INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Interoperability Between Information Systems of The Results Of Clinical Analysis And Electronic Record Of The Patient

Octávio Viana Jesus de Sousa

Master in Computer Science

Supervisor:
PhD, Carlos Eduardo Dias Coutinho, Assistant Professor
Iscte - Instituto Universitário de Lisboa

October, 2022

TECNOLOGIAS
E ARQUITETURA

Department of Information Sciences and Technologies

Interoperability Between Information Systems of The Results Of Clinical Analysis And Electronic Record Of The Patient

Octávio Viana Jesus de Sousa

Master in Computer Science

Supervisor:
PhD, Carlos Eduardo Dias Coutinho, Assistant Professor
Iscte - Instituto Universitário de Lisboa

October, 2022

Acknowledgment

First of all, to God, for giving me divine grace, life, giving me health and protecting me along this path.

My family for their understanding and patience in difficult times.

To the teachers of the Computer and Management course, namely Carlos Coutinho and Rúben Pereira.

Abstract

The issue of interoperability between institutions in underdeveloped countries always presents good opportunities for science to contribute to substantial improvements in real-world issues. Basic support systems, such as health institutions, are among those that can benefit most from scientific advances. This paper reports an analysis made on the lack of interoperability between health institutions, specifically with regard to the interaction between physicians' offices and clinical analysis laboratory. This research was validated with a real practical use case that is presented in this paper. In order to make the information stored in different computer applications of the health information system (HIS), more specifically in the Hospital Center Dr. Ayres de Menezes, in São Tomé and Príncipe, interoperable, two distinct applications were developed, Patient Management System (PMS) and Clinical Analysis Laboratory Management System (LMS) to implement interoperability between them. In this way, take advantage of the interoperability logic implemented between these two applications and later integrate other systems.

The PMS requests medical exams through your Medical Consultation panel. The second system, LMS receives the exam request and after the exams are processed and validated, the results are sent to the requesting system.

To make the interoperability service effective, the SOAP protocol was used, which allowed the exchange of information synchronously between these two applications, allowing for faster transactions of patients' pathological data, and greater confidentiality of this same information. And as the work progressed, we felt the need to create a functional prototype with REST to meet the cloud model proposal.

Keyword: Interoperability, Information System, Clinical Analysis

Resumo

A questão da interoperabilidade entre instituições em países subdesenvolvidos sempre apresenta boas oportunidades para a ciência contribuir para melhorias substanciais em questões do mundo real. Os sistemas de apoio básico, como as instituições de saúde, estão entre os que mais podem se beneficiar dos avanços científicos. Este trabalho relata uma análise feita sobre a falta de interoperabilidade entre instituições de saúde, especificamente no que diz respeito à interação entre consultórios médicos e laboratórios de análises clínicas. Esta pesquisa foi validada com um caso de uso prático real que é apresentado neste trabalho. De forma a tornar a informação armazenada nas diferentes aplicações informáticas do sistema nacional de saúde (SIS), mais concretamente no Centro Hospitalar Dr. Ayres de Menezes, em São Tomé e Príncipe, interoperáveis, foram desenvolvidas duas aplicações distintas, Sistema de Gestão de Pacientes (SGP) e Sistema de Gestão do Laboratório de Análises Clínicas (SGL) para implementar a interoperabilidade entre eles. Desta forma, aproveitar a lógica implementada entre estas duas aplicações e posteriormente integrar outros sistemas.

O SGP solicita exames médicos através do seu painel de Consulta Médica. O segundo sistema, o SGL, recebe a solicitação dos exames e após os exames serem processados e validados, os resultados são enviados ao sistema solicitante.

Para tornar o serviço de interoperabilidade efetivo, foi utilizado o protocolo SOAP, que permitiu a troca de informações de forma síncrona entre essas duas aplicações. Por outro lado, sentimos a necessidade de criar um protótipo funcional com REST para atender a proposta do modelo em nuvem.

Palavras-chave: Interoperabilidade, Sistema de informação, Análises Clinicas

Index

1 INTRODUCTION ... 1

1.1 MOTIVATION ... 2
1.2 RESEARCH QUESTION ... 2
1.3 HYPOTHESES ... 2
1.4 RESEARCH METHODOLOGY ... 3
1.5 APPLICATION OF THE INVESTIGATION METHOD ... 4
1.5.1 Problem Definition ... 4
1.5.2 Identification of Objectives ... 4
1.5.3 Development of model ... 4
1.5.4 Evaluation ... 5
1.5.5 Communication ... 5
1.6 REVIEW PROTOCOL ... 5
1.6.1 Criteria for Inclusion and Exclusion of papers ... 6
1.6.2 The databases used ... 6
1.6.3 Quality Criteria ... 6
1.6.4 Selection of studies ... 7
1.6.5 Analysis of the extracted data ... 7

2 LITERATURE REVIEW ... 8

2.1 INFORMATION SYSTEM ... 8
2.2 INTEROPERABILITY ... 8
2.3 INTEROPERABILITY STANDARDS AND PROTOCOLS ... 10
2.4 SOME INTEROPERABILITY BENEFITS ... 13
2.5 INTEROPERABILITY ISSUES ... 13
2.6 SOME INTEROPERABILITY CHALLENGES ... 13
2.7 DESCRIPTION OF SOME INTEROPERABILITY STANDARDS AND PROTOCOLS ... 14

3 DEVELOPMENT METHODOLOGY ... 15

3.1 BUSINESS MODELING ... 15
3.1.1 Business description ... 16
3.1.2 Patient ... 16
3.1.3 Employee ... 16
3.1.4 Doctor ... 16
3.1.5 Laboratory ... 17
3.1.6 Business actors ... 17
3.1.7 Diagram of business use cases ... 18
3.1.8 Activity diagram ... 19
3.1.9 Use case activity diagram request analysis ... 19
3.2 REQUIREMENTS ... 20
3.2.1 System Actors ... 21
3.2.2 Functional Requirements ... 21
3.2.3 Non-Functional Requirements ... 22
3.2.4 Appearance or external interface requirements ... 22
3.2.5 Usability requirements ... 22
3.2.6 Performance ... 22

3.2.7 *Portability* ... 22
3.2.8 *Security* ... 23
3.2.9 *Secure Socket Layer (SSL)* ... 23
3.2.10 *SQL Injection* ... 24
3.2.11 *Cryptography* ... 24
3.2.12 *System Use Case Model* ... 24
3.2.13 *Clinical Analyst use case diagram* ... 25
3.2.14 *General system use case diagram* ... 26
3.2.15 *Specification of the most relevant use cases for the problem domain* ... 26
3.2.16 *Package organization* ... 29
3.2.17 *Account Management* ... 29
3.2.18 *Exam Management* ... 29
3.2.19 *Medical Consultation Management* ... 30
3.3 ANALYSIS ... 30
3.3.1 *Analysis model* ... 30
3.3.2 *Interface Classes* ... 31
3.3.3 *Control Classes* ... 31
3.3.4 *Entity Classes* ... 31
3.4 DESIGN ... 33
3.4.1 *Sequence diagram request analysis* ... 34
3.4.2 *Class Diagram, PMS* ... 36
3.4.3 *System Class Diagram, Clinical Analysis Management* ... 36
3.5 IMPLEMENTATION ... 37
3.5.1 *Component Diagram* ... 37
3.5.2 *Deployment Diagram* ... 38
3.5.3 *The web service implementation* ... 38
3.5.4 *The Implementation of SOAP Web Services with PHP Language* ... 40
3.5.5 *Implementation of REST web services with PHP language* ... 41
3.5.6 *Advantages and Disadvantages of SOAP* ... 43
3.5.6.1 *Advantages of SOAP* ... 43
3.5.6.2 *Disadvantages of SOAP* ... 43
3.5.7 *Advantages and Disadvantages of REST* ... 44
3.5.7.1 *Advantages of REST* ... 44
3.5.7.2 *Disadvantages of REST* ... 44

4 PROJECT FEASIBILITY AND SUSTAINABILITY STUDY ... 45

4.1 PLANNING BY FUNCTION POINTS ... 45
4.2 TRANSACTION IDENTIFICATION ... 45
4.3 COUNTING FUNCTIONS BY COMPLEXITY ... 47
4.3.1 *External Entries* ... 47
4.3.2 *External Inquiries* ... 47
4.3.3 *External Outputs* ... 48
4.3.4 *Internal Logical Files* ... 48
4.3.5 *External Interface Files* ... 48
4.3.6 *Final Unadjusted Function Point Count* ... 48
4.3.7 *Calculation of adjusted function points* ... 49
4.3.8 *Estimated* ... 50
4.3.9 *Calculation of Effort and Time* ... 50
4.3.10 *Scale Factors* ... 50
4.3.11 *Scale Multipliers* ... 51

4.3.12 Summary of calculations 51
4.4 Tangible and Intangible Benefits 52
4.5 Cost-Benefit Analysis 52

5 PROPOSED CLOUD MODEL 53

5.1 Types of IaaS 53
5.2 Licensing type 53
5.3 Legal limitations 54
5.4 The institution already has a similar service in São Tomé and Príncipe 54
5.5 Architecture of a public IaaS 55
5.6 Comparison between Virtual and In-House Data Centers 55
5.7 Virtual Data Center Solution 56
5.8 In-House Data Center Solution 56
5.9 Benefits of the IaaS model: 58

6 CONCLUSION 59

6.1 Proposal for future work 60

Table index

Table 1.1: Inclusion and exclusion criteria ... 6
Table 1.2: The search results and the respective databases used. ... 7
Table 2.1: Synthesis of literature review, standards and protocol cited in the literature review ... 11
Table 2.2: Interoperability Benefits ... 13
Table 2.3: Interoperability Challenges ... 13
Table 2.4: The description of the technical terms mentioned ... 14
Table 3.1: Description of business actors ... 17
Table 3.2: Business Use Case Description ... 18
Table 3.3: Description of system actors. ... 21
Table 3.40: System Functional Requirements. ... 21
Table 3.5: Access Control Matrix ... 23
Table 3.6: Specification of the add user use case. ... 26
Table 3.7: Specification of the use case block user. ... 27
Table 3.8: Use case specification View the analysis requested ... 27
Table 3.9: Use case specification, View patient history. ... 27
Table 3.10: Use Case Specification, Send the analysis result. ... 27
Table 3.11: Use Case Specification, Consult patient. ... 27
Table 3.12: Use Case Specification, Print the result. ... 28
Table 3.13: Use Case Specification, Validation result. ... 28
Table 3.14: Use Case Specification, Process analysis. ... 28
Table 3.15: Use Case Specification, Request analysis. ... 28
Table 3.16: View the analysis Result ... 28
Table 3.17: Web service Specifications. ... 39
Table 3.18: Advantages of the SOAP protocol ... 43
Table 3.19: Some disadvantages of SOAP ... 43
Table 3.20: Some advantages of REST ... 44
Table 3.21: Some Disadvantages of REST ... 44
Table 4.1: Transactions. ... 46
Table 4.2: External Input Count ... 47
Table 4.3: External Query Count. ... 47
Table 4.4: External Output Count. ... 48
Table 4.5: Internal Logical File Count ... 48
Table 4.6: External Interface File Count. ... 48
Table 4.7: Final PFNA count. ... 48
Table 4.8: Adjustment Factor Calculation Table. ... 49
Table 4.9: Scale Factors ... 50
Table 4.10: Scale Multipliers. ... 51
Table 5.1: In-house data center cost description ... 56
Table 5.2: Total sum of in-house data center costs ... 57
Table 5.3: Some benefits of IaaS ... 58

Figure index

Figure 1.1: Phases of Design Science Research 3
Figure 1.2: Classification of the papers under analysis 6
Figure 1.3: Number of papers per year 7
Figure 2.1: Tthe four levels or types of interoperability 10
Figure 3.1: RUP Overview 15
Figure 3.2: Business Use Case Diagram 18
Figure 3.3:Use case activity diagram “Request analysis” 19
Figure 3.4: Use case activity diagram “Send analysis results” 20
Figure 3.5:Use Case Diagram of Lab Administrator System 25
Figure 3.6: Diagram of request analysis system use cases. 26
Figure 3.7:General system use case diagram 26
Figure 3.8:Account Control package. 29
Figure 3.9:Clinical analysis management package 29
Figure 3.10:Consultationy Management Package 30
Figure 3.11: Corresponding analysis class diagram, by use case Consult Patient 31
Figure 3.12: Analysis class diagram corresponding to the use case, process analysis. 32
Figure 3.13: Collaboration diagram corresponding to use case, Request exam. 32
Figure 3.14: Collaboration diagram corresponding to use case, Submit analysis results. 33
Figure 3.15: Activity diagram request analysis 35
Figure 3.16: Class Diagram, PMS 36
Figure 3.17: System Class Diagram, Clinical Analysis Management 36
Figure 3.18: Component diagram 37
Figure 3.19: Deployment diagram 38
Figure 3.20: SOAP server instantiation 40
Figure 3.21: SOAP client instantiation 40
Figure 3.22: Web service consumption using the SOAP protocol 41
Figure 3.23: Web service consumption using the REST protocol 42
Figure 5.1: Architecture of a public IaaS 55

List of abbreviations and acronyms

API	-	Application programming interface
CAPEX	-	Capital expenditure
CDA	-	Clinical document architecture
DICOM	-	Digital Imaging and Communications in Medicine
DL	-	Digital library
FCIL	-	The team support facilities
HIS	-	Health information system
HL7	-	Health level seven
ICT	-	Information and communications technology
IIIF	-	International image interoperability framework
IOT	-	Internet of Things
IS	-	Information system
JMS	-	Java Messaging Service
JSON	-	JavaScript Object Notation
Lang.	-	Language
LMS	-	Clinical Analysis Laboratory Management System
OAI-ORE	-	Open Archives Initiative Object Reuse and Exchange
OAI-PMH	-	Open Archives Initiative Protocol for Metadata Harvesting
OPEX	-	Operational expenditure
Org.	-	Organizational
P2P	-	Peer-to-Peer
PDIF	-	Platform difficulty
PERS	-	Personnel capability
PMS	-	Patient Management System
PREX	-	Personnel experience
R	-	Read
RCPX	-	Product reliability and complexity
RDF	-	Resource description framework
RED	-	Referenced Elementary Data
RLF	-	Referenced Logical Files
RSS	-	Rdf site summary
RUSE	-	The reuse required
RW	-	Read and write
SCED	-	Required schedule
SNOMED	-	Systematized Nomenclature of Medicine
SOA	-	Service oriented architecture
SOAP	-	Simple object access protocol
SoIS	-	Systems of information systems
SOS	-	Systems-of-systems
SRU	-	Search and retrieve url
SRW	-	Search/retrieve web service
SWORD	-	Simple web-service offering repository deposit
TCO	-	Total cost of ownership
Tech.	-	Technique
UC	-	Use cases
WSDL	-	Web service definition language
XML	-	Extensible markup language

1 INTRODUCTION

In recent years there has been an exponential growth of software to respond quickly and efficiently to various problems of organizations, especially with regard to data management. The health organizations such as hospitals, health centers and clinical analysis laboratories were also not far behind [1,7].

It happens, however, that the same organization uses heterogeneous software, built with different infrastructures, platforms and data formats [2], therefore, the synchronous sharing of this information becomes unfeasible and, consequently, leads to wasted time and data redundancy [3].

In the hospital center of São Tomé, Dr. Ayres de Menezes, there are some islands of software in some sections and in other sections there is no presence of any software, as is the case of the emergency service of patient care. The medical consultation management process is done in a traditional way (on paper), either for the registration, the patient's history and for the request of new clinical analysis.

Clinical analysis services are increasingly required, both by health technicians and by the users themselves who seek to monitor their health conditions. This fact reflects the importance that we must give to the automated and interoperable management of information, from the registration, collection of samples of biological products, and subsequent sending of the results to the user or the prescribing doctor [4, 5].

In this context, the objective of this work is to implement the interoperability between the Clinical Analysis Laboratory Management System (LMS) and Patient Management System (PMS), to facilitate and speed up the availability of information related to the patient's health status, in order to allow the health technicians can make decisions faster, thus avoiding some constraints or even the loss of human lives [6].

The remaining sections of this work are divided as follows, section 2 talks about literature review, section 3 development methodology, section 4 project feasibility and sustainability study, section 5 proposed cloud model and section 6 the conclusion.

1.1 Motivation

As an employee of the Ministry of Health of São Tomé and Príncipe, in the Department of the Health Information System (HIS), which is responsible for coordinating the entire health information system in the country, i was able to see in locus several islands of software in the general health service in São Tomé and Príncipe these software islands have caused some constraints, namely redundancy in data, lack of credibility, ambiguity, etc.

In order to overcome this situation without major constraints for users, we thought of implementing a service (interoperability) in order to consume the information found in these software in real time, establishing an interoperable platform to improve the health information circuit of the HIS.

In order to carry out this claim, we took as a starting point the improvement of the information circuit between the medical office and the clinical analysis laboratory. The sharing of information in real time between the medical office and the clinical analysis laboratory proves to be crucial in the process of requesting clinical analysis and sending the respective results, as they constitute a significant basis for decisions regarding the patient's health status. Currently, the process of requesting analysis and sending the result is done in handwritten form, therefore, there are certain constraints when the doctor requests a clinical analysis and the way the laboratory sends the results to the requesting physician, this practice provides illegibility, loss of time, opening of the analysis result by third parties (companions, medical assistant, family, etc), leakage of information about the patient's health, therefore leaving him in a very exposed and vulnerable situation.

1.2 Research Question

In search of a systematic perspective on IS interoperability and access to sensitive data, a literature review was carried out to answer the following research questions:
RQ1: How to improve interoperability between information systems (PMS and LMS)?
RQ2: How to make such sensitive patient data available to different stakeholders?

1.3 Hypotheses

H1: Participate in the process of developing interoperability standards.

H2: APIs, i.e., Open Application Programming Interfaces, will play a key role in achieving the interoperability.

H3: Plan to upgrade or replace old software.

H4: Data for diagnosing the patient's health may be used without the patient's prior consent, if they are to protect the patient's vital interests, under professional secrecy.
H5: Sensitive patient data may be used for preventive medicine purposes, the provision of health care, management of health systems and services.

1.4 Research Methodology

The research methodology used is design science research (DSR), interest in DSR began to grow in the 1990s, having its roots in engineering [8]. Unlike behavioral research, which deals with justifying and developing theories, explaining or predicting phenomena associated with a business need, DSR deals with the construction and evaluation of artifacts that are designed to meet a specific business need [8].

This research seeks to build and evaluate artifacts. These artifacts include elements named as constructs, models, methods and instantiations. Constructs provide the vocabulary to define and communicate problems and solutions. The models use constructs to represent a problem, expressing relationships between constructs. Methods provide guidance on how to solve problems, being a set of steps used to solve a problem. Instantiations are implementations of constructs, models or methods in a working system [8]. Based on these four artifacts, we will focus on Instantiations.

The DSR process consists of six phases [8], also known as sections. Section 1 presents the research problem, section 2 provides the study objectives, section 3 is the step of creating interoperability guidelines based on a literature review, section 4 constitutes demonstration of the artifact (guidelines), section 5 involves evaluation of the artifact using various methods to develop an understanding of the phenomena, section 6 involves communication of research.

The research process can be explained by the diagram represented in figure. 1.1

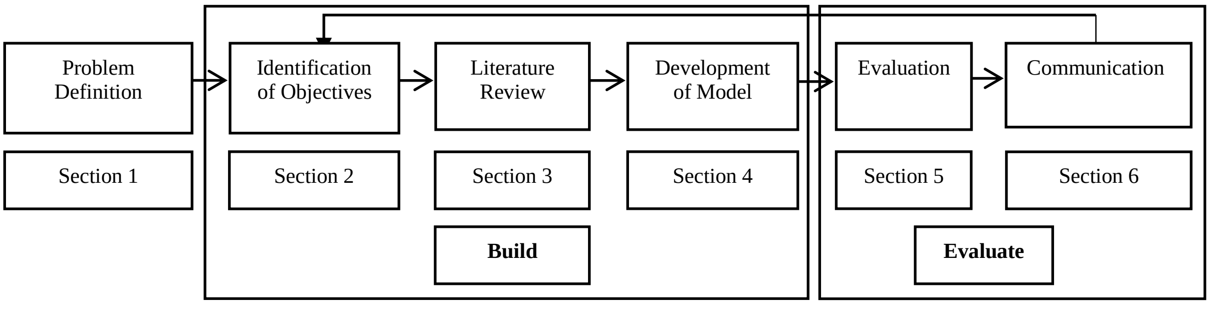


Figure 1.1: Phases of Design Science Research

1.5 Application of the investigation method

The DSR method will be presented in some detail about phases of the method adjusted to the context of the work.

1.5.1 Problem Definition

At this stage it is necessary to identify and understand the study problem for which a solution is intended to be presented [8]. This research aims to understand works previously carried out in the implementation of interoperability between heterogeneous software, thus serving as inspiration in the implementation of interoperability in ways to mitigate problems related to the discrepancy of data provided by various software installed in the same institution (Hospital Dr. Ayres de Menezes). Therefore, due to the need to obtain a sufficiently rich and structured base, to carry out the development of the different tasks, the research that subsidized the development of this work was carried out, the papers were selected applying the inclusion and exclusion criteria and the quality criteria [8].

1.5.2 Identification of Objectives

After the problem identification phase, it is intended to present the objectives defined for the realization of the dissertation, as well as to plan its development. In a first phase, a set of information related to the topic was selected, after the consolidation of the development idea, it was limited to the coding and implementation of interoperability between two different applications and the implementation steps were reported to the supervisor in order to obtain some insight into the work [8].

1.5.3 Development of model

The development stage is the effective construction of the artifact, to effect the process of construction and communication of data from different applications, we use the Rational Unified Process (RUP) development methodology, this term is a process that was created by the engineering company software, Rational Software Corporation, in order to guide the development of an application [33]. The development of this work involves creating a set of artifacts from the requirement gathering to its implementation [8].

1.5.4 Evaluation

Evaluation is an essential component of the research method. The evaluation of the DSR result is based on the business requirements that often occur in the context of utility, quality of the artifact produced, integration of the artifact with the technical infrastructure of the business environment. According to [3], in the IT environment, artifacts can be evaluated in terms of functionality, integration, consistency, precision, performance, cost, reliability, usability, among others. After creating the artifacts (the developed applications), we can say that they are interoperable, but we still need to make them available to end users and therefore make possible improvements.

1.5.5 Communication

The DSR Communication takes place, from a scientific point of view, to present the main results of the study. It is at this stage that the importance of the work carried out, its usefulness, rigor and the innovation it provides in the area of interoperability of heterogeneous information systems must be presented. The results obtained are presented and it can be inferred whether or not the development of the artifacts was successful. The communication will be presented at the end of the development of this dissertation [8].

1.6 Review protocol

The research process began with the search of papers in the following databases: 1) IEEE Xplore Digital Library; 2) ACM Digital Library; 3) Scopus; 4) Ebscohost. The searches of papers in the databases were selected from January 2017 to November 2021 and based on keywords for each of the following terms: 1) "Interoperability and Information System"; 2) "Interoperability and Clinical Analysis". Regarding the filtering processes, these were used papers written in English, papers that were published in the last 5 years, and manual exclusion of papers that had no relation to the objective of the work.

1.6.1 Criteria for Inclusion and Exclusion of papers

Studies for inclusion in this research must report empirical findings and may be journal and/or conference papers. The main inclusion and exclusion criteria are shown in Table 1.1.

Table 1.1: Inclusion and exclusion criteria

Inclusion criteria	Exclusion Criteria
Papers published between the years 2017 and 2021	Papers published before 2017
Papers in English	Papers not in English
Conference and newspaper papers	Books, non-academic papers
Papers that propose the IS interoperability solution	Similar Papers with the same results
Papers evaluating IS interoperability	Papers not available for download

1.6.2 The databases used

- ACM Digital Library
- IEEE Xplore Digital Library
- Ebscohost
- Scopus

1.6.3 Quality Criteria

Regarding scientific prestige indicators, the Scimago Journal & Country Rank and Conference Ranks were consulted. In figure 1.2, it can be seen that 43.8% of the papers belong to the 1st quartile, followed by 37.5% without rank, 12.5% belonging to the B1 conference and 6.2% belonging to the 3rd quartile.

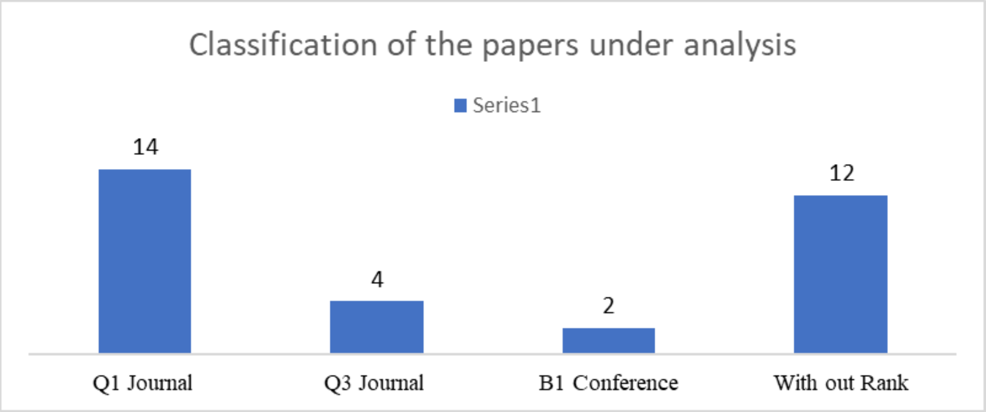


Figure 1.2: Classification of the papers under analysis

1.6.4 Selection of studies

Although the search showed thousands of results, the papers with the greatest relevance to the topic were collected, in a total of 45 papers for later choice. The choice initially included the analysis of abstracts, introduction, and conclusion, to verify the relevance of each paper and then a more detailed analysis which resulted in the exclusion of 13 papers, where 9 of which were inaccessible for download, and 4 of them had a lack of harmony to the topic under study.

Table 1.2: The search results and the respective databases used.

DB	Keywords	F1	F2	F3	F4	F5
IEEE	Interoperability and Information System	1462	542	39	11	8
	Interoperability and Clinical Analysis	1643	487	8	1	1
DL	Interoperability and Information System	4361	407	38	28	4
	Interoperability and Clinical Analysis	1067	569	17	10	4
SCOPUS	Interoperability and Information System	4448	2642	1839	58	8
	Interoperability and Clinical Analysis	568	20	13	6	1
EBSCOhost	Interoperability and Information System	1759	995	95	35	1
	Interoperability and Clinical Analysis	126	15	6	5	5
Total						**32**

F1 - Full text; F2 - All metadata; F3 – Abstract; F4-Title; F5 - Manual analysis.

1.6.5 Analysis of the extracted data

The papers under analysis are from the last 5 years, that is, from 2017 to 2021, it should be noted that since 2017 there has been a decline in published papers until the year 2020, and, from the year 2021 the numbers start to increase again, and the years with the highest number of published papers are the years 2017 and 2018 respectively.

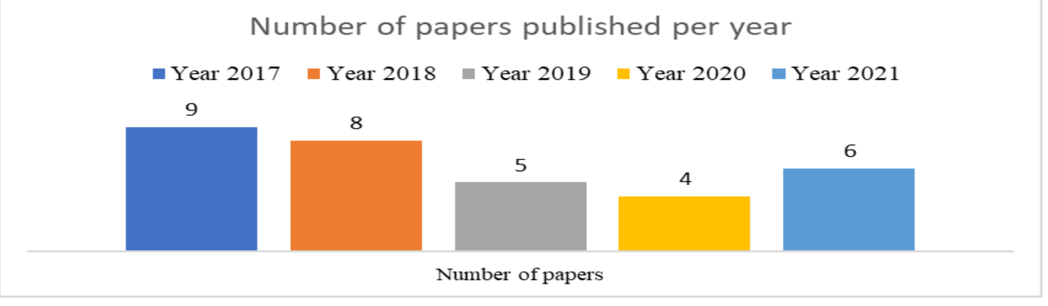


Figure 1.3: Number of papers per year

2 LITERATURE REVIEW

The scope of this section consists to highlighting the papers selected for this work, conceptualizing terms or expressions directly or indirectly related to interoperability, namely information system (IS), levels of interoperability, the most used protocols, advantage and challenge of interoperability.

2.1 Information system

Before approaching the topic under study, let's make an awareness of what an information system is. An Information System (IS) is a collection of dynamically interconnected components to collect, store, process data and provide information to support organizational decisions [23,22, 27].

2.2 Interoperability

Interoperability is about the interconnection between information systems to provide meaningful data sharing [1, 6]. In IS, interoperability is seen as the ability of different systems and technologies to communicate, exchange and use information effectively and efficiently [15, 16].

In our review, interoperability has been broadly classified into different types [10, 11, 12]. And in this work, we intend to describe the different types of interoperability, technical, syntactic, semantic, organizational, and legal [13].

Technical interoperability is related to the transmission of data between components or systems. It is associated with the hardware and software components, networks and equipment that allow machine-to-machine communication, including aspects such as open interfaces, data integration, middleware, data presentation, data exchange, accessibility, and security issues [4,14].

Syntactic interoperability is related to rules about the type of data that are exchanged, specifically, about how to group the data and in what order [4, 15]. This type of interoperability focuses on identifying elements and rules that structure the elements, well-defined syntax (i.e., structure of message content, size of headers, size of message body, fields contained in a message), and navigation between equivalent elements [4,16].

Semantic interoperability is related to the exact meaning of the data exchanged. In this type of interoperability, data are conceived as information to be shared, processed and well understood (without ambiguity) by the surrounding systems [16]. Thus, semantic interoperability is about ensuring that the precise meaning of the information exchanged is understandable by any other system that was not initially developed for that purpose.

Organizational interoperability is related to the coordination of distributed workflows and activities that are well understood by systems, organizations, or people interacting in business processes. This type of interoperability is related to the ability of two or more components/systems to provide services to and accept services from other components/systems and use those services to support them to operate together effectively [4,16].

Legal interoperability is the legal requirements and implications of making information items freely available [15].

The European Union (European Interoperability Framework or EIF) has classified interoperability in four levels, namely legal interoperability, organizational interoperability, semantic interoperability, and technical interoperability [15].

In this context, legal interoperability points to a coherence or alignment between legal issues that, in the case in question, are diffused between different countries, member states and regions. Organizational interoperability refers how business processes are aligned or integrated between different organizations and how relevant information is exchanged. Technical interoperability covers the applications and infrastructure that link systems and services. Aspects of technical interoperability include interface specifications, interconnection and integration services, among others. a Semantic interoperability seeks to ensure the proper format and meaning of the data exchanged, so that the information is preserved and understood between the parties. In this view, semantic interoperability covers both, syntactic and semantic aspects. The semantic aspect refers to the meaning of the elements and their relationships. It includes developing vocabularies and schemas to describe data exchange and seeks to ensure that the data is understood in the same way by all those involved in the communication. The syntactic aspect refers to the ability to describe the exact form of communication to be exchanged in terms of its grammar and format. [10,15].

Based on the papers selected in the research process, for the health area, the most outstanding interoperability are technical, syntactic, semantic and organizational [4,5].

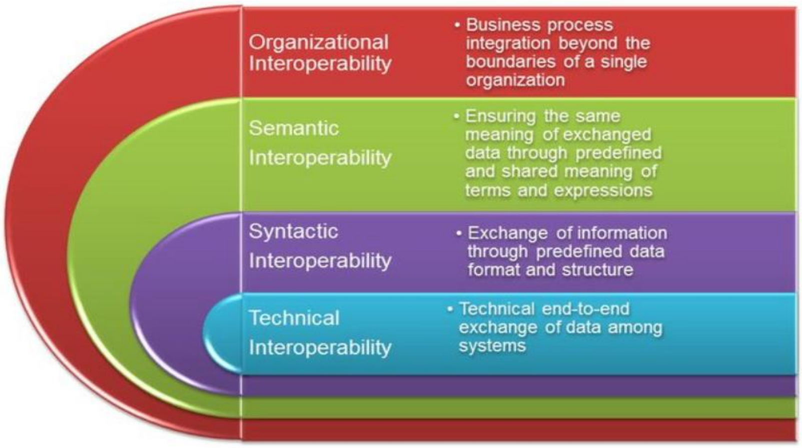


Figure 2.1: The four levels or types of interoperability

2.3 Interoperability standards and protocols

Protocols are a set of rules to define communication between information systems [10]. In table 2.1, we mentioned the most cited protocols in the papers selected in the process of this research.

In our literature review we found that different domains can have different types of interoperability, table 2.1 summarizes the literature review on the levels of interoperability.

Table 2.1: Synthesis of literature review, standards and protocol cited in the literature review.

Ref.	Type of communication				Standards of communication										Ling. for information exchange		Protocol for interoperability																	Domain	Topic	Classification
	Client - Server	JXTA	Server - Client	P2P	DICOM SR	CDA	CEN / EN 13606	OpenEHR	SNOMED	FHIR	PITUS	HL7	Dublin Core	AtomPub	XML	Serialized formats	IEOA	OAI-PMH	OAI-ORE	REST	Z39.50	SRU	OPC-UA	SRW	SWORD V2.	RSS	JSON	WSDL	JADE Gateway	IIIF	SOAP	SOA	ATOM			
[1]	X	X										X			X					X									X					Business	SoS	Tech.
[2]	X				X							X			X																			Health	SoIS	Semantics
[3]	X														X																			Health	SoS	Syntactic
[4]	X					X	X	X				X			X																	X		Health	IS	Tech., Semantics, Syntactic, and Organ.
[5]	X											X			X					X							X							Health	SoS	Tech., Syntactic, Semantics, Org
[6]	X											X			X																	X		eGov	SoIS	Tech. and Org.
[7]	X				X	X						X			X												X				X	X		Health	IoT	Tech. and Syntactic
[9]	X							X							X			X	X		X	X		X	X	X	X				X		X	Geral	IS	Tech.
[10]	X											X			X																			Geral	IS	Semantics
[11]	X								X			X																						Health	IS	Semantics
[12]	X											X			X		X																	Health	IS	Semantics
[13]	X														X																	X		Business	IS	Semantics
[14]	X														X																	X		Geral	IS	Tech. and Semantics

[15]	X														X																			Geral	ICT	Legal, Org. Sem. e Tech.
[16]	X					X									X																			Business	IS	Semantics
[17]	X					X																									X			Business	IS	Tech. and Semantics
[18]	X					X									X													X						Health	IS	Semantics
[19]	X														X					X			X								X			Business	IS	Legal, Org. Sem. e Tech
[20]	X															X					X											X		Health	IS	Syntactic
[21]	X					X																												Health	IS	Tech. and Semantics
[22]	X											X			X			X														X		Business	IOT	Tech. and Syntactic
[23]	X														X												X				X			Health	IS	Tech. and Semantics
[24]	X				X					X																								IS	IS	Semantics
[25]	X																													X		X	X	Geral	IS	Tech. and Semantics
[26]																																X		Geral	IS	Legal, Org. Sem. e Tecn.
[27]	X														X																	X		eGov	IS	Tech.
[28]																																		Geral	SoIS	Org.
[29]	X																																			Tech.
[30]	X				X	X			X			X																						e-Health	IS	Legal, Org. Sem. e Tecn
[31]	X														X					X											X	X		Geral	IS	Legal, Org. Sem. e Tecn
[32]	X				X	X						X			X					X											X			Geral	IS	Semantics and Syntactic
[37]			X						X			X			X					X							X									Tech. and Semantics

2.4 Some Interoperability Benefits

The table 2.2 shows some of the benefits obtained from having interoperability between the LMS and the PMS.

Table 2.2: Interoperability Benefits

Benefits	**Ref.**
Cost reduction for industry, healthcare institutions etc.	[20][4]
Decreased errors and data redundancy	[4]
Greater accessibility to patient information by specialist physicians increasing the quality of diagnosis and treatment.	[20][4][22]
Increases process efficiency	[4][6][28][9]
Allows you to query data in real time from other systems.	[4][19][20]
Facilitates data security and integrity	[15]

2.5 Interoperability issues

Interoperability is a concept that has different levels of application. At the level of technical interoperability, we can say that we face basic incompatibilities between different hardware, operating systems, and software, as well as in the structural part, in different models and data structures and schemas. Regarding syntactic interoperability issues, we can summarize the differences concerning encodings and data representations. Semantic interoperability has its problems with inconsistencies in terminology and meanings of terms used [14].

2.6 Some interoperability challenges

Table 2.3: Interoperability Challenges

Brief description of interoperability challenges	**Ref.**
Interoperability causes the standardization of vocabularies (performed through systems such as ICD-10, for example). However, the development of these standards can be slow and their adoption even slower.	[21][19]
Adequate training of professionals involved in the development of systems and IT and of those who use them (doctors, nurses, pharmacists, nutritionists, among other health professionals) is necessary.	[21][22][7]
One more factor that can make it difficult to adopt interoperability systems is the large number of existing standards. For this reason, it is difficult to know which one should be adopted to add more benefits to the institution.	[21][23] [24]
Combine multiple applications in a collaborative relationship as a single entity	[1]
Security and privacy, personal information must be kept confidential, and cannot even be shared with any authority without the patient's consent.	[1][4][15]

2.7 Description of some interoperability standards and protocols

In this section we described in table 2.4 some technical term that were cited in the table 2.1.

Table 2.4: The description of the technical terms mentioned

Protocol	Description	Ref.
OAI-PMH	It is an interoperability protocol aimed at the effective dissemination of digital content, it also works with import and export of data in Extensible Markup Language (XML) format used to store and transport data.	[22]
Z39.50	It is a computer-to-computer communication protocol designed to allow information search and retrieval in distributed computer networks.	[22]
SOAP	It is a protocol for exchanging structured information on decentralized and distributed platforms, using XML-based technologies.	[17]
WSDL	Is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.	[17]
SOA	Defines a way to make software components reusable and interoperable via service interfaces.	[18]
JSON	The JavaScript programming model uses conventions that are familiar to C, Java, JavaScript, Perl, Python, and many other languages. These properties make JSON an ideal data exchange format.	[15]
HL7	It is a not-for-profit organization dedicated to the development of standards to help exchange, integrate and share electronic health information that supports clinical practice and management.	[1][15]
SNOMED	This standard aims to unify the terminology used in sharing health information through EHRs.	[9][1]
SWORD	It is an interoperability standard that allows digital collections to accept the deposit of content from various sources in different formats, through a standardized protocol.	[9][4]
SRU	It is a protocol developed to work within the scope of the HTTP protocol and enables the retrieval of records in MARC (Machine Readable Cataloging) format.	[9]
SRW	It was developed with the same objective as the SRU, however it uses a computational architecture based on Simple Object Access Protocol (SOAP), and not a URL.	[9][13]
RSS	It is an XML-based distribution protocol, which is widely adopted by frequently updated data providers.	[9]
ATOM	It is designed to provide interoperability between extended implementations and other implementations that do not recognize these extensions.	[9]
IIIF	It consists of a set of application programming interfaces (APIs) based on open web standards for image interoperability.	[9]

3 DEVELOPMENT METHODOLOGY

A study of the different development methodologies, tools and standard architecture technologies was carried out, for use in software development according to their characteristics.

Due to the scale of this application, we understand that the most appropriate methodology to use is the Rational Unified Process (RUP), as it is an interactive and incremental software development process, which enables analysis, implementation and documentation of the system.

For the development of PMS and LMS in S. Tomé using the RUP methodology, there will be every need to go through its activities, which will be described in detail and presented in Figure 3.1. In principle, all activities of the RUP can be triggered at all stages of the process, naturally most of the effort is likely to be spent on activities such as business modeling and requirements early in the process.

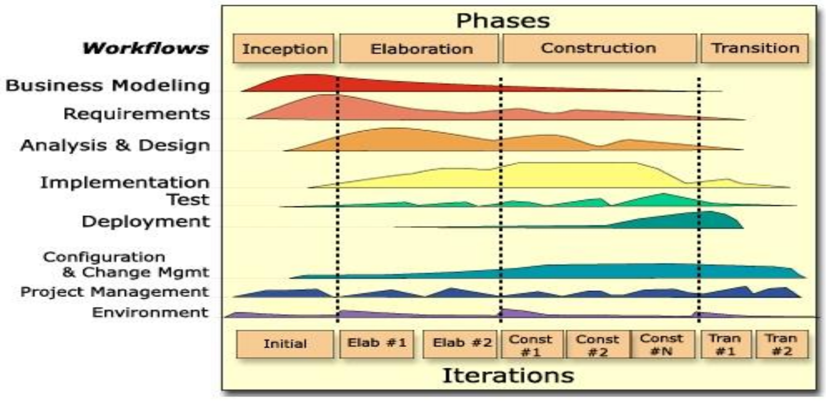


Figure 3.1: RUP Overview

3.1 Business Modeling

This activity is where the business analysis is carried out, before carrying it out, we have to delimit the scope of the proposed system. From this point, some steps will be possible, in most cases the workflow is suggested, this workflow includes enumerating candidate requirements, understanding the system context, capturing functional requirements and non-functional requirements [33].

3.1.1 Business description

The business modeling workflow teaches us to describe the current business and to model the proposed business, it gives a vision of what needs to be done to respond to the demand. This session will show how requirements are obtained and how they are handled in other workflows to turn them into a software product [33].

The Hospital Center Dr. Ayres de Menezes is a public institution that carries out a set of activities and services that constitute the management and organization of the same in favour of the patient's health, in order to respond to these activities, the aforementioned hospital center consists of 10 sections (functional units) which are: Psychiatry, Pediatrics I and II, Operating Room, Urgency, Intensive Care and Surgery, Pharmacy, Clinical Analysis Laboratory, X-Rays, Physiotherapy and Maternity.

3.1.2 Patient

When the patient arrives at the hospital in the emergency department, he is triaged and is given a card according to his health status. If his health condition is serious, he has the priority to go directly to the medical office, otherwise he must wait in a waiting room. All patients must have a registration form, if not, it is created, containing patient data such as name, date of birth, residence, affiliation, name of companion, contact, etc. Each patient is attended by one or more physicians. For patients who already have an open record, the search for the record is based on the clinical file that is on the shelves.

3.1.3 Employee

There are several employees in the hospital, each employee has a name, affiliation, date of entry, category (nurse, doctor, medical assistant, clinical analyst, administrative staff, etc.), section where he works and his academic qualifications. Each employee is linked to a specific section. All employees who do not have training are considered as medical assistants.

3.1.4 Doctor

Each doctor can consult one or more patients, all patients consulted by the doctors are recorded on A4 paper and placed inside an envelope identified by the case number and date. The record consists of the case number, patient's name, address, associated pathology.

3.1.5 Laboratory

The clinical analysis laboratory at the Dr. Ayres de Menezes is a hospital section that is responsible for helping doctors to identify pathologies through laboratory tests with biological material (blood, urine, feces, sputum and organic fluids). The clinical analysis is based on an exam ordered by the doctor, in the request for the medical exam it contains the name of the exams, patient's name, record number, date and the doctor's name (Signature). The laboratory can receive two types of exam requests, internal and external, the internal request is related to patients who are hospitalized in the hospital center or recommended by the doctor working in the hospital, and the external request refers to patients clinics. All examinations carried out are recorded in a book for each current year.

The performance of analyzes is sometimes conditioned by the fact that the laboratory has difficulties in managing the stock of reagents efficiently. The results of the analyzes are folded and stapled on A4 paper and made available to patients, family members or the medical assistant to be delivered to the requesting doctor, this practice has caused some constraints, namely, opening the result before going through the doctor, disclosing of the patient's health situation, thus entailing all the implications for the psychological state of the same.

3.1.6 Business actors

Table 3.1: Description of business actors

Actor's name	Description
Clinical analyst	Laboratory employee who processes and validates the analyzes
Docter	It is the person who treats patients and requests the analysis
Patient	The sick person who provides data or information in order to be treated and often transports the results of analyzes
Medical assistant	It is the person who takes the test results to the medical office

3.1.7 Diagram of business use cases

A business use case diagram describes the actions that are triggered by each business actor, allowing the identification of the main functionality for the system that will be designed [33].

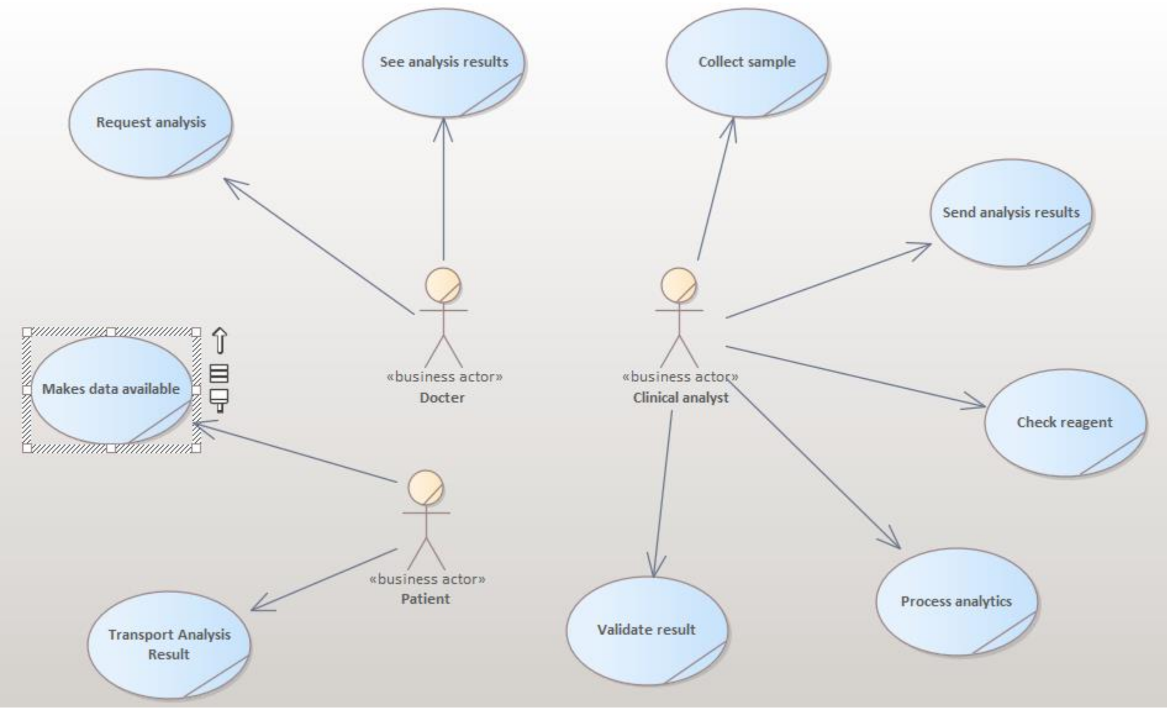


Figure 3.2: Business Use Case Diagram

The table 3.2 gives a brief description of business use cases.

Table 3.2: Business Use Case Description

Use case	**Description**
Check reagent	The laboratory technicians, before confirming the performance of the analyzes, verify the existence of the reagent
Request analysis	The doctor requests analysis to help diagnose the patient
Send analysis results	Analysts make analysis results available
Process analytics	After confirming the existence of the reagent, the analysts process the analyzes
Collect sample	Analysts collect samples from patients or users
Validate result	The Clinical Analyst responsible to validate the results of the analyses
Transport analysis result	The patient or medical assistant takes the results of the analysis to the medical office that requested the respective analysis
Makes data available	The patient provides data or information in order to be treated

3.1.8 Activity diagram

This section describes the occurrence of the most relevant use case for the problem domain through activity diagrams, where it is possible to show how and who starts the use cases, as well as the generated objects and the realization of their activities.

3.1.9 Use case activity diagram request analysis

In order to determine with greater precision, the disease that plagues the patient, the doctor requests the necessary analyses, describing their names on a paper, and guides the patient to go to the clinical analysis laboratory with the respective request for analysis.

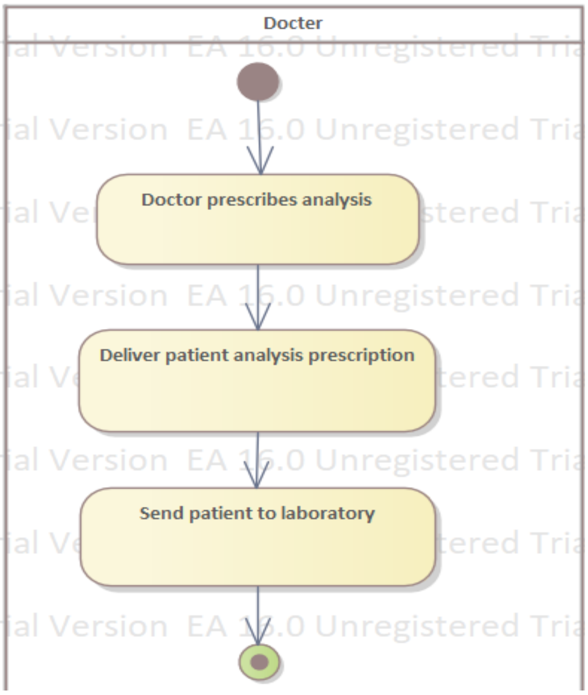


Figure 3.3:Use case activity diagram “Request analysis”

After processing the analysis, the clinical analyst transcribes the result on an A4 sheet, then it is folded, stapled and made available to the user. The user or medical assistant, after obtaining the results of the analyses, are forwarded to the doctor's office.

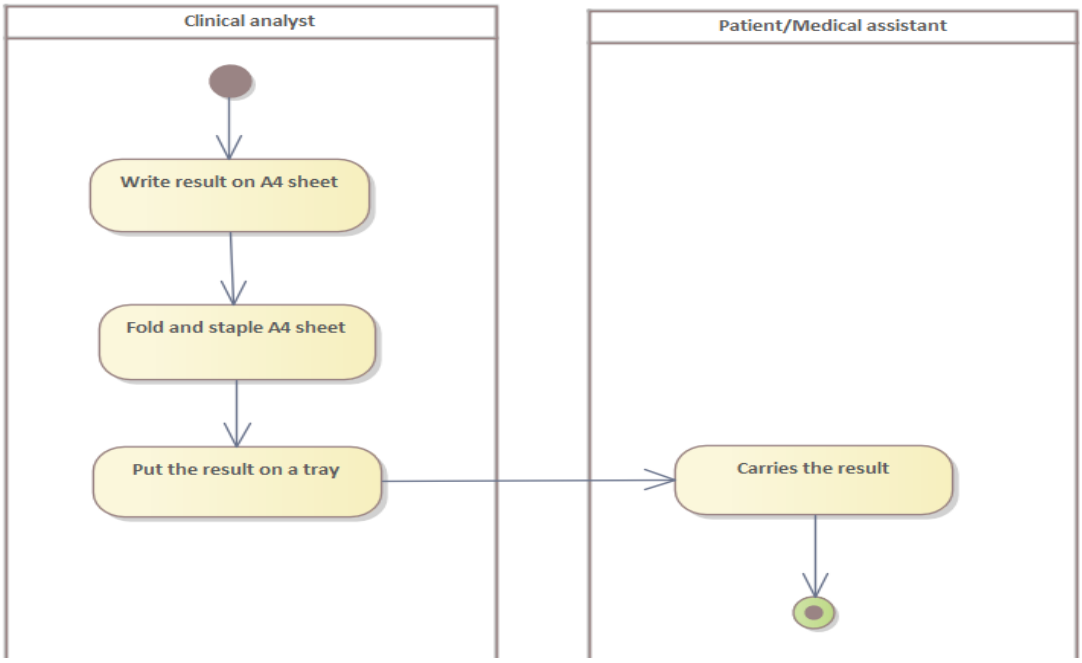


Figure 3.4: Use case activity diagram “Send analysis results”

3.2 Requirements

After the textual description and business representation through business use case diagrams and activities, in this section we seek to identify the potential requirements for building our system. During the requirements gathering process, the customer plays an essential role, becoming one more member of the project team, therefore, the requirements gathering results must be written in a language understandable by all [33].

The workflow that is performed in the requirements gathering process, it is important to emphasize that it is necessary to evaluate the feasibility of the system contemplating each of the desired requirements according to the scope of the system [33].

All ideas that the customer, users and project team members have about what the system should to be analyzed as candidate requirements. Requirements can be classified into functional and non-functional. Functional requirements are capabilities or conditions that the system must meet, while non-functional requirements refer to qualities inherent in the system [33].

3.2.1 System Actors

Since an actor represents a cohesive set of actions that the users of the use cases perform [33], several roles of the users of the system to be developed were identified, as shown in table 3.3.

Table 3.3: Description of system actors.

Actor's name	Description
Administrator	System user with privileges to add and remove other users and/or records, manage accounts and maintain established security policies.
Doctor	User with access to services such as showing patient history, prescribing prescriptions, admitting, and discharging patients, requesting analyses, issuing medical reports, etc.
Clinical analyst	User with access to the services that the system provides, such as seeing the exam requested by a doctor, seeing the amount of reagent stock, sending the analysis results, validating the analysis results.

3.2.2 Functional Requirements

In the execution of the business use cases, the activities that will be the object of automation are obtained, which are not exactly the functional requirements, but are starting points to identify what the system must do. Functional requirements are all the needs, functionalities expected in a process that can be fulfilled by the software [33].

Table 3.4: System Functional Requirements.

N. RF	Requirement Name	Priority
RF1	Register the analysis request	Essential
RF2	Request analysis	Important
RF3	Automatically assign the code to each requested exam	Essential
RF4	Associate the doctor with each request	Essential
RF5	Send the test results to the requesting doctor	Important
RF6	Record the analysis performed	Essential
RF7	Authenticate user	Desirable
RF8	Remove user	Desirable
RF9	Register patient	Desirable
RF10	Triage	Essential
RF11	Issue report	Desirable
RF12	Remove records	Desirable
RF13	Update patient data	Essential
RF14	View patient history	Essential

3.2.3 Non-Functional Requirements

Non-functional requirements are properties or qualities that the product must have, these properties must be thought of as the characteristics that make the product easy to use, secure, interactive, fast and reliable, for example, one may want the system responds within a specified time interval. In many cases, non-functional requirements are fundamental to the success of the product, they are usually linked to functional requirements, that is, once we know what the system should do, we can determine how it should behave, which qualities should be added. [33].

There are multiple categories to classify non-functional requirements, the following being representative of a set of aspects that must be taken into account, although they do not limit the definition of others.

3.2.4 Appearance or external interface requirements

This type of requirement describes the appearance of the product. It is important to note that they do not deal with the design of the interface in detail but specify what the external interface of the product should look like. Based on this criterion, we can say that the product is readable, simple to use, authoritative so that users feel confident and interactive [33].

3.2.5 Usability requirements

The application is a tool to be used by anyone who wants to join it. That's why we designed an interface that, in our opinion, is pleasant and objective, that is, its functionalities and information are very intuitive. System error messages are precise and constructive, making the user identify their origin and how to proceed after their occurrence [33].

3.2.6 Performance

In the tests carried out, the system guarantees the greatest possible efficiency regarding the processing of information so that the processing speed is also as high as possible, ensuring an adequate response time. It guarantees the consistency and availability of information at all times according to user requests, in addition, it supports the concurrency of several users, remaining stable and reliable [33].

3.2.7 Portability

The system is compatible with the most used operating system, such as UNIX (Linux), Windows (versions 2000 to Windows 10) [35].

3.2.8 Security

This is perhaps the most difficult type of requirement, which will cause the greatest risks if not handled correctly. Security was addressed in three different aspects:

1. **Confidentiality** the information processed by the system is protected against unauthorized access and disclosure. Due to the confidentiality of the information to be transacted, it is the object of careful protection interventions. To implement this non-functional requirement, the access control mechanism was used.

The System will have an interface for authentication of its users. During identification, the user informs the system of his name and password and the authentication process verifies the data provided.

To specify the user groups and their permissions, the access control matrix was used. This requirement defined a set of resources that are subject to the access control mechanism and their protection status, describing precisely the protection status through the Access Control Matrix.

Table 3.5: Access Control Matrix

Nº	Requirement Name	Administrator	Doctor	Clinical analyst
RF 1	Request analysis	----------	RW	----------
RF2	See required analysis	----------	RW	R
RF3	Process analytics	R	R	RW
RF4	Send analysis results	R	R	R
RF5	See analysis result	----------	R	R
RF5	Validate analysis results	----------	----------	RW

2. **Integrity:** Information managed by the system will be subject to careful protection against corruption and inconsistent states, just as it will be considered the same as the data source [35].

3. **Availability:** Authorized users will have guaranteed access to information and the devices or mechanisms used to obtain security will not hide or delay users in obtaining the desired data at any time [35].

3.2.9 Secure Socket Layer (SSL)

To provide privacy and data integrity between the two applications that communicate and exchange data, we use the SSL protocol. This protocol helps prevent intermediaries between the two ends of the communication from gaining undue access to the data being transacted [36].

3.2.10 SQL Injection

SQL Injection attacks are quite common on the web, although their prevention can be done relatively easily. This attack is interesting for attackers, since in the health area, databases contain privileged and confidential information, which can be exploited for various malicious purposes [33].

To avoid these types of attacks, instead of dynamic queries, 'prepared statements' (parameterized queries) were used, more specifically PDO (PHP Data Objects) with parameterized queries. This implied that, at the development level, the SQL code was first defined, then passed the user parameters in the query. They have the advantage of isolating SQL from the data that the user enters into the interface, preventing intruders from

System administrators are responsible for creating users in the systems for health professionals and these in turn change their passwords, thus preventing users from registering as health professionals undue.

3.2.11 Cryptography

We use RSA algorithm encryption protection, even if attackers get the contents of our files, they will be unreadable.

3.2.12 System Use Case Model

System actors

Every business participant that has activities to automate is a candidate system actor. If any business actor is going to interact with the system, it will also be a system actor. System actors do not necessarily have to be part of the business, they can exchange information, be passive receivers of information, they can represent the role played by one or more people, a team or an automated system [33].

System use cases

System use cases are narrative artifacts that describe, in the form of actions and reactions, the behaviour of the system from the user's point of view. To do so, it establishes an agreement between customers and developers about the conditions and possibilities (requirements) that the system must meet [33].

Candidate use cases are also among the activities to be automated. This does not mean that an activity becomes a use case, because a use case is a process that gives a result of value to a certain actor and a sequence of activities to automate, may involve steps within a use case. In the example, the functional requirements, described above, can be grouped in the process, which is presented as a use case [33].

3.2.13 Clinical Analyst use case diagram

The clinical analyst responsible for the laboratory service, after viewing the requested analyses, he has the possibility to follow the analysis processing status through the functions process analysis, validate result, and send the result of the analyzes through his system, the figure 3.5 presents the duties of the clinical analyst.

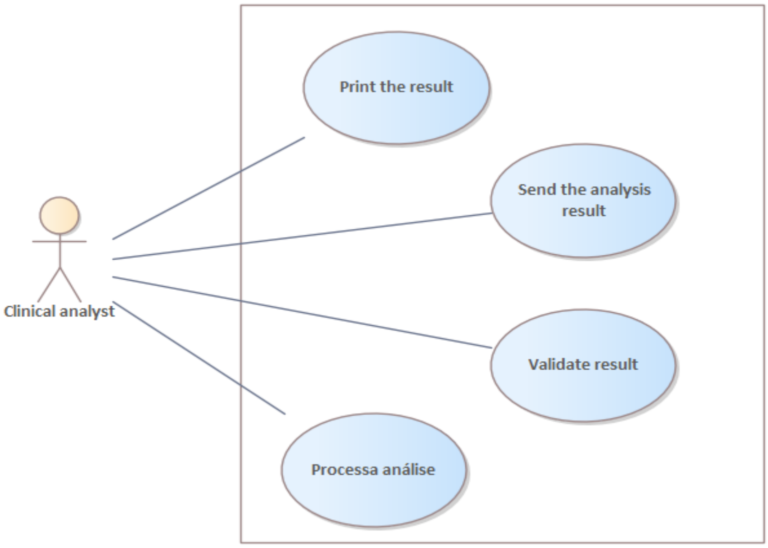


Figure 3.5:Use Case Diagram of Lab Administrator System.

During the consultation with the patient, the doctor can request analysis through the Request analysis function and can view the analysis results through the See analysis result from the PMS dashboard.

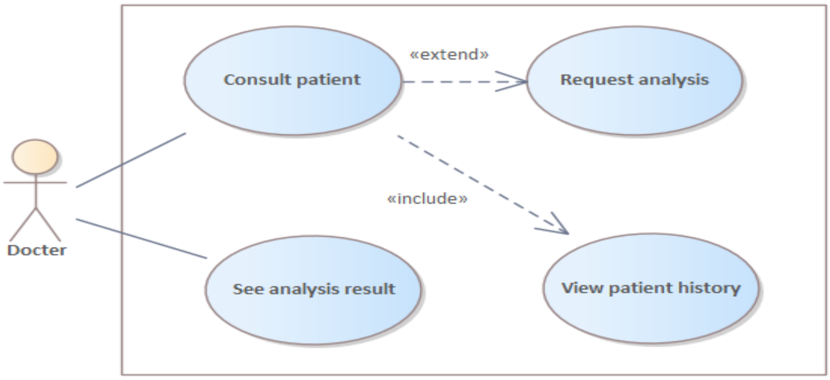


Figure 3.6: Diagram of request analysis system use cases.

3.2.14 General system use case diagram

Figure 3.7 represents all the features that the PMS and LMS, this diagram only demonstrates what the systems do and not how they do it.

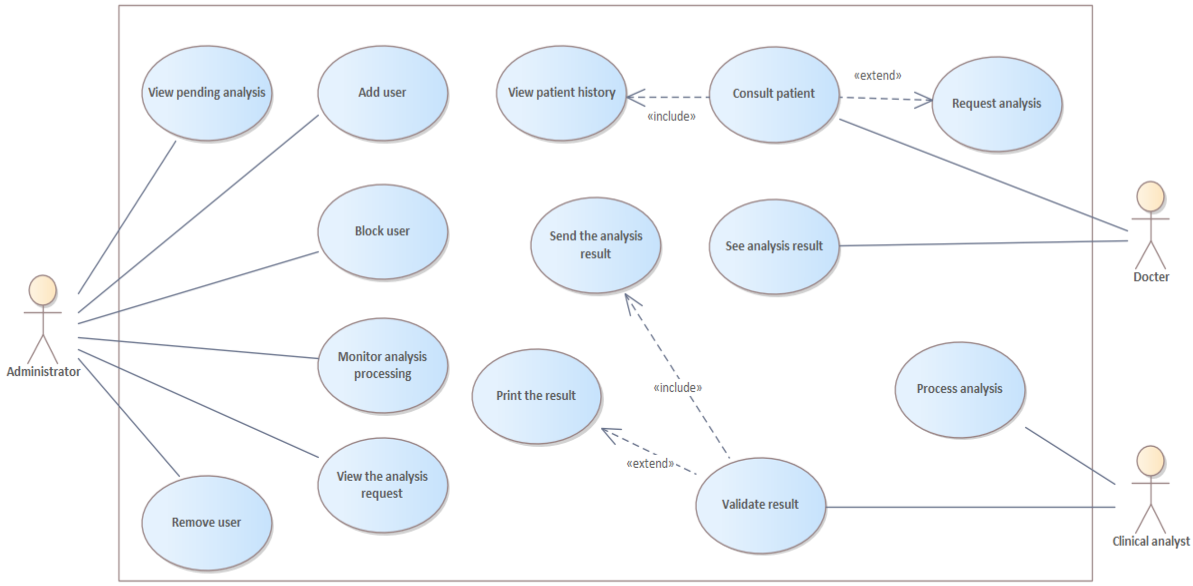


Figure 3.7:General system use case diagram

3.2.15 Specification of the most relevant use cases for the problem domain

Table 3.6: Specification of the add user use case.

Add user	
Precondition	The administrator must be logged in
Description	1. The use case starts when the administrator accesses the system. 2. The administrator adds a new user and informs name and temporary password 3. The system registers a new user.
Post condition	New user registered in the system.

Table 3.7: Specification of the use case block user.

Block user	
Precondition	The administrator must be logged in
Description	1. The administrator makes a list of users. 2. The system shows user list. 3. The administrator locks the user selected in the list.
Alternate paths	If the administrator does not want to block the user, he can simply change the login data.
Post condition	User blocked from the system or some data changed.

Table 3.8: Use case specification View the analysis requested

View the analysis requested	
Precondition	The Clinical Analyst must be logged in
Description	1. Clinical analyst is notified upon arrival of required analyses. 2. Clinical analyst lists the required analyses.
Post condition	Records of required analyzes

Table 3.9: Use case specification, View patient history.

View patient history	
Precondition	The doctor must be logged in
Description	1. In the consultation process, the doctor can view his patient's history through the system's functionality, View patient history. 2. The system makes the clinical history available to the physician.
Alternate paths	The doctor may order tests
Post condition	The result of analysis or consultation is added to the patient's history

Table 3.10: Use Case Specification, Send the analysis result.

Send the analysis result	
Precondition	The clinical analyst must be logged in
Description	1. After validating the result, the analyst clicks on send result. 2. The system sends the analysis result to the requesting system.
Alternative paths	Clinical analyst prints analysis result
Post condition	The respective analysis is recorded as sent to the applicant.

Table 3.11: Use Case Specification, Consult patient.

Consult patient	
Precondition	The doctor must be logged in
Description	1. In the consultation process, the doctor can view his patient's history through the system's functionality, View patient history.
Alternative paths	2. The system makes the clinical history available to the physician.
Post condition	The doctor can request the analyzes from the LMS, from his consultation panel.

Table 3.12: Use Case Specification, Print the result.

Print the result	
Precondition	The clinical analyst must be logged in
Description	1. After validating the results of the analyses, the analyst can print the first copy of the result.
Alternative paths	The system sends the analysis result to another system (Patient Management).
Post condition	The first analysis printout is registered

Table 3.13: Use Case Specification, Validation result.

Validate result	
Precondition	The clinical analyst must be logged in
Description	1. The analyst validates the analysis processing steps.
Post condition	2. The system provides the status of the analyzes being processed.

Table 3.14: Use Case Specification, Process analysis.

Process analysis	
Precondition	The clinical analyst must be logged in
Description	1. The analyst updates the data for each processing phase in the system. 2. The system provides data on the processing phases.
Post condition	Analysis processing is recorded

Table 3.15: Use Case Specification, Request analysis.

Request analysis	
Precondition	The doctor must be logged in
Description	1. The doctor requests analysis in the system through the Request analysis option. 2. The PMS sends the user number and the names of the exams to the LMS.
Post condition	A new analysis request is registered

Table 3.16: View the analysis Result

View the analysis Result	
Precondition	The doctor must be logged in
Description	1. The doctor visualizes the results of the analysis through the option view the analysis result that is found on the consultation dashboard. 2. The PMS displays the results of the analyzes that have been requested.
Post condition	New analysis results are added to the patient's history

3.2.16 Package organization

Packages are a mechanism for organizing elements that subdivide the model into other smaller models that collaborate with each other, in which their contents must be closely related and minimize dependencies between packages (Loosely coupled) [50], in this section We present the following packages.

3.2.17 Account Management

This package is responsible for all the features related to the management of the System's users. Here, users and their access permissions to certain system resources are defined.

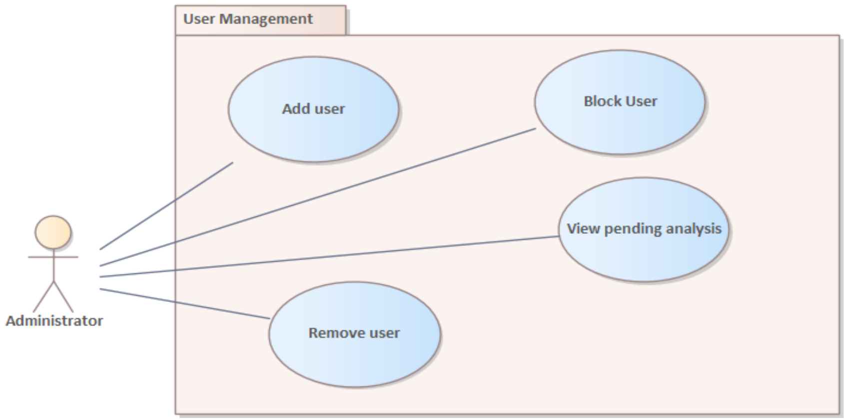


Figure 3.8:Account Control package.

3.2.18 Exam Management

This package aims to guarantee the identification and processing of analyzes requested by the doctor and subsequent sending of the results to the requesting system through an interoperability mechanism, enabling the synchronous communication of data and the pathological situation of the patients.

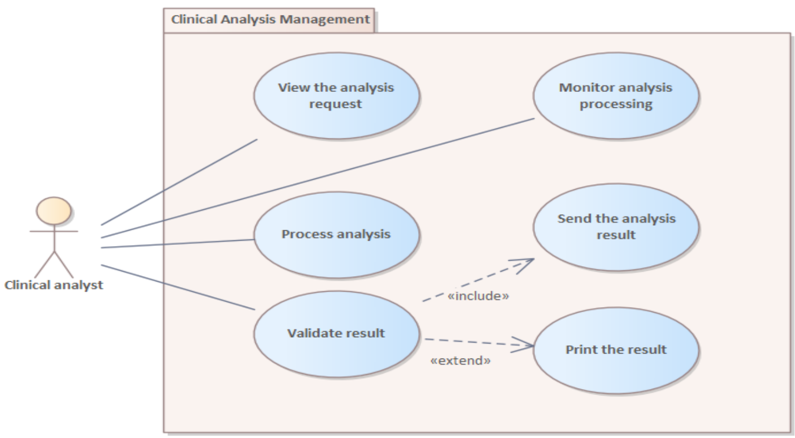


Figure 3.9:Clinical analysis management package

3.2.19 Medical Consultation Management

This package generates information associated with medical consultations, placing greater emphasis on the direct request of clinical analyzes to the system that manages the clinical analysis laboratory.

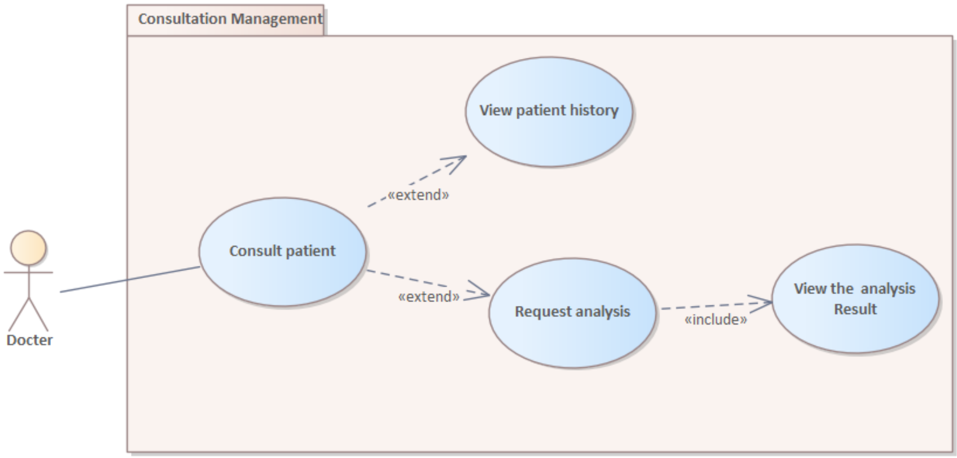


Figure 3.10:Consultationy Management Package

3.3 ANALYSIS

As a result of the requirements workflow, an external view of the system was obtained, which was described in a customer language, to elucidate the Use Case diagram. In the analysis model, the use cases go deeper, detailing them in a way that reflects an internal view of the system described in the developers' language. In this internal view, the use cases are better specified and the classes needed to perform the functionalities contained therein [33].

This analysis model activity is mainly developed in the RUP elaboration phase, and corresponds mainly to the analysis and design workflow, as illustrated in figure 3.1.

3.3.1 Analysis model

An analysis model structures the requirements in a way that facilitates their understanding, preparation, modification and in general their maintenance [33]. It can be considered as a first approximation to the design model and is, therefore, a fundamental input when contouring the system in design and implementation. Analysis classes always fit into one of three basic stereotypes, Interface, Control and Entity [33].

3.3.2 Interface Classes

The interface class models the interaction between the system and the system's actors, they can be identified from a class that models the user actor's interaction with the system, that is, a class for each actor that represents a device on which the system acts on or receives information [33].

3.3.3 Control Classes

Control classes coordinate the work of one or a few use cases, coordinating the activities of objects that implement the functionality of the use case, thus defining the flow of control and transactions within a use case, delegating the work to other objects [33].

3.3.4 Entity Classes

This class models enduring and often persistent information, phenomena, concepts and events that occur in the real world [33].

The medical actor interacts with the interface class, Medical appointment dashboard, to carry out the consultation, which in turn interacts with the Control class, Analysis control, whose mission is to coordinate the requests of the interface classes and carry out registration. in the entity class, request analysis, and its subsequent notification to the interface class, view request analysis, as shown in figure 3.11.

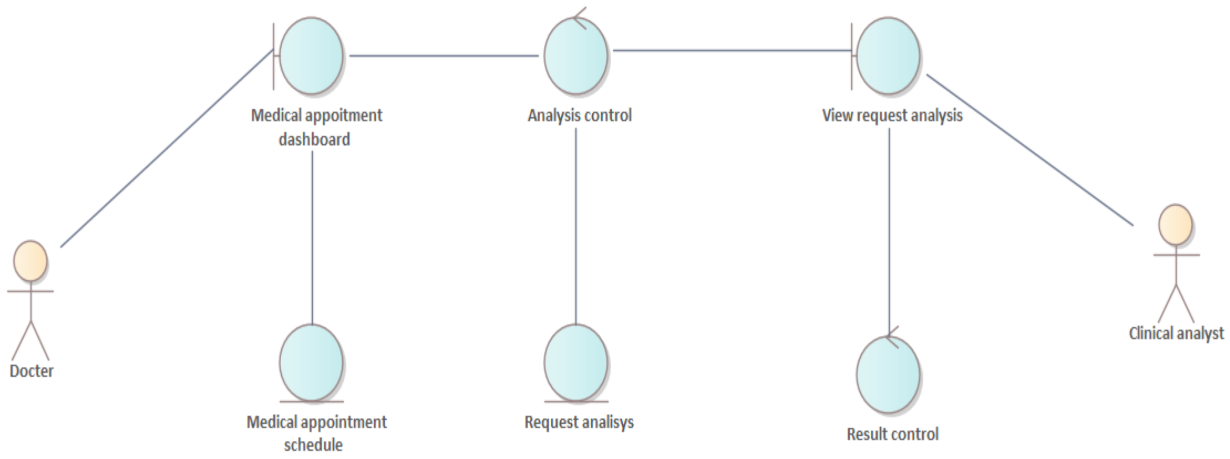


Figure 3.11: Corresponding analysis class diagram, by use case Consult Patient

The Clinical Analyst begins to record patient data and requested analyses, this information is verified by the control class, verify data, and modeled by the entity class, Exams processed until its validation, as illustrated in figure 3.12.

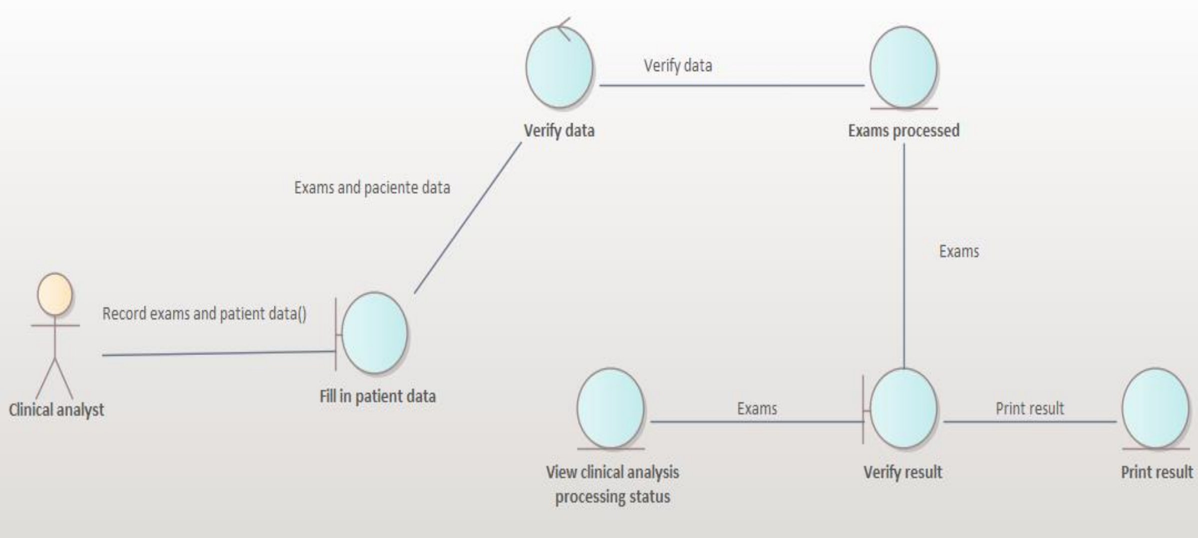


Figure 3.12: Analysis class diagram corresponding to the use case, process analysis.

The physician interacts with the interface, Request exam, and requests new analyses, and this in turn arrives at the LMS screen, where it is possible to see the analyzes requested by the re-questing physician, through the interface, view requested exam

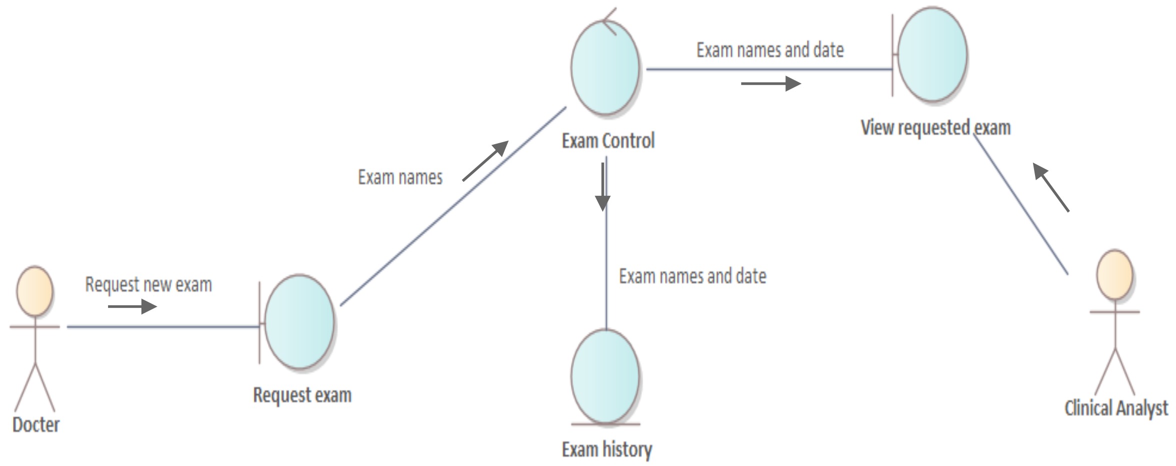


Figure 3.13: Collaboration diagram corresponding to use case, Request exam.

The clinical analyst then sees the analyzes required by the physician, he inserts the respective results after they have been validated and submits them for the applicant application.

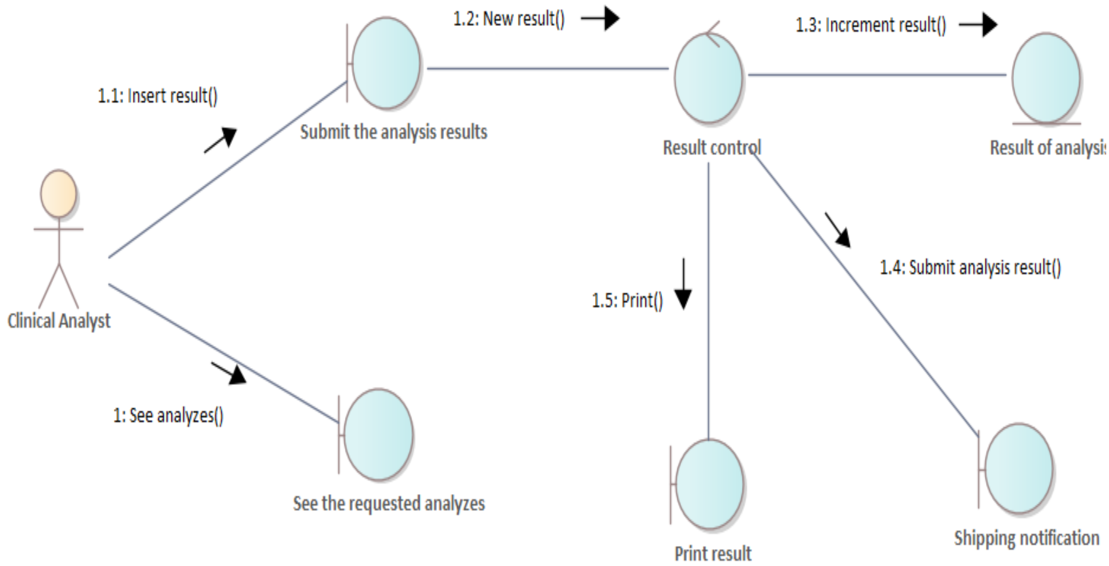


Figure 3.14: Collaboration diagram corresponding to use case, Submit analysis results.

3.4 Design

The design model is an object model that describes the physical realization of use cases centered on functional and non-functional requirements, along with other constraints related to the implementation environment, impacting the system to be considered. Furthermore, the design model serves as an abstraction of system implementations and is thus used as a fundamental input in implementation activities [33].

The design model is represented by design system which denotes the higher-level subsystem of the same in more manageable portions. Design subsystems and design classes represent subsystem abstractions and system implementation components. These abstractions are straightforward and represent a simple correspondence between design and implementation [33].

The activities contemplated in the analysis started to be deepened in the problem to be solved, so through them we represent an internal view of the system in which, using the language of the developers, the requirements are refined and structured based on classes and packages. This process continues in the design until the objects that interact to meet the functional and non-functional requirements are obtained [33]. In this session, some artifacts incorporated into the design model are demonstrated.

3.4.1 Sequence diagram request analysis

This diagram represents how the event to request analyzes from the LMS takes place, this interaction begins between the medical user and the PMS, through the medical consultation dashboard, data such as exam names are sent. Data are sent to the LMS, to be processed and subsequently sent to the requesting system, PMS.

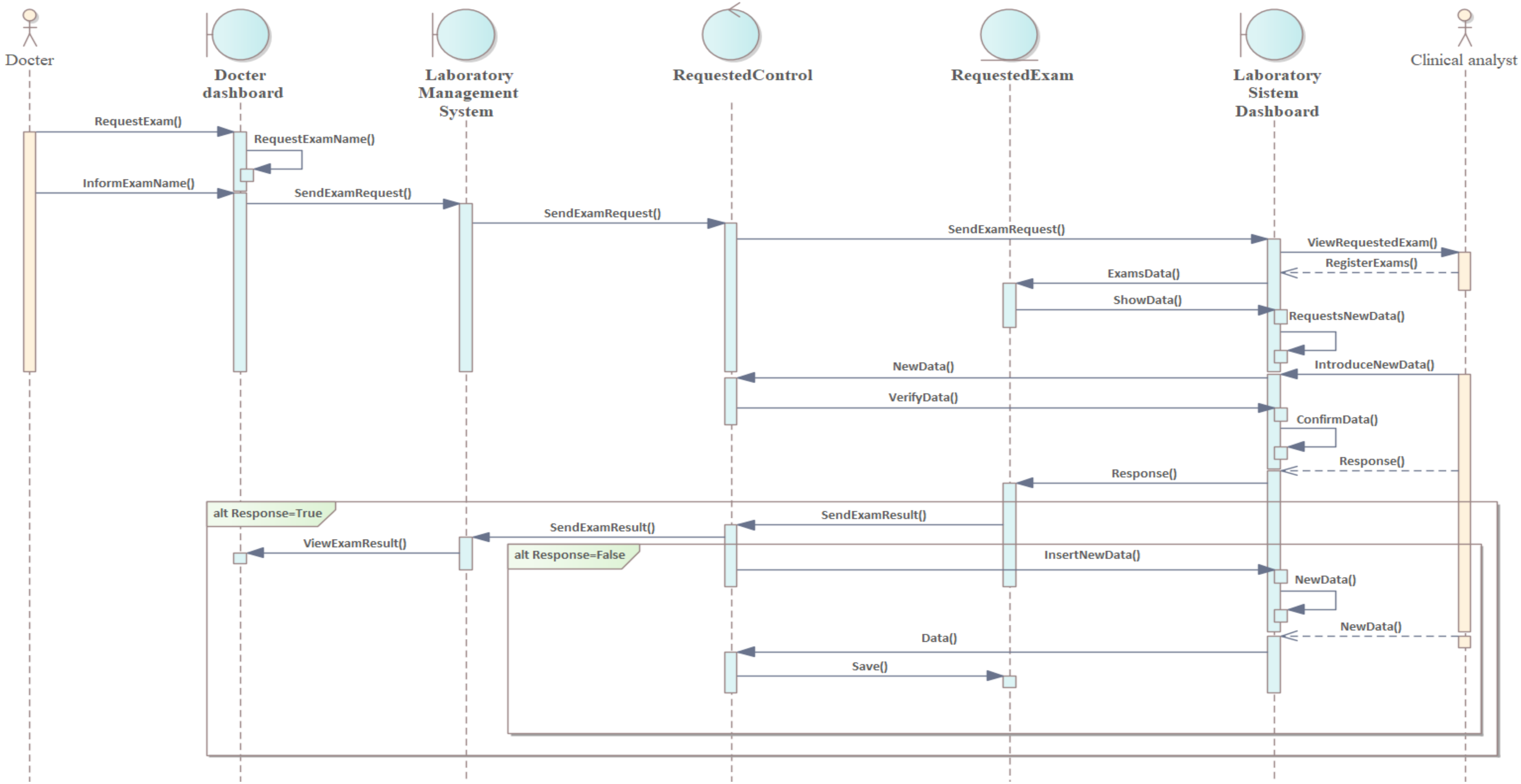


Figure 3.15: Activity diagram request analysis

3.4.2 Class Diagram, PMS

Class diagrams clearly map the structure of the system when modeling, its classes, attributes, operations and relationships between objects, as shown in figure 3.16 and figure 3.17.

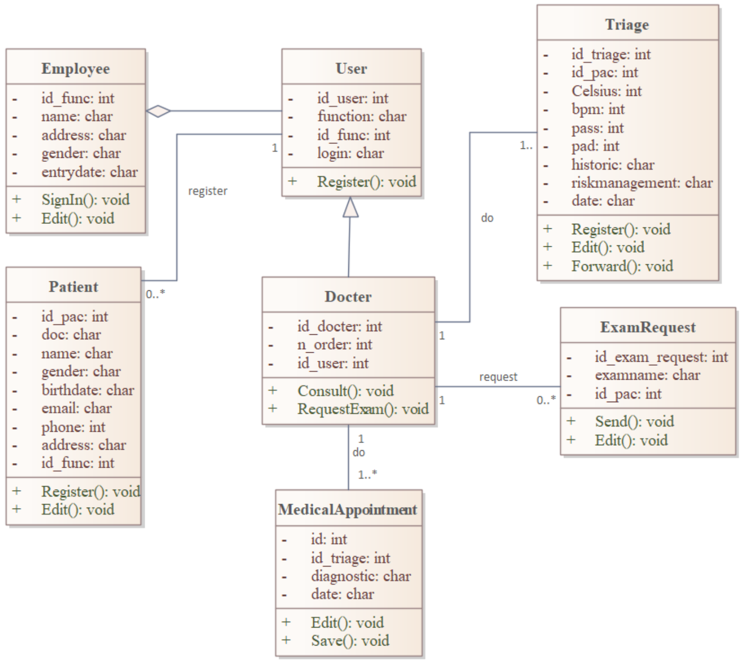


Figure 3.16: Class Diagram, PMS

3.4.3 System Class Diagram, Clinical Analysis Management

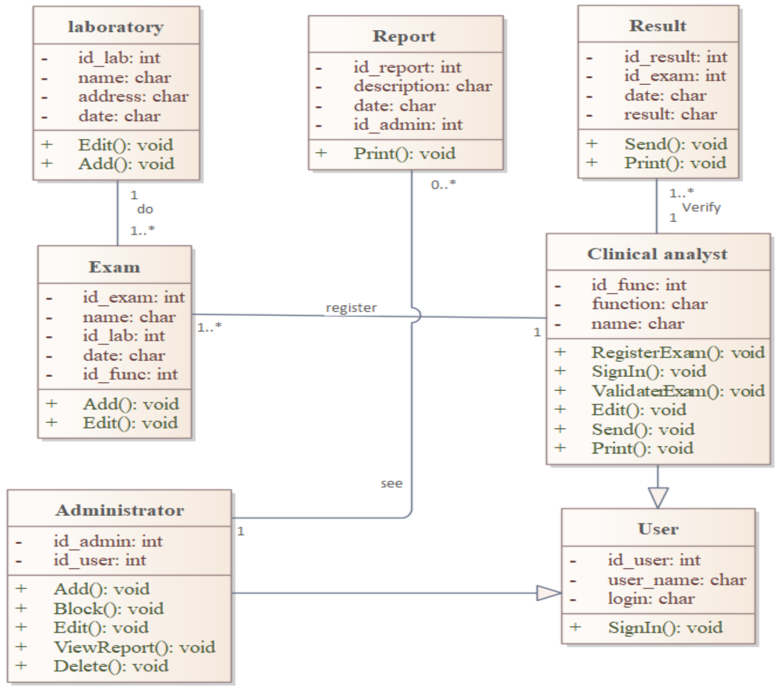


Figure 3.17: System Class Diagram, Clinical Analysis Management

3.5 Implementation

Following the steps previously presented (requirements, analysis and design), the physical configuration of the network on which our application will work must be evaluated due to the influence it has on the software architecture. Furthermore, in the case of client-server applications, the implemented layers condition or influence the determination of the nodes that are defined in the architecture. For other types of systems, it is equally important to define how the application will work and whether it is related to other devices or applications that work on a different node than the one intended for our application [33]. The implementation of tiered applications is based on message delivery and represents a modular structure that improves usability, flexibility, interoperability and scalability [33].

3.5.1 Component Diagram

The component diagram gives an overview of the software system, allows understanding the exact behavior of the service provided by each software involved in the interoperability process [51]. The login and REST API components are the components without which it would not be possible to carry out any type of operation in the system, that is, figure 3.18 illustrates that there is a dependency on the first component, which is preliminary for any subsequent action in the system. . Regarding the second component, REST API, there is a dependency that to request analyses, see the analysis request, see the results of the analyzes and send the results of the clinical analyses, this only happens with the presence of the REST API component, that is, it is the component that allows data interoperability as illustrated in figure 3.18.

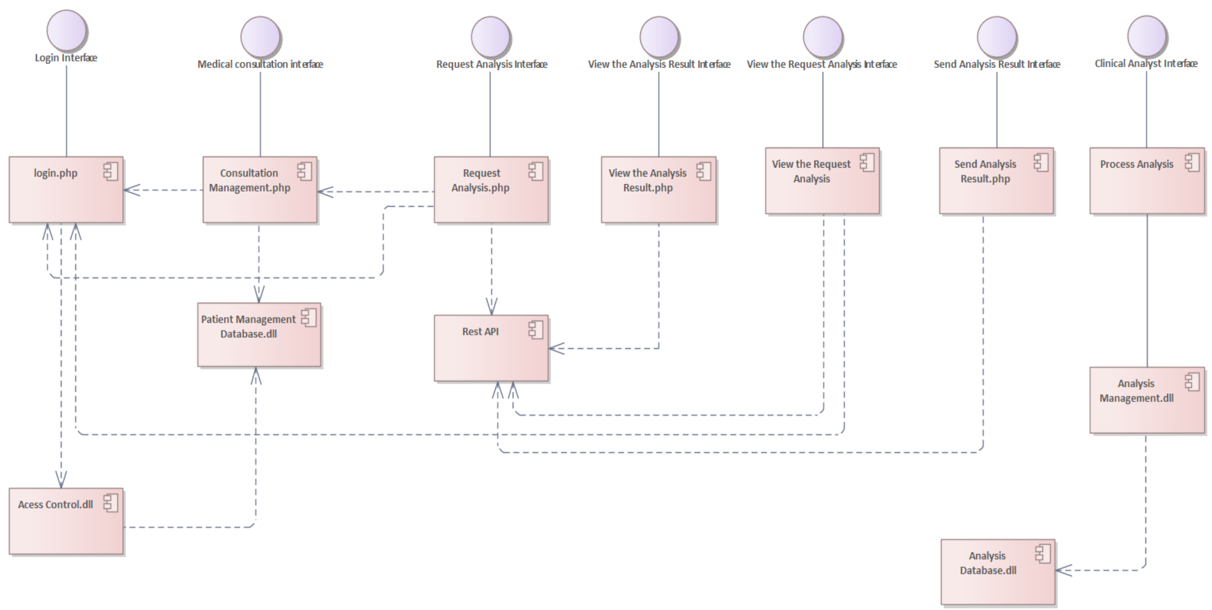


Figure 3.18: Component diagram

3.5.2 Deployment Diagram

A deployment diagram is a type of diagram that shows the architecture execution of a system, including nodes, such as hardware or software execution environments, and the middleware that connects them [51]. Using it, one can understand how the system will be physically deployed on the hardware. We have two web applications, with client-server architecture, Patient Management and Clinical Analysis Management. The application servers, Hospital Local Server and Clinical Analysis Server, communicate synchronously through the server, REST API Server, and the json format was defined for data exchange between the aforementioned servers.

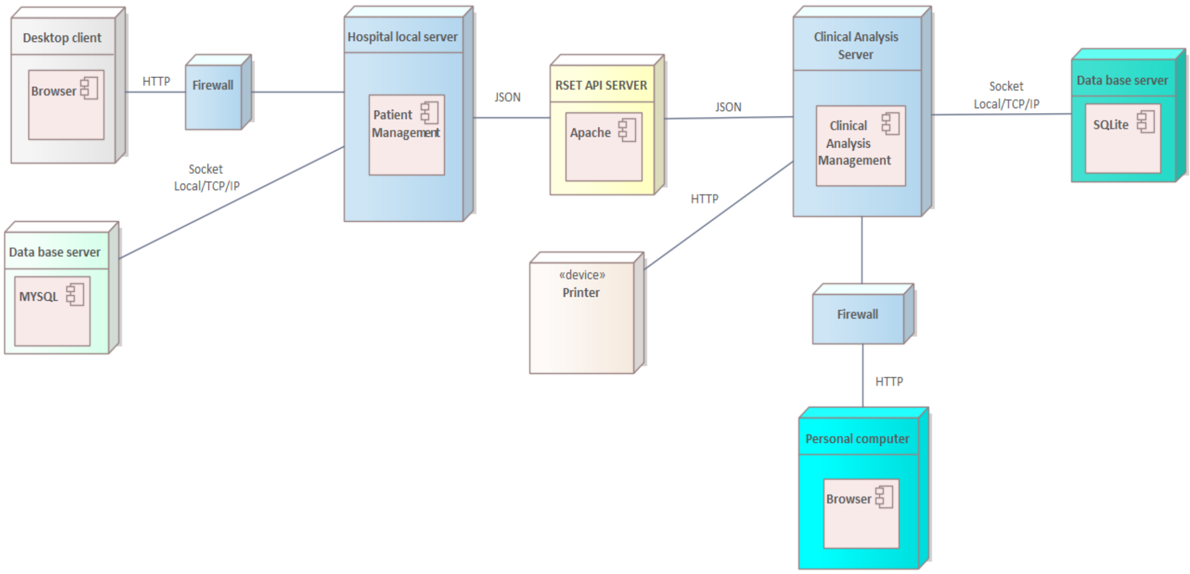


Figure 3.19: Deployment diagram

3.5.3 The web service implementation

It is important to briefly emphasize the emergence of interoperability standards since their creation and some evolutions. When talking about technical interoperability, it is inseparable to talk in some detail about SOAP and REST standards. Therefore, in the second half of the 1990s, distributed computing models such as RMI, DCOM and CORBA emerged, and were successful in the integration of applications in local and homogeneous network environments [52].

With the advance of time and the expansion of the internet, applications appeared in several heterogeneous environments, carrying the aforementioned standards that worked in a homogeneous environment no longer suited what were the requirements of the evolution of the new corporate model of trafficking. data [53]. It was then that web applications evolved (JSP, ASP, PHP) but at the time there was no consensual or single standard which made it difficult to interact between applications that had been created to be interoperable. In this adverse context, companies such as IBM, Microsoft, BEA among other W3C participants decided to standardize the implementation of these web applications, using the SOAP standard [58].

From that moment on, the W3C researchers began to create specifications for web services, involving all kinds of standardization that they considered interesting for this technology. Table 3.17 shows some specifications for web services created by the W3C.

Table 3.17: Web service Specifications.

Specification	Description	Paper
WS-Addressing	Specific transport protocol that allow web services to exchange messages.	[54,57]
WS-Eventing	Defines a protocol that a web service can both send and receive messages from other web services.	[55,57,58]
WS-Security	Protocol that provides security mechanism to web services.	[53,58]
WS-SecureConversation	It aims to allow secure conversations between sites that use web services as a means of communication.	[53,54]
WS-Policy	Specification that allows web services to use XML to publish and consume their policies (security policies, quality of service, etc).	[57,58]
WS-AtomicTransaction	It defines two types of transaction coordination: Atomic Transaction (AT) for individual operations and Business Activity (BA) for long-running transactions.	[52,55]
WS-Coordination	Defines and proves protocols that coordinate the actions of distributed applications, including distributed transactions.	[53,55]
WS-ReliableMessaging	Describes a protocol that allows SOAP messages to be reliably delivered between distributed applications in the presence of software or network failures.	[53,55]

The REST was proposed by Roy Fielding in 2000, one of the authors of the HTTP protocol, in his doctoral thesis. The REST standard was not proposed by the W3C, it was born from a doctoral thesis. Unlike SOAP, not a huge amount of specifications were created for the REST protocol, in fact, no specifications were created [54,58].

3.5.4 The Implementation of SOAP Web Services with PHP Language

We have the first system involved in interoperability, the PMS, it should be noted that this has the following workflow, the patient is registered in the system and sent for triage, after verifying their health status a risk level is assigned, from the attribution of a risk level, the patient enters a queue to be seen by the doctors, in the act of consultation the doctor can prescribe the prescription, hospitalize the patient or make the request for analyses clinics.
Our focus is on requesting clinical analyzes through the first system, Patient Management. The second system involved in interoperability is the LMS, which is instantiated in the Clinical Analysis Laboratory Center.

We start by instantiating our SOAP SERVER server, where it was necessary to configure our server, xampp, in order to recognize the SoapServer method, which will be responsible for serving, this function being a service operation, we assign the access URI so to the web service as shown in figure 3.20, so this will be the address for accessing the web service, once our server is instantiated, it is ready to be consumed by the web service in SOAP protocol.

```
$servidor = new SoapServer( null, ['uri' => 'http://localhost:80/Ap2/interop/comunicacao_covid.php'] );
$servidor->addFunction( 'comunicacaoCovid' );
$servidor->handle();
```

Figure 3.20: SOAP server instantiation

Having published the service, we can start using the PMS, we instantiate a SOAP Client (SoapClient), which also receives some parameters and we need to pass some parameters inside a vector, namely Location and the URI, these are two different options, the first is the service location and the other is the access URI. These can be the same or different. In our case, the URI and Location have the same address, this is because these operations are being performed on the same machine (localhost), once this is done, we can call the operation by triggering the method defined and passing the parameters as shown. the fig 25.

```
$cliente = new SoapClient( null, [ 'location' => 'http://localhost:80/teste/examples/comunicacao_resultado.php',
'uri' => 'http://localhost:80/teste/examples/comunicacao_resultado.php' ] );
$cliente->comunicacaoResultado($nome_analise,$id_utente,$nome);echo "oct";
//
//}
print 'Inserido os resultados de análise com sucesso.';
```

Figure 3.21: SOAP client instantiation

Two web applications were built with different databases where it was possible to implement interoperability between them using the SOAP protocol, the first application, a PMS, the doctor in the consultation process with the patient, through the consultation system dashboard, he clicks on the request analysis option, then he can enter the names of the analyses to be performed, and these data will be sent to another system, a LMS, this in turn, the clinical analyst can now see the names of the requested analysis, and it is up to the user to insert the respective result and click on send the result, and these data will be forwarded to the requesting system. At this point, the doctor, through his dashboard, can now click on the option view the analysis result, to see the results of the analyses. Figure 3.22 illustrates an application server that provides the service to be consumed by these two systems, thus allowing interoperability between them.

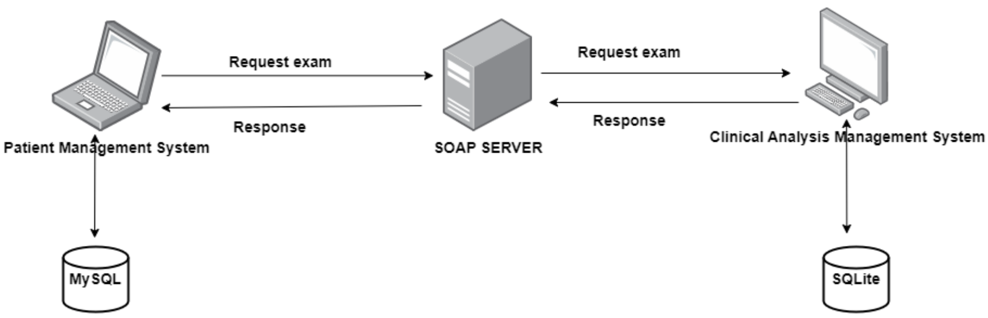


Figure 3.22: Web service consumption using the SOAP protocol

3.5.5 Implementation of REST web services with PHP language

We are presenting an application of system interoperability in this case study we perform the interoperability of Rest service and PHP language. So here we have the first system involved in interoperability, it's about workflow management in the Health Center, PMS. This system registers new patients, once registered they go to a waiting list for triage, where vital signs data and a patient history are informed, after triage the patient is sent to a medical care queue and is after the medical consultation, in the consultation process it is recurrent for the doctor to request clinical analysis exams.

The second system involved in interoperability is the system that manages data from sample collection and processing of patient or user exams at the Clinical Analysis Center (LMS).

How to send the exam request and receive the respective results if we are dealing with two different systems? This is where the possibility of an interoperability application with a

REST-type web service comes in, we will publish a service in the LMS, which sends the results of the exams upon a request made by the requesting doctor. This service will be consumed at the time of requesting exams and sending results by the PMS and LMS, as shown in Figure 3.23.

We start by creating a web service by writing a code that captures any input data through the function, $json = file_contents('php://input'), we transform this json structure into a vector to be able to manipulate it within the program, to that we use the function, json_decode($json, true), we capture in our vector, the name of the analyzes and the user number through their respective indexes, having this data added, we can now visualize them, therefore process and enter the results of the analyses.

There is a very important aspect when we are going to consume a web service, we need to know two things, the location of the web service and the input and output data format supported by the service. So we store the service access url in a variable, as the two systems are located on the same machine, the service access address is on localhost, we also know that the data structure expected as input by the web service needs to have the indices (analyze and num_uente), these indexes will receive from the request variable, the name of the analyzes and the user number. Then we convert the data structure to json format, using the function json_encode($array) passing the data vector of the request as a parameter. And now we have the two crucial pieces of information for us to consume the published service, where it is located and the data structure in json format. We then made an http request to the service with the function, curl_init($url) passing as a parameter the url available for access to the service, as shown in figure 3.23.

```
$url = "'http://localhost:80/Ap2/interop/comunicacao_covid.php'";
$array = ['analise'=>$_REQUEST['nome_analise'],'num_utente'=>$_REQUEST['id_utente']];
$json = json_encode($array);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_POSTFIELDS, $json);
curl_setopt($curl, CURLOPT_HTTPHEADER,['Content-type:application/json']);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$json = curl_exec($curl);
```

Figure 3.23: Web service consumption using the REST protocol

3.5.6 Advantages and Disadvantages of SOAP

3.5.6.1 Advantages of SOAP

The SOAP protocol continues to be widely used, table 3.18 briefly presents some of the advantages of the SOAP protocol.

Table 3.18: Advantages of the SOAP protocol

Description	Paper
SOAP web services provide built-in security and compliant transactions that meet many business needs.	[53]
SOAP can use any existing transport means to send its request, from SMTP to even JMS.	[53]
SOAP is an industry standard, with well-defined protocols and a well-established set of rules.	[54]
It fits best for applications that are not installed in the cloud and low request that may consume less bandwidth.	[54]

3.5.6.2 Disadvantages of SOAP

Despite some advantages that were mentioned in table 3.18, SOAP also has its limitations, in table 3.19 we present the disadvantages of SOAP.

Table 3.19: Some disadvantages of SOAP

Description	Paper
SOAP adds considerable overhead, because it's in XML and because it adds a lot of metadata tags. Also, serializing and deserializing messages can be time consuming, making it a slower protocol.	[53, 54]
Integrations with high volume of requests are unfeasible in SOAP	[53]
The data request is sent to a SOAP API, it can be processed through any application layer protocol, HTTP, SMTP, TCP etc. However, once the request is received, SOAP messages must be returned in XML format.	[53]
Difficult to implement and not so popular with web and mobile developers	[53,54]
Strictly define standards to effect interoperability	[52]
A browser cannot cache a completed request to a SOAP API, so it cannot be accessed later without resubmitting to the API.	[52]
Requires a lot of bandwidth for multiple requests	[54]
SOAP is not natively supported on Android	[54]

3.5.7 Advantages and Disadvantages of REST

3.5.7.1 Advantages of REST

Table 3.20: Some advantages of REST

Description	paper
REST meets more and better the needs of mobile applications and web services.	[52,53,54]
The data sent to a REST API is normally done by HTTP protocol. But REST APIs are designed to return messages in a variety of formats (HTML, XML, plain text, and JSON). JSON is a more appropriate and lightweight message format, and can be read by any person, machine and programming language, therefore it makes RESTful APIs more flexible and easier to configure.	[52,53,54]
Web Services that use REST are lightweight, flexible, faster, and ideal for more modern contexts such as IoT.	[52,53]
REST is able to handle large volume and complexity of data without difficulties.	[53]
REST makes use of a URL pattern to call and initialize the web service.	[52,53]
REST is simple to understand and can be adopted by virtually any client or server that supports HTTP/HTTPS.	[52,53,54]
REST is generally faster and uses less bandwidth. It is also easier to integrate with existing sites, without having to restructure the existing infrastructure. This allows developers to work faster and spend less time deploying the service.	[52,53,54]
The return structure is in any format defined by the developer and any browser can be used.	[52,57,58]
Rest can reduce development complexity, improve system scalability.	[53]

3.5.7.2 Disadvantages of REST

Table 3.21: Some Disadvantages of REST

Description	Paper
REST seems simple, but it is not, it requires us to think carefully about its implementation (identify resources, externalize state transitions, etc.), otherwise we will see all its benefits denied.	[54,57]
Most REST implementations do not support asynchronous invocations and events. Therefore, event-driven architecture is not for REST.	[55,58]
It only works on top of the HTTP protocol.	[52]

Based on the advantages and disadvantages of these two protocols, we can infer that both are good, but it depends a lot on the problem that is proposed to be solved, if it is for an academic, “business” environment or organization such as NATO, the use of SOAP is recommended. [54], on the other hand, when we are faced with a large number of requests, low bandwidth, use of web applications and mobile developers, it is convenient to use REST [52,5354].

4 PROJECT FEASIBILITY AND SUSTAINABILITY STUDY

The development of any project requires some care, as it may suffer from a shortage of resources and its delivery date may be compromised, so it is necessary to evaluate the feasibility and sustainability of the project before its execution, thus avoiding commitment and waste of resources. Computer projects are not exempt from this risk, to mitigate this situation, human effort, cost and development time required for their execution are estimated [39].

To carry out the project estimation, there are several methods: Analysis of Use Case Points, there is a possibility to predict the size of a system from the characteristics of its requirements, expressed in the use cases, for projects with more than 20 Use Case [39].

Function Point Analysis and COCOMO II, this method proposes a procedure to measure the effort called COCOMO II. It is based on mathematical equations that allow the effort to be calculated from certain estimated size metrics, such as Function Point Analysis and Source Line of Code (SLOC) [39].

4.1 Planning by Function Points

For the evaluation of the project, the COCOMO II model was used, applied to the RUP, estimating the effort, deadline and the average team for the Elaboration and Construction phases. The study was carried out in adjusted function points, the use cases were used to calculate the source instructions. In this way, the magnitude of the system is estimated and indicators such as the number of men, the effort, duration and cost of the project are obtained.

4.2 Transaction Identification

The system functions or functionalities obtained from the list of the system's functional requirements are listed here as transactions that will serve to estimate the size of the project in Function Points.

Transaction Table

Table 4.1: Transactions.

Transactions	Fields	Files
Authenticate Administrator (Login).	Username, Password, Command.	Administrator Account
Authenticate Doctor (Login).	Username, Password, Command.	Doctor Account
Authenticate Clinical Analyst (Login).	Username, Password, Command.	Account Analyst
Add User.	Username, Password, Category, Command.	users
Remove User.	User number, Command.	users
Open the patient's record.	Name, Date of birth, girlfriend, gender, parentage, nationality, telephone number, Command.	record
Issue Reports.	Date, Command.	Patient, Consultation, Exams
View Patient History	User number, date, Command.	Form, Exam Consultation
Consult patient	Query name, patient name, date, Command.	Consultation
Bock user	User number, Command.	User
Monitor analysis processing	Analysis number, Command.	Analysis
View the analysis request	Analysis number, Command.	Analysis
View pending analysis	Analysis number, Command.	Analysis
Send the analysis result	Result, User Number, Command.	Result
Print the result	User number, Command.	Result
Validate result	Analysis number, Command.	Validation
Process analysis	Analysis number	Analysis
Request analysis	User number, analysis names	Solicitation

4.3 Counting functions by complexity

Next, the system functions are represented, which are grouped into External Inputs, External Outputs, Queries, Internal Files, and External Interfaces. All of them are classified by their level of complexity, which can be Low, Medium and High according to the reference tables of complexity and weight of the functions [39].

4.3.1 External Entries

Table 4.2: External Input Count

The complexity of the External Input is calculated from the number of Elementary Data (RED) and the number of Referenced Logical Files (RLF)

Job Description	RED	RLF	Complexity	FP
Remove records;	3	1	Low	3
Request exams;	3	1	Low	3
Open the patient's file;	2	2	Average	4
Add user;	3	1	Low	3
Remove User;	3	1	Low	3
Consult;	4	1	Low	4
Register Doctors;	3	1	Low	3
Register Clinical Analyst.	3	1	Low	3
Remove Records;	4	1	Low	4
Update Records;	2	2	Average	4
Total	47	21		33

.

4.3.2 External Inquiries

Table 4.3: External Query Count.

Job Description	RED	RLF	Complexity	FP
View patient history	2	2	Average	4
View pending analysis	2	2	Average	4
Authenticate Administrator (Login)	3	2	Low	3
Authenticate User (Login)	3	1	Average	3
Authenticate User (Login)	4	1	Average	4
View the analysis request	3	1	Low	3
Total				21

4.3.3 External Outputs

Table 4.4: External Output Count.

Job Description	**RED**	**RLF**	Complexity	FP
Issue Reports.	5	1	Medium	5
Make information available	2	2	simple	4
Total				9

4.3.4 Internal Logical Files

Table 4.5: Internal Logical File Count

Job Description	**RED**	**RLF**	Complexity	FP
Access Control	5	1	Low	5
User Management	5	1	Low	5
Records	7	1	Medium	7
Total				17

4.3.5 External Interface Files

Table 4.6: External Interface File Count.

Job Description	**RED**	**RLF**	Complexity	FP
None	0	0	0	0
Total				0

4.3.6 Final Unadjusted Function Point Count

Table 4.7: Unadjusted Function Points Count

FP Gross values are obtained from tables 4.2, 4.3, 4.4 and 4.5.

Item	FP Gross
External Entries	33
External Outputs	9
External Inquiries	21
Internal Logical File	17
External Interface File	0
Unadjusted Function Points (UFP)	80

4.3.7 Calculation of adjusted function points

Adjustment Factor Calculation

Table 4.8: Adjustment Factor Calculation Table.

Feature	Degree
Data communication	4
Distributed Processing	3
Performance	4
Highly used configuration	2
Transaction volume	3
End user efficiency	3
complex processing	0
Reuse	3
Ease of installation	4
Ease of Operation	3
Multiple locations	4
Requested Modification	1
NI	34

AF = (NI * 0,01) + 0,33

AF = (34 * 0,01) + 0,33 = 0,67

where AF is the Adjustment Factor

a. Calculation of adjusted function points.

AFP = UFP * AF; where AFP is the Adjusted Function Point

AFP = 80 * 0,67= 53.6

AFP = 53.6 function points

b. Project Lines of Code Calculation

To calculate the source instructions, the following formula was used:
SLOC = AFP * RATIO
Ratio is a constant for the SLOC of each programming language, in this case it has a value for PHP of 75
AFP = 53.6; Ratio = 75
Calculation
SLOC = 53.6 x 75 = 4020

KSLOC = 4020 (Millions of lines of code)

4.3.8 Estimated

It was obtained from PMS, LMS and an Application Programming Interface (API) of interoperability 53.6 FP, developed by a person with a productivity of 7 hours / FP with a 6 hours workday.

4.3.9 Calculation of Effort and Time

After calculating the number of lines of code, the value of the effort calculation given by Bohem's formula was used:

$PM = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$ Formula to Get Effort, where:

$E = B + 0.01 \times \sum_{J=1}^{s} SF_j$ Formula to get Exponent of Size.

Furthermore, we have the values of A and B as constant values of 2.94 and 0.91 respectively.

To calculate the time, the formula was implemented:

$TDEV = C \times (PM)^{F}$ Formula to get the Development Time, where:

$F = D + 0.2 \times 0.01 \times \sum_{J=1}^{s} SF_j$ or $F = D + 0.2 \times (E - B)$

We also have the values of C and D as constant values of 3.67 and 0.28 respectively. To obtain the results of the previously exposed formulas, the values of each scale factor (SFj) and of each effort multiplier (EMi) were calculated.

4.3.10 Scale Factors

Table 4.9: Scale Factors

Scale Factor	Value	Justification
PREC	6,2	there is no other
FLEX	5,90	Flexible
RESL	5,65	Between 5-10 critical risks and less than 40% architect availability
TEAM	2,19	Mainly cooperative interactions, medium experience operating as a team.
PMAT	4,68	Level 3 (Nominal).

4.3.11 Scale Multipliers

Multipliers reflect capability of developers, non-functional requirements, familiarity with development platform, etc.

Table 4.10: Scale Multipliers.

IN	PERS	RCPX	RUSE	PREX	FCIL	SCED	PDIF
Value	1.26	1.00	1.00	1.12	0.87	1.43	1,00

4.3.12 Summary of calculations

- **Effort** $PM = A \times Size^{E} \times \prod_{i=1}^{n} EM$

  Data: A = 2,94; B = 0,91; Size = 5,22

  $\sum_{J=1}^{s} SF_j$ = PREC+FLEX+RESL+TEAM+PMAT= 24,62

  $E = B + 0.01 \times \sum_{j=1}^{s} SF_j$ = 1,1562

  $\prod_{i=1}^{n} EM$ = 1,75567392

  PM = 34.878 ≈35 Person / Month

- **Development time** $TDEV = C \times (PM)^{F}$

  Data: B = 0,90; C = 3,67; D = 0,28

  $F = D + 0.2 \times 0.01 \times \sum_{j=1}^{s} SF_j$ = 0,32924

  $TDEV$ = 11.831≈12 month

  Number of person NP = PM / TEDV
  NP = 35 / 12
  NP = 2.948 3≈3

- **Cost** NPM = person x average salary(AS)
  C = NPM x PM; AS=900€

  NPM = 1 x 900€

  C = 900€ x 34.878

  C = 31390,2 €

4.4 Tangible and Intangible Benefits

a. Tangible Benefits

Through the use of the system, the following benefits can be obtained in a confining way:

- Easy access to information.
- The use of a very powerful database manager that guarantees speed in processing, security and data protection.
- Reduction of physical files (papers, shelves, covers and envelopes of processes, etc).
- Increase in physical space
- Reduction in the acquisition of material to save the clinical process of patients.

b. Intangible Benefits

The systems developed will be a tool for daily use with a high level of functionality, usability, security and flexibility, allowing interoperability between two different applications, increasing the availability of information to end users in real time and, therefore, the reuse of information contained in software islands.

4.5 Cost-Benefit Analysis

The system implemented, will considerably alleviate working conditions and the disposition of information from different systems in real time, optimizing the delivery time of exams that were previously done.

In this cost-benefit analysis, taking into account the project cost, human effort and estimated development time, as well as the tangible and intangible benefits that were specified above, it can be stated that the implementation of interoperability between different computer systems in the Hospital Center Dr. Ayres de Menezes in São Tomé and Príncipe is feasible.

5 PROPOSED CLOUD MODEL

We intend to implement a business model, Infrastructure as a Service (IaaS) for São Tomé and Príncipe, with the rental of virtual servers in the cloud. We propose this solution because the power failure has been recurrent.

In the IaaS model, we have full autonomy and flexibility to increase and reduce resources, perform infrastructure configurations, firewall configurations, network management and various configurations. This means that in order to operate an IaaS environment, it is essential to have knowledge or even follow-up by a specialized IT team to carry out its maintenance and management [40].

5.1 Types of IaaS

Public IaaS includes features that all of the vendor's customers can utilize. It is, in a way, a catalog of services that users order according to their needs [40].

Private IaaS is similar to creating a virtual private datacenter for a customer, all resources are dedicated to it. To implement a private cloud, it is necessary to invest in servers and other equipment and in personnel to install, maintain and manage the cloud functionalities [40].

Hybrid IaaS is a combination of public IaaS and private IaaS. In this way, the user can maintain control of their critical applications and services, while benefiting from on-demand resources to complement specific needs [40].

5.2 Licensing type

We propose the SPLA (Services Provider License Agreement) agreement when purchasing the licenses. The Microsoft SPLA is used by service providers and independent software vendors (ISVs) to license Microsoft software and provide software services and hosted applications to customers [41].

One of the great advantages of this model is the fact that it allows the shared commercial use of the manufacturer's technological resources, that is, it offers more than one client the same infrastructure and the same manufacturer's license, which results in a better use of resources [41].

SPLA licensing has no initial cost, no joining fee and no consumption commitment. Microsoft software is provided on a similar basis to a monthly lease for a three-year contract.

The model offers several products available, from Dynamics management tools to Exchange Hosted Services and Server, Fore-frontTM, Office System, SharePoint Server, SQL Server and System Center [40].

5.3 Legal limitations

The customer may not resell or redistribute Microsoft Azure Services, nor may multiple users directly or indirectly access any functionality of Microsoft Azure Services without being authenticated to do so [40].

5.4 The institution already has a similar service in São Tomé and Príncipe

Currently there is the Institute of Innovation and Knowledge (INIC), which is under the tutelage of the Prime Minister's office, was created in 2008, whose main purpose is to implement the Information and Knowledge Society in São Tomé and Príncipe, to promote the Scientific Research, Development and Technological Innovation [42].

The INIC is responsible for formulating and implementing strategies for the integration of New Information and Communication Technologies in the processes of human activities that should contribute to sustainable development in the fields of public administration, education, health, employment, tourism, environment, agriculture, livestock, fisheries, science and to support companies in their modernization [42].

Since its creation until the present date, INIC has undertaken some actions, having successes in some and failures in others. The failure stems from the lack of a policy of centralized coordination of the entire process of implementing electronic or digital governance to avoid the multiplicity of Communication Infrastructures, Information Systems, Data Processing Centers (CPD), inefficiency in the use of resources (material, human and financial) and inefficiency in the provision of public services [42].

5.5 Architecture of a public IaaS

In this session we present a public IaaS made available through internet access with shared resources in the cloud, the infrastructure is contracted as a service and not as a product, has high scalability and is hosted on the Azure server located in South Africa. The database engine is SQL Server and the operating system to be used is Windows Server.

Windows Server is an operating system created by Microsoft to serve companies and corporations with the aim of sharing services with multiple users and provides extensive administrative control of data storage, applications and corporate networks [40].

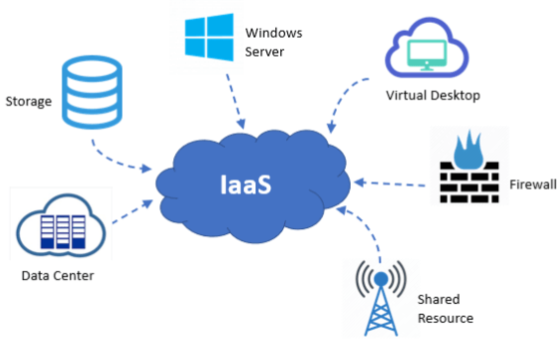


Figure 5.1: Architecture of a public IaaS

Source: https://brasiline.com.br/blog/o-que-e-nuvem-privada

To start the implementation process, we propose renting 1 virtual server with 64 GB of RAM, 16 cores CPU(s), 400 GB temporary storage in Microsoft Azure at a cost of €0.9960/hour for 3 years reserved [40], we can add more features as needed.

5.6 Comparison between Virtual and In-House Data Centers

To effectively implement interoperability between the two applications mentioned above, it is important to define the infrastructure model that meets our reality. In this session, we will briefly demonstrate the cost-effectiveness of Virtual Data Center and In-House Data Center.

5.7 Virtual Data Center Solution

This solution consists of renting a data center on the Azure cloud for the next 3 years in order to support our entire infrastructure.

Azure data center rental cost:

1 h --------0,996€

24 h-------x

In 1 day we will have a rental cost of: €23,904/server;

1 server -------23,904€/day

1 day ------- 23,904€

365 days-----x

X=8724.96€/year

In 3 years we will have a rental cost of:

1 year----- 8724,96€

3 year ----x

x = 26174,88€

For 1 cloud server we will have a cost of €26174.88 over the next 3 years.

5.8 In-House Data Center Solution

Table 5.1 shows a description of server costs with the same characteristics of the data centre in the cloud, energy consumption costs and other inherent costs.

Table 5.1: In-house data center cost description

Items	**Energy consumption**	**Paper**
Server NAS QNAP TVS-h1288X	97,4W/h	[44]
Refrigeration system	48,6 W/h	[45]
Network components	6W/h	[45]
Lighting	2w/h	[45]
Electrical security system	10 Wh	[45]
Fire prevention	3 Wh	[44]
Four technicians	Installation and configuration cost	
A 1MW generator A 1MW generator	A generator to fill the power failure	
Diesel/L	700L/month	
Air conditioning	Acquisition of 2 air conditioners, 7500 BTUs	

TCO = Capex + Opex

TCO = Server cost + Refrigeration system + network components + Lighting + electrical security system + Fire prevention + Technical + Generator Acquisition + Diesel + Acquisition of Air Conditioning.

Table 5.2: Total sum of in-house data center costs

Itens	**Average consumption**	**Value**
Cost of Purchasing a Server, NAS QNAP TVS-h1288X	2315€	2315€
Server NAS QNAP TVS-h1288X	0,13€/h	9116,64€
Refrigeration system	0,13€/h	4550€
Network components	0,13€/h	561,6€
Lighting	0,13€/h	187.2€
Electrical security system	0,13€/h	936€
Fire prevention	0,13€/h	280,8€
Four technicians for installation and maintenance	0,13€/h	4400€
A 1MW generator	75000€	75000€
Diesel/L	1,2€/L	840€
Two air conditioners	279€	558€
Total		**98821,2€**

Based on the calculations of the data presented, for the next 3 years we will have the cost of IaaS with data center rental in the cloud in the amount of €26,174.88 (twenty-six thousand, one hundred and seventy-four euros and eighty-eight cents), while the in-house solution is implemented at €98,821.2 (ninety-eight thousand, eight hundred and twenty-one euros and two cents) for the initial phase.

Therefore, for a country with so much energy instability and limited financial resources, the proposal for an in-house solution would not be economically viable. With this amount it would not be sustainable to keep this infrastructure in full operation, on the other hand there is no way of obtaining revenue from this infrastructure in order to meet the expenses inherent to the data centre, hence our proposal of Current solution is Microsoft Azure data center rental.

With the implementation of a proposed solution, the IaaS model of cloud computing will make available computing resources such as processing, memory, storage, network, database and servers accessed via the internet.

5.9 Benefits of the IaaS model:

Table 5.3: Some benefits of IaaS

Description	**Paper**
There is no need for large initial investments, you can set up the infrastructure and grow according to demand.	[43]
IaaS accompanies the customers' needs, it is possible to easily reduce the number of contracted resources, thus avoiding the idleness of the equipment.	[47]
New resources such as servers, databases, can be hired in a matter of seconds or minutes.	[46]
The Customer may choose to replicate its data in other physically separate Availability Zones, which reduces the risk of its business being affected by disaster, equipment failure, network, etc.	[43]
The customer can also reduce latency by using servers in regions where they want to make their applications available.	[47]
Better use of the IT team, with the provider being responsible for maintaining the infrastructure, the IT team can take care of other tasks that bring more value to the business.	[47]
IaaS is applied in a variety of ways to optimize and solve business problems, namely storage, backup, big data analysis, high-performance computing, web application support and website hosting.	[46]

After an exhaustive feasibility study in which the estimation method based on function points was applied to estimate the human effort, development time and cost that will have to be assumed in carrying out the project, the main benefits, both real and as intangibles, a cost-benefit analysis was carried out.

6 CONCLUSION

In this work we proposed to establish the interoperability between two heterogeneous software, PMS and LMS. To achieve the objective, it was necessary to understand several levels of interoperability and the protocols that best suit the problem to be solved. For this one in particular we apply the SOAP protocols and then REST. The greatest emphasis was given to technical-level interoperability, as this is the focus of this work, but there are dependencies and challenges to be overcome between the different levels of interoperability.

A challenge highlighted in the research question, on access to sensitive patient data, where we found that despite the legal implications, there is an exception when it comes to taking care of patients' health, in this way, manipulation of sensitive data without authorization is allowed, but keeping this data solely and exclusively for professional health care purposes [49].

With the implementation of an interoperability service between the aforementioned systems, it is possible to safeguard sensitive information, such as the patient's pathological state, as São Tomé and Príncipe is a small country with 1001 km^2 and about 197700 (one hundred ninety-seven thousand and seven hundred) inhabitants [48], therefore, any information, whether good or bad, can easily be extended to the national territory, so the results of clinical analyzes that were once folded and stapled on paper no longer will be unclipped and opened by third parties (medical assistant, family members, patient companions, etc.) without first going through the referring physician, in this way we would literally eliminate the possibility of disclosure of patient health data by a third party without clinical responsibility.

Therefore, having understood and implemented the interoperability between two heterogeneous software, we will follow the same logic to effect the integration of other software within the national health system of São Tomé and Príncipe, reducing duplication and discrepancy between data, thus increasing greater availability of real-time data from other platforms.

6.1 Proposal for future work

We intend to extend the integration of information contained in several islands of existing software, whether public or private, through the consumption of interoperability service in the national health service of São Tomé and Príncipe.

We also intend to implement a function that, with the interruption of the internet, it will be possible for users to continue the process of inserting and editing data, and subsequent synchronization of the data with the restoration of the internet.

REFERENCES

[1] A. Mendes, S. Loss, E. Cavalcante, F. Lopes, and T. Batista, “Mandala: An agent-based platform to support interoperability in systems-of-systems,” *Proc. - Int. Conf. Softw. Eng.*, pp. 21–28, 2018, doi: 10.1145/3194754.3194757.

[2] A. L. Barbieri, O. Fadare, L. Fan, H. Singh, and V. Parkash, “Challenges in communication from referring clinicians to pathologists in the electronic health record era,” *J. Pathol. Inform.*, vol. 9, no. 1, pp. 1–24, 2018, doi: 10.4103/jpi.jpi.

[3] R. Meyer and C. Lovis, “Interoperability in hospital information systems: A return-on-investment study comparing CPOE with and without laboratory integration,” *Stud.* Health Technol. Inform., vol. 169, pp. 320–324, 2011, doi: 10.3233/978-1-60750-806-9-320.

[4] N. Zeinali, A. Asosheh, and S. Setareh, “The conceptual model to solve the problem of interoperability in health information systems,” *2016 8th Int. Symp. Telecommun. IST 2016*, pp. 684–689, 2017, doi: 10.1109/ISTEL.2016.7881909.

[5] F. Lopes, S. Loss, A. Mendes, T. Batista, and R. Lea, “SoS-centric Middleware Services for Interoperability in Smart Cities Systems,” Proc. 2nd Int. Work. Smart, Smart Cities 2016, 2016, doi: 10.1145/3009912.3009917.

[6] M. M. El Benany and O. El Beqqali, “SOA based e-government interoperability,” Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA, vol. 2016-July, pp. 1–6, 2016, doi: 10.1109/AICCSA.2015.7507251.

[7] E. Lee, Y. D. Seo, S. R. Oh, and Y. G. Kim, “A Survey on Standards for Interoperability and Security in the Internet of Things,” *IEEE Commun. Surv. Tutorials*, vol. 23, no. 2, pp. 1020–1047, 2021, doi: 10.1109/COMST.2021.3067354.

[8] J. vom Brocke, A. Hevner, and A. Maedche, “Introduction to Design Science Research,” no. September, pp. 1–13, 2020, doi: 10.1007/978-3-030-46781-4_1.

[9] J. E. S. Segundo, M. F. Silva, and D. L. Martins, “Revisiting interoperability in the context of digital accounts,” *Inf. e Soc.*, vol. 29, no. 2, pp. 61–84, 2019, doi: 10.22478/ufpb.1809-4783.2019v29n2.38107.

[10] M. L. de A. Campos and N. T. Barbosa, “Interoperabilidade semântica: proposta metodológica para o mapeamento semântico entre SOCs em sistemas heterogêneos,” *Informação Soc. Estud.*, vol. 30, no. 4, pp. 1–21, 2020, doi: 10.22478/ufpb.1809-4783.2020v30n4.57046.

[11] R. Alonso-calvo et al., “A semantic interoperability approach to support integration of gene expression and clinical data in breast cancer,” Comput. Biol. Med., vol. 87, no. January, pp. 179–186, 2017, doi: 10.1016/j.compbiomed.2017.06.005.

[12] R. A. Hardie, D. Moore, D. Holzhauser, M. Legg, A. Georgiou, and T. Badrick, “Informatics External Quality Assurance (IEQA) down Under: Evaluation of a pilot implementation,” *J. Lab. Med.*, vol. 42, no. 6, pp. 297–304, 2018, doi: 10.1515/labmed-2018-0319.

[13] T. M. de Farias, A. Roxin, and C. Nicolle, “A brief overview of semantic interoperability for enterprise information systems,” Lect. Notes Bus. Inf. Process., vol. 273, no. Mdi, pp. 38–52, 2018, doi: 10.1007/978-3-319-73459-0_3.

[14] N. Bin M. Ibrahim and M. F. Bin Hassan, “A survey on different interoperability frameworks of SOA systems towards seamless interoperability,” Proc. 2010 Int. Symp. Inf. Technol. - Syst. Dev. Appl. Knowl. Soc. ITSim’10, vol. 3, pp. 1119–1123, 2010, doi: 10.1109/ITSIM.2010.5561617.

[15] A. Kouroubali and D. G. Katehakis, “The new European interoperability framework as a facilitator of digital transformation for citizen empowerment,” *J. Biomed. Inform.*, vol. 94, no. April, p. 103166, 2019, doi: 10.1016/j.jbi.2019.103166.

[16] T. M. de Farias, A. Roxin, and C. Nicolle, “A brief overview of semantic interoperability for enterprise information systems,” *Lect. Notes Bus. Inf. Process.*, vol. 273, pp. 38–52, 2018, doi: 10.1007/978-3-319-73459-0_3.

[17] A. Fay, “Interoperability rules for heterogenous Multi-Agent Systems.”

[18] R. Ellerweg, D. Reuter, and P. Weir, “Architecture of a web-based DICOM viewer showing segmentations and simulations,” *2016 IEEE 18th Int. Conf. e-Health* Networking, Appl. Serv. Heal. 2016, pp. 6–10, 2016, doi: 10.1109/HealthCom.2016.7749521.

[19] S. C. Jepsen, T. I. Mørk, J. Hviid, and T. Worm, “A Pilot Study of Industry 4. 0 Asset Interoperability Challenges in an Industry 4. 0 Laboratory,” pp. 571–575, 2020.

[20] M. Buzzi, F. Gennai, C. Petrucci, and I.- Roma, “Interoperability Challenge of Certified Communication Systems via Internet,” pp. 166–171, 2017.

[21] G. Santos, T. Pinto, Z. Vale, and J. M. Corchado, “Semantic Interoperability for Multiagent Simulation and Decision Support in Power Systems,” *Commun. Comput. Inf. Sci.*, vol. 1472 CCIS, pp. 215–226, 2021, doi: 10.1007/978-3-030-85710-3_18.

[22] A. Shajjar, N. Khalid, H. F. Ahmad, and H. Suguri, “Service interoperability between

agents and semantic web services for nomadic environment," *Proc. - 2008* IEEE/WIC/ACM Int. Conf. Intell. Agent Technol. IAT 2008, pp. 583–586, 2008, doi: 10.1109/WIIAT.2008.327.

[23] A. Hussain and W. Wu, "Sustainable interoperability and data integration for the IoT-based information systems," Proc. - 2017 IEEE Int. Conf. Internet Things, IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, vol. 2018-January, pp. 824–829, 2018, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.126.

[24] M. Alhajeri and S. G. S. Shah, "Limitations in and Solutions for Improving the Functionality of Picture Archiving and Communication System: an Exploratory Study of PACS Professionals' Perspectives," *J. Digit. Imaging*, vol. 32, no. 1, pp. 54–67, 2019, doi: 10.1007/s10278-018-0127-2.

[25] S. Lu, S. Repo, M. Salmenperä, J. Seppälä, and H. Koivisto, "Using IEC CIM standards and SOA Technology for Coordinated Voltage Control Application," *Proc. 2019 IEEE PES Innov. Smart Grid Technol. Eur. ISGT-Europe 2019*, no. Cvc, 2019, doi: 10.1109/ISGTEurope.2019.8905541.

[26] E. Emanuela Boros, A. Toumi, E. Rouchet, B. Abadie, D. Stutzmann, and C. Kermorvant, "Automatic page classification in a large collection of manuscripts based on the international image interoperability framework," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 756–762, 2019, doi: 10.1109/ICDAR.2019.00126.

[27] R. E. Izzaty, B. Astuti, and N. Cholimah, "済無No Title No Title No Title," Angew. Chemie Int. Ed. 6(11), 951–952., pp. 5–24, 1967.

[28] J. Fernandes, F. Ferreira, F. Cordeiro, V. G. Neto, and R. Santos, "How can interoperability approaches impact on systems-of-information systems characteristics?" *ACM Int. Conf. Proceeding Ser.*, 2020, doi: 10.1145/3411564.3411621.

[29] L. C. Weiss, "Interoperabilidade semântica: uma análise das perspectivas teóricas dos estudos desenvolvidos na área de Ciência da Informação," *Em Questão*, vol. 27, no. 3, pp. 431–457, 2021, doi: 10.19132/1808-5245273.431-457.

[30] F. Adebesin, R. Foster, P. Kotzé, and D. Van Greunen, "A Review of Interoperability Standards in E-health and Imperatives for their Adoption in Africa," *South African Comput. J.*, vol. 50, no. July, 2013, doi: 10.18489/sacj.v50i1.176.

[31] H. L'Amrani, B. E. Berroukech, Y. El Bouzekri El Idrissi, and R. Ajhoun, "Toward interoperability approach between federated systems," ACM Int. Conf. Proceeding

Ser., vol. Part F129474, 2017, doi: 10.1145/3090354.3090391.

[32] S. Dowell, A. Barreto, J. B. Michael, and M. T. Shing, “Cloud to cloud interoperability,” Proc. 2011 6th Int. Conf. Syst. Syst. Eng. SoSE Cloud Comput. Smart Grid, Cyber Secur. SoSE 2011, pp. 258–263, 2011, doi: 10.1109/SYSOSE.2011.5966607.

[33] T. Report, A. Hern, and A. Echeverr, “Autora : Dra . Anaisa Hernández González Facultad de Ingeniería Industrial Instituto Superior Politécnico José Antonio Echeverría Mayo 2004,” no. January 2004, 2017.

[34] B. P. Management, Evolution of Enterprise Systems Architectures. 2007.

[35] V. Lalanne, M. Munier, and A. Gabillon, “Information security risk management in a world of services,” Proc. - Soc. 2013, pp. 586–593, 2013, doi: 10.1109/SocialCom.2013.88.

[36] N. Tutubala and T. E. Mathonsi, “A Hybrid Framework to Improve Data Security in Cloud Computing,” Big Data, Knowl. Control Syst. Eng. - Proc. 7th Int. Conf. BdKCSE 2021, pp. 11–15, 2021, doi: 10.1109/BdKCSE53180.2021.9627294.

[37] R. Saripalle, C. Runyan, and M. Russell, “Using HL7 FHIR to achieve interoperability in patient health record,” *J. Biomed. Inform.*, vol. 94, no. May, p. 103188, 2019, doi: 10.1016/j.jbi.2019.103188.

[38] S. Al Idrus, W. Nur Hidayat, and A. Hamdan, “Naya as a tool of software cost automatic analysis,” *4th Int. Conf. Vocat. Educ. Training, ICOVET 2020*, pp. 258–263, 2020, doi: 10.1109/ICOVET50258.2020.9230249.

[39] W. D. Sunindyo and C. Rudiyanto, “Improvement of COCOMO II Model to Increase the Accuracy of Effort Estimation,” *Proc. Int. Conf. Electr. Eng. Informatics*, vol. 2019-July, pp. 140–145, 2019, doi: 10.1109/ICEEI47359.2019.8988909.

[40] M. Corporation and E. Contrato, “Contrato de Subscrição Online da Microsoft – Nuvem Governamental dos Estados Unidos,” pp. 1–24, 2022.

[41] D. De Grupo, “Histórico do Windows Server,” vol. 2003, pp. 2008–2009, 2008.

[42] http://inic.gov.st/inic.php/ (Accessed: 21 March 2022).

[43] https://azure.microsoft.com/pt-pt/pricing/details/virtual-network/ (Accessed: 2 April 2022).

[44] G. Zhabelova, A. Yavarian, and V. Vyatkin, “Data center power dynamics within the settings of regional power grid,” IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA, vol. 2015-October, 2015, doi: 10.1109/ETFA.2015.7301633.

[45] J. Choi, S. Govindan, J. Jeong, B. Urgaonkar, and A. Sivasubramaniam, “Power consumption prediction and power-aware packing in consolidated environments,” *IEEE Trans. Comput.*, vol. 59, no. 12, pp. 1640–1654, 2010, doi: 10.1109/TC.2010.91.

[46] V. Getov, “Security as a service in smart clouds - Opportunities and concerns,” *Proc. - Int. Comput. Softw. Appl. Conf.*, pp. 373–379, 2012, doi: 10.1109/COMPSAC.2012.112.

[47] M. Bouchaala, C. Ghazel, L. A. Saidane, and F. Kamoun, “End to end cloud computing architecture based on a novel classification of security issues,” *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2017-October, no. September 2011, pp. 303–310, 2018, doi: 10.1109/AICCSA.2017.136.

[48] https://www.ine.st/index.php/o-pais/sobre-o-pais

[49] C. Stach and B. Mitschang, “The secure data container: An approach to harmonize data sharing with information security,” Proc. - IEEE Int. Conf. Mob. Data Manag., vol. 2016-July, pp. 292–297, 2016, doi: 10.1109/MDM.2016.50.

[50] S. K. Y. Donzia, H. K. Kim, and Y. P. Geum, “Implementation of Autoware Application to real-world Services Based Adaptive Big Data Management System for Autonomous Driving,” Proc. - 2021 21st Int. Conf. Comput. Sci. Its Appl. ICCSA 2021, pp. 251–257, 2021, doi: 10.1109/ICCSA54496.2021.00042.

[51] H. Sastypratiwi, “Web Application Development using MVC- component-based approach,” no. Mvc, 2019.

[52] H. Su, B. Cheng, T. Wu, and X. Li, “Mashup service release based on SOAP and REST,” Proc. 2011 Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2011, vol. 2, pp. 1091–1095, 2011, doi: 10.1109/ICCSNT.2011.6182150.

[53] X. Shi, “Sharing service semantics using SOAP-based and REST Web services,” IT Prof., vol. 8, no. 2, pp. 18–24, 2006, doi: 10.1109/MITP.2006.48.

[54] T. H. Bloebaum and F. T. Johnsen, “Exploring SOAP and REST communication on the Android platform,” Proc. - IEEE Mil. Commun. Conf. MILCOM, vol. 2015-December, pp. 599–604, 2015, doi: 10.1109/MILCOM.2015.7357509.

[55] Y. Y. Peng, S. P. Ma, and J. Lee, “REST2SOAP: A framework to integrate SOAP services and RESTful services,” IEEE Int. Conf. Serv. Comput. Appl. SOCA’ 09, pp. 106–109, 2009, doi: 10.1109/SOCA.2009.5410458.

[56] H. E. S. Wr et al., “0ljudwlqj :he $ssolfdwlrqv wr &orxgv zlwk 0lfurvhuylfh

$ufklwhfwxuhv," pp. 7–10.

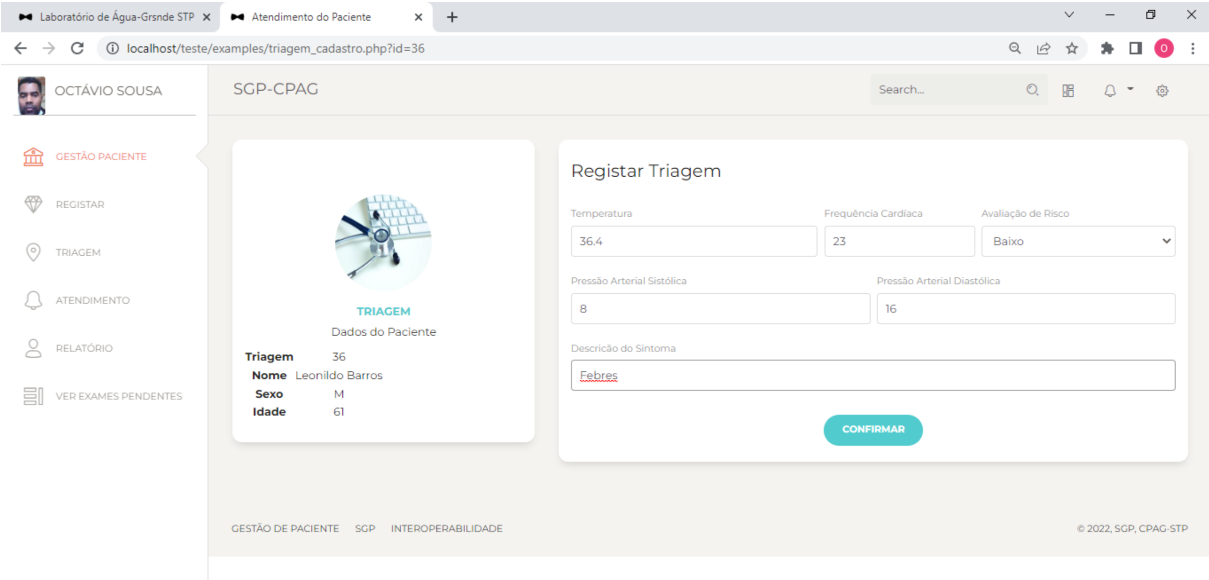
[57] X. Shi, "Sharing service semantics using SOAP-based and REST Web services," no. April, 2006.

[58] M. Pagni, J. Hau, and H. Stockinger, "A multi-protocol bioinformatics web service: Use SOAP, take a REST or GO with HTML," Proc. CCGRID 2008 - 8th IEEE Int. Symp. Clust. Comput. Grid, pp. 728–734, 2008, doi: 10.1109/CCGRID.2008.28.

Attachment

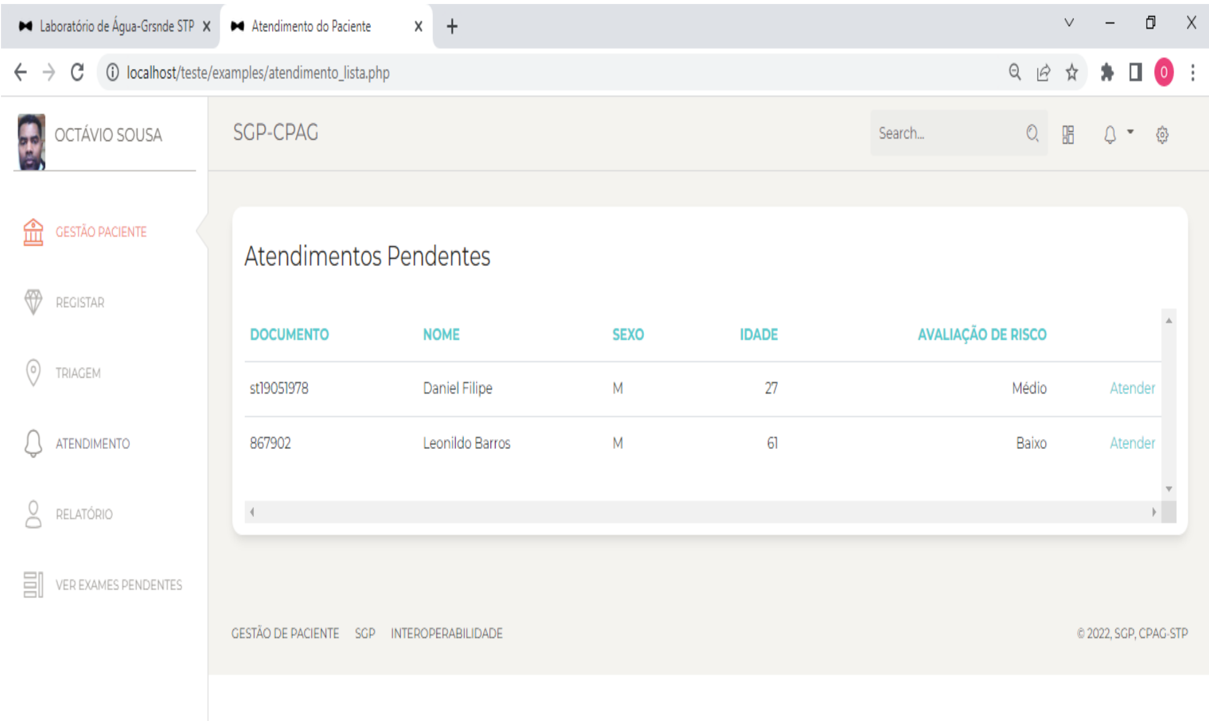
Triage

This is the page that allows us to triage patients admitted to the emergency room.



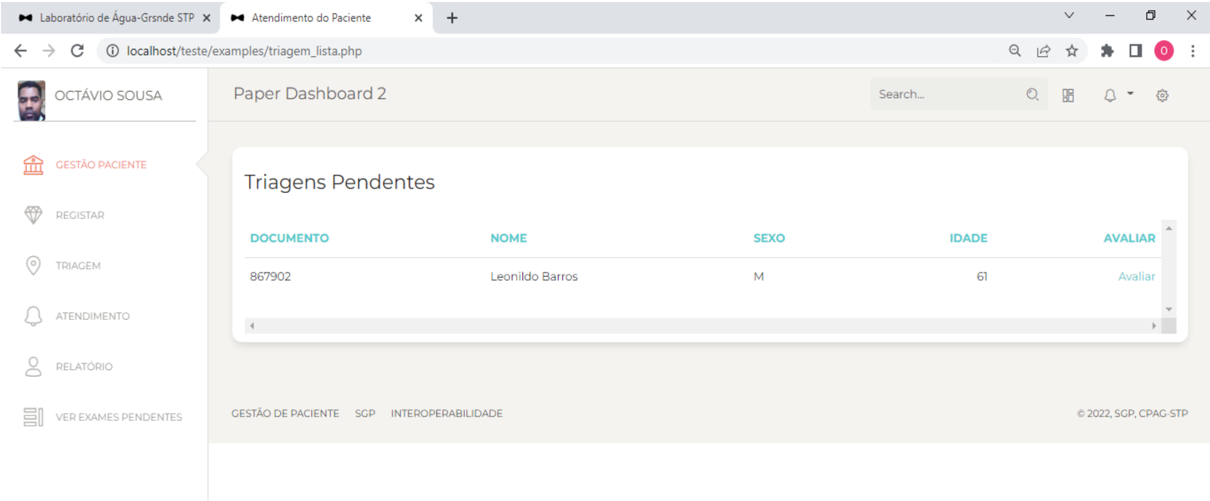
Consultation waiting list

Through this interface, it is possible to list and organize the patients according to their priorities.



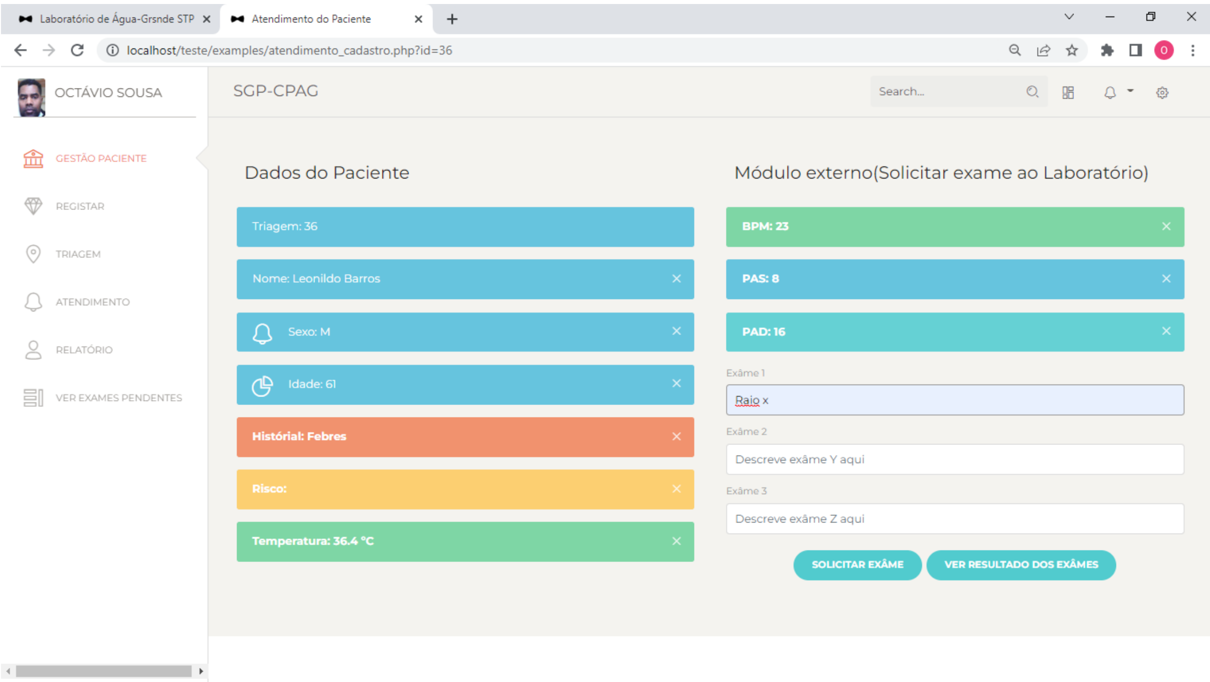
Waiting List

The drop-down list allows us to list all pending screenings.



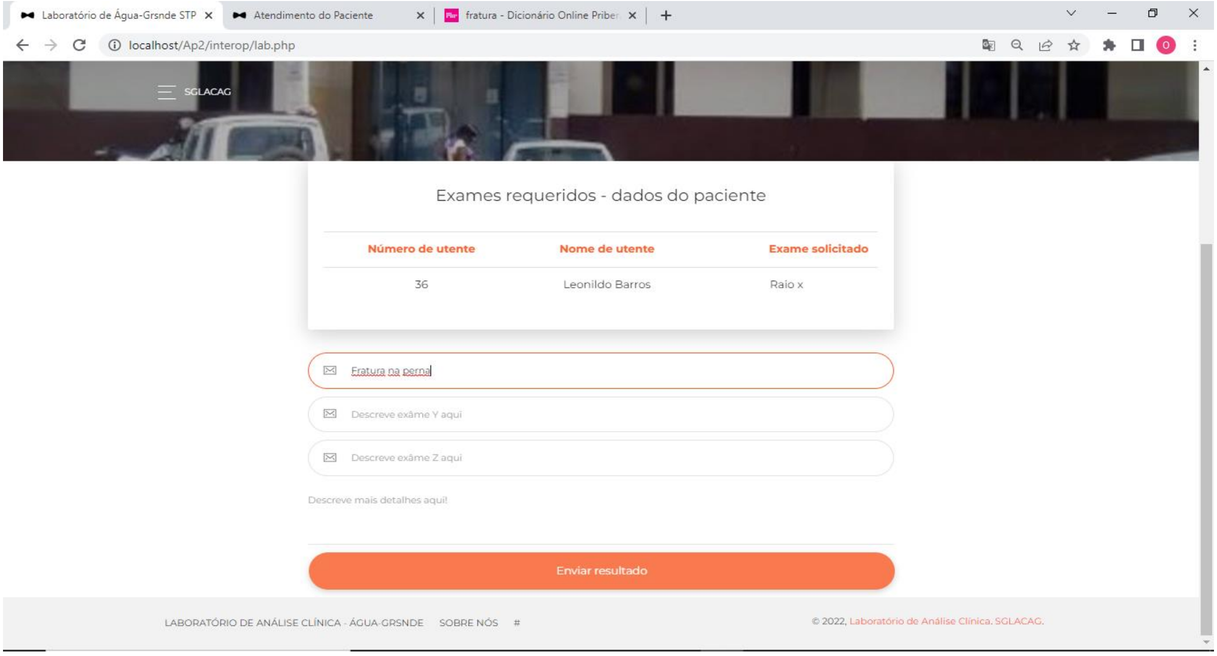
Request new exam

The doctor in his office can see the patient's history, being able to request new exams through the request exam button and see the respective results by clicking on the option “Ver resultados dos exames”.



Send exam results

After processing the test results, the clinical analyst sends them through this interface of the Laboratory Management System, which is instantiated in the clinical analysis laboratory.



See exam results

This dashboard allows you to see the results of the exams that were requested by the doctor, through the view results button.