Device with Risk Prediction using Cloud Computing and Machine Learning. , [S.l.], p. 1–6, 2021.

FITBIT. **Developer**. Available at: <https://dev.fitbit.com/build/reference/>. Accessed in: 13 November 2022.

GARMIN. **Developer**. Available at: <https://developer.garmin.com/>. Accessed in: 13 November 2022.

GRIFFIN, A. C.; HE, L.; SUNJAYA, A. P.; KING, A. J.; KHAN, Z.; NWADIUGWU, M.; DOUTHIT, B.; SUBBIAN, V.; NGUYEN, V.; BRAUNSTEIN, M.; JAFFE, C.; SCHLEYER, T. Clinical, technical, and implementation characteristics of real-world health applications using FHIR. **JAMIA Open**, [S.l.], v. 5, n. 4, 10 2022. ooac077.

HL7. **HL7 International**. Available at: <http://www.hl7.org/index.cfm>. Accessed in: 19 July 2023.

HUSSAIN, I.; PARK, S. J. Big-ECG: cardiographic predictive cyber-physical system for stroke management. **IEEE Access**, [S.l.], v. 9, p. 123146–123164, 2021.

JALEEL, A.; MAHMOOD, T.; TAHIR, A.; ASLAM, S.; FAYYAZ, U. U. Autonomic interoperability manager: a service-oriented architecture for full-stack interoperability in the internet-of-things. **ICT Express**, [S.l.], v. 8, n. 4, p. 507–512, 2022.

LOINC. **Loinc**. Available at: <https://loinc.org/atlas/>. Accessed in: 19 July 2023.

MICROSOFT. **Design a DDD-oriented microservice**. Available at: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>. Accessed in: 25 July 2023.

MICROSOFT. **What is a relational database?** Available at: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-relational-database/examples>. Accessed in: 25 July 2023.

MUDAWI, N. A. Integration of IoT and Fog Computing in Healthcare Based the Smart Intensive Units. **IEEE Access**, [S.l.], v. 10, p. 59906–59918, 2022.

MUKATI, N.; NAMDEV, N.; DILIP, R.; HEMALATHA, N.; DHIMAN, V.; SAHU, B. Healthcare Assistance to COVID-19 Patient using Internet of Things (IoT) Enabled Technologies. **Materials Today: Proceedings**, [S.l.], v. 80, p. 3777–3781, 2023. SI:5 NANO 2021.

NANDA, S. K.; PANDA, S. K.; DASH, M. Medical supply chain integrated with blockchain and IoT to track the logistics of medical products. **Multimedia Tools and Applications**, [S.l.], Mar 2023.

PLASTIRAS, P.; O'SULLIVAN, D. Exchanging personal health data with electronic health records: a standardized information model for patient generated health data and observations of daily living. **International Journal of Medical Informatics**, [S.l.], v. 120, p. 116–125, 2018.

QADRI, Y. A.; NAUMAN, A.; ZIKRIA, Y. B.; VASILAKOS, A. V.; KIM, S. W. The Future of Healthcare Internet of Things: a survey of emerging technologies. **IEEE Communications Surveys Tutorials**, [S.l.], v. 22, n. 2, p. 1121–1167, 2020.

RAHMANI, A. M.; GIA, T. N.; NEGASH, B.; ANZANPOUR, A.; AZIMI, I.; JIANG, M.; LILJEBERG, P. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: a fog computing approach. **Future Generation Computer Systems**, [S.l.], v. 78, p. 641–658, 2018.

REHMAN, O.; FARRUKH, Z.; AL-BUSAIDI, A.; CHA, K.; PARK, S.; RAHMAN, I. IoT Powered Cancer Observation System. In: THE 9TH INTERNATIONAL CONFERENCE ON SMART MEDIA AND APPLICATIONS, 2021, New York, NY, USA. **Anais...** Association for Computing Machinery, 2021. p. 313–318. (SMA 2020).

ROHMETRA, H.; RAGHUNATH, N.; NARANG, P.; CHAMOLA, V.; GUIZANI, M.; LAKKANIGA, N. R. AI-enabled remote monitoring of vital signs for COVID-19: methods, prospects and challenges. **Computing**, [S.l.], Mar 2021.

SANTANA, V. F. de; OTANI, L. M. F. Measuring Quantitative Situated User Experience with a Mobile Galvanic Skin Response Sensor. In: XX BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2021, New York, NY, USA. **Proceedings...** Association for Computing Machinery, 2021. (IHC '21).

SARIPALLE, R.; RUNYAN, C.; RUSSELL, M. Using HL7 FHIR to achieve interoperability in patient health record. **Journal of Biomedical Informatics**, [S.l.], v. 94, p. 103188, 2019.

SNOMEDINTERNATIONAL. **Snomed International**. Available at: <https://www.snomed.org/>. Accessed in: 19 July 2023.

VORISEK, C. N.; LEHNE, M.; KLOPFENSTEIN, S. A. I.; MAYER, P. J.; BARTSCHKE, A.; HAESE, T.; THUN, S. Fast Healthcare Interoperability Resources (FHIR) for Interoperability in Health Research: systematic review. **JMIR Med Inform**, [S.l.], v. 10, n. 7, p. e35724, Jul 2022.

WU, F.; QIU, C.; WU, T.; YUCE, M. R. Edge-Based Hybrid System Implementation for Long-Range Safety and Healthcare IoT Applications. **IEEE Internet of Things Journal**, [S.l.], v. 8, n. 12, p. 9970–9980, June 2021.

XIAOMI. **Xiaomi Open Platform Documentation Center**. Available at: <https://dev.mi.com/docs/>. Accessed in: 14 November 2022.

ZEPPOS. **Development Documentation**. Available at: <https://docs.zepp.com/docs/intro/>. Accessed in: 13 November 2022.