IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

JOÃO FILIPE
MARTINS PEREIRA

SIREPH: An Integrated Approach to the Pre-Hospital Emergency Framework

IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

Dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Science in **Software Engineering**

Adviser: João Ventura
Assistant Professor, Polytechnic Institute of Setubal

Co-adviser: José Sena Pereira
Assistant Professor, Polytechnic Institute of Setubal

November, 2022

SIREPH: An Integrated Approach to the Pre-Hospital Emergency Framework

To Kika, forever in my heart.

Acknowledgements

First of all, I would like to thank the *Instituto Politécnico de Setúbal*, as it was in this establishment that I pursued my academic studies and grew as a person and as a professional.

I wholeheartedly thank professor André Sabino, for the availability, dedication and and competence, which was instrumental for the conclusion of this milestone in my academic path. The time invested in our weekly meetings is something that I believe I will never quite be able to repay.

To my esteemed colleague and friend André Alves for being present in this stage of my life. With him I learned, laughed, vented and shared all the ups and downs of our academic journey.

To my lifelong friends, namely Diogo Costa, Costel Tabarcea, André Vaz, Miguel Machado, Inês Henriques a sincere thanks for all the beautiful moments that we spent together and that we will keep on spending.

To my dearest Rafaela Reis, for all the patience, guidance and love with which you support me every day to achieve my goals. One day we shall have our animal farm.

Last but not least, an enormous thank you to my parents, sister and grandparents for always being there when it matters and providing unconditional love. To all the rest of the family, thank you for always believing in me.

To all, a big thank you.

"And this I believe: that the free, exploring mind of the individual human is the most valuable thing in the world. And this I would fight for: the freedom of the mind to take any direction it wishes, undirected. And this I must fight against: any idea, religion, or government which limits or destroys the individual. This is what I am and what I am about." (John Steinbeck)

Abstract

Pre-hospital emergency care is one of the most important phases in the healthcare system framework, as it is the first point of contact between patients and healthcare professionals. In Portugal, there is still a considerable amount of human dependent processes that could be automated to avoid unnecessary miscommunication and to improve the quality of the service provided. The main areas of challenge lie on the communication between emergency centrals, between emergency centrals and the technician's on the field and in the patient pre-hospital emergency information handover phase to the target hospital.

As these interactions take place in environments where time is essential and the possibility of committing errors is considerable, relying exclusively on human processes may hinder the entire framework. This thesis presents the analysis, design and implementation of a proof of concept solution that tackles the deficiencies identified in process automation in the pre-hospital emergency framework, and aims to improve the service quality provided by the professionals in this field.

The system follows a service-oriented architecture, were a REST API supports several user interfaces. All web-based interfaces are implemented with a low code platform. This thesis is focused on 3 main distinguishable actors: administrative emergency centrals (emergency dispatch centers at national level); non-administrative emergency centrals (firefighter and red cross stations); and hospitals.

We conducted trials with 25 users to validate the reliability of the system, particularly of the API, and the suitability of the user interfaces. Results show that the system accurately captures the pre-hospital emergency information workflow, and the several user interfaces were positively reviewed by all users.

Keywords: Emergency Management, Low-Code Development, Service Oriented Architecture, Software Engineering

Resumo

Os serviços de urgência pré-hospitalares são uma das fases mais importantes no âmbito do sistema de saúde, uma vez que são o primeiro ponto de contacto entre os doentes e os profissionais de saúde. Em Portugal, existe ainda uma quantidade considerável de processos dependentes do ser humano que poderiam ser automatizados para evitar erros de comunicação indesejados e para melhorar a qualidade do serviço prestado. As principais áreas de desafio residem na comunicação entre os centros de emergência, entre os centros de emergência e os técnicos no terreno e na fase de entrega de informação de emergência pré-hospitalar do paciente ao hospital de destino.

Como estas interações têm lugar em ambientes onde o tempo é essencial e a possibilidade de cometer erros é considerável, confiar exclusivamente em processos humanos pode dificultar todo o funcionamento do serviço. Esta tese apresenta a análise, concepção e implementação de uma solução de prova de conceito que aborda as deficiências identificadas na automatização de processos no quadro de emergência pré-hospitalar, e visa melhorar a qualidade do serviço prestado pelos profissionais nesta área.

O sistema segue uma arquitectura orientada para serviços, onde uma API REST suporta várias interfaces de utilizador. Todas as interface web são implementadas com uma plataforma de low-code. Esta tese centra-se em 3 intervenientes principais distintos: centrais administrativas de emergência (centros de despacho de emergência a nível nacional); centrais não administrativas de emergência (estações de bombeiros e de cruz vermelha); e hospitais.

Realizaram-se testes com 25 utilizadores de modo a validar a fiabilidade do sistema, particularmente da API, e a adequação das interfaces de utilizador. Os resultados mostram que o sistema capta com precisão o fluxo de trabalho de informação de emergência pré-hospitalar, e as várias interfaces de utilizador foram avaliadas positivamente por todos os utilizadores.

Palavras-chave: Gestão de Emergência, Desenvolvimento em Low Code, Arquitetura Orientada para Serviços, Engenharia de Software

Contents

List of Figures

List of Tables

List of Source Codes

Acronyms

ACID	atomicity, consistency, isolation, durability 39
ANTEPH	*Associação Nacional dos Técnicos de Emergência Pré-Hospitalar* 45, 99
API	Application Programming Interface 6, 17, 25–28, 35, 38, 39, 44–47, 57, 58, 67–69, 72, 75, 99, 100, 104, 105
CODU	*Centros de Orientação de Doentes Urgentes* 3, 4, 50, 99
CPU	Central Processor Unit 37
CRUD	Create Read Update Delete 19, 20, 23, 71
csrf	Cross-site request forgery 72
CSS	Cascading Style Sheets 18, 33
CSV	Comma-Separated Values 85
CVP	*Cruz Vermelha Portuguesa* 3, 4, 50, 51, 99
EMDC	Emergency Medical Dispatch Center 1, 9
FK	Foreign Key 58–61
GQM	Goal-Question-Metric 49
HIS	Health Information System 12
HTML	HyperText Markup Language 18, 21, 33
HTTP	Hypertext Transfer Protocol 19, 20, 25–27, 37, 38, 68, 69, 72
INEM	*Instituto Nacional de Emergência Médica* 2–4, 58–61, 99
JSON	JavaScript Object Notation 14, 27
KPI	Key Success Indicator 41
LAN	Local Area Network 10
LCDP	Low-Code Development Platform 29, 30, 34, 39, 45, 99, 100
MVC	Model View Controller 18, 19
MVT	Model View Template 18
ORM	Object Relational Mapper 18, 19

RACE	Rapid Arterial Occlusion Evaluation 78
RCP	Reanimação Cardiopulmonar 61
REST	Representational State Transfer 17, 25–28, 33, 38, 39, 47, 57, 58, 67, 68, 72, 99, 100
RTS	Revised Trauma Score 78
SIEM	*Sistema Integrado de Emergência Médica* 2, 3, 99
SIREPH	*Sistema Integrado de Resposta à Emergência Pré-Hospitalar* 5, 9, 12, 17, 23, 39, 43, 46, 58, 59, 63, 65, 72, 101, 103, 104, 106
SNS	*Sistema Nacional de Saúde* 2
SOA	Service-Oriented Architecture 14
SONHO	*Sistema Integrado de Informação Hospitalar* 12
SQL	Structured Query Language 18, 19
SRS	Software Requirements Specification 49
UI	User Interface 6, 29–34, 36, 45, 46, 57, 62, 99, 106
UIX	User Interaction and Experience 57
UML	Unified Modeling Language 33
URL	Uniform Resource Locator 19, 21, 22, 26, 57, 72, 105
WSGI	Web Server Gateway Interface 38

Chapter 1

Introduction

In this chapter the author introduces the main contents contemplated in this thesis, starting with a brief background of the scientific areas in which the project will be inserted, followed by a concise problem statement detailing the issues aimed at being solved/improved and listing the objectives to be achieved to fix the problems described. The last part of the chapter will be dedicated to describing the structure of the document.

1.1 Background

Wilson et al. [1], regarded the pre-hospital emergency care as the treatment given to patients prior to their arrival at the referring hospital, after the activation of the emergency medical services. It customarily encompasses a wide scope of procedures and participants with different areas and levels of expertise, ranging from bystander resuscitation to legal emergency medical treatment and the patient allocation to the adequate and available medical facility for further health care. There is a persistent need to promote service excellence from all the parties involved (especially in cases of very serious diseases, or injured patients) in the pre-hospital period, as the decision-making and actions that take place during this period highly impact the outcomes. There is a critical time dependency to guarantee the success of the pre-hospital service, since early and basic procedures will frequently have a stronger effect on the patient's health when compared to later and more significant interventions.

Since time is invaluable in the success of a pre-hospital emergency service, it is vital to note the importance that dispatchers play in this framework, considering that they are the point of connection between the victims/patients and the pre-hospital emergency professionals and services. Mortaro et al. [2], recognized the importance of a Emergency Medical Dispatch Center (EMDC) in ensuring an organized and secure pre-hospital care, as well as the need for the dispatcher to be able to perform a quick and trustworthy diagnosis based on the information received from the caller. From the EMDC, it is required an expertly coordination to face the demanding and dynamic setting of an extra-hospital emergency, where actions have to be taken quickly and accurately due to severe time limitations and critical patient conditions that could be susceptible to unpredictable alterations. As seen in figure 1.1 the EMDC is an active participant in the entirety of the emergency process, playing a pivotal role in coordinating the ground resources and information triage. As this is meant to represent an overview of all EMDC systems, some steps are not accounted for, as they are dependant on the country and its emergency management structure.

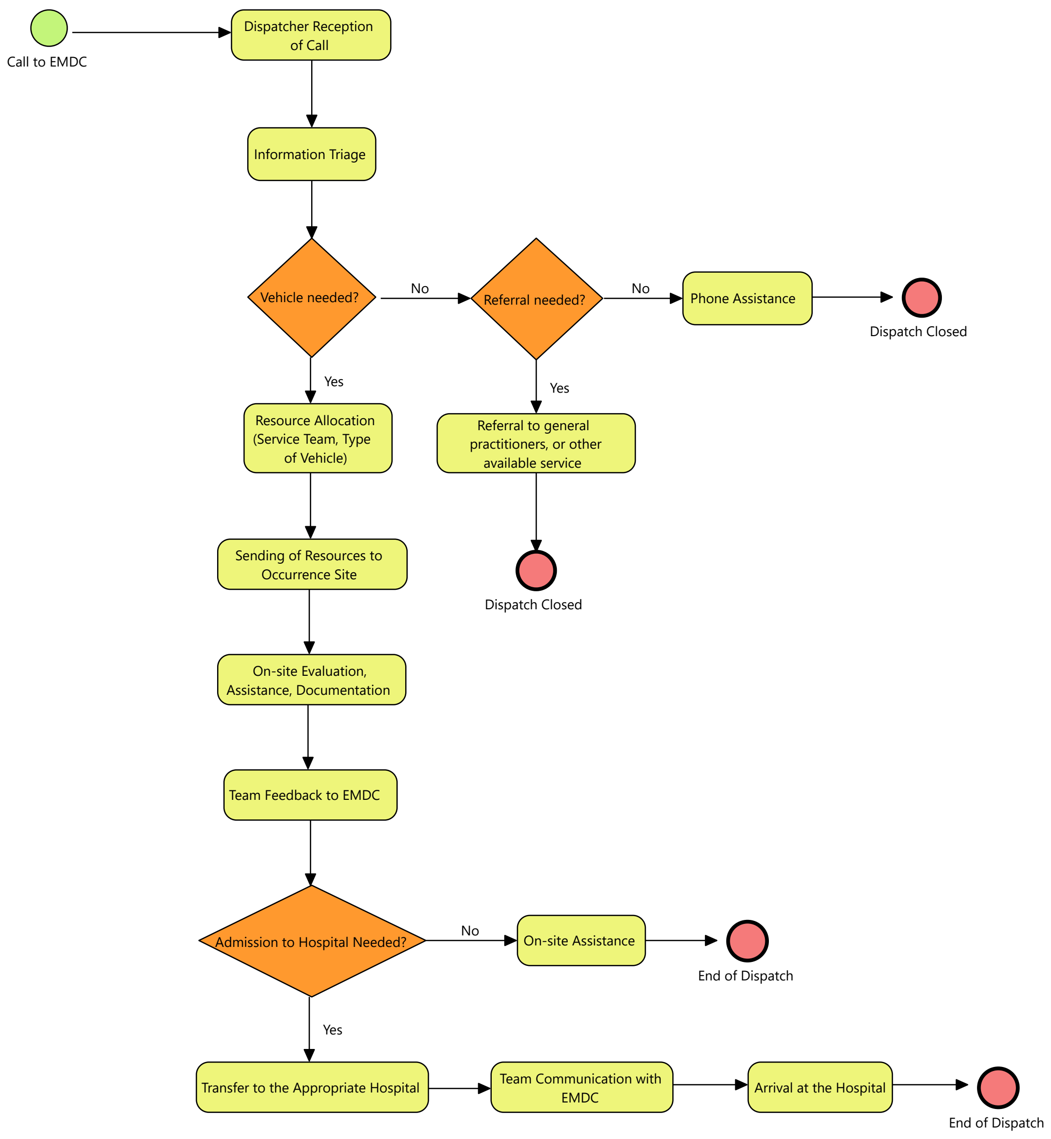

Call to EMDC
Dispatcher Reception of Call
Information Triage
Vehicle needed?
No
Referral needed?
No
Phone Assistance
Dispatch Closed
Yes
Yes
Resource Allocation (Service Team, Type of Vehicle)
Referral to general practitioners, or other available service
Sending of Resources to Occurrence Site
Dispatch Closed
On-site Evaluation, Assistance, Documentation
Team Feedback to EMDC
Admission to Hospital Needed?
No
On-site Assistance
End of Dispatch
Yes
Transfer to the Appropriate Hospital
Team Communication with EMDC
Arrival at the Hospital
End of Dispatch

Figure 1.1: Basic EMDC dispatch management process diagram

The enactment of the decree-law n^{o} 234/81 [3] by the Portuguese Government, oversaw the creation of the *Instituto Nacional de Emergência Médica* (INEM), body of the Portuguese Ministry of Health and its integrated medical emergency system, also known as *Sistema Integrado de Emergência Médica* (SIEM), forever changing the pre-hospital emergency and hospital emergency landscape in the country. According to a document published by the INEM [4], the SIEM is responsible for a group of coordinated actions set in a extra-hospital, hospital and inter-hospital capacity that result from the intervention of the various components of the portuguese national health system, the *Sistema Nacional de Saúde* (SNS), with the goal of providing fast and effective action in medical emergency situations. It comprises the entirety of emergency activity, from the pre-hospital aid system, transportation, hospital reception and the appropriate referral of the patient. The participants in this system, hail from very different areas of expertise, namely: dispatch operators, firefighters, ambulance crew members, physicians and nurses, hospital technical staff, technical staff for telecommunications and IT, among others. Together, they comprise a multidisciplinary team, each with their own tasks, to guarantee proper emergency care.

In Portugal, when an medical emergency situation arises, a call is made to the European Emergency Number (112), being attented by the police in the Emergency Dispatch Centers. After a preliminary situation checkup is complete and the emergency situation is found to be medical related, the call is rerouted to the *Centros de Orientação de Doentes Urgentes* (CODU), responsible for the medicalization of all 112 calls. The caller then, informs the CODU operators of certain relevant information to be passed on to the technical and medical staff. After the information reception, the process is similar to the one described in figure 1.1. Whenever the CODU calls upon an emergency service, it seeks to ensure that this service is currently available and close to the emergency situation, regardless of the entity to which it belongs to (whether it's the INEM, fire department or the *Cruz Vermelha Portuguesa* (CVP). In cases that the INEM can directly ensure the mobilization of resources for the execution of the service, the INEM is simultaneously the coordinator of the service and its main player, using the Institute's own resources. However, in most cases the Institute can't guarantee the direct execution of the service, as there isn't an INEM national ambulance network, so the INEM protocols with the fire department and CVP the creation of medical emergency stations, giving emergency vehicles for them to use in emergency situations and payment to the teams working in said stations. This constraint leads to the necessity of the CODU to contact a third emergency central, typically a central designated to the fire department or the CVP, in charge of the closer emergency station to the occurrence and it's this station that mobilizes the pre-hospital emergency resources available. It's in the connection of the second to the third central and its participants that a part of this project will focus on.

In 2014, the decree law n^o 10319/2014 [5] promulgated by the Portuguese Government determined the structure of the SIEM at the hospital responsability level and its interaction with the pre-hospital level, the responsability levels of the Emergency Services and the establishment of minimum standards in certain areas and indicators. It provided an improvement to the decision making process in the last steps of the pre-hospital emergency stage and a clearer definition of how the hospitals should participate in the SIEM system. It improved the flux of information regarding the patients condition when he was delivered to the health care unit, as well as an improvement in the definition of which health care unit is best suited to each patient's unique condition, trying as much as possible to avoid confusions regarding the transportation of the victim. Again, as was mentioned earlier, when there's a need to resort to the third emergency central, responsible for the coordination of the closer emergency station, there's still work to be improved.

1.2 Problem Statement

As stated in the previous section 1.1, there are problems in certain key areas in the portuguese pre-emergency hospital scenario. Much of these problems are not only related to the handover of information when going from the pre-hospital stage to the hospital stage regarding patient information but most importantly related to the operational conditions of the personnel working in these occurrences.

When an available technician goes to his fire station's central office, it provides him with a physical document, linked to a call from the CODU, with information regarding the victim(s) and their state of health. The technician then takes the occurrence paper, which contains only the file number, the victim's data and the location (address/street name/street number) and begins their service.

During the service handover phase, there is a problematic human dependence. The vast majority of situations in Portugal are still not covered directly in their entirety by the INEM and their resources, and most of the situations where the INEM have to rely on the fire and CVP stations, the occurrences are still heavily dependent on human processes.

When the CODU operator calls to the fire station to confirm their availability, he verbally passes on the required information. The person receiving the call, writes down on a piece of paper, or manually into the fire department system and hands it to another technician or team that will be responsible for the service. Any dyslexia is very likely to happen due to the great human dependence, such as a wrong street name or a number being swapped, etc. These situations are still quite common. Thus, it it is important to automate these processes, so that the technicians receive in their application what they require and that this information is registered in an applicational system related to their central, in order to avoid mismanagement's.

Another problematic phase of the pre-emergency hospital framework comes after the on-site assistance phase and the handover of information to the hospital stage. There is still constraints in the hand-off of the patient to the hospital staff as well as maintaining track of the patient's applied procedures and health state. When the pre-hospital team is en route to an health care unit, the information from the pre-hospital scenario should be sent directly to the referral hospital. Due to the human and analogical dependence stated previously, there is a problem when it comes to sharing the information generated since it is retained for the organization that generates it, creating limitations that doesn't benefit the well being of the patient.

Lastly, in the pre-hospital environment there is a lot of dispersed information, that isn't stored, handled nor analyzed, making it almost impossible to produce any actionable insights to aid in the decision making process, namely in the planning stage of the pre-hospital emergency framework. There is also no possibility of making any assessment, quality control or register any relevant casuistic about the technician's or the station's work, such as the number of services performed, mean time of service, mean time of occurrences related to cardiac arrests, mean time of arrival at the site, among others. This lack of information handling is a barrier to the improvement of the workflow of all the interveners in this complex system, which then results in weaker service being conducted due to lack of planning.

1.3 Objectives

This project is part of a larger system that its currently in development denominated *Sistema Integrado de Resposta à Emergência Pré-Hospitalar* (SIREPH). One part of SIREPH, not contemplated in this report is linked to a mobile application to be used by civilians to call and register an occurrence in progress, while the other is focused in the development of a mobile application to be used by the pre-emergency technicians to aid them before, during and after their services are complete. The project contemplated in this report is aimed at developing a web based system to be used by the emergency centrals and hospitals, each with their own role and place in the system. **Regarding the application subsystem for the emergency centrals, its main goals are to:**

- Receiving directly all the related occurrence information, coming from the civilian mobile application.
- Automate the processes in the transfer of information to the emergency technicians.
- Keep an historical record of each central's occurrences as well as those in progress.
- Provide assistance to the technicians in service as well as keeping record of all the status coming from the teams.

At the hospital application subsystem level, its main objectives are to:

- Receive incoming pre-hospital emergency service information inbound to the assigned hospital.
- Enable information consultation relative to a pre-hospital emergency service that checked-in the assigned hospital.
- Allow the update/handle of the information received relative to a pre-hospital emergency service that checked-in the assigned hospital.

It's important to emphasize that for the objectives outlined in this report to be achieved, it is necessary that all of subsystems (the ones described in this report and the one's that aren't) are perfectly integrated with each other to form the unified system that comprises the SIREPH framework.

1.4 Document Structure

This thesis will follow a clear structure, dividing each chapter according to the topics covered in it.

Chapter 1 introduces the main contents contemplated in this thesis, providing a brief overview of the background surrounding the project's theme, the problem statement, the project objectives, the structure of the document and a Gantt chart detailing the project's activity schedule.

Chapter 2 describes the published findings, works and research papers, representative of the highest level of overall development related to the project's scope.

Chapter 3 details the technologies used in the development of the project as well as research into analogous technologies that were considered.

Chapter 4 outlines the project's proposed stages and methodologies used, its planning activities, its success criteria, the software requirements specification, User Interface (UI) design, system architecture and the data model diagram.

Chapter 5 highlights the implementation efforts, taken in the development of the project, namely the Application Programming Interface (API) implementation, and the application implementation.

Chapter 6 describes the validation approaches that were undertaken to verify the suitability of the developed application.

Chapter 7 details the final project conclusions, its limitations, contributions and future work.

1.5 Time Planning

The time planning conveyed in this section was constructed during the research stage and implemented for the intermediate delivery of the project. As will be made aware in Chapter 4, the activities highlighted and the Gantt chart represented in figure 1.2 is not representative of the definitive planning of the study but remained as a part of the final document to illustrate the changes verified during the project planning stage.

Although not the definitive version, the scheduling efforts at this stage served as an outline for the implementation of the planning that would later guide the system's development activities.

1. The **planning** stage will comprise of:
   i. Requirements Gathering (for the entire system as well as for the different subsystems).
   ii. Scope Study, to better understand the end users needs and technical knowledge.
2. The **intermediate delivery** stage will comprise of:
   i. Introduction Draft of this document, where it is detailed the introductory themes of this thesis.
   ii. Literature Review of the project-related works and studies as well as the study of the main components of the Django web development framework.
3. The **development** stage will comprise of:
   i. System Design.
   ii. System Implementation.
4. The **testing** stage will comprise of:
   i. Software Tests
   ii. Result Analysis
5. The **deployment** stage will comprise of:
   i. Server Configuration
   ii. Application deployment

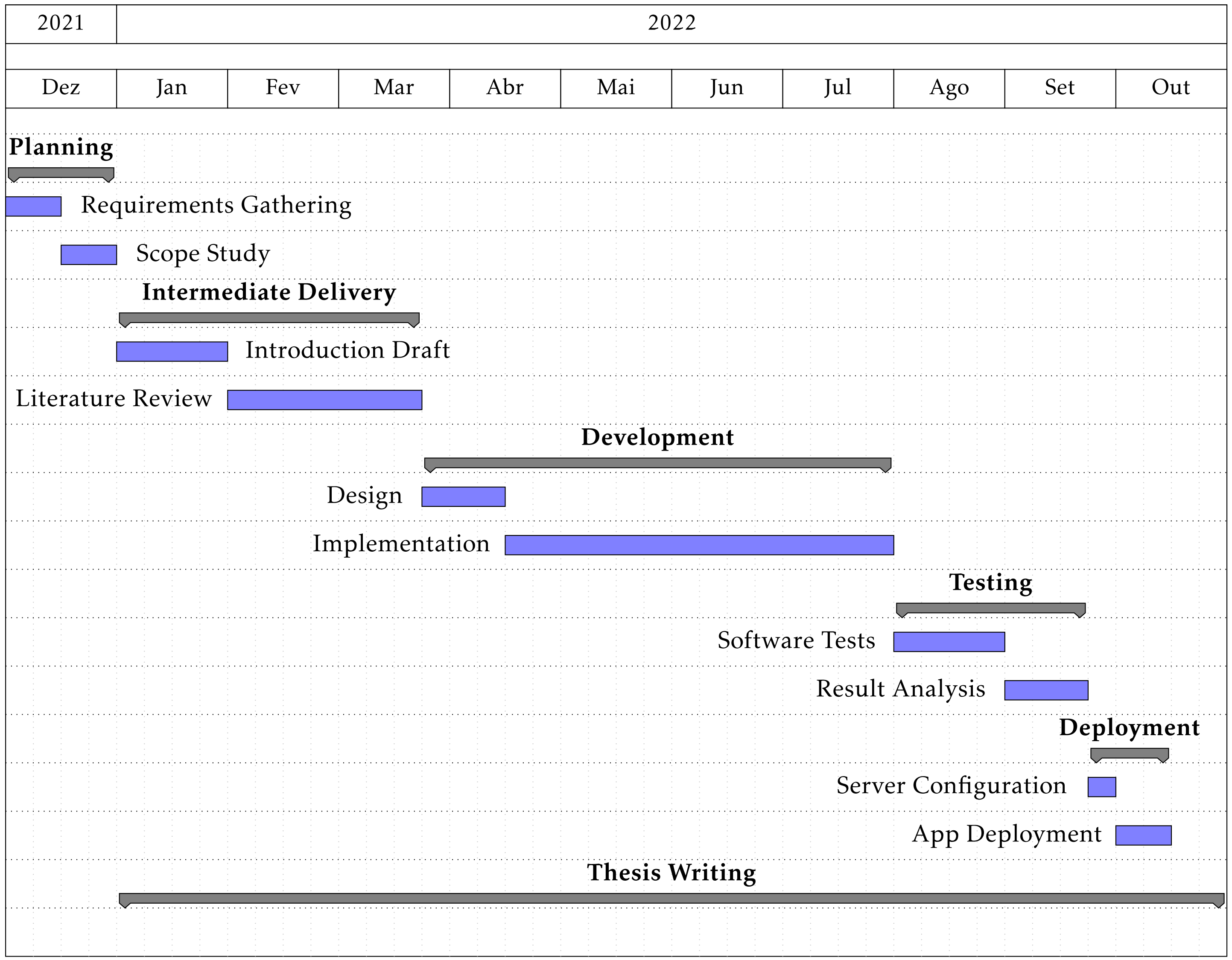


Figure 1.2: Gantt Chart of the Project's Time Planning

Chapter 2

State of the Art

This chapter will cover the study of related work, published findings and research papers representative of the highest level of overall development related to the project's purview.

2.1 Literature Applied to Real Case Studies

Chapter 1 introduced the scope of pre-emergency hospital care in general and in the Portuguese case. This thesis is focused on the identification of each participant's needs and issues in the exercise of their duties in the pre-hospital emergency environment, and also in a multitude of real case studies performed in varied countries with the most diversified teams. To this effect, and to better understand the role of the SIREPH system within these published works and studies, this section presents several case studies, divided by each of the SIREPH subsystems covered in this thesis.

2.1.1 Emergency Central Subsystem

As stated in Chapter 1, the role of the emergency medical dispatch centers is crucial for the success of a pre-hospital occurrence. Mortaro, et al. [2], present a study performed in Verona, Italy, in which the authors use an information retrieval system to analyze the role of the EMDC in the safety of pre-hospital emergency care. The EMDC is responsible for the reception of all medical emergency calls in Italy. This study intended to identify and record the list of potential error modes in the six stages of the EMDC (Emergency Call; Event site Identification; Dispatch; Resource allocation and Timing; Clinical evaluation/transfer; Hospital Admission), from February to November 2010 by using form information filled out by the EMDC professionals. The main findings concluded that:

1. The majority of errors were found in the **resource allocation and timing** stage (34.2%) and **dispatch** stage (31.0%).

   i. In the dispatch stage, poor communication with the caller was the main reason behind the errors, notably in **unreliable report of data by the caller** (18.6%) and the **caller being unable to see or interact with the person in need of assistance** (6.2%).

   ii. In the resource allocation and timing stage, the main errors were found in the **departure delay of the rescue vehicle** (13.7%) and **problems related to location names** (4.3%).

   iii. In both these stages, the great majority of errors were due to human factors, with 31% in the **dispatch** stage (all the problems were human related) and 24.8% in the **resource allocation and timing stage**.

2. The study reports 161 error modes.
   i. 125 error modes were found to be due to **human factors** (77.6%).
   ii. Nearly half of the errors were either considered **moderate** (27.9%) where there was potentially temporary damage, or **severe** (19.9%) where there was potentially permanent damage or death.

The authors concluded that the implementation of an efficient system that could record the information gathered from the pre-hospital occurrences, to monitor the quality of the service being performed, was a major driver for the development of the organization and its professionals, and a way to overcome the negative blame culture that still prevails in medical environments. The percentage of errors caused by human factors showed that in this setting, there was a low safety culture that causes more errors, and the existence of an information system could be helpful for the staff to develop critical thinking and sense of ownership.

Another study by Kao, et al. [6], in Taipei, Taiwan, verified an increased number of medical occurrences, stemming from the growth of traffic in the country, with medical emergency dispatches exceeding 300 000 and growing with each passing year. The study focused on acquiring a symptom-based medical emergency dispatch guide system that met Taiwan's rules and emergency procedures. The study identified the emergency dispatcher as the key participant in the process, and so, must be a passionate and highly trained individual, able to give medical advice and guidance to the public in cases of emergency occurrences. The main challenges faced by these dispatchers and that restrict the potential for emergency medical systems as a business development option, stem from:

- Lack of knowledge from the general population regarding the medical emergency care system.
- Lack of funding for the development of these systems.
- Callers do not know the required information to pass to the dispatchers.
- The dispatcher may not have had the proper training.

The system was built using C#, ASP.net and the server database was built upon Microsoft SQL Server, requiring permanent Internet access through an wireless Local Area Network (LAN), and enabling users the possibility to use mobile devices to enter the system. The system main functions, were divided into modules:

1. **Management Interface**, divided into 3 features:
   i. **Adding new rules**, regarding the identification of symptoms for the reporting of the patient's condition and medical needs and dispatching rules in accordance to the symptoms verified.

    ii. **Rule modification**, when there is need to change the dispatch rules.
    iii. **User Management**, allowing the managers to add and change users, broker the user's main functions according to the user type and selection of user accounts for other functions in the system.

2. **Query function to send system rules**, that permit:
    i. **Sending new rules**, that were designed according to queries of symptoms.
    ii. **Enter symptom reports**, allowing the users to input the symptom names, while the system operates the query results for the rules to be performed according to the symptom.

3. **Report query functions**, that permit:
    i. **User consultation**, allowing users to consult and print functions.

The main conclusions gathered, showed an increase of personnel availability and a time decrease in the decision-making process, and the functions built into the system helped the dispatchers in improving their service quality. Also, it was possible to ensure accountability, as the system requires a login so that the users can participate.

A study by Kyriacou, et al. [7], situated in Limassol, Cyprus, aimed at the creation of an electronic system, denominated eEmergency system, to provide support in emergency dispatches. The system was successfully used in tackling the COVID-19 pandemic since its early stages. Recent technological and scientific developments actively helped with emergency dispatch calls, but there are still critical logistical problems regarding coordination and resource distribution. To address this issue, the team devised a computer based-system that implemented priority dispatch protocols in order to automate processes and minimize human error rates. The main Use Cases were centered around the roles of shift manager, unit manager and dispatchers. Specifically:

- Provide support to the emergency dispatchers when in a call, using the protocols designed for the triage.
- Organize patient transportation from and to different places.
- Terminate an emergency occurrence or transportation.
- Monitor the occurrence status through real time information exchange between the on-site units and the call dispatch center.

The system architecture was partitioned into 2 subsystems: one was designed for incident and ambulance vehicles management, and for recording initial occurrence information. The second part is related to the data exchange between the different participants. The ambulance units possess an android tablet, that will be used to get initial information from the incident and report meaningful information to the dispatchers. This information could include patient status information or occurrence information. This information, coupled with device and user authentication are lodged on a control server with the aid of a database server. The data coming from the ambulances are sent to the server using web services. The system also possesses a dedicated database to store system information, related to users, their groups and permission levels, ambulance entity records, patient entity records and incident entity records. The access to the system is monitored with the use of session keys, that are dynamically generated after a user is successfully authenticated. The system has been in practice since 2020 and during its first thirteen months, were completed around 62 thousand occurrences.

2.1.2 Hospital Subsystem

The hospital subsystem in the SIREPH system revolves around the information hand off phase during the patient's transportation to the referral hospital. To this subsystem, what matters the most is the information gathered by the various pre-hospital emergency participants regarding the victim and occurrence status, to serve as a bridge between the pre-hospital and the in-hospital phase of the health care system.

As a way of introduction, the first work contemplated in this subsection [8] studied the key issues verified in the Portuguese Health Information System (HIS). The main objectives were to identify and describe the main issues that affected the daily work of the HIS managers and to propose improvements. Regarding the information systems in Portuguese hospitals, it has been since the 90's, adopted the *Sistema Integrado de Informação Hospitalar* (SONHO), and other complimentary systems, that have fallen outdated in a functional and technological way. The biggest problem, this study found, was related to the absence of a well-integrated HIS, able to reliably respond to the different information needs of the health care system and its participants. So far, an effort has been made in an attempt to revitalize the SONHO with the development of SONHO v2, although it's still fairly under implemented in the Portuguese health institutions. The SONHO v2 aims to tackle the technical issues experienced by hospital units, caused from the obsolescence of the SONHO application. Its main changes have been [9]:

- Technology upgrade to Oracle Database 11g R2, to ensure a greater compatibility with the majority of the applications on the market.
- Development of a new service-oriented integration layer.
- Provision of a Reporting database, from which management reports can be extracted, taking the burden off the operational database.
- New functional features, such as the use of identification cards in patient identification.

Continuing the study mentioned [8], the major challenges found in the HIS were:

- Lack of a clearly delineated strategy at the national level, which should be reviewed.
- Lack of technological and functional evolution of the systems in place, particularly the SONHO, that hinder the service quality of the health care professionals.
- Scarcity of human resources and difficulty in hiring skilled staff for the information systems in hospitals.
- Limited financial investment in resources that aid the response to the challenges of the institutions.

- Minimal acknowledgment of the importance of information systems in the current activity and management of a hospital unit.

It is concluded that there is a grave need of financial investment in the development of new applications to aid in the hospital operations, but also a clear need to educate the staff in the advantages of the use of these systems, because if the staff doesn't recognize its necessity, no matter how good a system is implemented, it will never function the way it was designed to.

The next study conducted by Murad et al. [10], sought to create a way to facilitate the passage of information between care providers, from the emergency dispatch staff to the emergency medical system personnel. The information handover was described as very dependant of verbal and written information exchanges between the different participants of the pre-hospital emergency setting. In the study's environment, the information handover was very similar to what happens in the majority of occurrences in Portugal, as a pre-hospital unit collects patient and incident information, and writes said information in an available physical paper. The paramedic unit would then contact the emergency dispatch and relay the collected information, sometimes not even having time to transmit all the relevant information, and frequently the patient will arrive at the handoff staging location before the paramedic unit has had time to transfer the relevant information. In these cases, a verbal information exchange is performed in a setting that is not suitable to hearing and catching important insights, and the written and verbal information is in many situations overlooked, omitted, misplaced or in an unreadable state [11]. A study [12], concluded that 67.6% of the reports didn't contemplate a thing as important as the victim's name, giving clues to the lack of organization in this handout phase, not boding well for a proper treatment of the patient. The main challenges observed in the handoff phase were due to [10]:

- Restricted time frame for the paramedics to collect data at the scene.
- Resistance of the end users to utilize technology due to perception that it will disrupt patient care.
- Limited number of available electronic tools for paramedics to correctly collect information.
- Absence of standards in the information exchange process.
- Missed, wrong or unreported information to the emergency dispatchers, that then in turn, will delegate poor information.

The solution developed in the study aimed to be simple to use but effective at the same time, facilitating the exchange of patient and occurrence information between the entities of the pre-hospital scene and the hospital providers and enforcing the standardization of data collection techniques in order to avoid missing and poor information problems that are linked to the number of medical errors.

For the standardization of data collection it was used the five-Ps technique (**Patient -** its name and identifiers such as age, gender and location; **Plan -** diagnostics, treatment plan and next steps to be taken; **Purpose -** justification for the treatment plan; **Problems** - report of the patient's issues; **Precautions -** explanation of the necessary precautions to take with the specific patient). The main features of the system, to tackle the challenges of the handover phase were:

- Touch screen technology, large interface buttons and few tabs and clicks, in order to deal with the limited time that paramedics have for data collection at the site.
- Asynchronous two-way communications and data record for later retrieval by the health care workers, to deal with the gaps in the information flow, due to interruptions.
- Built-in camera and audio device to capture surrounding information that add important contextual data.
- Possibility of gathering in the system, multiple techniques of information collection.
- Web interface to allow the users to access the information when needed.

The study concluded by stating that there is still challenges to be fixed in terms of system usability, and emphasized the importance of web and mobile based systems in aiding the participants in this important and complex stage of the pre-hospital emergency setting.

The following study [13] was performed in Guangdong, China, a country with wildly unequal regional distribution of medical resources, that lacked an integrated planning and digital health care system. This hindered the exchange of patient records between different institutions, resulting in a gross amount of repetitive processes that increased the burden of hospitalizations, physical injuries as well as medical and human resources. To avoid this predicament, the team proposed the creation of a collaborative medical information system, to maximize the use of medical resources and the reduction of the costs related to medical care.

It was used Web Service technology, to establish a middle layer between the public management application and the the various database platforms, and a Service-Oriented Architecture (SOA) where the features present in the system had to be made available in the form of services, usually accessible through the use of web services. The figure 2.1 represents the proposed system architecture. Each subsystem supplies interface and logic to operate data, allowing external systems to interact with the platform as long as they find the services to connect. The results coming from the platform were converted into JavaScript Object Notation (JSON), in order to be read from web browsers.

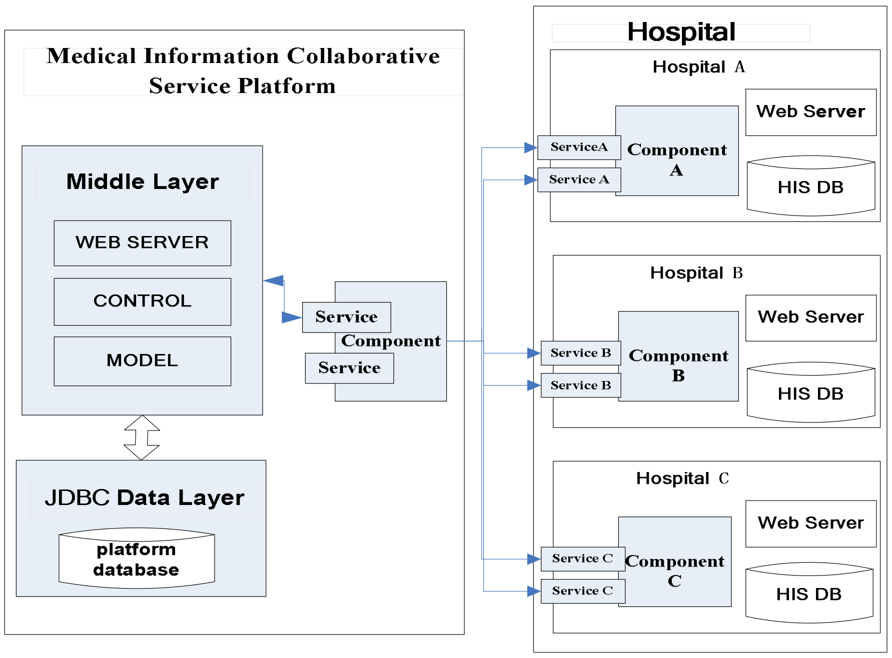


Figure 2.1: System Architecture. Extracted from [13].

Chapter 3

Technologies

This chapter will cover the study of the technologies employed during the project's development cycle.

3.1 Django Framework

This section is concerned with exploring the Django framework, employed in the development of the SIREPH backbone and its present API.

3.1.1 Introduction

As it was mentioned before, the backbone of the SIREPH system is written in Django, a Python-based high-level web framework, know for providing fast, secure and maintainable website development. Since its initial release in 2005, Django has been at the forefront of web development, proving its reliability and consistency. When trying to fit Django in the web development scene, Shaw et al. [14] states that one might consider JavaScript frameworks like React, Angular, or Vue, as these are used to increase interativity to previously generated web pages, when in reality Django sits in the layer underneath these frameworks, being responsible for URL routing, database data fetching, template rendering and handling of user form inputs. Nevertheless, these are not mutually exclusive, and can be utilized simultaneously with ease, as these Javascript frameworks may be used to improve the Django output, or by interacting with a Representational State Transfer (REST) API generated by Django. For instance, the software company Instagram [15], presently boasts the world's largest deployment of the Django web framework. The choice behind Django as their core for backend development relies on its simplicity and practicality. They prioritized the web services efficiency, to quell the influx of a vast number of new users over the years, allowing their system to keep scaling smoothly. They also make use of JavaScript frameworks such as React, since as mentioned before, couple conveniently with Django.

3.1.2 Model View Template

Django follows the Model View Template (MVT) software design pattern [14][16]. Although similar in its inception to the Model View Controller (MVC) design pattern in which its 3 main components (Model, View, Controller) work seamlessly, the MVT follows a slightly different paradigm. In MVC, the model (storage of data) of the application is exhibited in one or more views and a controller is responsible for the interaction between the model and the view components, whereas in the MVT standard, a view will consult a model (also used for the storage of data) and render it with a template. One important distinction between these 2 design pattern relies on the fact that with MVC all 3 elements are required to be developed in the same programming language, while MVT allows the template to be written in a different language. So to that effect, in Django the models are written in Python and the Template utilizes HyperText Markup Language (HTML), enabling team diversification, in the sense that a Python developer might focus on the models and views, while a HTML/JavaScript/Cascading Style Sheets (CSS) (and others) developer can work on the templates, but all inside the same Django project. The figure 3.1 captures the Django framework MVT architecture.

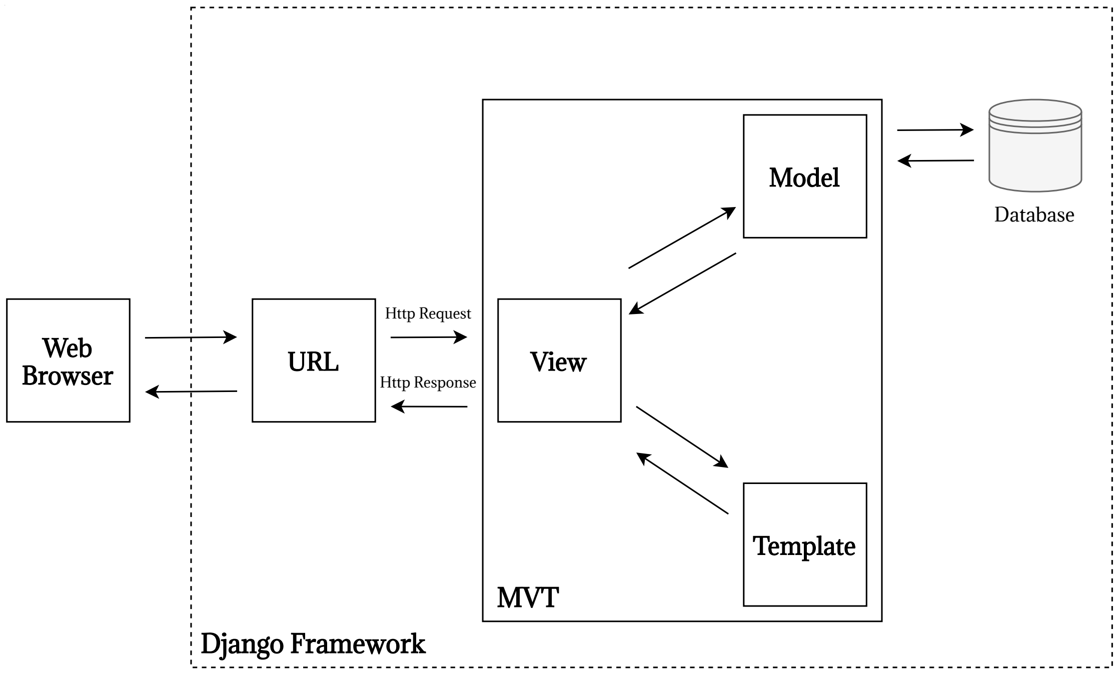


Figure 3.1: Django MVT Architecture

3.1.2.1 Models

Django models are fundamentally classes that contain the blueprint for the creation of tables, their associated fields, and a variety of constraints, in short, are responsible for establishing the application's data. Django models are mapped to a single database table each and supply an abstraction layer to an Structured Query Language (SQL) database access through an Object Relational Mapper (ORM) [14]. AN ORM, allows the developer to define the data schema (classes, fields and relationships) using primitive code, without requiring an understanding of the subjacent database. Lorenz et al. [17] states in his work that an ORM is a layer responsible for the management of the link between objects and tables, with the main goal of automating the mapping processes and hiding technical details of the mapping implementation through abstractions. When querying a database, the output is generally rudimentary Python objects with different data types (integers, floats, strings). By utilizing the ORM, the output is automatically transformed into instances of the defined model classes. It also protects the application from some SQL injection attacks, since it further decreases the use of explicit SQL code. There's however enough literature to point out the impact of ORMs [18][19][20] on the performance of relational queries, notably Karwin [20] that states the importance of the Active Record design pattern to the ORM tools. This pattern maps an object to a particular row in a table, but in a MVC or similar architecture the models are intimately connected with database schemas, so in cases that a change arises in the schema the model becomes inoperative. Other issue lies in classes with Create Read Update Delete (CRUD) features, as it exposes these features to any subclass that inherits from it, enabling direct database access, lowering cohesion.

The Django ORM specifically, transforms object-related Python code, into proper database structures, such as tables with specific data types and oversees all the database related activities using basic Python code [14]. Due to this, the developers can focus on the application development and upkeep, without having to worry about SQL commands while interacting with the database. The code snippet 3.1 highlights a practical example of Django models.

```python
from django.db import models

class Author(models.Model):
    first_name = models.CharField(max_length=10)
    last_name = models.CharField(max_length=15)

class Book(models.Model):
    writer = models.ForeignKey(Author, on_delete=models.CASCADE)
    name = models.CharField(max_length=60)
    release_date = models.DateField()
    review_rate = models.IntegerField()
```

Source Code 3.1: Django models code snippet

3.1.2.2 Views

According to the Django Documentation [21], the view fuction is the structure in a Django application that receives a web request and returns a web response. This view will receive the request in the format of a Python object (an HttpRequest object). The view then, must return an HttpResponse object that captures the entire information to be delivered to the requester: its content, Hypertext Transfer Protocol (HTTP) status, and other meta information, and can be anything a web browser can display, being entirely up to the developer to decide what to respond back to the requester. [14]. Alternatively, the view can receive information embedded in the Uniform Resource Locator (URL) from the request, most commonly, an identifier. A standard design pattern of a view is to query the application's database through the Django's ORM using an identifier that is handed to the view. This view, can then render a template by supplying it with data from the model that was fetched from the database. This rendered template then becomes the content of the HttpResponse object and is returned from the view. Django is then able to communicate the response back to the requester (in this instance, a browser).

Views are one the most crucial components of a Django app, on account of being the place where the application logic is placed. This applicational logic marshals the interactions with the database (CRUD operations). It is also in charge for the way the data can be displayed to the requester [14]. There are 2 main types of views:

Function-Based Views - As the name suggests, these views are implemented through Python functions. The view function is specified and named, and takes a request object as a function argument, returning an HttpResponse object with the desired information [21].In the code snippet 3.2, it's possible to see an example of this type of view, where the function name **functionName**, takes the request object as argument and returns an HttpResponse object with the message **"This is a function view"** if the HTTP method is GET. Function views have the advantage of being simpler to code and read, since they implement plain Python functions, but on the other hand they don't permit code reutilization for general use cases, and the handling of HTTP methods can only be made through conditional branching [22].

```python
from django.http import HttpResponse


def functionName(request):
    if request.method == 'GET':
        example_message = "<html><h1>This is a function view></h1></html>"
    return HttpResponse(example_message)
```

Source Code 3.2: Django function-based view snippet.

Class-Based Views - These views are implemented through the use of Python classes. Through class inheritance, these classes are built as subclasses of the Django's base view classes. In the case of function-based views, where the logic is written inside functions, Django has generic view classes with pre-constructed methods and properties, that allow for a much cleaner and reusable code. It also allows the use of Mixins, reducing code duplication. In terms of code structure, the handling of the HTTP methods isn't made with conditional branching but by using different class instance methods. Their main advantages is the need for fewer lines of code to implement the same features as in function-based views making the code much cleaner and concise. However, they are harder to implement and to read [14][21] [22]. The code snippet 3.3 is a simple implementation of a class-based view, similar to its function-based counterpart in 3.2.

```python
from django.http import HttpResponse
from django.views import View

class ClassView(View):
    def get(self, request):
        example_message = "<html><h1>This is class-based view></h1></html>"
    return HttpResponse(example_message)
```

Source Code 3.3: Django class-based view snippet.

3.1.2.3 Templates

A Django template is a text document using Django's template language, in order to dynamically generate HTML. This text document contains unique placeholders to be replaced by the application's variables, which then in turn get replaced with values when the template is generated. It can be then said that the Django template system was designed to convey information in a presentable way to the end user and not applicational logic, thus separating them [21]. Writing HTML code inside Python modules is considered to be a poor practice, due to the amount of code necessary as the web page expands. This blend of HTML and Python makes the code harder to read and upkeep, leaving possible scalability and performance issues in the future. Important to note, that almost any front-end programming language can be written in this template system, if properly integrated with Django, so the versatility of this component is considerable. The snippet 3.4 is an example of a Django template. In this particular example, Bootstrap framework is in use, and the colored parts of the code refer to code parts written in the Django template language. The extends and block statements refer to the Django template inheritance that can be studied in detail at [21]. The variables in this system are placed between { { } } referring to a specific variable in the application.

```
{% extends 'base.html' %}
{% block title %}<h1>Header Title</h1>{% endblock %}
{% block content %}
    <h1>Player Id: {{ player.id }} - {{ player.username }} </h1>
    <div class="col-sm-3">
        <ul class="list-group">
            <li class="list-group-item"><b>First Name:</b> {{ player.first_name }}</li>
            <li class="list-group-item"><b>Surname:</b> {{ player.last_name }}</li>
        </ul>
    </div>
{% endblock %}
```

Source Code 3.4: Django template snippet.

3.1.3 URL Configuration

A URL is the web address that is used to access a web page and there's many components that make up a good URL [23]:

- Its structure must be logical and intuitive so that the end users can gather quickly the link between the various pages of the web application.
- It must possess relevant keywords (matching the content of the page).
- It must not comprise of spaces nor underscores to detach words. If need arises, hyphens should be employed, making the words in URL much easier to read.
- It must not have stop words nor uppercases letters (URLs are case sensitive).
- It should not change. A good URL doesn't have the need to be changed, even after a website redesign. As stated by Tim Berners-Lee, "cool URLs don't change"[24].

Django comprehends the importance of a clean URL scheme in high-grade web applications. To that, Django allows a clear URL configuration without framework restrictions, with its URL dispatcher [21].

Django views are incapable of working by themselves in a web application. Django's URL configuration oversees the request routing to the target view function or class to handle the request. An example of a common URL configuration in Django can be seen in snippet 3.5. The **urlpatterns** is the variable responsible for setting the URL path list, and the **path()** sets the path to be matched. When there is a URL match, the view function invoked is **views.ExampleView** and the **name="example_view"** is the name of the view function used when referring to the specific view. In cases of class-based views, it is possible to connect a class with a URL by using the **as_view()** method, as seen in the example **views.LibrariesList.as_view()**. It is also possible to pass arguments in the URLs, most commonly integers that act as primary keys of certain Django models. In the example, **'libraries/<int:library_pk>/books/'**, the **<int:library_pk>** part will prompt Django to search for URLs that contain a library primary key (identifier) in this position of the string. Then the corresponding view will search for a specific library identifier in the database and return the corresponding data, in this case, the books present in said library. This subsection only covers the crucial parts of Django's URL configuration, and there's a great deal of literature detailing it [14] [21][22].

```python
from django.urls import path

from . import views

urlpatterns = [
    path('url-path/', views.ExampleView, name="example_view"), # path generic example
    path('libraries/', views.LibrariesList.as_view(), name="libraries_list"),
    path('libraries/<int:pk>/', views.LibrariesDetails.as_view(), name="libraries_detail"),
    path('libraries/<int:library_pk>/books/', views.BooksList.as_view(), name="library_books"),
]
```

Source Code 3.5: Django URL configuration snippet

3.1.4 Django Admin

The Django admin site is one of the most powerful components of Django and another main reason for its choice as the backbone of the SIREPH system. With it, it's possible to perform a myriad of features, such as [14]:

- Read metadata from the application's existing models (administrative access to the model data)
- Manage and customize the application's content.
- Authenticate users.
- Create, view, update and delete records.

This component is not intended for the non-privileged users who interact with the website, but for its administrators. This admin interface allows a bird's eye view of the intricacies of the model data, facilitating its management and CRUD operations in the model's records. First and foremost, it is necessary to create a **superuser** account with elevated privileges. Then, it is necessary to make use of the **ModelAdmin** class (a representation of the model in the admin interface). These are stored in the file **admin.py** in the application. In the snippet 3.6, it is represented a possible admin representation of the models created in 3.1. Since the book model has a foreign key from the author model, it is best to create an **InLineModelAdmin** class for it. This way, it is possible to edit in the admin interface the books written by a specific author while on the author page. In the end, it can be used the **admin.site.register** code to register the **ModelAdmin** classes created, in order for them to appear on the admin page.

```python
from django.contrib import admin
from .models import Book, Author


class BookInline(admin.TabularInline):
    model = Book

class AuthorAdmin(admin.ModelAdmin):
    inlines = [
        BookInline,
    ]

admin.site.register(Author, AuthorAdmin)
admin.site.register(Book)
```

Source Code 3.6: Django **admin.py** example snippet

3.1.5 Authentication and Session

Presently, security is increasingly a concern when developing software. Protecting applications against unwanted use has become a crucial point in any project. To that end, it is important to guarantee that the creation, modification and removal of content are restricted to authenticated users who are registered in the system. Django possesses an authentication system, that provide both authentication and authorization combined, as these functionalities are in a certain way related and add an extra layer of security to the system. This authentication system provides Django with the models for representation of users, groups, and permissions. All the users are none other than the people that interact with the system, each with their own group and permissions, but within the same user class, only being separated through different permissions applied to each other [21]. For instance, the **superuser** mentioned in the previous subsection, is nothing more than a normal user with a special set of attributes. Django has in-built a permissions system to delineate what is permissible to specific users and groups, being normally used at an administrative level, through the Django admin to customize the permissions levels to each User object. In most cases, it is much cleaner to integrate different users to specific groups and then assign its permissions. Groups make it much easier to model roles and organizational structures, because if a new permission is added, it's much simpler to modify the groups than each user individually [14]. This authentication system is managed by Django's middleware. According to [25], middleware is a collection of services located between software components and the platform they are running on. It delivers a unique identifier for each applicational object that is present and supported in the system and keeping record of the physical location of specific resources used by the system. In short, middleware plays a substancial role as a communication facilitator, not being part of the applications that it supports but being an external service used to glue software together. In the case of Django's middleware, it refers to a myriad of software components that are present in the request and response process, and aid in crucial features like authentication and security. There is a list of middleware modules that are added in the beginning of a project and others that can be added later on as necessity arises, that can be found in the **settings.py** file as seen in the snippet 3.7. By adding authentication functionalities, it can be added Python code in the views to assure that a view is only available to an authenticated user, or a user with the correct identifier, as seen in snippet 3.8, a non-authenticated user is redirected to the login page and a user with a incorrect user identifier isn't authorized to complete a request. In the templates, it is also added a registration folder, to place the templates related to user authentication. The sessions in Django are controlled by the **'django.contrib.sessions.middleware.SessionMiddleware'** in the Middleware section. This is a highly complex topic, that can be studied in detail at [21].

```python
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

Source Code 3.7: Django Middleware settings example snippet

```python
if not request.user.is_authenticated:
        return redirect('login')

    if request.user.id != user_id:
        return HttpResponseForbidden()
```

Source Code 3.8: Django view authentication control example snippet

3.1.6 Django REST Framework

This subsection is dedicated to a brief study of the Django REST Framework, an important part of development in Django and the toolkit used in the building of SIREPH's API.

3.1.6.1 Introduction

Modern day applications have a very clear structure. It is common for a web application to have their frontend developed using powerful libraries that allow for the construction of dynamic user interfaces, most notably JavaScript libraries such as ReactJS, AngularJS, VueJS and others. However, these libraries do not communicate directly with the backend code nor with the database, and therefore do not have direct access to any data running on the backend server. So, it is essential the creation of components whose function is to connect this frontend code to the backend code. In Django, this is where the REST APIs come into place.

An API facilitates the communication between different software artifacts, using the standard protocol between servers and clients, the HTTP, that is crucial for the exchange of information on the web. These APIs are responsible for receiving HTTP requests and sending HTTP responses, and focus on the communication and synchronization of object states between the database server and the frontend client [14]. These software artifacts are immensely helpful in cases where the frontend code must be changed or created anew, since they prevent the need for any changes to be made to the backend code, as long as the new frontend is built upon the same API requests as the previous one. As these REST APIs are stateless, the communication between client and server is made without any need to store any states in order to communicate. This means that whenever a request is submitted, the data is handled, and the response is submitted without having to store any intermediate data. The API handles each request as isolated, not needing to store information regarding the session itself [14]. As will be studied posthumously, a collection of REST APIs is denominated as RESTful web service, and the creation of these artifacts is exactly the function of the Django Rest Framework.

3.1.6.2 API HTTP Methods

A RESTful API has many HTTP methods for handling requests and its content, of which we can highlight:

POST - This method is mainly used of top of resource collections. These structures have their own hierarchy, defined by the developer. When a developer intends to create a new child resource in the structure, applying the POST method to the parent resource cues it to create the new resource, coupling it with the structure's hierarchy [26]. Using as an example the model's structure studied above, a POST method in the Author model would prompt the creation of a single author, that when after created, would be linked to the collection of authors present in the database. It is also important to note that the use of a POST method always causes a server state change, meaning that if a POST method is declared 10 times, it will create 10 different resources, each with their own URL, meaning that it isn't idempotent [27].

PUT - This method is responsible for updating a single resource's content. In a RESTful API it is the most frequent method used in revising a resource's information. Unlike the POST method, the PUT method is idempotent, meaning that the result of a successful request is independent of the number of the times it was performed, so a single resource can be updated in an indefinite number of requests [27].

GET - This method is the most commonly used in HTTP method. When applied, it returns the current view of a resource's content and data, in a read-only capacity (protecting the data and the idempotency of the resource) [26]. In cases where no changes have been made prior to the use of GET methods, it is necessary to verify that the content remains unaltered in every request.

DELETE - This HTTP method is responsible for the complete removal of a single resource, meaning that after its use, all references to the resource must be also expunged [26].

3.1.6.3 Integration with Django

The Django REST Framework is a Python library that is used to build REST APIs to be used in a Django project. To integrate it with Django, it is just needed to run the following code in the terminal app with PyCharm **pip install djangorestframework** and make sure to have placed in the **settings.py** file, in the **INSTALLED_APPS** section the code **'rest_framework'**, as seen in snippet 3.9.

```
INSTALLED_APPS = [
    ...
    'rest_framework'
]
```

Source Code 3.9: Django settings.py installed apps snippet

3.1.6.4 API Views

According to the Django REST framework documentation [28] these views are built differently from regular django views, as they use the REST framework in-built APIView classes, which are subclasses of the main View class from regular Django. In REST framework, the requests passed to the handler methods are Request instances instead of Django's HttpRequest instances. The same goes for responses. In the snippet 3.10 it is possible to see an example of an API View. As can be seen in the example, the argument **APIView** is present in the class definition (instead of View in regular Django) and there is a periodic use of serializers. Also, the responses are returned through Response instances instead of HttpResponse instances, as was stated before. In the post method, if the serializer is valid, it is saved and returns a Response instance with the serializer's data and a 201 HTTP status, which means that the resource was created. If the serializer isn't valid, it will return a Response instance with the serializer's errors and a 400 HTTP status, which means that it was unable to process the request. To better understand the code present in the snip 3.10, it is necessary to study the function of the serializer's in the scope of the REST framework, which it will be done in the next subsection.

```python
class LibrariesList(APIView):
    """List all libraries"""

    def get(self, request):
        libraries = Library.objects.all()
        serializer = LibrarySerializer(libraries, many=True)

        return Response(serializer.data)

    def post(self, request):
        serializer = LibraryDetailsSerializer(data=request.data.copy())

        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)

        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Source Code 3.10: Django API View example snippet

3.1.6.5 Serializers

As mentioned in the previous subsection, serializers play an important role in the Django REST Framework. They permit the conversion of objects into data types comprehensible by frontend applications. For example, in the case of a website built with the intent of displaying a list of libraries present in an application, it would be necessary to call the **name** property of each library instance, for the application to know which string to present to the user. However, the frontend application can't properly read this information, and requires an HTTP request to retrieve this data, in order to return it as string, implying that any given information transposed between Django and the frontend through the created API, must be done by converting the information to the JSON format. So, in order to return information from the database using the API created, it is necessary to convert Python objects into recognizable formats to the front-end frameworks, such as JSON, for them to be disseminated beyond the web using HTTP protocols. Serializers can also execute deserializations, which essentially means converting the parsed data back into its original state, in this case Python objects, so that it can be handled in the application. The snippet 3.11 represents an example of a declaration of a serializer. In this example, it is used a **ModelSerializer** class, that creates automatically a **Serializer** class with fields that correspond to the model fields. In this case, the model is a library, and its fields are represented inside square brackets. The serializers are a crucial part of the Django REST Framework, with a lot of literature to explore. However, this subsection, was meant for giving a brief introduction to the theme and not detailing it expansively. More information on serializers can be found at [28].

```python
class LibrarySerializer(ModelSerializer):
    id = IntegerField()

    class Meta:
        model = Library
        fields = ['id', 'name', 'location', 'books', 'year_opened']
```

Source Code 3.11: Django ModelSerializer class declaration example snippet

3.1.6.6 API Documentation

As early as 2001, research such as Hoffman [29], reinforces the concept that proper API documentation translates into an effective use by its various users. Moreover, its importance rapidly increases when building or consuming complex APIs, such as frameworks or elaborate libraries. In this way, an effort must be made to correctly document the specifications and functionality of the API, in order to avoid possible misinterpretations between API developers and their end users.

The interpretability of an API is highly dependent on the accessibility of its research material, and despite the developer's reliance on its authoritative documentation, it recurrently proves to be inadequate or insufficient. Therefore, and to overcome these shortcomings, there has been considerable growth in a plethora of online websites and community-driven documentation projects that help tackle the issues present in the official documentations [30].

The most notable example of this growth, has been represented by the OpenAPI specification, formerly known as the Swagger specification. This specification, is undoubtedly the reference choice for defining and describing RESTful APIs, not only to humans, but also computers, to ascertain and comprehend the abilities of a service, foregoing the need to access its source code or documentation. When accordingly specified, a consumer gains the knowledge to interact with the remote service with a minimal knowledge of its implementation logic [31]. The decisive choice to make use of this specification for the project in study, can be ultimately substantiated by Casas et al. [32], in a study conducted with forty seven articles dated between 2011 and 2020. The conclusions gathered from the sample size, were that the OpenAPI specification was responsible for a considerable improvement in the software development activities documentation by 43%, that 77% of the study sample verified that the specification provided solutions to the problems and requirements of the API consumers and the automation tasks were improved by 68%. Moreover, this specification also allows the use of powerful generation tools to display the API and testing tools for developers and consumers alike.

In this regard, Django REST Framework possesses the ability to utilize the OpenAPI specification, by way of community-driven generator applications that follow its standard practices. For this project, it was researched thoroughly which of these applications to use, and ultimately it was decided in favour of a Swagger generator application for Django, named **drf-yasg**. This generator was comprised of a methodical and complete documentation and its use proved to be beneficial for the project's development, as it will be made aware in the coming sections of this document. More information about this application can be found at [33].

3.2 Low-Code Development Platforms

A Low-Code Development Platform (LCDP) can be succinctly described as a compilation of toolkits with the purpose of empowering visual application development through a graphical UI. At its core, it facilitates the conception and delivery of powerful applications by visual methods, most specifically through a drag-and-drop interface [34]. According to Gomes [35], these platforms can be referred to as solutions with a high-level programming abstraction, contemplated for end-user development via Model-Driven Engineering principles. Its rapid growth can be explained from the burgeoning demand for the development of different and complex technological applications with complex business logic.

In the upcoming years, it is projected that its use will become increasingly prominent, as reported by Gartner [36] in 2021, in which it was concluded that low-code will be accountable for 65% of application development activities by 2024. This trend has grown significantly in the last couple of years, due to the Covid-19 pandemic, providing the ideal environment for these platforms to thrive, as the need for the quick iteration of adequate solutions became increasingly important in the medical area, such as in the case of the biotechnological field in which applications built with low-code platforms helped onboard staff during the crisis [37].

In a broader context, the usage of low-code have plenty benefits and limitations. Following the study cited earlier [35], the mainly results gathered concluded that the main benefits of LCDPs were:

- Faster development.
- Easier to use and research.
- Decreased operational costs.
- Proper and ready-to-use components.
- Beginner friendly.
- Considerable integration, deployment and expansion efficiency.

From these results, it can be extrapolated that LCDPs allow for a expeditiously application development and for it to be a less laborious endeavor, contributing to an agile development.

However, the same study reported certain limitations, in which can be highlighted the following:

- Steep learning curve.

- High pricing.
- Lack of customization.
- Slow loading and publishing times.
- Limited programming capabilities.
- Considerable complexity.

From the results, it can be concluded that LCDPs have a considerable learning curve and its full capabilities are usually locked behind a paying version. One of the phenomenons also reported with this type of platforms is called "vendor lock-in", where essentially a developer or company becomes subordinate to the proprietor of these platforms and is unable to use other alternatives without substantial costs associated with revocation of the current plan.

Later in this section, 2 of these types of platforms will be studied, Mendix and Outsystems, with special regard given to Outsystems, as it was the project's chosen platform for the front-end screens. This choice will also be substantiated later in the document.

3.2.1 Outsystems

The Outsystems platform was the one chosen for the creation of the project's application (business logic and user interface). It is a LCDP founded in 2001 in Lisbon, Portugal. As a LCDP it possesses the main characteristics described in the previous section. One of its main advantages is the availability of a free plan for users to experience the platform's capabilities, being the one applied in the installation process. The following subsections will detail the study of Outsystems main components and features, which mainly contributed to the choice of using it in the project to be developed.

3.2.1.1 Components and Tools

Within the Outsystems platform, there is a broad range of components and tools made available for users to build and develop their applications. The components mentioned in this section are available to paying and free customers alike, only except that the free version has a limit of 100 end users to the application, whereas the paying version has no limitations whatsoever. It can be highlighted the following:

Platform Server - The Platform Server [38] is Outsystems server. It is comprised of a group of servers, responsible for the compilation, management, deployment, monitorization and runtime for all applications that are present within the user's infrastructure. It allows for any application to be used in real time as soon as possible, dispensing the need of a localhost. It is automatically installed with the environment's download.

Service Studio - Service Studio [39] is Outsystems development environment for building applications. It is in this component that users can create and publish their applications to the platform server. With each publish action, Outsystems stores a different version of the application being developed in the Platform Data database. As the application is publishing, the platform server will then compile and generate the code for the application and deploy it to a conventional application server. This application server makes use of databases and other external systems, as well as the logic and UI from the code generated to run the application created on a web browser. All the development efforts are made within this component, and as will be seen later in this document, it is one of the most important architectural components of the Outsystems platform.

Integration Studio - Integration Studio is another Outsystems development environment that is mainly used to create and manage extensions to the platform, providing catalysts for integrating with other enterprise systems, such as external code or databases [40]. As it was not used in the project's development, it will not be studied further.

Service Center - Service Center [41] is a management web application used with the purpose of managing and configuring the platform server from an administrative and operational viewpoint. Built into it, are many features useful in the development process. In the Factory tab, it is possible to check the applications available in the user's environment as well as its modules (will be elaborated further) and extensions. Furthermore, it is also possible to monitor the environment and inspect the logs that are generated by the platform and by the running of applications It is a useful tool to analyse when an error or a incorrect behaviour in the application is encountered during the development process, as the Service Center Monitoring saves logs of every request and response sent by the platform. It also has an administration tab, destined to the customization of certain settings within the platform. In short, it is not mandatory to interact with this component during the development process, but its utilization has numerous advantages to its users.

LifeTime - LifeTime [42] is yet another web application Outsystems provides, destined to the management of the application's lifecycle across its distinct stages. It allows User Management capabilities, by managing user permissions and teams and creating new users. It also allows complete visibility over all environments present in the user's infraestructure and manage them accordingly through their lifecycle.

Additionally to these components, Outsystems has also another component called Forge. It essentially functions as a repository that promotes code reusability, in which any user can create code, UI components or other connectors and share their work with other users. Furthermore, it has an extensive community page, where queries can be made about various topics, to exchange knowledge and search for solutions to a problem that has arisen.

3.2.1.2 Service Studio Overview

In the preceding section, it was listed the key components of the Outsystems platform. The one in which developers will allocate more time is undoubtedly the service studio platform. As previously stated, the service studio is where developers create applications and modules on the server. Outsystems benefits from a modular architecture. Modules are reusable objects that contain code specific to certain features of an application. By dividing key areas of an application into modules, its easier to maintain and reuse its code in various different applications. The development capabilities performed inside service studio are always contained within the scope of a module, and its inside it that the different features will be developed.

After the creation of a module, the user can enter the service studio workspace. This is where the main work will be conducted. Its layout is visually oriented, and benefits from a drag and drop structure. The main distinguishable areas in its layout are:

Layers Tab - Each tab bookmarks an application layer in Outsystems: Processes, Interface, Logic and Data. Switching between tabs will highlight elements that correspond to the precise layer. Selecting each element present in the layer, will highlight its properties. The properties grant customization and behavioural capabilities to each element independently.

True Change - This capability, encapsulated in a message list, is responsible for validating the application module. It has 3 possible states: if any error is detected, a button with a red and a cross symbol is shown at the top of the page and the error list is displayed. If no error is detected, and there are changes made in the application that haven't been published to the server, a green button with the number 1 is shown, symbolizing the 1-click publish feature that will be studied further. After successfully publishing the application module to the server, then a blue button appears at the top of the page, and if clicked, will open the application in the machine default web browser.

Canvas - The canvas is located in the center of the workspace and is the main component of the UI. Depending on the type of applicational layer element opened, the canvas will be responsible for displaying its content to the user. If an Interface screen element is selected, a live preview of the screen is displayed. If a logic or processes element is selected, it will display the actions screen, with a visual representation of the logic flow created. If an element from the data layer is selected, the entity attributes will be highlighted.

Toolbox - At the left of the canvas it is located the toolbox. It is always present, unless the user clicks on an element from the data layer. In the interface layer, the toolbox has widgets, which are the constituents of a screen, such as buttons, containers, forms, tables, etc. If an element from the processes or logic layer is selected, then the toolbox changes to accommodate different actions within a logic flow, such as if and switch statements or variable assignments.

Important Mentions - Other workspace elements that are relevant to mention is the element breadcrumb that facilitates navigation through the element hierarchy. Also important to mention the Widget Tree, as it is an important part in the construction of screens, giving users the ability to quickly view the widget hierarchy. Without it, it could be quite challenging to view and manage complex screens where a considerable amount of different widgets are present.

All the material described in this subsection can be found in detail at [43].

3.2.1.3 Application Layers

As previously stated, one of the focal features of Outsystems is presented in the form of application layers. In this section, a brief introduction will be given to them, and their main applications.

Processes Layer - It is divided into 2 distinct categories of elements, namely the Processes and Timers. The processes category, is composed of business processes, representing either human tasks or automated tasks and all the different decisions they may entail. It can also perform events and waits during the duration of these tasks.

The second category, the timers, is essentially is an action with a certain degree of scheduling associated, being it daily, weekly, monthly, or any frame of time. It also allows for the setting of priorities for these timers, in which certain events will take precedence over less important ones, when they have to run concurrently. Additionally, it also permits the creation of timeouts in order for the events not looping indefinitely.

Interface Layer - This layer contains all the components that allow for the creation of the user graphical interface. UI flows, is a category of this layer, comprising of Screens and Blocks. Screens are the *de facto* component that contain all the widgets used in the making of a web page. In HTML terms, they contain all the elements that can be placed inside a web page, such as inputs, containers, labels, tables, lists, checkboxes and many others. A block, in Outsystems, is essentially a reusable screen element with the ability to implement its own logic. They are especially useful in situations where a part of a screen must appear in a given time (or at all times, regardless of the page currently active).

Another category is Images, that can be either pictures or icons, to be used in the screens.

It is also in this layer, that the overall theme of the application can be structured. In CSS terms, it is where the stylesheet comprising the overall aesthetic appearance of the application is constructed, and where all the other stylesheets descend from.

Lastly, there is also the Scripts category, that represent JavaScript resources, that may be required for the adequate use of any element present in a screen.

Logic Layer - This layer is responsible for the logic of the application, both on the client and the server side. It contains Client Actions, that run solely on the client-side, namely a web browser. It also contains Server Actions, that run solely on the server-side of the application. These action can possess input (provides data to an action element to be used inside the action's scope) and output (parameters that are able to return computed action values) parameters It also possesses integrations, in which external systems can be integrated into the application, such as RESTful services.

Data Layer - This layer is responsible for the definition of the different entities, available in a database or locally on the device storage. It also allows for the creation of entity diagrams, similar to Unified Modeling Language (UML) Class Diagrams. It also possesses Structures, that are essentially representations of in-memory data and compound data types. Lastly, and one of the most important features, is Client Variables. These types of variables allow the storage of user-specific data on the client-side, such as login authorization tokens or other informations that require persistence during the entire usage lifecycle of the application.

The material described can be found in detail at [44].

3.2.1.4 1-Click Publish

The 1-Click Publish capability is Outsystems way of storing and uploading the current state of the application that is currently executing on the Platform Server to Service Center. When the button is clicked, the Outsystems creates a new version of the application in the platform data database, and readily deploying and compiling its code onto the application server. After the publish is completed, service studio will display a **Done** message in the publish log, signaling that the application is ready to be used. When that happens, the usual green button with the number 1 is replaced by a blue button, signaling its readiness for use. When the blue button is clicked, it redirects the application to the default browser. With every 1-click publish, Outsystems creates a new version of the developed application, however, if the user has the need to access previous versions, it can be done through the Module Menu. In it, a list with all the existing versions and their dates and times will be shown to the user. Finally, it may occur after the request for a publish, a compare and merge message, to highlight that there versions with differences between them. In these cases, Outsystems allows for the comparison of the differences and to merge them or overwrite them, in order to avoid loss of work.

The material described can be found in detail at [45].

3.2.1.5 Modular Programming

As was referenced earlier, Outsystems makes use of modules to separate different application functionalities. Modules are able to encapsulate all the needed features to execute one aspect of functionality and seperate different functionalities by independent and reusable pieces of code. This way, the code reusability process is made simpler, due to the fact that each module is independent from each other. It is also where the UI and business logic is developed. These modules can have different types depending on the function they provide to the overall system, so it can be concluded that an application is nothing more than a group of correlated modules.

A module can be a Producer, if they are responsible for sharing features, and Consumers in cases where they use features from other modules.

The material described can be found in detail at [46].

3.2.2 Mendix

Another LCDP that was considered being used, was Mendix. This platform was founded in Netherlands in the year 2005, with the goal of providing a faster yet robust way for organizations to create internal software applications [47]. In certain areas, it is quite similar to certain Outsystems features, yet maintaining a certain degree of innovation. It also benefits from an available free version, just like Outsystems, however, it is restricted to 10 maximum users, a lower number when compared to the maximum 100 users of its counterpart.

In this subsection, some of the key features of Mendix will be studied, albeit in a smaller scope when compared to the one given to Outsystems, due to not being chosen for the project's development.

3.2.2.1 Mendix Studio

Mendix Studio is Mendix development environment. It serves the same purpose as Outsystems Service Studio. Its workspace layout is similar to Outsystems, but with certain unique features. The figure 3.2 is representative of the layout of the Mendix Studio Workspace, with highlight on its main areas.

Top Menu Bar - The main features of this area are the preview button and the publish button. The preview, as the name indicates, allows for a preview of the application being developed before being published to the server. The publish button, like in Outsystems publishes the application to the server to be used in a browser.

Left Menu Bar - This area is where the main features of Mendix are found. There it allows access to Pages, domain model, microflows, workflows and search for different elements present in the application, open settings and customize the theme of the application.

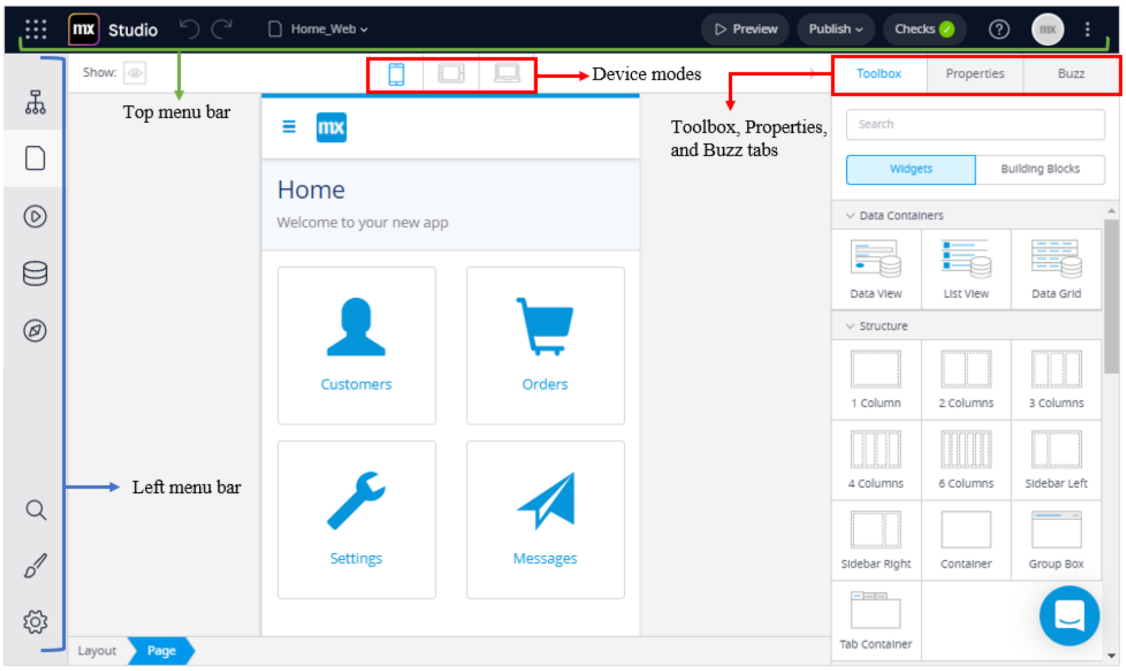


Figure 3.2: Mendix Studio Layout. Extracted from [48].

Device modes - This area is responsible for changing the view of the current page in different application modes: Mobile, Tabled and Responsive. It is helpful to see how the pages will look in different devices.

Toolbox, Properties and Buzz Tabs - The toolbox is similar to Outsystems, where widgets that compose the pages can be chosen, such as lists, tables, columns and others. The properties tab is also similar, highlighting the properties of the selected element. Buzz is a concept not found in Outsystems, that allows for the insertion of comments to different pages, microflows and domain models, permiting them to interact with each other.

3.2.2.2 Workflows and Microflows

Workflows in Mendix are used to build and represent business processes. It has integration with other Mendix features such as microflows. In short, workflows are a visual approach to develop logic in the application. Similar to the actions flow screens in Outsystems, it also has a start and an end, with a multitude of possible activities between them to create a flow chart. Workflows require an input from an end-user, such as a request to trigger the start event of the workflow. Then, the workflow is paused until the request is handled, usually by a widget with logic associated (for example, a button). The logic encapsulated in the button is in the scope of microflows. The figure 3.3 shows an example of a workflow in Mendix.

In Mendix, microflows are nothing more than a visual tool to add custom logic to elements, such as creating, changing or showing objects. It is mainly used in cases where it is needed to change or extend the behaviour of elements, by adding custom logic throughout the application, or to integrate it with other external systems, such as databases or APIs. Like the workflows, they are displayed as flowcharts, with a start and end event. In between, it is possible to connect new events and activities that will mirror the desired

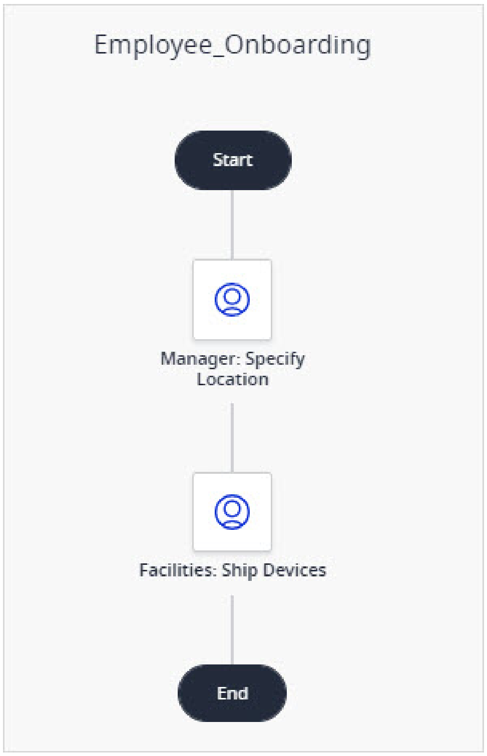


Figure 3.3: Mendix Workflow. Extracted from [49].

logic for those elements. The figure 3.4 represents the start of a microflow, with the activities section on the right corner. Besides the General section displayed, microflows also integrate activities that interact with one or more objects, such is the case of **Object Activities**, **Client Activities** (activities performed in the client-side of the application) and **Workflow Activities** (contain particular activities with the ability to interact with workflows, as mentioned previously).

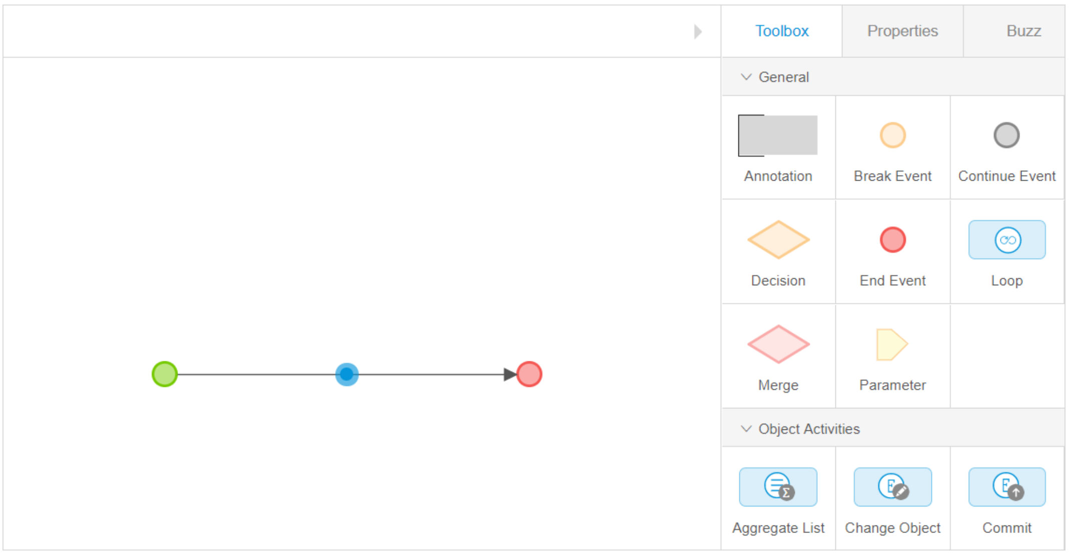


Figure 3.4: Mendix microflow. Extracted from [50].

3.2.2.3 Pages

Pages are the equivalent of the interface layer in Outsystems, representing the UI of the application. It also has widgets, that represent UI elements, such as lists, cards, tables and others. It also has blocks, that are pre-constructed groups of widgets that accelerate the development of the pages. Much like Outsystems, the overall experience of designing pages in Mendix is dragging and dropping widgets to the desired places and modifying its properties when needed, which is why it will not be developed further.

3.2.2.4 Data

Every platform needs to be able to work with data. In Mendix approach, the data is highlighted in a visual editor denominated domain model, where it is possible to see the data model for the application. Much like Outsystems, it is possible to import spreadsheets that represent the data structure or it is possible to work with external data, using Mendix Data Hub.

Mendix also makes use of modules. A module is a unit responsible for dividing functionality of the application into distinct parts. Each module has its own domain model. The domain model is a data diagram that defines the information present in the application domain, consisting of entities (representative of real-world objects with its own attributes) and associations (relation between entities) [51]. Its Outsystems counterpart is the data diagram.

3.3 Nginx

Nginx is an open source software that was initially used to serve as a web server. With its success, their capabilities were increased and can now be used for reverse proxying, load balancing and others [52]. It was chosen over its competitors such as Apache, Cherokee, Varnish and others due to the superiority of its benchmark results in both speed and memory usage [53].

It is the first contact between requests from the Internet and the web application. Nginx handles these requests, and only allows requests that conform to the applied configuration to reach the web application.

It can also be configured to act as a reverse proxy for HTTP requests, to distribute the load between available servers, in order to maintain performance.

It is also responsible for [54]:

- Handle multiple concurrent internet requests.
- Manage request routing, deciding where requests should go and trigger error responses when appropriate).
- Manage slower requests.
- Manage computing resources, mostly Central Processor Unit (CPU) and memory.
- Distribute static files.
- Manage other implemented configurations related to load balancing, reverse proxying, caching

However, nginx isn't able to run Python web applications. To that effect, another component will have to be used.

3.4 Gunicorn

Gunicorn is a Web Server Gateway Interface (WSGI) HTTP server. According to PEP 3333 [55], a WSGI is a specification that characterizes the approach taken to allow communication between a web server and web applications and the way web applications can be linked in order to process a request.

As soon as nginx allows passage to a internet request, it is passed on to gunicorn, according to the rules configured by the user.

Gunicorn is responsible for [54]:

- Operating a set of worker processes. Requests and responses are completely managed by these worker processes For more information [56].
- Convert incoming nginx requests to WSGI compatible requests.
- Transform the application WSGI responses into appropriate HTTP responses.
- Call the appropriate Python code to manage incoming request.
- Interact with different web servers.

As it can be seen, gunicorn excels at what nginx is unable to do, and is incapable of replicating what nginx is able to do. Thus, they complement each other and are commonly used together.

The figure 3.5 is a broad representation of the way nginx and gunicorn interact with a Django application and a PostgreSQL database.

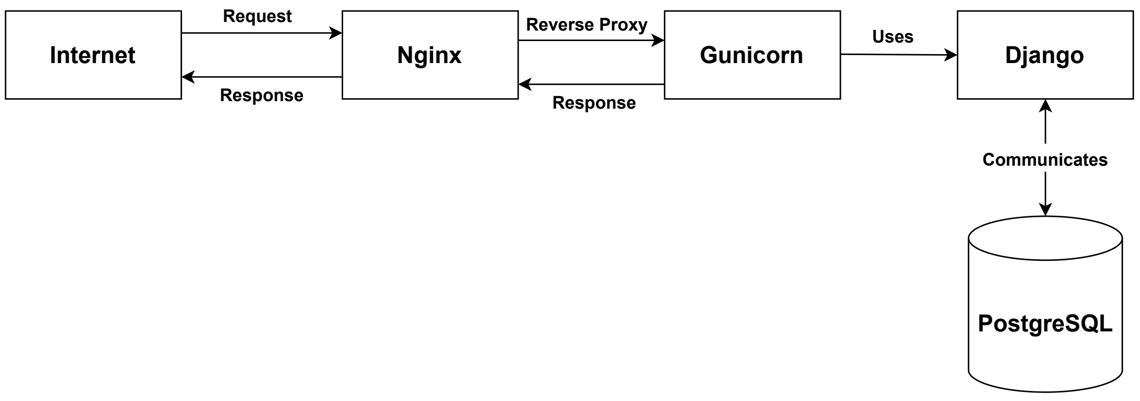


Figure 3.5: Django application deployment with Nginx and Gunicorn.

3.5 PostgreSQL

In the previous chapters it was addressed the technologies used for the backend, the development of the RESTful API and the front-end of the application. All that remains to be covered in this subject is the technology chosen for the application's database system.

The choice settled on PostgreSQL. The main reasons for this selection were based on the fact that Django framework officially supports PostgreSQL and that it is a market proven technology and completely open source. In 2000, Egan [57] considered it to be the most advanced open source relational database in existence, and while since then much has happened in the technology field, PostgreSQL continues to prove its merit by remaining ever relevant until this day.

PostgreSQL has also been atomicity, consistency, isolation, durability (ACID) compliant since 2001, proving that can be relied upon preserving the application's data integrity and security. Also, one of the key factors for this choice, lies on the fact that the server where the database and the Django application is hosted, is managed by a data center company designated Hetzner, that has an extensive documentation on how to create a PostgreSQL database [58].

3.6 Final Considerations

This chapter was focused on exploring the different technologies used or considered to be used in the project's development that this document aims to report. In conclusion, it was opted for the Django Framework for the backbone of the application and the Django REST framework for the development of SIREPH's REST API, as it is a market proven technology and one of the most popular choices for API development. The fact that is run on Python was also taken into consideration, over other programming languages.

For the implementation of the application itself it was opted for a LCDP, due to the increase of its popularity in recent times and the fact that the development efforts are greatly reduced by the use of these platforms. Ultimately, it was chosen Outsystems, due to its interesting free capabilities, the fact that its a leader in the low-code development scene and its high market visibility in Portugal. Although Mendix was considered an option, its free version, documentation and tutorials are significantly lacking when compared to Outsystems, making Outsystems an evident choice for someone who has no knowledge of either of them.

Chapter 4

Planning, Analysis and Design

This chapter will detail the project's planning and analysis efforts. This includes its requirements specifications, the effective project Gantt chart (versus the anticipated plan shown in Chapter 1), and the first steps taken to designing the application's mock up screens.

4.1 Project Management Background

Project Management, as a science, evolved from the growing necessity in guaranteeing a rigorous planning in all phases of a project's life cycle. This necessity, hasn't been borne out of coincidence, but through a substantial amount of research literature on the factors that contribute to a project's success.

Thus, a project success can be viewed as the accomplishment of a singular set of objective and subjective Key Success Indicator (KPI)s, substantiated in the project's success criteria (what must occur for the project to be considered successful) and calculated at the end of a project [59].

However, and it is important to emphasize, that quantifying a project's success is not a menial task, but rather a quite complex concept, further noting that distinguishing between project success and project management success is relevant, and that the success of the former is not entirely dependant on the success of the latter [60]. To further this concept, project performance indicators are typically linked to the compliance with the outlined project schedule and oversight of the project's tasks in a coordinated, time and cost effective way, in order for the project's development to be in line with its proposed purview [61]. Conversely, project management performance is defined as a crucial step in achieving a project's objectives by applying a set of tools and techniques as well as controlling certain factors that can contribute to the failure of a project, such as defective foundation, shortage of technical project management, absence of responsibility to the project, time, cost and end user satisfaction [61].

To conclude, it is advisable to separate project success from project management success, as both phases have different sets of goals [62] and the success of the project is not dependent in its entirety to its planning and management efforts but also other external factors, such as customer satisfaction and perceived value of the project [63]. It is incumbent upon the project managers to deal with these challenges, and to hone their communication skills in terms of customer relationships, in order to guarantee the project's scope relevancy to its intended audience. Failure to do so, and to recognize and calculate the project risks ahead of its development phase, may lead to the organization incurring in significant expenses, project failure and potentially leading to the non-fulfilment of the organization's business goals [61].

4.1.1 Project Management Success Indicators

Project Management, as stated in the previous section, is responsible for planning and organizing all facets of a project. This includes the incentive to achieve the delineated project goals, within the established time frame, budget and performance criteria.

The Iron Triangle model is considered to be the first attempt at producing a prototype that could quantify a project's management success [64]. In the vertex of the triangle, lies the 3 main constraints of a project management scenario: Cost (representing the monetary resources needed), Scope (representing the functionality of the project) and Time (pertains to the scheduling activities). The quality of the project management (and its sub-consequent success), was then dependant upon the equilibrium between these 3 indicators. However, as projects became increasingly larger and complex, and more research was performed on the subject, it was proven to be just a fraction of what could influence the success or failure of the project management.

Since the continuous decline in the use of this model, researchers have been able to conclude that a project's success can be achieved even when an unsuccessful project management has occurred. This happens owing to the fact that wider array of goals can be satisfied, resulting in a long term success, but in the short term, the restricted set of goals that project management incorporate, can be unsuccessful [65]. The same can be happen vice-versa.

Other approaches have been taken to attempt to quantify the success of the project management [66], but it is unanimous in the scientific community, that an important aspect for this success, is the human resources, risk management and procurement management [65].

Lastly, it is difficult to adequately evaluate the success of project management efforts, as it brings both quantifiable and unquantifiable rewards to the project [67], and its success or failure may not be directly linked to the achievement of the project's long term goals.

4.1.2 Project Success Indicators

Although a project's ability to accomplish its goals can't be instantly determined by the quality of its management, it can significantly increase its odds, as it is considerably harder for a project to achieve success without proper management [65].

A project's success is the combination of all work performed by the participants in its life cycle and constraints present in its environment, ranging from the capabilities of the assigned team, to the capabilities of its project manager, as well the organizational culture. Much of the success is dependent on human factors as well as process factors, and their separation is hard to make [61].

To an extent, a project success is also dependant on the stakeholders perception of success, and in this case, as it is of subjective nature, many contradictory opinions can be formulated about the same project. In attempts to categorize success factors in relation to long and short-term goals, according to the time span of expected results, Hyvari [68] broke them down into the following:

- An immediate objective of project efficiency is to satisfy costs, deadlines, and objectives.
- A medium-term objective of customer success is to successfully achieve the technical requirements, functional performance, and fixing the client's problem that initiated the project.
- A long-term objective of a corporation is to achieve commercial success and greater market share, and if that is not applicable to the project's scope or industry (p.e aid initiatives), to build trust, satisfaction, and influence.
- A very long-term objective of planning for the future in the development of new methods, strategies, goods, and markets.

It can be then concluded from this analysis, that the success of a project may also be measured considering certain time frames, and that its failure in a specific interval does not equate to a failure in all periods of time [61]. From this conclusion, it must also be noted that a project may succeed in the long term, but fail in the short term. This phenomenon can appear in industries where product maturation can be beneficial, as a certain portion of the target audience may be resistant to change. The inverse may also happen, in products where time influences negatively the project's perception of success. These industries typically rely on intensive advertisement over their products in the short period of time to build profits as their relevancy tends to wane in the mid to long term.

4.1.3 Overall Planning of the Project Developed

The SIREPH Hospitals and Centrals, as mentioned previously, is the software project that is being studied in this document. In Chapter 1, it is present a Gantt chart with the proposed planification phases of the project. As time progressed, it proved necessary to perform a more extensive planning phase, as there were many gaps that needed correction prior to the beginning of the development process.

For this purpose, another Gantt chart was developed. The chart contained in figure 4.1 is the definitive version of overall planning of the project.

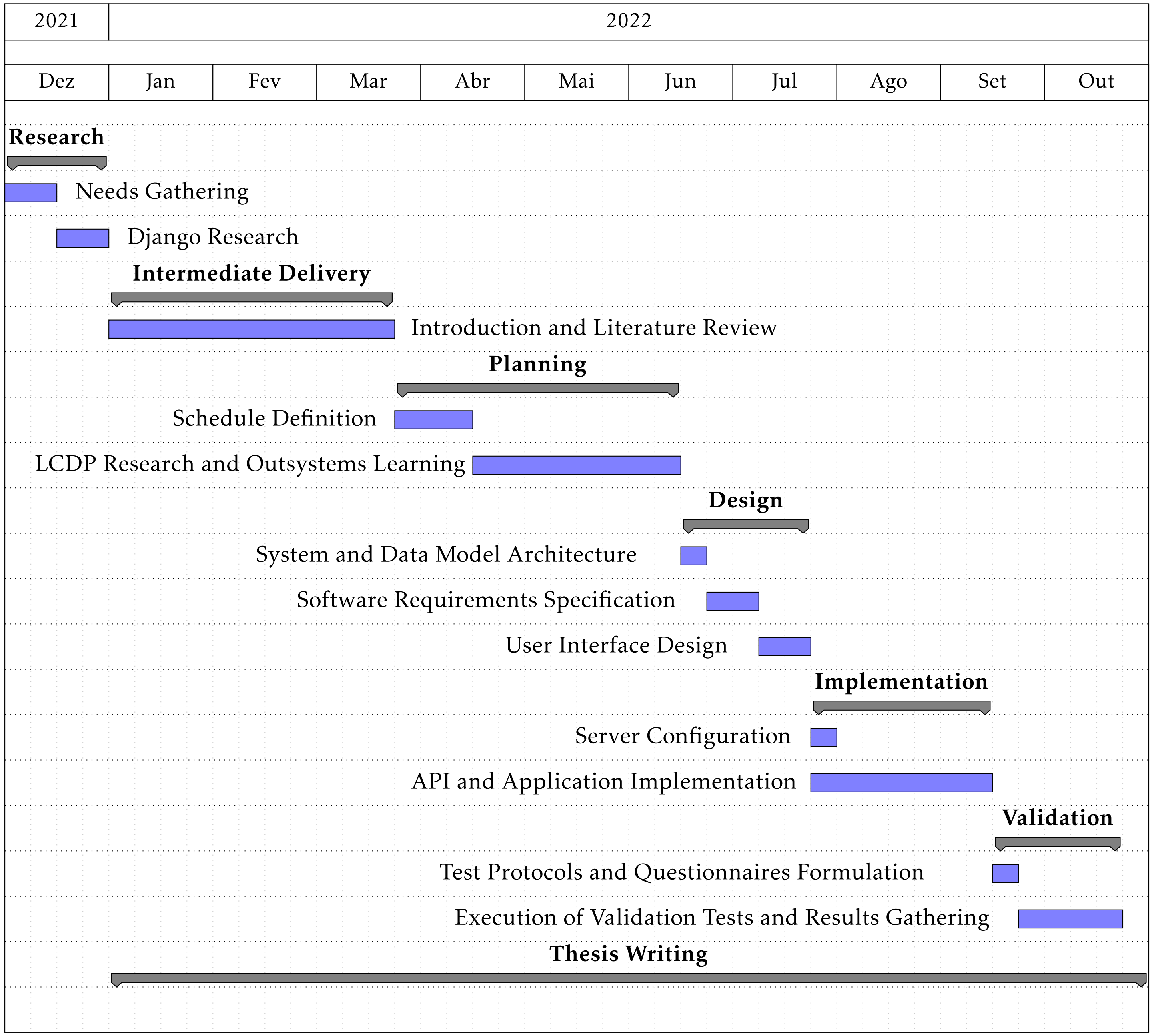


Figure 4.1: Project Real Planning Gantt Chart

Emphasis was given to the correct planning of the project, with a clear schedule and stage definition. The planning stage also saw a technology paradigm change occur. Initially, in the research phase, it was outlined the use of Django in all phases of the system's implementation. Its influence on the project was restricted, and Django's views (only API views from the Django REST framework were used) and templates capabilities were not applied. Instead, all the application logic (business and user interface) was implemented relying on the use of a LCDP platform, which after research, the Outsystems platform was chosen.

The design stage saw the the development of the system architecture and data model, requirements specification and the design of the UI screens.

The technological decision of using Outsystems enabled the decrease in the project implementation time and the deprecation of the deployment stage, as Outsystems allowed for real-time application deployment during implementation. The server configuration was moved to the implementation stage, because it was considered as the natural time for its realization and its was essential for the application implementation that the Django code was in production.

After implementation, the system validation started with the formulation of test protocols and questionnaires to be executed by users to validate the suitability of the application.

Therefore, the decisions taken that resulted in the evolution of figure 1.2 into figure 4.1, made the implementation time decrease and removed the deployment phase. Furthermore, it enabled the acquisition of knowledge related to Outsystems technology and LCDP platforms in general.

4.1.3.1 Development Methodology Employed

During the entire development process of the software in study, a specific methodology was not followed, but rather a compilation of good practices drawn from various studied methodologies, adapted to the characteristics of the project.

So instead of choosing and following a single methodology, research was performed on the subject to try and maximize the development efforts. To this effect, the development process was divided into stages, each with their own unique scopes, time frames and deliverables, but nonetheless related to each other on the broader scale of the project. The stages highlighted were:

A. *Research*

Even prior to the start of the project, contacts were made to *Associação Nacional dos Técnicos de Emergência Pré-Hospitalar* (ANTEPH) [69] to exchange information about the possibility of the project going forward. As this is a non-profit organization, the interest for now was purely academical, as a proof of concept software with the possibility for continuous growth in the long term. After the project was accepted, the research phase began. It was in this stage that the first technological and infrastructural decisions were made, namely the choice in favour of Django framework for the system backbone and API development, PostgreSQL as the relational database system and the use of Nginx and Gunicorn. It was also in this stage that the needs that later would be materialized into requirements were specified.

B. *Planning*

During the planning phase, scheduling and organization efforts were performed and the project management methodology aligned. The weekly meeting schedules and deliverables were proposed, and the tuning of any pending issues from the previous stage was performed. This stage also saw the decision of using Outsystems for the application implementation instead of Django's views and templates. In the end of this phase, a Gantt chart with all the planned stages and their actions was constructed, that would later guide the work on the future stages.

C. *Design*

During the design phase, the system architecture was delineated as well as its data model diagram. The final requirements draft were specified and the design of the UI screens was performed. All the deliverables produced in this stage were displayed to the interested parties, so that tweaks could be made before the start of the implementation stage.

D. *Implementation*

The implementation stage was characterized by the development of the SIREPH API and the application logic (business and user interface logic). During this period, as was with the other stages, weekly meetings were scheduled to monitor and update the interested parties on the project progress. It was also performed API documentation, debugging processes to remove application errors or incorrect behaviour.

E. *Validation*

This stage saw the formulation of test protocols and questionnaires to be given to test users, to ascertain if the project could be considered a success given the criteria proposed during its planning.

From the outset of the project, it was sought to ensure that all stages of development were carried out rigorously. To that end, time was invested in researching development methodologies and their best practices. In the end, not only one methodology was strictly followed, but rather a group of good practices pertaining to different methodologies. The ones that can be highlighted are:

A. *Waterfall*

To a certain extent, the stages delineated followed the waterfall methodology. This methodology applies an ordered and linear approach to software development, where each stage starts when the previous is finished, flowing from one to the other, much like the cascading layers of a waterfall [70]. The project is divided into a progression of activities, also known as phases or stages. In the traditional sense, this methodology aims at delivering the full system at once. There are many advantages in its use, namely the fact that requirements are completed early in the project, which in turn enables the team to describe the project scope, schedule and design the application. It also allows for a more thorough knowledge of the application's requirements and deliverables [71]. However its use also has downsides. It is quite common for the requirements to change during the project's life cycle, it requires a very meticulous analysis of the activities and deliverables for the overall project, which may prove to be above the team's competence in the beginning of the project and it is common for project's following this methodology to fail meeting the projected schedules and budgets due to trying to deliver everything simultaneously [71].

In the project in study, this methodology was used in the stages formulation, and in its attempt of following a sequential approach to development where the later stages of the project would only commence when the earlier stages were completed (something that did not happen entirely). As the software being built was grounded on institutional practices and rules, it was predicted that the requirements would not change during the entire development, so a first draft was created in the early stages (as will be seen later, it was not the finished document). To avoid the constraints of this methodology, its influence was limited to the steps detailed.

B. *Prototyping*

This methodology is based on a prototype, derived from its selected function, that is created, tested and reconstructed until an adequate outcome is attained from which the complete software can be developed [72]. It must be built promptly and frequently, without regard for the best programming practices. After the prototype is finished, the team resumes to the current stage and applies the knowledge gathered in the creation of the prototype in the real project.

In the project in study this method was applied in certain areas during the implementation stage, mostly in frond-end implementations. One instance of a prototype being created was in trying to solve a connection problem between the REST API and the Outsystems platform. A new project was built, and simple implementations were conducted to find the missing feature that allowed the connection to be established. After the prototype was tested and deemed successful, it was deprecated and the knowledge gathered from it applied in the real project. It was also applied prototyping in the construction of front-end screen logic. As the Outsystems platforms saves old configurations, there were times where the real project screens were put on hold, and tests were conducted in a controlled environment in these old configurations. After the corrections were made, the real implementation could resume.

C. *Iterative and Incremental*

A logical alternative to the waterfall model is the incremental model. In it, multiple iterations of smaller cycles of each stage yields a software deliverable, that enhances and expands upon the previous one [73]. This deliverable is then presented to the project owner and if no objection is met, the development process continues. It differs from the prototyping model, as the deliverable is added to the real project after completion, and not deprecated. The development process continues until the real software application is finished [72].

This methodology was followed throughout the entire project. When the stages were planned, it sometimes proved beneficial to divide them into smaller cycles of development, and increment deliverables with each meeting. The requirements specification was created using this model, in which were created iterations that when finished composed a finished requirements draft as a deliverable. Also in the design and implementation stages, the work was presented incrementally and not everything simultaneously. This provided a degree of quality control over the work being performed, that would not be possible if only the waterfall approach was took.

4.1.3.2 Project Management Success Criteria

To assess whether the management of the project under study was successful, it proved crucial to establish a success criteria. As this project is purely academical certain success indicators couldn't be considered, such as financial resources.

During the planning phase, it was devised how the project's management tasks would materialize in all stages of development. To that end, great importance was given in scheduling tasks, concerning both the time of the project and its scope.

It was planned weekly sessions with the project's interested parties, where a detailed account of the work performed between meetings would be given. The subject of these meetings would change according to the development stage the project was in and in the work performed. In cases where a deliverable had to be reworked or improved upon, a portion of the next week's work would have to be dedicated to improving it.

In general, the criteria outlined to measure the success of the project management were:

- At the project start, all development stages must be delineated and comprehended. Time estimation to complete the tasks required to finish each stage must also be calculated and acceptable to the interested parties.
- The planned time estimation vs actual execution must be deemed acceptable by the project's interested parties.
- The number of adjustments to the schedule of the project as a whole must be acceptable to the interested parties.
- The on-time task completion percentage must be acceptable to the interested parties.
- The number of change requests must be acceptable to the interested parties.

Important to refer, as was mentioned previously, that the success of the project in the short term is not dependent on its successful management, however, in order to guarantee an acceptable final result, importance was given not only on its development but also to its short and long term management efforts. In the results discussion part of this document, it will be possible to discuss if the project management was successful, according to the delineated criteria.

4.1.3.3 Project Success Criteria

The success of the project under study is deeply rooted in its suitability to improve upon the existing pre-hospital emergency processes and to fix as much as possible the limitations that stem from the human processes that are currently in place. In order to quantify if the project meets its success criteria, the application would be user-tested and its results would be studied and their suitability deemed acceptable or not.

In general, the criteria outlined to measure the success of the project were:

- The application's acceptance tests are deemed approved if at least 85% of the users consider the application suitable to improve the current human processes.
- The time required to complete the compulsory tasks are deemed acceptable by the interested parties.
- The project, as a proof of concept, is considered acceptable by the interested parties to achieve future project feasibility.
- The project functionalities conform to the outlined requirements at a level that is acceptable to the interest parties.

4.2 Software Requirements Specification

This section delves into disciplines pertaining to requirements engineering, described in detail in ISO/IEC/IEEE 29148:2018. This standard served as a basis for the Software Requirements Specification (SRS) activities that were employed during the development of the project in study.

4.2.1 SRS and Project Success

The link between SRS and project success is one that has been studied in depth throughout the years, and extensive literature on the subject illustrates that most certainly a quality SRS may have a meaningful impact on a project's ability to achieve success. However, it still proves difficult to objectively determine if a project SRS fails to reach a pre-determined standard, as there is no quality model universally defined and inordinate amounts of empirical data are required to provide correlation between project success and SRS quality standards [74].

Knauss and El Boustani [74] formulated a set of hypothesis that aimed at estimating the quality of SRS and the link between project success. Their research hinted at a substantial link between quality of SRS and project success. To measure the results, they proposed a technique to quantify SRS quality based on the Goal-Question-Metric (GQM) method. Their final results on 40 projects concluded that 87% of the projects that scored more than 44 points were successful (software used in the intended manner) and all projects that failed (some of the proposed goals or failed to deliver a working software entirely) scored below 40 points, suggesting a possible correlation between quality SRS and the outcome of the project.

Related literature on the subject has been published [75][76] with similar conclusions albeit with different approaches taken. Limitations to these approaches are connected to the projects unique characteristics and the difficulty to define what is considered a quality flaw in SRS.

4.2.2 Needs vs Requirements

First and foremost, the requirements definition stage starts with the declaration of the stakeholder needs (or objectives), that must be refined and matured before being considered valid stakeholder requirements. These needs can't be defined as requirements as they lack definition and eventual consistency and feasibility [77]. Through requirement engineering methods, the initial needs are transformed into clear, objective and structured declarations of the stakeholder requirements and objectives.

The first step to understand the project's stakeholder needs was booking a meeting with their representative. This meeting proved to be useful, as the knowledge of the pre-hospital environment at the beginning of the project was limited, and a clear idea of what was needed was not in place. From the meeting, it was possible to ascertain the flow of a pre-hospital occurrence, since it is reported by the victim until it reaches the adequate healthcare unit. It was also possible to conclude the following:

1. After a victim has made aware that it needs medical assistance, an administrative emergency central (in Portugal this falls under the purview of CODU) must receive the occurrence.

2. After the reception of the occurrence is made, an administrative emergency central user must assign the occurrence to a non-administrative emergency central, such as a CVP station or a fire station present in the system. This assignment cannot be made automatically as there are many constraints besides geographical that could impact this choice. Beyond this required action, an administrative emergency central user can also view and update pertinent occurrence information, as well monitor the states of the occurrence.

3. After the assignment to a non-administrative station is made, the user belonging to that CVP or fire station must verify the admission of the occurrence into his workspace. It must also record the means of assistance that will be used in the handling of the occurrence and assign a team, that is comprised of pre-emergency personnel, belonging to that same station, responsible for the onsite victim assistance. Beyond these required actions, a non-administrative central user may also view and update pertinent occurrence information, and monitor the states of the occurrence.

4. After the assignment of a team to the occurrence, the non-administrative emergency central users must monitor the occurrence status, geographically, as well as duration of the occurrence states, until it is completed and no longer active in the system.

5. After the victim assistance is completed, and the adequate health unit chosen, the technicians must send the victim information to the hospital that will be responsible for the treatment (this hospital must be present in the system and the act of sending the information is outside this application's purview).

6. The hospital user must verify the reception of the information, and check the victim's information, as well as the information generated by the technicians during the handling of the occurrence (evaluations, symptoms, administered drugs and procedures).

These needs were the first step to define the system's requirements, as it will be later seen.

4.2.3 User Characterization

Knowing the end-users of one's software is crucial to its success. This stage in requirements engineering, characterizes the generic characteristics of the desired audience for the software, taking into account factors that may affect usability like educational attainment, work history, disability, and technological know-how [77].

As already mentioned, the project in study has 2 subsystems: one is destined to emergency centrals and another for hospitals. The subsystem dedicated to emergency centrals has 2 types of users, one for administrative centrals and another for non-administrative centrals, whereas the hospital subsystem only has one type of users.

In general:

A. *Administrative Emergency Central User*

This type of user is able to observe all pre-hospital emergency occurrences that are currently in the system, and allocate them to a non-administrative emergency central. It is also able to monitor the status of all occurrences and check the history of all completed occurrences.

It is a user that must have proper knowledge of the pre-hospital emergency framework, adequate computer literacy and good interpersonal relationship skills.

These users have as their main task the allocation of created occurrences, which will appear on their main page, to the appropriate non-administrative emergency central. They may also change the information received from occurrences if there is new information regarding them. They may also monitor any occurrence if it is deemed necessary, to assist in the execution of the occurrence.

B. *Non-Administrative Emergency Central User*

This type of user is only able to interact with occurrences allocated to his/her emergency station (fire station or CVP station), and in turn, only these will be available for monitoring and will only be able to see the history of completed occurrences from his/her central.

It is a user that must have proper knowledge of the pre-hospital emergency framework, adequate knowledge of how their central operates in order to correctly assign occurrences to a team, adequate computer literacy, good interpersonal relationship skills to aid the technician's on-site if need arises.

These users have as their main function the allocation of an active team from their corporation, so that it can handle the occurrence. They can also change the information received, if new information arises. They can also monitor the status of the allocated occurrence with the help of the map in the monitoring page until the occurrence is deemed completed.

C. *Hospital User*

As with users of non-administrative centrals, these users can only see information regarding victims on the way to, or that arrived at their hospital. They can also only see the history of victims treated at their hospital.

It is a user that must have proper knowledge of medical related terms, regarding the evaluations, symptoms, pharmacies and procedures performed on the victim, or is able to correctly transmit this information to staff that have. These users can be from nurses, to doctors to hospital administrative staff. So they must be able to work under pressure, have adequate computer literacy and good interpersonal relationship skills.

The main function of hospital users is to fill in the date and time when the patient was registered in the hospital. Being the final users of the pre-hospital emergency environment before entering the hospital environment, they must also consult and check the information received about the patient to correctly treat him or give the knowledge to who will.

4.2.4 Software Requirements

A requirement is declaration that expresses a need and its associated constraints and conditions [77]. A constraint is a externally imposed restriction on a system's development or modification process, its design, or implementation. A condition is a quantifiable property that is defined for a requirement, that identifies a situation under which a requirement applies [77].

So, it can be concluded that a requirement is a capability that a system must satisfy in order to comply with an agreement, standard, specification or any other contractually issued document [78]. 3 types of software requirements will be detailed: functional, non-functional and user requirements.

The requirement priority approach taken for the project in study was the MoSCoW method.

4.2.4.1 Functional Requirements

The tasks or functions that the system must implement are described by its functional requirements [77].

The tables ranging from 4.1 to 4.10 pertain to the **Emergency Central subsystem** and table 4.11 to 4.16 pertain to the **hospital subsystem** .

Table 4.1: Emergency Central Subsystem Functional Requirement 1 - Use Restriction

ECSFR1	**Use Restriction**	**Priority**	Must Have
The application must restrict non-registered users in the SIREPH system access to its content. Details: Users must acquire credentials to access the system. This process is managed by the administration team, and there is no register operation available to the general public.			

Table 4.2: Emergency Central Subsystem Functional Requirement 2 - Dispatcher Profile Restriction

ECSFR2	**Dispatcher Profile Restriction**	**Priority**	Must Have
The application grants emergency central subsystem access only to dispatchers registered in the SIREPH system. Detail: Users must be given dispatcher roles to access the subsystem. This process is managed by the administration team.			

Table 4.3: Emergency Central Subsystem Functional Requirement 3 - Visualization Authorization

ECSFR3	**Visualization Authorization**	**Priority**	Must Have
Access to occurrences information must be subject to authorization roles. Detail: Users from non-administrative centrals, must only see occurrences that are assigned to their emergency central. Users from administrative centrals, must see all occurrences present in the SIREPH system.			

Table 4.4: Emergency Central Subsystem Functional Requirement 4 - View Occurrence Info

ECSFR4	**View Occurrence Info**	**Priority**	Must Have
The application must allow the visualization of all the information related to a specific occurrence.			

Table 4.5: Emergency Central Subsystem Functional Requirement 5 - Update Active Occurrence Info

ECSFR5	**Update Occurrence Info**	**Priority**	Must Have
The application must allow the update of active occurrence related information.			

Table 4.6: Emergency Central Subsystem Functional Requirement 6 - Occurrence Allocation Restriction

ECSFR6	**Occurrence Allocation Restriction**	**Priority**	Must Have
Permission to assign a central to an occurrence is dependent on the user's role. Details: Only administrative central users are able to assign occurrences to a non-administrative central.			

Table 4.7: Emergency Central Subsystem Functional Requirement 7 - Occurrence Team Allocation

ECSFR7	**Occurrence Team Allocation**	**Priority**	Must Have
Permission to assign a team to an occurrence is dependent on the user's role. Detail: Only non-administrative central users are able to assign an occurrence to an active team belonging to their emergency central.			

Table 4.8: Emergency Central Subsystem Functional Requirement 8 - Occurrence State Monitoring

ECSFR8	**Occurrence State Monitoring**	**Priority**	Should Have
The application must allow geographical and temporal monitoring of occurrence state updates.			

Table 4.9: Emergency Central Subsystem Functional Requirement 9 - Occurrence Current State

ECSFR9	**Occurrence Current State**	**Priority**	Should Have
The application must display the current occurrence state.			

Table 4.10: Emergency Central Subsystem Functional Requirement 10 - Occurrence History List

ECSFR10	**Occurrence History List**	**Priority**	Must Have
Authorized users are able to view occurrences historical data. Detail: For non-administrative central users the application must keep a record of all completed occurrences assigned to their central. For administrative central users the application must keep record of all completed occurrences within the system.			

Table 4.11: Hospital Subsystem Functional Requirement 1 - Use Restriction

HFR1	**Use Restriction**	**Priority**	Must Have
The application must restrict non-registered users in the SIREPH system access to its content. Details: Users must acquire credentials to access the system. This process is managed by the administration team, and there is no register operation available to the general public.			

Table 4.12: Hospital Subsystem Functional Requirement 2 - Hospital Staff Profile Restriction

HFR2	**Hospital Staff Profile Restriction**	**Priority**	Must Have
The application grants hospital subsystem access only to registered hospital staff in the SIREPH system. Detail: Users must be given hospital staff roles to access the subsystem. This process is managed by the administration team.			

Table 4.13: Hospital Subsystem Functional Requirement 3 - Visualization Restrictions

HFR3	**Visualization Restriction**	**Priority**	Must Have
The application must solely display patients that are assigned to the logged user's hospital.			

Table 4.14: Hospital Subsystem Functional Requirement 4 - Consult Patient Info

HFR4	**Consult Patient Info**	**Priority**	Must Have
The application must allow the consultation of all patient related information.			

Table 4.15: Hospital Subsystem Functional Requirement 5 - Patient Check-in

HFR5	**Patient Check-in**	**Priority**	Must Have
Hospital staff must be able to check a patient in. Detail: When a patient is assigned to an hospital, a hospital staff user must place the date and time of arrival, in order for the patient to be registered in the hospital history.			

Table 4.16: Hospital Subsystem Functional Requirement 6 - Patient History List

HFR6	**Patient History List**	**Priority**	Must Have
The application must keep a record of all patients that were attended at the logged user's hospital.			

4.2.4.2 Non-Functional Requirements

These requirements should be identified prior to starting requirements specification. Include a set of criteria that define the use of a system, or what the system must be, such as transportability, reusability, reliability, security, integrability and other "ilities"[77].

The non-functional requirements specified were:

Table 4.17: Non-Functional Requirement 1 - Responsive

NFR1	**Responsive**	**Priority**	Must Have
The application must be developed as a Reactive Web Application so that it is optimized for all computer devices.			

Table 4.18: Non-Functional Requirement 2 - Usability

NFR2	**Usability**	**Priority**	Must Have
The application graphical interface must be efficient to use, intuitive and have minimal apparent [79] workload.			

Table 4.19: Non-Functional Requirement 3 - Availability

NFR3	**Availability**	**Priority**	Must Have
The application must be available 90% of the time.			

Table 4.20: Non-Functional Requirement 4 - Security

NFR4	**Security**	**Priority**	Must Have
The application must implement security measures to minimize unauthorized access to functionality and stored data.			

Table 4.21: Non-Functional Requirement 5 - Integrability

NFR5	**Integrability**	**Priority**	Must Have
The application must consult and persist information in the SIREPH's system.			

Table 4.22: Non-Functional Requirement 6 - Integrity

NFR6	**Integrity**	**Priority**	Must Have
The application must accurately display users, patients, and occurrence's details.			

Table 4.23: Non-Functional Requirement 7 - Performance

NFR7	**Performance**	**Priority**	Must Have
The application must ensure web response times ranging between at least 100 milliseconds and 1 second [80].			

Table 4.24: Non-Functional Requirement 8 - Reliability

NFR8	**Reliability**	**Priority**	Must Have
The application must be reliable, allowing the users to correctly perform their tasks without failure 99.9% of the time.			

Table 4.25: Non-Functional Requirement 9 - Extensibility

NFR9	**Extensibility**	**Priority**	Should Have
The application should be able to accommodate future updates without major architectural, design and implementation changes.			

Table 4.26: Non-Functional Requirement 10 - Scalability

NFR10	**Scalability**	**Priority**	Must Have
The application must be able to scale to accommodate the expected service load for medical emergency response at the national level.			

4.3 Project Design Overview

This section details the design approaches followed in this project.

4.3.1 System Architecture

The proposed system architecture required a server to host the Django code and the PostgreSQL database to be used while in production. To this effect, it was bought a single server from the Hetzner company, to store these different components. After the server's purchase, it was studied the installation of nginx [81] and the use of gunicorn.

Initially, if the costs were acceptable, it was formulated the purchase of 3 servers and one load balancer. 2 web servers hosting the Django application and the load balancer distributing incoming application traffic between them and the other server hosting the PostgreSQL database. In the 2 web servers it would be used nginx as a reverse proxy for the Django and gunicorn [82]. Gunicorn would then be used to manage the communication between both web servers, distributing the traffic caused by concurrent web requests and maintaining multiple web application executing.

As the costs were too high, it was bought a single server as mentioned before, but the installation of nginx and gunicorn was still verified. In the server, all the components are within the same server running on different processes where each exposes its port. As seen in figure 4.2, nginx maps the server's local access port 8000, where gunicorn serves the Django application, to its port 82, which allows external access. The database process exposes the port 5432 and Django uses this port to connect the application to the database.

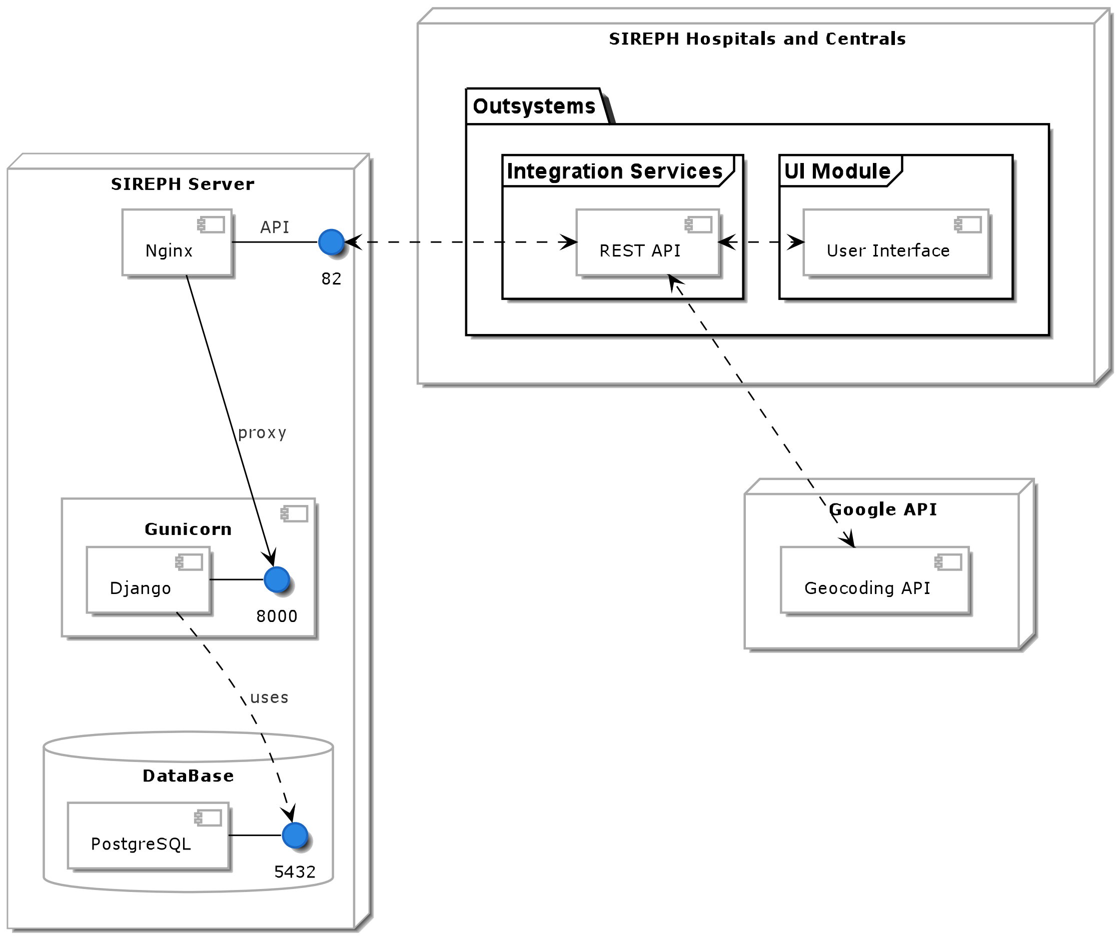


Figure 4.2: Project System Architecture

Besides the reasons stated in Chapter 3, the choice behind these 2 artifacts was also based on their extensive documentation and the fact that they are the most used solutions for hosting Python-based applications.

While using the Outsystems platform, it was created a module specifically to integrate the API with the Outsystems platform. To that end, it was necessary to create API documentation that could be understood by the Outsystems platform. As mentioned in Chapter 3, it was opted for the OpenAPI specification. In the Django urls, it was devised an endpoint for the swagger (78.46.215.82:82/api/swagger/) and it was this URL that was inserted in Outsystems to consume the developed REST API. The UI module was created to develop the UI and User Interaction and Experience (UIX) of the application. Through Outsystems capabilities the UI module communicates with the integrations module and any changes applied in the API must be refreshed in the integration services module for it to be used in the user interface. It is common practice to create an Outsystems module designated for "Core services", due to the fact that Outsystems can use its local database.

As the data model used in the application was in its entirety the data model present in the PostgreSQL database, this capability was dismissed for the remote database.

Lastly, it was used the Google Geocoding API to read the coordinates of an occurrence state and to accurately display it on a Google Maps interface. This API was connected to the REST API through the integration services module.

4.3.2 Data Model

A data model is an artifact responsible for representing a system's data structures and their relationships as well as its business rules embodied by the business requirements [71]. In the first stages of the project development, a data model was constructed following the proposed requirements and institutional practices in place. As the project in study was proposed as an alternative/complementary system to an already established practice, the data model created sought to mirror the current established relationships.

The main basis for the data model developed was the INEM's procedure document, in Portuguese denominated as *"Verbete INEM"*. This document is used as a standard in all pre-hospital emergency related capacities, as a means to document an occurrence's execution and its pertinent information. This document can be found in annex I.

After the entities found in the document were specified in the data model, it was sought to implement additional entities that participate in the pre-hospital environment, such as the emergency centrals, dispatchers, technicians, teams, hospitals, hospital staff and others.The final data model is represented in appendix A.

From the data model, it can be extracted the following entities:

A. *User*

Django's default User model present in its authentication system **django.contrib.auth**. It represents a user, of any type, registered in the SIREPH database. For more information about Django's authentication system, refer to the Django's documentation at [21].

B. *Central*

This entity represents an emergency central. It has fields related to its designation, address, area of action and contact. Moreover, it possesses a Boolean field denominated **is_administrative** that allows the distinction between an administrative and a non-administrative emergency central.

C. *Technician*

This entity represents a specific type of user, known in the SIREPH system as Technician. These users are the ones responsible for the on-site occurrence execution. They are the main actors in the technician's application, an application outside this project's purview but related nonetheless to the entire SIREPH system and to the project in study. As fields, it possesses a central identifier and user identifier (known in this entity as technician_id to differentiate between other entities) as Foreign Key (FK), as well as an active Boolean field that represents if the technician is active inside the system.

D. *Team*

This entity represents a group of technicians, that join together during the execution of an occurrence. As fields, possess a central identifier as FKs and an active Boolean as well. As a technician can be a part of more than one team and a team can be composed of more than one technician, it proved necessary the creation of a intermediary entity between the Technician and Team entity, as will be seen next.

E. *TeamTechnician*

Intermediary entity between Team and Technician. Represents a technician's team. As fields, it possesses a Team and technician identifier as FKs, an active and team_leader Boolean. If the latter is True, the technician is the leader of the team he is a part of.

F. *Dispatcher*

Entity that represents a specific type of user, known in the SIREPH system as Dispatcher. These dispatchers are the end users of the project's emergency central subsystem and represent the workforce in the emergency centrals responsible for receiving and monitoring the inbound or assigned occurrences. As fields, it possesses a central and a user identifier (in this entity it is known as dispatcher_id) as FKs and a Boolean denominated active.

G. *Hospital*

This entity represents an hospital. It has fields relating to its name, address, contact, capacity (integer that represents a hospital supported capacity) and current_capacity (integer that represents the hospital's current capacity).

H. *HospitalStaff*

Entity that represents another specific type of user, known in the SIREPH system as Hospital Staff. They are the end users of the project's hospital subsystem and represent the workforce in the hospitals responsible for receiving and consulting the entirety of the occurrence victims information generated during the pre-hospital emergency environment. AS fields, it possesses a hospital and user identifier (in this entity known as employee_id) as FKs and a Boolean denominated active.

I. *Occurrence*

This entity is responsible for storing all occurrence related information. Its fields were extracted from the INEM document mentioned previously, such as: Occurrence Number, mean of assistance, motive, number of victims, local, parish, municipality, team and central identifier as FKs, and Boolean fields such as active (if true occurrence is active) and alert mode (pertains to the civilians application outside of the project in study purview). The FKs exemplify the relationship between entities, as an occurrence has but one team assigned but a team can have multiple occurrences assigned and a occurrence has but one emergency central assigned but an emergency central can have multiple occurrences.

J. *State*

This entity represents a state. In the SIREPH system, a state pertains to the current designation of the position a team can have during a occurrence. It has 5 possible values:

A caminho do local: This state is triggered when the occurrence's assigned team is on their way to the emergency site.

Chegada à vítima: This state is triggered when the occurrence's assigned team arrives at the victim(s) location.

A caminho do hospital: This state is triggered when the occurrence's assigned team starts the victim(s) mobilization efforts from the emergency site to the target healthcare unit.

Chegada ao hospital: This state is triggered when the occurrence's assigned team arrives at the target healthcare unit.

Fim da ocorrência: This state is triggered when the occurrence's assigned team declares the end of the occurrence.

K. *OccurrenceState*

Intermediary entity between the Occurrence and State entities. As an Occurrence can have multiple states and a state can exist in multiple occurrences, this entity allows to uniquely identify a state within an occurrence. Its fields are: occurrence and state identifier as FKs, longitude and latitude as the geographical coordinates to where the state was triggered and date_time that represents when the state was triggered.

L. *TypeOfTransport*

Entity that represents the type of transport used in the victim(s) transportation. Its values are derived from the possible choices present in the INEM's document:

Não Transporte: In cases where victim(s) transportation doesn't occur.

Primário: In cases where victim(s) is(are) transported between the emergency site and the target healthcare unit [83].

Secundário: In cases where victim(s) is(are) transported between healthcare units [83].

M. *NonTransportReason*

In cases where the type of transport selected was "Não Transporte", the document requires a reason for the victim's non transportation, and specifying it, is the purpose of this entity. Its values are also derived from the possible choices found in the INEM document:

Abandonou o local: In cases where the victim(s) abandoned the emergency site.

Decisão Médica: In cases where the transportation didn't occur due to a medical decision.

Morte: In cases where the victim(s) died in the emergency site.

Recusou e assinou: In cases where the victim(s) refused transportation and signed a form.

Recusou e não assinou: In cases where the victim(s) refused transportation and didn't sign a form.

Desativação: In cases where the occurrence was deactivated.

N. *Victim*

This entity represents a victim. Its fields are derived from the INEM document as well. As they are too numerous to state individually, it will only be mentioned a few: occurrence, non_transport_reason, type_of_transport and hospital identifiers as FKs and fields such as name, birthdate, age, identity number and circumstances. It is related to the Occurrence entity as an Occurrence may have more than one victim but a victim can only have one occurrence (at the same time) and the same relationships can be defined to the remaining FKs.

O. *Evaluation*

Entity that represents an evaluation performed a specific occurrence victim. It has a victim identifier as a FK due to the fact that a victim may have multiple evaluations but an evaluation can only belong to a single victim. The other fields can be found in the INEM document in the evaluation tab.

P. *Glasgow Scale*

Entity that represents information pertaining a Glasgow Scale evaluation. It has as FK an evaluation identifier. Glasgow Scale is a clinical scale used to determine the consciousness level of a victim of a brain injury.

Q. *Pharmacy*

Entity that represents a pharmacy administered to a specific occurrence victim. Like the evaluation entity it has a victim identifier as FK. Its fields can also be found in the document in the pharmacy tab.

R. *Symptom*

Entity that represents the symptoms of an occurrence victim. It has a One-to-One relationship to the Victim entity.

S. *Trauma*

Entity that represents a subset of symptoms, related to trauma injuries. It has as FK a symptom identifier.

T. *ProcedureRCP*

Entity that represents information pertaining to Cardiopulmonary resuscitation procedures, in Portuguese is known as Reanimação Cardiopulmonar (RCP). All the procedures are detailed in the document. Relationship of One-to-One with the Victim entity.

U. *ProcedureVentilation*

Entity that represents information pertaining to Ventilation procedures. All the procedures are detailed in the document. Relationship of One-to-One with the Victim entity.

V. *ProcedureCirculation*

Entity that represents information pertaining to Circulation procedures. All the procedures are detailed in the document. Relationship of One-to-One with the Victim entity.

W. *ProcedureProtocol*

Entity that represents information pertaining to Protocol procedures. All the procedures are detailed in the document. Relationship of One-to-One with the Victim entity.

X. *ProcedureScale*

Entity that represents information pertaining to Scale procedures. All the procedures are detailed in the document. Relationship of One-to-One with the Victim entity.

Y. *News*

Entity that doesn't possess any relationship to the remaining entities and is only used by the civilian application, outside the project in study purview.

4.3.3 User Interface

One of the activities performed in the design stage was the creation of mockups for the application screens. These mockups were the first visual representation of the outlined requirements and served as the basis for the UI. It is also important to refer, that as the requirements were refined, the final screens became slightly different from the produced mockups, but nevertheless the latter continued to serve as a guideline in their creation. As the UI of the application had significant changes according to the type of institution the current user was logged on (emergency central or hospital), the design process was divided according into 2 categories: the mockups for the emergency centrals UI and for the hospitals UI.

It only remains to mention that the software used in the creation of these mockups was Balsamiq, a company responsible for creating website wireframes. It was chosen due to its simplicity and quality in producing interactive UI previews. More about Balsamiq can be found at [84].

4.3.3.1 Emergency Central Subsystem

The emergency subsystem screen hierarchy is as depicted in figure 4.3. These screens together, compose the UI of the emergency central subsystem. Important to refer that the hierarchy is the same in administrative an non-administrative centrals, changing only the screen logic and certain layout widgets.

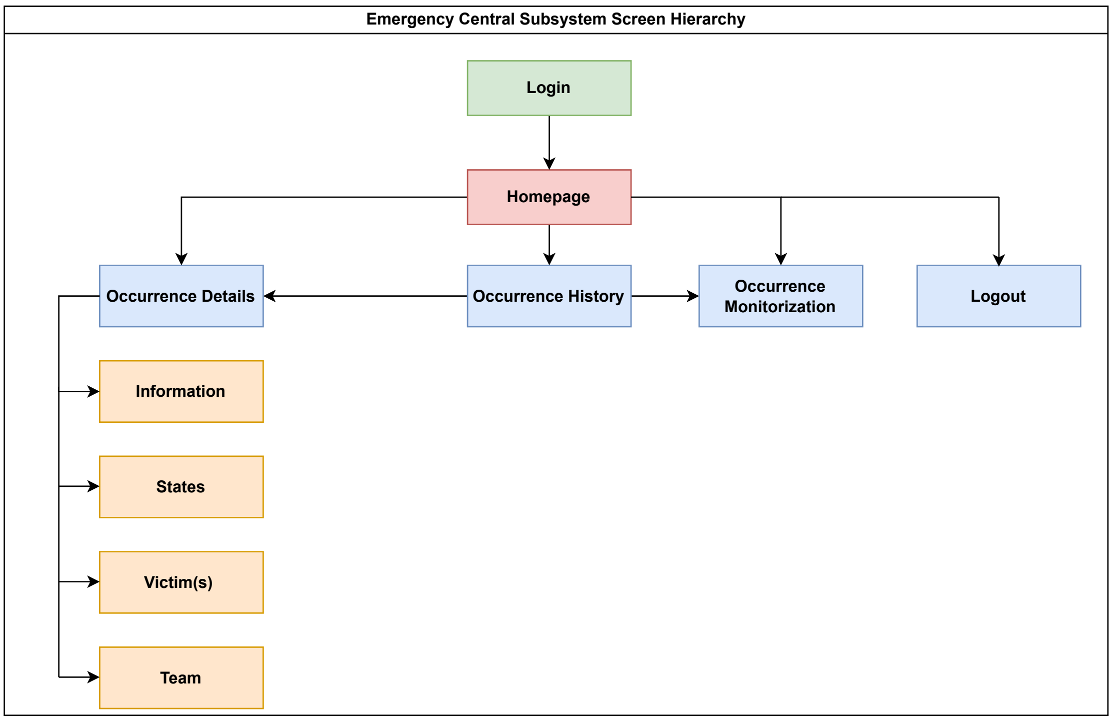


Figure 4.3: Emergency Central Subsystem Screen Hierarchy.

Initially the design process contemplated a screen dedicated to occurrence statistics, that was later deprecated due to the fact that the information generated by the proposed system was not from real occurrences and any attempt at building visualizations from them, was considered not applicable to the accomplishment of the system's objectives. The mockups created for the emergency central subsystem user interface can be found in appendix B.

The function of each screen is as follows:

A. *Login*

Page responsible for the user login. Only users that are registered by the SIREPH system administrator are able to login.

B. *Homepage*

The Homepage is where it can be found the active occurrences table. It is possible to search through the table, if there is need to find a specific occurrence and tooltips throughout the page were designed to help the user performing the required actions. For an administrative central user, the entire list of active occurrences present in the SIREPH system is shown, but for a non-administrative central user only active occurrences assigned to its station appear. From the homepage is possible to navigate to 4 pages: the Logout page, the occurrence monitorization page, the occurrence history page and occurrence details page.

C. *Occurrence Details*

This page pertains to a specific Occurrence. It is possible to view and update information regarding certain aspects of an occurrence. This page has the same interface in both types of emergency centrals.

D. *Occurrence History*

In this page it is possible to see the completed occurrences (not active). It follows the same principle as the homepage, users from administrative centrals can see all completed occurrences and users from non-administrative centrals can only see completed occurrences from their station. From this page it is possible to navigate to the occurrence details and monitorization page.

E. *Occurrence Monitorization*

This page, accessible to both types of emergency central users, is responsible for aiding the emergency dispatchers monitoring the occurrences states timelapse and current geographical location of the field team.

F. *Logout*

While a part of the homepage, this button is responsible for logging the user out, and as such is considered to be of relevant mention.

4.3.3.2 Hospital Subsystem

The emergency subsystem screen hierarchy is as depicted in figure 4.4. These screens together, compose the UI of the hospital subsystem.

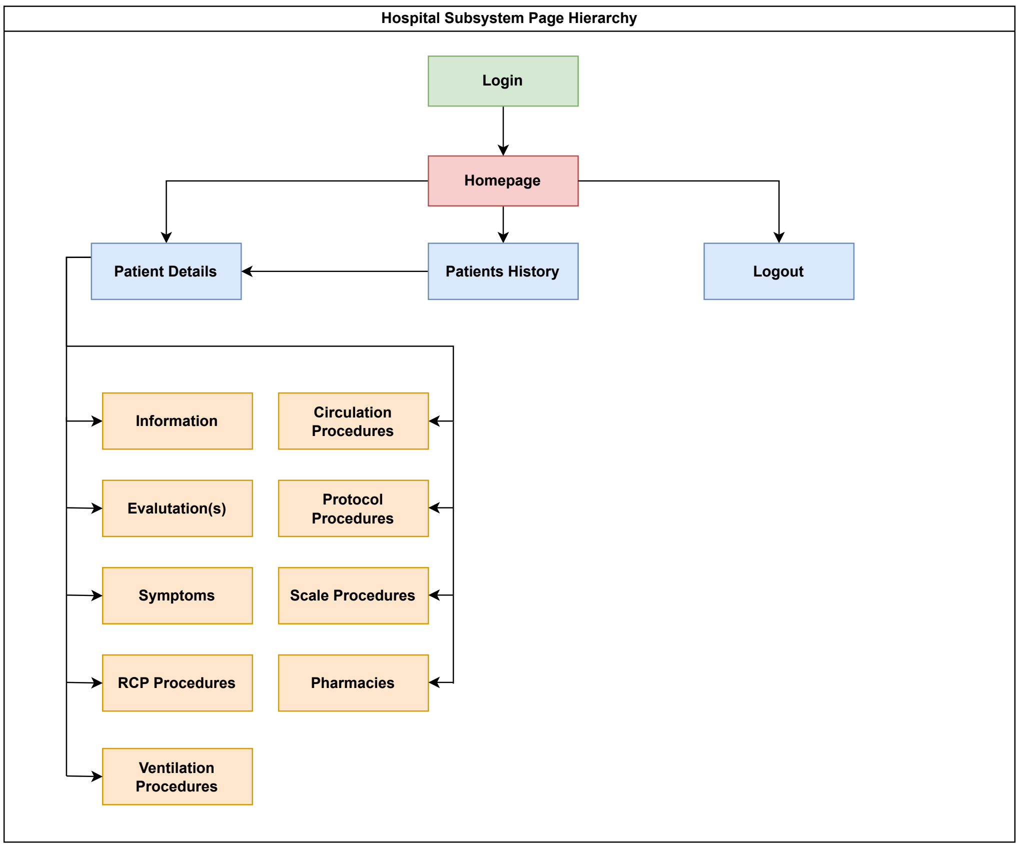


Figure 4.4: Hospital Subsystem Screen Hierarchy.

As happened with the previous subsystem, a screen dedicated to patient statistics was contemplated, that was later deprecated due to the same fact that the information generated by the proposed system was not from real patients and any attempt at building visualizations from them, was considered not applicable to the accomplishment of the system's objectives. The mockups created for the hospital subsystem user interface can be found in appendix C.

The function of each screen is as follows:

A. *Login*

   Page responsible for the user login. Only users that are registered by the SIREPH system administrator should be able to login.

B. *Homepage*

   The Homepage is where it could be found the patients that hadn't been processed in the user's hospital. Possible to search through the table, if there was need to find a specific patient and tooltips throughout the page were designed to help the user performing the required actions. From the homepage is possible to navigate to 3 pages: the Logout page, the patient history page and patient details page.

C. *Patient Details*

   This page pertained to a specific patient. Possible to view information regarding certain aspects of a patient that was involved in a pre-hospital emergency occurrence.

D. *Patient History*

   This page was destined to the viewing of the list of patients that were attended by the user's hospital.

E. *Logout*

   While a part of the homepage, this button is responsible for logging the user out, and as such is considered to be of relevant mention.

Chapter 5

Implementation

This chapter will cover the study of the implementation techniques and efforts of the project in study.

5.1 API Implementation

The API implementation started with the integration of Django REST framework to the Django project created. After the data model was designed it started the development of the Django models that would implement the designed diagram. Although not direcly related to the API, it was the first step taken in its creation. A partial Victim Model is highlighted in snippet 5.1, as the full Model was too sizeable to be represented in this document (as it is possible to verify through the Victim entity of the data model).

```
class Victim(models.Model):
    name = models.CharField(max_length=50, null=True, blank=True)
    birthdate = models.DateField(null=True, blank=True)
    age = models.PositiveSmallIntegerField(null=True, blank=True)
    gender = models.CharField(max_length=25, null=True, blank=True)
    identity_number = models.CharField(max_length=30, null=True, blank=True)
                            ...
    occurrence = models.ForeignKey(
        Occurrence,
        on_delete=models.RESTRICT,
        related_name='victims',
        null=True,
        blank=True)

        hospital = models.ForeignKey(
        Hospital,
        on_delete=models.RESTRICT,
        related_name='victim_hospital',
        null=True,
        blank=True )
```

Source Code 5.1: Victim Model code snippet

After the completion of the Django Models, it started the implementation of the serializers. The technical viewpoint of serializers was already given in Chapter 3 so a simple project example is given in snippet 5.2.

```python
class VictimDetailsSerializer(serializers.ModelSerializer):
    type_of_transport = serializers.ReadOnlyField(source='type_of_transport')
    non_transport_reason = serializers.ReadOnlyField(source='non_transport_reason')
    occurrence = OccurrenceSimplifiedSerializer(read_only=True)
    evaluations = EvaluationSerializer(read_only=True, many=True)
    pharmacies = PharmacySerializer(many=True, read_only=True)
    procedure_rcp = ProcedureRCPSerializer(read_only=True)
    procedure_ventilation = ProcedureVentilationSerializer(read_only=True)
    procedure_protocol = ProcedureProtocolSerializer(read_only=True)
    procedure_circulation = ProcedureCirculationSerializer(read_only=True)
    procedure_scale = ProcedureScaleSerializer(read_only=True)
    symptom = SymptomDetailsSerializer(read_only=True)

    class Meta:
        model = Victim
        fields = ['id', 'name', 'birthdate', 'age', 'gender', 'identity_number',
                  'address', 'circumstances', 'disease_history', 'allergies',
                  'last_meal', 'last_meal_time', 'usual_medication',
                  'risk_situation','medical_followup', 'hospital_checkin_date',
                  'episode_number','comments','type_of_emergency',
                  'type_of_transport','non_transport_reason', 'occurrence',
                  'evaluations', 'symptom','procedure_rcp','procedure_ventilation',
                  'procedure_protocol','procedure_circulation', 'procedure_scale',
                  'pharmacies', 'hospital']
```

Source Code 5.2: Victim Detail Serializer code snippet

After the completion of the Django serializers, it started the implementation of the API views. The technical viewpoint of views was also already given in Chapter 3 so a simple project example from a PUT method is given in snippet 5.3. It is possible to see from the snippet, that the authentication classes require token authentication for access to this view (and this happens with all created views). In this type of authentication, the token key should be inserted in the Authorization HTTP header, and its syntax be "Token "+ Token. Through the Django Admin it is possible to generate a user token, but during implementation it was developed a way to also be possible to generate or return the user token through the login method, depending if the user has already signed in or not, using the Django's class Token from **authtoken/models.py** and **Token.objects.get_or_create** method. The permission classes, coupled with the authentication define whether a request should be granted or denied access. In this case, to grant access to this API view the user must be authenticated, otherwise access is denied. More information on Django's REST framework Authentication and Permissions can be found at [85] and [86] respectively.

```
class VictimDetails(APIView):
authentication_classes = [TokenAuthentication]
    permission_classes = [IsAuthenticated]
                ....
    def put(self, request, victim_id):
        victim = get_object_or_404(Victim, pk=victim_id)
        data = request.data.copy()
        serializer = VictimDetailsSerializer(victim, data=data)

        if serializer.is_valid():
            serializer.save()
            return Response(data=serializer.data,
                            status=status.HTTP_201_CREATED)

        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Source Code 5.3: Victim Detail View code snippet

5.1.1 API Endpoints

An API endpoint is a digital address made available by the API through which requests are made and responses are returned. A URL identifies each endpoint as the place where a resource is located on the API structure [87].

During the API implementation a set of endpoints were created. This project didn't use the entirety of endpoints that were implemented, as some weren't applicable to the application meeting the established requirements. Others were implemented, but eventually neglected due to Outsystems' filtering capabilities, as application performance would improve due to fewer API calls. The complete list of the used endpoints and their associated API HTTP methods were:

A. login

   POST */login/*

   Login method, responsible for the creation of the user authentication token, that will uniquely identify the user logging in. Sends 3 booleans in the Response (is_technician, is_dispatcher, is_hospital_staff) that identify the logged user role in the system. If superuser all 3 booleans are set to True.

B. logout

   GET */logout/* - Logs out the current user.

C. centrals

   GET */centrals/* - Returns the list of emergency centrals.

   GET */centrals/{central_id}/occurrences/* - Returns the list of occurrences assigned to a specific emergency central.

D. hospitals

   GET */centrals/* - Returns the list of hospitals.

E. occurrence

   POST */occurrence/* - Creates an occurrence object.

F. occurrences

   GET */occurrences/* - Returns the occurrences list.

   GET */occurrences/{occurrence_id}/* - Returns a specific occurrence object.

   GET */occurrences/{occurrence_id}/states/* - Returns the list of states of a specific occurrence object.

   GET */occurrences/{occurrence_id}/victims/* - Returns the list of victims of a specific occurrence object..

G. teams

   GET */team/* - Returns the teams list.

H. user

   GET */user/{user_id}/central/* - Returns the assigned emergency central of a specific user object.

   GET */user/{user_id}/hospital/* - Returns the assigned hospital of a specific user object.

I. victim

   POST */victim/* - Creates a victim object.

J. victims

   GET */victims/* - Returns the victims list.

   GET */victims/{victim_id}/* - Returns a specific victim object.

   PUT */victims/{victim_id}/* - Updates a specific victim object.

Much more endpoints were created but not used. Some proved to be inapplicable to the project's requirements and others were disregarded over filters performed in Outsystems. For example, as a victim object has multiple objects associated to it (evaluations, procedures, symptoms, pharmacies) and in turn these objects have other associated objects, it proved unnecessary the use of endpoints such as:

GET */victims/{victim_id}/evaluations/* - Returns the evaluations of a specific victim.

GET */victims/{victim_id}/evaluations{evaluation_id}* - Returns a specific evaluation of a specific victim.

GET */victims/{victim_id}/pharmacies/* - Returns the pharmacies of a specific victim.

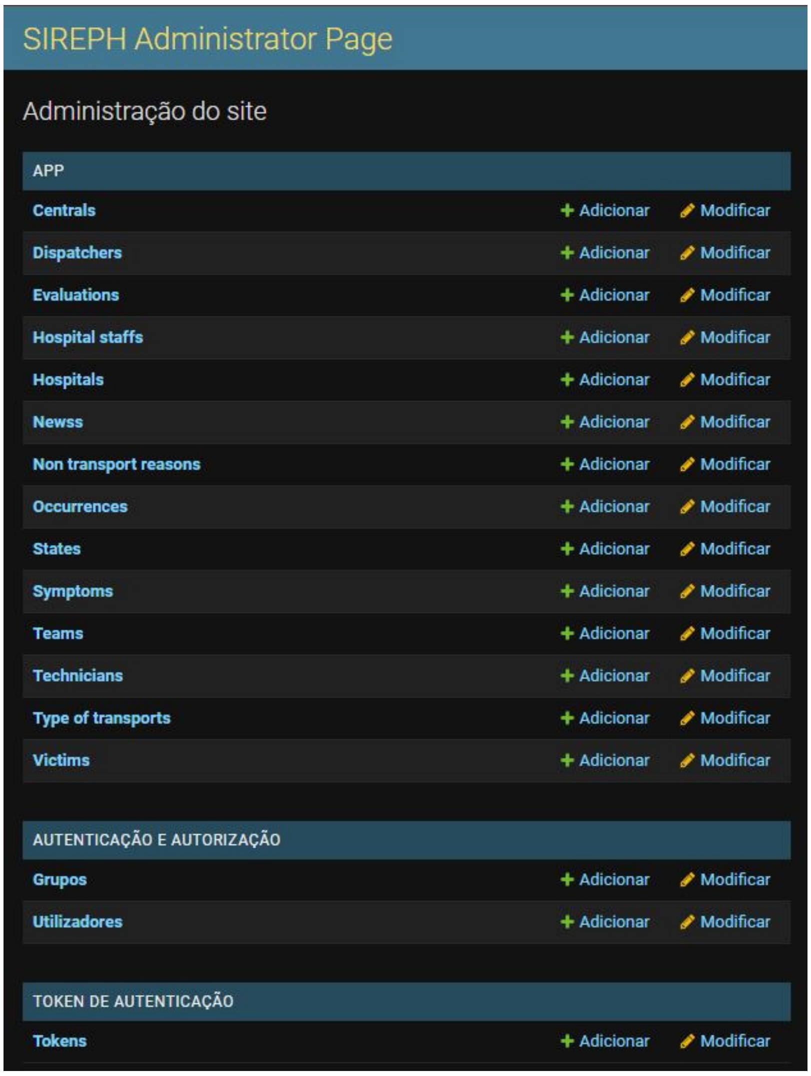
GET */victims/{victim_id}/pharmacies{pharmacy_id}* - Returns a specific pharmacy of a specific victim.

5.2 SIREPH Administrator Page

The SIREPH administrator page was a crucial component during the project's implementation. With it, it was possible to easily create objects to populate the database. As the application studied in this document was responsible for receiving/allocating occurrences and victims, there was need to create these entities manually as there was no option to create them otherwise, without interference from the other applications within SIREPH. To this end, a Django administration page was created to aid in the implementation stage, by allowing CRUD operations on the Django models that would later would materialize in the development testing (testing performed during implementation) efforts.

It was also in this page, that users were registered and given a role. This decision was due to the fact that emergency centrals and hospitals are institutions that need to be accommodated before partnering with SIREPH. In this idea, any user would have to be registered by the SIREPH administrator as being part of said institution. User registration into the system must always be accompanied by the specification of the institution they belong to, and no individual users are able to be registered without this contact.

The SIREPH Administrator Page was conceived due to the capabilities that the Django Admin provided, as it was referenced in the Chapter 3. The figure 5.1 represents the SIREPH Administration Homepage. The App tab represents the entities created in the Django Models. The Authentication and Authorization tab is related to Django's UserAdmin and AuthUserAdmin classes from its authentication system that was imported from **django.contrib.auth.admin**. The authentication token tab is related to TokenAdmin class imported from **rest_framework.authtoken.admin** and represents all the registered user tokens (unique).


SIREPH Administrator Page

Administração do site

APP

Centrals	+ Adicionar	Modificar
Dispatchers	+ Adicionar	Modificar
Evaluations	+ Adicionar	Modificar
Hospital staffs	+ Adicionar	Modificar
Hospitals	+ Adicionar	Modificar
Newss	+ Adicionar	Modificar
Non transport reasons	+ Adicionar	Modificar
Occurrences	+ Adicionar	Modificar
States	+ Adicionar	Modificar
Symptoms	+ Adicionar	Modificar
Teams	+ Adicionar	Modificar
Technicians	+ Adicionar	Modificar
Type of transports	+ Adicionar	Modificar
Victims	+ Adicionar	Modificar

AUTENTICAÇÃO E AUTORIZAÇÃO

Grupos	+ Adicionar	Modificar
Utilizadores	+ Adicionar	Modificar

TOKEN DE AUTENTICAÇÃO

Tokens	+ Adicionar	Modificar

Figure 5.1: SIREPH Administrator Page

5.3 Application Implementation

Once the API implementation had reached a stage of near completion and the documentation of the methods already created had been done, the Outsystems platform was employed to consume the SIREPH REST API.

In Outsystems' integrations features, it was selected the option to consume multiple REST API methods. The URL containing the Swagger specification was inserted (*78.46.215.82:82/api/swagger/*) and the methods to be used were chosen.

After the selection, Outsystems was fully integrated with the chosen REST API methods and their request and response data structures. The data structures present in the REST methods were also generated and ready to be used locally. During the API documentation, it was specified in each API view class an Authorization HTTP header, meaning that every REST API method integrated in Outsystems would have to have an Authorization header value in the form of a string (*auth = openapi.Parameter('Authorization', openapi.IN_HEADER, type=openapi.TYPE_STRING*). This authorization header, would later materialize in the expression "Token "+ Token, with the Token being the user authentication token that would uniquely identify the logged user, avoiding Cross-site request forgery (csrf) attacks. This token, as mentioned earlier, is part of Django's authentication system.

5.3.1 Login

The login is one the most important components of the application, and as such, the login implementation efforts of both subsystems will be addressed simultaneously. As specified in the requirements, users had to be subjected to various types of login validations:

- Only registered users in the SIREPH system may have access to the application.
- Only registered users with the dispatcher role may access the application's emergency central subsystem.
- Registered users with the dispatcher role should only have access to their emergency central's occurrences (important to remind that dispatchers from administrative emergency centrals can see all the system's occurrences).
- Only registered users with the hospital staff role may access the application's hospital subsystem.
- Registered users with the hospital staff role should only have access to their hospital's patients.

5.3.1.1 Login_IS server action

The first thing to implement these requirements was creating the server action 'Login_IS' that would receive the login API method request data (in this case the user's username and password) and assign its values to the server action's input parameters (Username and Password). Afterwards, the API method would have in its response body a user token that would be assigned to an output parameter from the server action called Token. With this token it would be possible to validate which role the user had in the system through its token value. This validation would be possible through 3 booleans in its response body (is_dispatcher, is_hospital_staff, is_technician). If the dispatcher role validated True, it would use the **GET** */user/{user_id}/central/* method to verify the user_id and with it, find the identifier of the central the user belonged to and the value of the is_administrative boolean from the central entity (if True the user belonged to an administrative central). These values would then be assigned to the server action's CentralId and IsAdmin output parameters.

If the hospitall staff role validated True, it would use the **GET** */user/{user_id}/hospital/* method to verify the user_id and with it find the identifier of the hospital the user belonged to. Its value would then be assigned to the server action HospitalId output parameter.

5.3.1.2 DoLogin server action

The second thing to implement, was the 'DoLogin' server action (action to authenticate the user in the Outsystems' side). This action had as input parameters: Username, Password and a boolean called isCentral. As output parameters had: Token, HospitalId, CentralId, IsAdmin, and IsSuccess (as True default value).

This server action would call the Login_IS server action (reusing all its implemented logic) and verify if the token received was different from null. If it was it would verify the following:

(isCentral and Login_IS.CentralId <> NullIdentifier()) or (not isCentral and Login_IS.HospitalId <> NullIdentifier())

If False, the isSuccess boolean would assign False and end the flow branch. If True, it would search Outsystems users by their username. If the username didn't exist it would create the user and then perform the login through the Outsystems in-built User_Login server action. If the username existed, the User_Login server action would simply be called. In the end, it would assign the Token, HospitalId, CentralId and IsAdmin values coming from the Login_IS server action to the Token, HospitalId, CentralId and IsAdmin output parameters of the DoLogin server action.

5.3.1.3 Client Variables

Client variables were then created to store client-side data to be persisted throughout the application use. So it was created 4 client variables: Token (to persist the token value in the client-side), HospitalId (to persist the user's hospitalId in the client-side if applicable), CentralId (to persist the user's CentralId in the client-side if applicable) and isAdmin (to persist if the user's central is administrative in the client-side).

5.3.1.4 Logging Client Action

In Outsystems, server actions can't be called from the Interface tab while creating screen logic. To bypass this issue, a client action is created to call the server action logic to be used in the screen.

The client action 'Logging' with Username, Password and isCentral as input parameters and isSuccess as an output parameter calls the 'DoLogin' server action (reusing all its implemented logic) and assigns the 'DoLogin' output parameters values to the client variables created:

```
Client.Token = DoLogin.Token ; Client.HospitalId = DoLogin.HospitalId ;
Client.CentralId = DoLogin.CentralId ; Client.IsAdmin = DoLogin.IsAdmin
```

And the isSuccess output parameter is assigned the value of the isSuccess output parameter coming from the 'DoLogin' server action.

5.3.1.5 Login Screen Action

Finally, in the Login screen the 'Logging' client action is called into a screen action called 'Login' and a if validation is performed to check if the 'isSuccess' parameter is True. If it is, the user logs in. If it isn't, an exception is raised signifying that the user doesn't have permission to access an hospital/emergency central. This happens in cases where the user tries to login in a subsystem with the credentials of a role with no permissions (i.e: a dispatcher tries to log in the hospital subsystem and vice-versa).

5.3.2 Emergency Central Subsystem

This section will address the implementation efforts regarding the emergency central subsystem, minus the login aspects, that were covered in the earlier section. The page structure followed was the one mentioned in the 4 during the design stage. However, as the requirements were refined, the designed mockups underwent changes to better mirror the final requirements specified. The emergency central subsystem definitive user interface pages can be found in appendix D.

5.3.2.1 Emergency Central Login

This page is dedicated to the login efforts of the emergency central subsystem users, namely the dispatchers.

5.3.2.2 Homepage

The emergency subsystem homepage can be divided into 2 sections:

- The first section is composed by a greeting to the logged user 'Hello, username'. Below it, 4 cards with the number of active occurrences currently in the system (administrative central user) or assigned to the user's non-administrative central, the user's central phone contact, the user's central area of action and the user's central designation can be found.

- The second section is dedicated to the active occurrences table. This table returns the response of the **GET** */occurrences/* method, filtered through a Data Action (action created when data from other sources is fetched after a Screen loads) where only occurrences that have the active attribute set as True are shown. The information contained in the table headers regarding the active occurrences is: occurrence number, creation date and time, local, parish, municipality, motive, number of victims, details and monitorization. The table is sorted to show the most recent active occurrences. Above the table there's a search bar that allows the search for a specific occurrence through its occurrence number value (unique to each occurrence). The details header row contain an icon with a link that navigates the user to the details of the selected occurrence record through its identifier. The monitorization header row contain an icon with a link that navigates the user to the monitorization page of the selected occurrence record through its identifier.

5.3.2.3 Occurrence Details Pages

Although the occurrence details pages were created independently from each other, their logic is similar and can be navigated freely from one into another. To that reason, they will be described together. The occurrence details are divided into 4 pages:

- The first page is a form related to the Occurrence information. It returns the occurrence attributes from the **GET** */occurrences/{occurrence_id}/* to be filled by a dispatcher user. The only practical difference between a non-administrative central user and a administrative central user, is that the latter can only assign occurrences to emergency centrals while the former can only assign an occurrence to a team of technicians. The assignment is made through a dropdown widget that displays a list, where in the case of an administrative central the available emergency centrals list are displayed through the **GET** */centrals/* method and in the case of a non-administrative central the available teams from the same emergency are shown through **GET** */team/* method and then filtered to only show the active teams of the assigned emergency central.

- The second page has a form pertaining to the information of 1 or more occurrence victims. This information is returned through the same method as the previous section but without any filtering.

- The third page has a table highlighting the list of states of the occurrence. Returns information from the **GET** */occurrences/{occurrence_id}/states/* method without any filtering.

- The fourth page has a table highlighting the team assigned to the occurrence. Returns the same method as in the first section with the same filtering that was applied in the teams list dropdown.

5.3.2.4 Monitorization Page

This page was implemented so that emergency central dispatchers could monitor occurrences both geographically and temporally, in order to aid the on-site technicians in successfully executing the occurrence.

This page is divided into 4 sections:

- The first section has a geographical monitoring board. The map was constructed using an API key from Google's Geocoding API. The map also possesses a marker that highlights the current occurrence state location (using its latitude and longitude attributes to ascertain the coordinates) in a Google maps interface.

- The second section has occurrence information, to aid dispatchers in easily identify the occurrence being monitored beyond its occurrence number.

- The third section contains a form wizard that represents the occurrence states. It has 5 possible states (specified previously in Chapter 4). When the technician's application triggers an occurrence the wizard receives this information and displays it in the wizard. Each state trigger activates its correspondent wizard item. As the occurrence victim may not be transported to the hospital, before reaching the state *'A caminho do hospital'* the wizard only shows 3 possible states (*'A caminho do local'*, *'Chegada à vítima'*, *'Fim da Ocorrência'*). If the fourth state is triggered then the wizard starts displaying the 5 possible states, for if the victim is inbound to the hospital, he will also have to arrive.

- The fourth section contains a notification table. This table possesses information regarding the occurrence states, the date and the time at the state start, its duration and coordinates. The duration is calculated by calculating the time difference between the beginning of the previous state and the moment when the current state started. In the duration row of the last occurrence state *'Fim da Ocorrência'*, is always stated *'Finalizado'*. If a state is currently underway, the duration row for that state highlights *'Em curso'*.

5.3.2.5 Occurrence History Page

The occurrence history page is fairly similar in its design to the homepage. The main difference arises from the fact that the occurrence history page only highlights completed occurrences and therefore inactive. So the table records are filtered to only highlight occurrences with the active attribute set as False. Like the homepage, it also allows searching through an occurrence number to find it in the table.

5.3.3 Hospital Subsystem

This section will address the implementation efforts regarding the hospital subsystem, minus the login aspects, that were covered earlier. The page structure followed was the one mentioned in the 4 during the design stage. The hospital subsystem definitive user interface pages can be found in appendix E.

5.3.3.1 Hospital Login Page

This page is dedicated to the login efforts of the hospital subsystem users, namely the hospital staff.

5.3.3.2 Homepage

The hospital subsystem homepage can be divided into 2 sections:

- The first section is composed by a greeting to the logged user 'Hello, username'. Below it, 4 cards with the the number of patients who have not yet been checked into the hospital (still unassisted), the phone contact of the user's hospital, the capacity of the user's hospital (total capacity - current capacity) and the name of the user's hospital.

- The second section holds a table dedicated to the patients that haven't been checked into the user's hospital, that is, remain unassisted. This table returns the response of the **GET** */victims/* method, filtered through a Data Action where only patients that have the 'hospital_checkin_date attribute set as null are shown. The information contained in the table headers regarding the patients is: name, identity number, age, gender, circunstances, address and details . The table is sorted to show the most recent patients that arrived in the hospital. Above the table there's a search bar that allows the search for a specific patient through its name value. The details header row contain an icon with a link that navigates the user to the details of the selected patient record through its identifier.

5.3.3.3 Patient Details Pages

The patient details pages will be described along the same lines as with the occurrence details pages. The patient details are divided into 9 pages.

- The first page displays patient information. It is divided into 3 sections: identification (personal patient information like name, age, address, etc), clinical history (patient's medical information like allergies, disease history, usual medication, etc) and transport/other observations (patient transport information and observations like non transport reason, type of transport, observations and medical followup. It is also in this section that the patients are formally checked into the hospital by the user, with the date and time that they started being assisted.

- The second page displays patient evaluation information. As a patient can have more than 1 evaluation, each evaluation is separated by accordion widgets. Inside each accordion is information regarding a specific evaluation like date and time, temperature, glycemia, systolic blood pressure, etc. It also has a section dedicated to Glasgow Scale information, a specific evaluation that can be performed to assess the level of conscience of a trauma victim.

- The third page displays symptom information. Throughout this page, tooltips were implemented to provide context information to the user in understanding the information values. Information regarding noticeable symptoms and % burned area (in burning victims) can be found. This page, has another section, pertaining to a specific group of symptoms, denominated Trauma. This section possesses 2 anatomical drawings for trauma documentation adapted from the Istanbul Protocol [88] to aid the hospital user's in identifying the body part location that was subjected to trauma. It also has information regarding the type of trauma injury, if the injury is closed or not, and the burn degree (in burning victim).

- The fourth page displays information regarding cardiopulmonary resuscitation procedures performed on the patient. It has information regarding date and time of basic and immediate life support, if mechanical compressions were performed and other attributes from the procedure RCP entity.

- The fifth page displays information regarding ventilation procedures. The information is presented in a boolean format. It is contemplated procedures such as mechanical ventilation, endotracheal tube, laryngeal tube/mask and other attributes from the procedure ventilation entity.

- The sixth page displays information regarding circulation procedures. It is also represented through boolean formats. Procedures such as temperature monitoring, compression, pelvic belt, venous access and tourniquet are displayed.
- The seventh page displays information regarding protocol procedures. It is also represented through boolean formats. Procedures such as immobilization, *Via Verde Coronária*, *Via Verde Trauma*, *Via Verde Sépsis* [89] and others.
- The eighth page displays information regarding scale procedures. It is entirely represented through numerical values. Tooltips were implemented throughout this page, giving context information on the scales, to aid the user in interpreting the displayed values. Procedures such as Cincinatti, Revised Trauma Score (RTS), Rapid Arterial Occlusion Evaluation (RACE) and others.
- The ninth page displays information regarding the pharmaceuticals administered to the patient on a table. The patient may be administered more than 1 pharmaceutical. The information contemplated concerns the date and time of pharmaceutical administration, its name, its dose, route and adverse effects.

5.3.3.4 Patient History Page

The patient history page is fairly similar in its design to the homepage. Patient records that were already checked into the hospital are moved to the patient history page, where they can be consulted but no longer appear on the homepage.

Chapter 6

Validation

This chapter will detail the project's validation efforts. After the implementation was completed, it was necessary to ascertain if the developed application served the purpose for which it was created. To that end, test protocols and forms were formulated to understand if the project reached its proposed objectives. Once the number of test results was deemed acceptable, they underwent statistical analysis to verify the results obtained.

6.1 User Validation

User validation is a crucial stage in project development, responsible for ensuring delivery quality, by discovering errors in the software and reducing the risk of a user encountering liabilities when using it [90]. However, it is a considerable endeavor, in terms of resources, as imitating all the possible scenarios and configurations that may arise from the use of the software require complex test protocols and infrastructure [91].

The user validation comprised of the following steps:

A. *Test Protocols*

   3 documents were formulated for each of the identified type of user that would use the application. In it, a brief description of the application and their main actors was given, in order for the users to better understand what was implemented. Then, according to the type of user the document was for, test protocols were formulated and access credentials to the application given. For the test protocols, certain steps were timed to understand if the user interface was seamless to use.

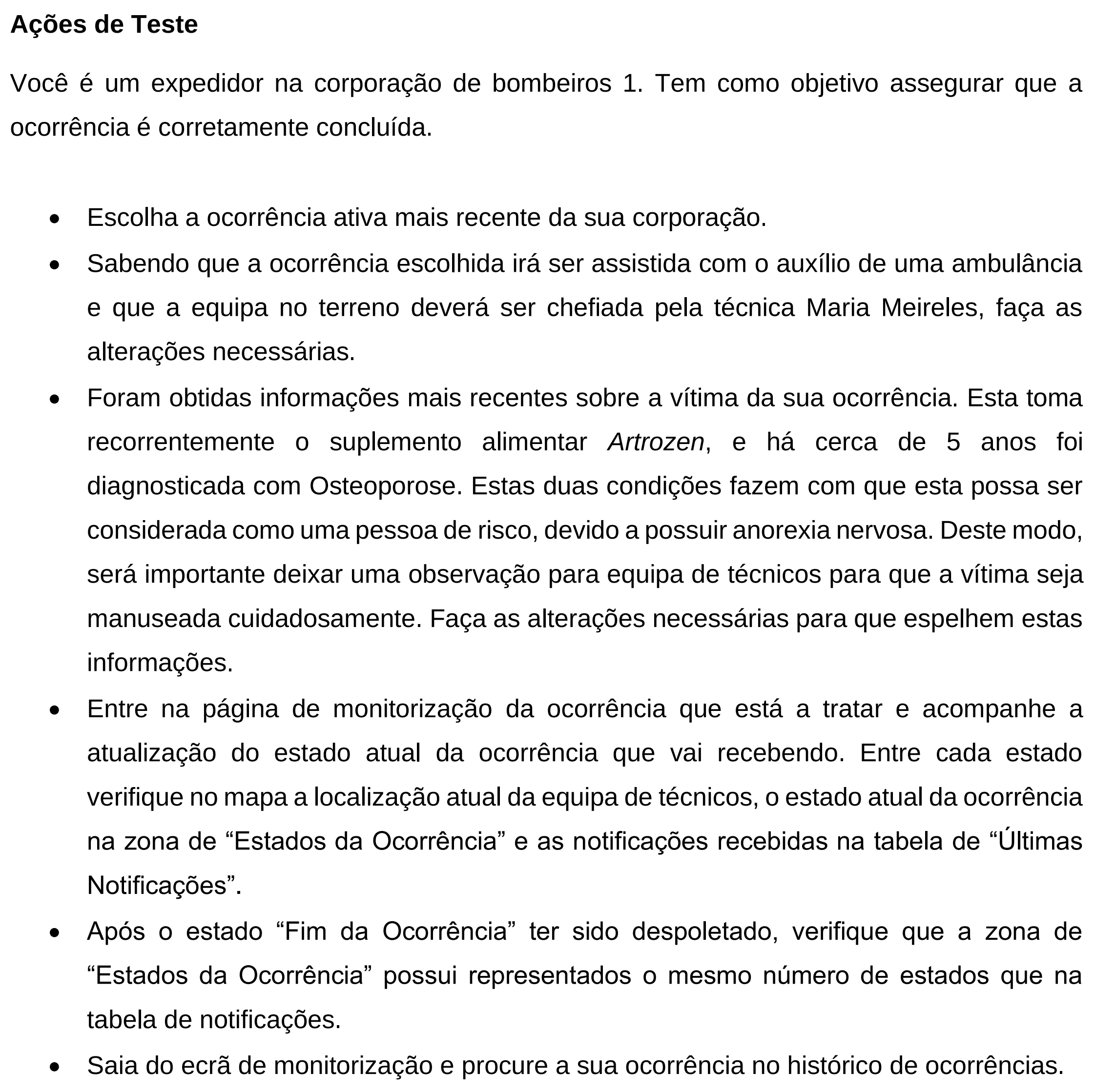
B. *Questionnaires*

   After the testing protocols were completed, the users would fill a form, that aimed to detail their experience using the application. The forms were developed using Google Forms and an access link was granted inside the test protocols document.

After these steps, the results gathered were to be analyzed and later presented, to validate if the project achieved its proposed goals.

6.1.1 Testing Protocols

The testing protocols formulated aim to present plausible scenarios for each type of user. These scenarios are time sensitive, and so we recorded the time spent by the user on each task, and the overall time for the scenario completion. Figure 6.1 is an example of the test tasks presented to the users of non-administrative emergency centrals. The remaining test protocols are present in appendix I.



Ações de Teste

Você é um expedidor na corporação de bombeiros 1. Tem como objetivo assegurar que a ocorrência é corretamente concluída.

- Escolha a ocorrência ativa mais recente da sua corporação.
- Sabendo que a ocorrência escolhida irá ser assistida com o auxílio de uma ambulância e que a equipa no terreno deverá ser chefiada pela técnica Maria Meireles, faça as alterações necessárias.
- Foram obtidas informações mais recentes sobre a vítima da sua ocorrência. Esta toma recorrentemente o suplemento alimentar *Artrozen*, e há cerca de 5 anos foi diagnosticada com Osteoporose. Estas duas condições fazem com que esta possa ser considerada como uma pessoa de risco, devido a possuir anorexia nervosa. Deste modo, será importante deixar uma observação para equipa de técnicos para que a vítima seja manuseada cuidadosamente. Faça as alterações necessárias para que espelhem estas informações.
- Entre na página de monitorização da ocorrência que está a tratar e acompanhe a atualização do estado atual da ocorrência que vai recebendo. Entre cada estado verifique no mapa a localização atual da equipa de técnicos, o estado atual da ocorrência na zona de “Estados da Ocorrência” e as notificações recebidas na tabela de “Últimas Notificações”.
- Após o estado “Fim da Ocorrência” ter sido despoletado, verifique que a zona de “Estados da Ocorrência” possui representados o mesmo número de estados que na tabela de notificações.
- Saia do ecrã de monitorização e procure a sua ocorrência no histórico de ocorrências.

Figure 6.1: Example of non-administrative central user test tasks.

6.1.2 Questionnaires

The questionnaires were created to validate the user experience when using the application. They were divided into sections, each related to a specific topic to be evaluated. Common to all the 3 forms are the user personal information section, represented in table 6.1 and the opinions section, represented in table 6.2. The remaining sections are form-specific, therefore they were addressed separately.

Table 6.1: Personal information section across all forms

Question	Answer Format
Email	Short-answer text
Age	Multiple choice (5 choices, with ages ranging from 18 to 51+, ++7 years per choice)
Gender	Drop-down (3 choices - Masculine, Feminine, Another)
Academic Background	Drop-down (6 choices - Basic or Secondary Education, Without a Bachelor's Degree, Bachelor's Degree, Master's Degree, PhD
Occupation	Short-answer text
Years in the Occupation	Multiple choice (4 choices, ranging from >3 years to <11 years, ++3 years per choice)

Table 6.2: Opinions section across all forms

Question	Answer Format
Does the application seem useful in a real-life context?	Multiple choice (5 choices - Completely Disagree, Disagree, Neutral, Agree, Completely Agree)
Do you consider that the realization of the scenario was?	Multiple choice (5 choices - Very Hard, Hard, Neutral, Easy, Very Easy)
Do you have any comments to add about the application (e.g.: improvements)?	Long-answer text

6.1.2.1 Emergency Central Subsystem Forms

These 2 forms are specific to users who participated in the test protocols for the emergency central subsystem. They are almost identical except for 2 questions that are highlighted as footnotes. Every section will be highlighted (minus the 2 sections already covered in the last section). Although not referenced in the tables, between every question, the users have the option to make additional comments about the last answered question in a long-answer text. Also important to point out, that as the answer format is the same to every question in these form-specific sections (5 choices - Completely Disagree, Disagree, Neutral, Agree, Completely Agree) it will be omitted from the tables. The tables 6.3 to 6.6 are the questions found in the remaining sections of the emergency subsystem forms.

Table 6.3: Emergency Central Subsystem Forms - Home page section

Question
It is easy to understand which occurrences do not yet have a central assigned [1]
It is easy to understand which occurrences do not yet have a team assigned [2]
The search bar is useful
The table of active occurrences has the appropriate information
The graphical interface is aesthetically appealing
It is simple to find the button for the details of an active occurrence
It is easy to find the monitoring page

Table 6.4: Emergency Central Subsystem Forms - Details page section

Question
The graphical interface of the detail pages is aesthetically appealing
It is easy to read and change the information received
It is easy to assign an occurrence to an emergency central [1]
It is easy to assign an occurrence to a team of technicians [2]
Navigation through the detail pages is intuitive
The information presented is intuitive

Table 6.5: Emergency Central Subsystem Forms - Monitorization page section

Question
The graphical interface of the monitorization page is aesthetically appealing
The map present in the geographical monitoring panel is useful
The marker present on the map, regarding the location of the current state of occurrence is useful
The information about the occurrence being monitored is adequate
The schematic regarding the occurrence states is useful and appropriate
The notification table is useful and appropriate
I consider the monitoring page an asset to ensure that the execution of the occurrence proceeds smoothly [2]

Table 6.6: Emergency Central Subsystem Forms - History page section

Question
It is easy to find the occurrence history page
The information presented is adequate
The search bar is useful

[1]Only related to the administrative central form
[2]Only related to the non-administrative central form

In appendix F it is possible to view the non-administrative central form and in appendix G the administrative central form template.

6.1.2.2 Hospital Form

This form is specific to users who participated in the test protocols for the hospital subsystem. Every form section will be highlighted (minus the 2 sections already covered that are common to all forms). The same structure is employed in this form as it was in the other 2, and the same answer format was used (5 choices - Completely Disagree, Disagree, Neutral, Agree, Completely Agree) so it will be omitted from the tables. The tables 6.7 to 6.9 are the questions found in the remaining sections of the hospital form.

Table 6.7: Hospital Subsystem- Home page section

Question
It is easy to understand which victims have not yet been checked-in by the hospital
The search bar is useful
The hospital's patient table has the appropriate information
The graphical interface is aesthetically appealing
It is easy to find the button for your hospital's patient details

Table 6.8: Hospital Subsystem- Details page section

Question
The graphical interface of the detail pages is aesthetically appealing
The information received about the patient is adequate and well-structured
It is easy to assign the date and time that the patient was checked into the hospital
The information received on a patient's evaluation page is understandable and well-structured
The information received on a patient's evaluation page is understandable
The human image present in the symptoms page is useful for understanding a patient's traumas
Information regarding the procedures applied on the victim is easy to understand
The tooltips present in the scale procedures are useful and sufficient for understanding the displayed values
The information in the table of administered pharmaceuticals is adequate and easy to read

Table 6.9: Hospital Subsystem- Patient history page section

Question
It is easy to find the patient history page
The information presented is adequate
The search bar is useful

In appendix H it is possible to view the hospital form template.

6.2 Results

The results obtained were based on the outlined user validation tasks, both from the collection and analysis of the user's form answers and the experience accumulated during the participation in all of the user's test protocols.

6.2.1 Test Sample Size and Constraints

In total, 25 users were involved in the validation process. As every user performed the 3 testing protocols and submitted the 3 forms, the total number of tests performed amounted to 75 (25 for the administrative central test protocol, 25 for the non-administrative test protocol and 25 for the hospital test protocol).

This application is not intended for general public use. Due to its specificity, attempts were made to integrate in the tests, users who complied with at least one of the characteristics outlined in end user characterization performed in Chapter 4. This attempt proved challenging, as all tested users performed the 3 protocols, and in some instances, one might be completely suitable for one subsystem and not for the other (i.e: a nurse performing the test protocols could perfectly be an end user for the application's hospital subsystem but not for the application's emergency central subsystem. The same could be applied vice-versa in the case of a firefighter).

Important to also mention that the author was present in the execution of all test protocols, to correctly measure each step completion time and the total test time. No user was excluded from the initial sample, since all form answers were correctly submitted.

6.2.2 Test Sample Characterization

Since the test sample was the same in all test protocols, the personal information section of all forms yielded the same results and thus made it easier to characterize it.

The interval frequency distribution of the test users age were:

- 12 users were aged between 26 and 33 years, 48% of the total.
- 6 users were aged between 18 and 25 years, 24% of the total.
- 4 users were aged between 34 and 42 years, 16% of the total.
- 1 user was aged between 43 and 50 years, 4% of the total.
- 2 users were older than 51 years, 8% of the total.

As for the gender, the numerical and percentage proportion of the test sample was:

- 14 users identified as male, 56% of the total.
- 11 users identified as female, 44% of the total.
- 0 users identified as other, therefore not represented in the figure.

Regarding the highest academic background, the numerical and percentage proportion of the test sample was:

- 14 users have a bachelor's degree, 56% of the total.

- 7 users have a master’s degree, 28% of the total.
- 2 users have a doctoral degree, 8% of the total.
- 2 users have secondary education, 8% of the total.
- 0 users had basic education or without a bachelor’s degree, therefore not represented in the figure.

In terms of professions, due to the different spelling used in specifying the same profession (gender changes, lower or uppercase letters) and the high variance of professions, they were grouped into professional sectors.

- 11 users worked in the healthcare sector, 44% of the total.
- 7 users worked in the technological sector, 28% of the total.
- 5 users worked in the civil protection sector, 20% of the total.
- 2 users worked in the education sector, 8% of the total.

Years in Professional Activity is defined as the time the user has been employed in the profession it has submitted in the form. The interval frequency distribution of the users’ years of professional activity was:

- 9 users have been working in their profession between 4 and 7 years, 36% of the total.
- 8 users have been working in their profession between 0 and 3 years, 32% of the total.
- 6 users have been working in their profession for more than 11 years, 24% of the total.
- 2 users have been working in their profession between 8 and 11 years, 8% of the total.

6.2.3 Questionnaire Results

Through the Google Forms capabilities, all responses to the 3 questionnaires were collected to a separate Comma-Separated Values (CSV) file. The responses present in each file underwent statistical analysis to extract relevant and actionable information from them.

First, each questionnaire was handled independently, separating it into sections and analyzing the answers to the questions within each section individually.

Secondly, the responses to the questions were cross-referenced with the professional sector of the users who answered them, in order to identify the responses by professional sector. This analysis was conducted to validade whether there would be variations in satisfaction with the application in each sector. Since not all test users belonged to the intended professional sector of the application, it was important to validate if the application was comprehensible above all else to the target professional sectors and if its design was intuitive enough for people with no pre-emergency or medical background to be able to use it adequately.

After the frequency of response by professional sector to each of the options in the questions and their total frequency percentage was calculated, the results were gathered into 3 tables, each table for each questionnaire.

The tables with each questionnaire's results are as follows:

Figure 6.2 represents a table with the results of the Administrative Central questionnaire.

Figure 6.3 displays a table with the results of the Non-Administrative Central questionnaire.

Figure 6.4 highlights a table with the results of the Hospital questionnaire.

		Completely Disagree/Very Hard[1]					Disagree/Hard[1]					No opinion					Agree/Easy[1]					Completely Agree/Very Easy[1]					
		%					%					%					%					%					
		S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	Σ
Section 2	Q1: It is easy to understand which occurrences do not yet have a central assigned	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	8	4	4	**28**	32	20	4	16	**72**	100
	Q2: The search bar is useful	0	0	0	0	**0**	4	0	4	0	**8**	0	0	0	0	**0**	20	8	0	12	**40**	20	20	4	8	**52**	100
	Q3: The table of active occurrences has the appropriate information	0	0	0	0	**0**	0	0	0	0	**0**	8	0	0	0	**8**	20	8	8	12	**48**	16	20	0	8	**44**	100
	Q4: The graphical interface is aesthetically appealing	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	8	8	8	**36**	32	20	0	12	**64**	100
	Q5: It is simple to find the button for the details of an active occurrence	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	4	16	8	4	**32**	40	12	0	16	**68**	100
	Q6: It is easy to find the monitoring page	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	16	8	8	**48**	28	12	0	12	**52**	100
Section 3	Q1: The graphical interface of the details page is aesthetically appealing	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	12	4	8	**32**	36	16	4	12	**68**	100
	Q2: It is easy to read and change the information received	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	12	8	8	**44**	28	16	0	12	**56**	100
	Q3: It is easy to assign an occurrence to an emergency central	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	8	4	4	**24**	36	20	4	16	**76**	100
	Q4: Navigation through the detail pages is intuitive	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	24	8	8	12	**52**	20	20	0	8	**48**	100
	Q5: The information presented is adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	20	8	8	4	**40**	24	20	0	16	**60**	100
Section 4	Q1: The graphical interface of the monitorization page is aesthetically appealing	0	0	0	0	**0**	4	0	0	0	**4**	0	0	0	0	**0**	12	8	4	8	**32**	28	20	4	12	**64**	100
	Q2: The map present in the geographical monitoring panel is useful	0	0	0	0	**0**	0	0	4	0	**4**	0	0	0	0	**0**	12	8	0	8	**28**	32	20	4	12	**68**	100
	Q3: The marker present on the map, regarding the location of the current state of the occurrence is useful	0	0	0	0	**0**	0	0	0	0	**0**	0	0	4	0	**4**	12	8	0	8	**28**	32	20	4	12	**68**	100
	Q4: The information about the occurrence being monitored is adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	8	8	8	**40**	28	20	0	12	**60**	100
	Q5: The schematic regarding the occurrence states is useful and appropriate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	12	4	8	**40**	28	16	4	12	**60**	100
	Q6: The notification table is useful and adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	20	16	4	4	**44**	24	12	4	16	**56**	100
Section 5	Q1: It is easy to find the occurrence history page	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	20	8	4	8	**40**	24	20	4	12	**60**	100
	Q2: The information presented is adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	28	8	8	12	**56**	16	20	0	8	**44**	100
	Q3: The search bar is useful	0	0	0	0	**0**	0	0	4	0	**4**	0	0	0	0	**0**	28	8	0	16	**52**	16	20	4	4	**44**	100
Section 6	Q1: Does the application seem useful in a real-life context?	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	8	8	8	**40**	28	20	0	12	**60**	100
	Q2: Do you consider that the realization of the scenario was?	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	40	12	4	20	**76**	4	16	4	0	**24**	100

Professional Sectors

S1 - Healthcare **S2** - Technology **S3** - Education **S4** - Civil Protection

[1] Option only applicable in section 6.

Figure 6.2: Administrative Central Form Results

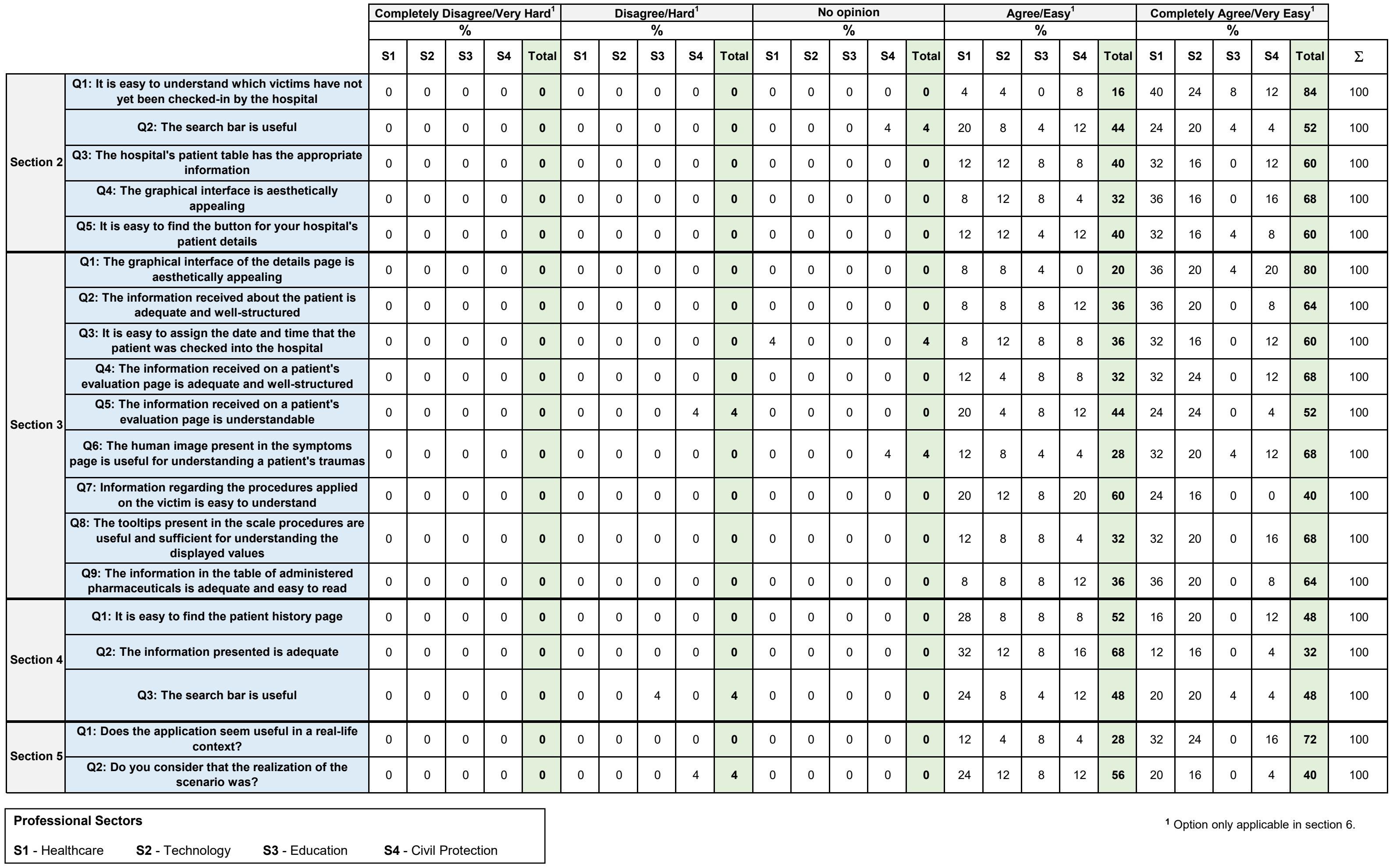
		Completely Disagree/Very Hard[1]					Disagree/Hard[1]					No opinion					Agree/Easy[1]					Completely Agree/Very Easy[1]					
		%					%					%					%					%					
		S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	Σ
Section 2	Q1: It is easy to understand which occurrences do not yet have a central assigned	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	8	4	4	**28**	32	20	4	16	**72**	100
	Q2: The search bar is useful	0	0	0	0	**0**	4	0	4	0	**8**	0	0	0	0	**0**	20	8	0	12	**40**	20	20	4	8	**52**	100
	Q3: The table of active occurrences has the appropriate information	0	0	0	0	**0**	0	0	0	0	**0**	8	0	0	0	**8**	20	8	8	12	**48**	16	20	0	8	**44**	100
	Q4: The graphical interface is aesthetically appealing	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	8	8	8	**36**	32	20	0	12	**64**	100
	Q5: It is simple to find the button for the details of an active occurrence	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	4	16	8	4	**32**	40	12	0	16	**68**	100
	Q6: It is easy to find the monitoring page	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	16	8	8	**48**	28	12	0	12	**52**	100
Section 3	Q1: The graphical interface of the details page is aesthetically appealing	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	12	4	8	**32**	36	16	4	12	**68**	100
	Q2: It is easy to read and change the information received	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	12	8	8	**44**	28	16	0	12	**56**	100
	Q3: It is easy to assign an occurrence to an emergency central	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	8	4	4	**24**	36	20	4	16	**76**	100
	Q4: Navigation through the detail pages is intuitive	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	24	8	8	12	**52**	20	20	0	8	**48**	100
	Q5: The information presented is adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	20	8	8	4	**40**	24	20	0	16	**60**	100
Section 4	Q1: The graphical interface of the monitorization page is aesthetically appealing	0	0	0	0	**0**	4	0	0	0	**4**	0	0	0	0	**0**	12	8	4	8	**32**	28	20	4	12	**64**	100
	Q2: The map present in the geographical monitoring panel is useful	0	0	0	0	**0**	0	0	4	0	**4**	0	0	0	0	**0**	12	8	0	8	**28**	32	20	4	12	**68**	100
	Q3: The marker present on the map, regarding the location of the current state of the occurrence is useful	0	0	0	0	**0**	0	0	0	0	**0**	0	0	4	0	**4**	12	8	0	8	**28**	32	20	4	12	**68**	100
	Q4: The information about the occurrence being monitored is adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	8	8	8	**40**	28	20	0	12	**60**	100
	Q5: The schematic regarding the occurrence states is useful and appropriate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	12	4	8	**40**	28	16	4	12	**60**	100
	Q6: The notification table is useful and adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	20	16	4	4	**44**	24	12	4	16	**56**	100
Section 5	Q1: It is easy to find the occurrence history page	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	20	8	4	8	**40**	24	20	4	12	**60**	100
	Q2: The information presented is adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	28	8	8	12	**56**	16	20	0	8	**44**	100
	Q3: The search bar is useful	0	0	0	0	**0**	0	0	4	0	**4**	0	0	0	0	**0**	28	8	0	16	**52**	16	20	4	4	**44**	100
Section 6	Q1: Does the application seem useful in a real-life context?	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	16	8	8	8	**40**	28	20	0	12	**60**	100
	Q2: Do you consider that the realization of the scenario was?	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	40	12	4	20	**76**	4	16	4	0	**24**	100

Professional Sectors

S1 - Healthcare **S2** - Technology **S3** - Education **S4** - Civil Protection

[1] Option only applicable in section 6.

Figure 6.3: Non-Administrative Form Results



		Completely Disagree/Very Hard[1]					Disagree/Hard[1]					No opinion					Agree/Easy[1]					Completely Agree/Very Easy[1]					
		%					%					%					%					%					
		S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	S1	S2	S3	S4	Total	Σ
Section 2	**Q1: It is easy to understand which victims have not yet been checked-in by the hospital**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	4	4	0	8	**16**	40	24	8	12	**84**	100
	Q2: The search bar is useful	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	4	**4**	20	8	4	12	**44**	24	20	4	4	**52**	100
	Q3: The hospital's patient table has the appropriate information	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	12	8	8	**40**	32	16	0	12	**60**	100
	Q4: The graphical interface is aesthetically appealing	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	12	8	4	**32**	36	16	0	16	**68**	100
	Q5: It is easy to find the button for your hospital's patient details	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	12	4	12	**40**	32	16	4	8	**60**	100
Section 3	**Q1: The graphical interface of the details page is aesthetically appealing**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	8	4	0	**20**	36	20	4	20	**80**	100
	Q2: The information received about the patient is adequate and well-structured	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	8	8	12	**36**	36	20	0	8	**64**	100
	Q3: It is easy to assign the date and time that the patient was checked into the hospital	0	0	0	0	**0**	0	0	0	0	**0**	4	0	0	0	**4**	8	12	8	8	**36**	32	16	0	12	**60**	100
	Q4: The information received on a patient's evaluation page is adequate and well-structured	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	4	8	8	**32**	32	24	0	12	**68**	100
	Q5: The information received on a patient's evaluation page is understandable	0	0	0	0	**0**	0	0	0	4	**4**	0	0	0	0	**0**	20	4	8	12	**44**	24	24	0	4	**52**	100
	Q6: The human image present in the symptoms page is useful for understanding a patient's traumas	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	4	**4**	12	8	4	4	**28**	32	20	4	12	**68**	100
	Q7: Information regarding the procedures applied on the victim is easy to understand	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	20	12	8	20	**60**	24	16	0	0	**40**	100
	Q8: The tooltips present in the scale procedures are useful and sufficient for understanding the displayed values	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	8	8	4	**32**	32	20	0	16	**68**	100
	Q9: The information in the table of administered pharmaceuticals is adequate and easy to read	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	8	8	8	12	**36**	36	20	0	8	**64**	100
Section 4	**Q1: It is easy to find the patient history page**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	28	8	8	8	**52**	16	20	0	12	**48**	100
	Q2: The information presented is adequate	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	32	12	8	16	**68**	12	16	0	4	**32**	100
	Q3: The search bar is useful	0	0	0	0	**0**	0	0	4	0	**4**	0	0	0	0	**0**	24	8	4	12	**48**	20	20	4	4	**48**	100
Section 5	**Q1: Does the application seem useful in a real-life context?**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	12	4	8	4	**28**	32	24	0	16	**72**	100
	Q2: Do you consider that the realization of the scenario was?	0	0	0	0	**0**	0	0	0	4	**4**	0	0	0	0	**0**	24	12	8	12	**56**	20	16	0	4	**40**	100

Professional Sectors

S1 - Healthcare **S2** - Technology **S3** - Education **S4** - Civil Protection

[1] Option only applicable in section 6.

Figure 6.4: Hospital Form Results

6.2.4 Chronometric Results

In addition to calculating the results of each questionnaire, the chronometric results of each test protocol were also subject of statistical analysis and each would later integrate a table.

The clocking activities were performed by the author of this thesis and were divided in phases and measured according to the time that a user took to complete the tasks present in each phase.

Regarding the administrative emergency central test protocol, the phase tasks delineated were:

1. Pick the most recent occurrence that doesn't yet have an assignment, and assign it to the *Bombeiros 1* central.
2. The occurrence started in the parish of *Santa Susana*, belonging to the munipality of *Alcácer do Sal*. Make the changes that mirror this information.
3. The victim of the occurrence chosen possesses a risk situation that is aggravated due to having a weakened social condition and suffers from gluten allergy. Make the necessary changes.
4. Enter the monitoring page of the occurrence you just handled.
5. Enter the occurrence history page.

Regarding the non-administrative emergency central test protocol, the phase tasks delineated were:

1. Pick the most recent occurrence of your corporation.
2. Register an ambulance as mean of assistance and assign the occurrence to the team led by technician *Maria Meireles*.
3. The occurrence's victim usually takes the food supplement *Artrozen*, and has been diagnosed with *Osteoporose*. It is a person at risk, due to having nervous anorexia. This way, leave an observation to the technician's team to handle the victim with care. Make the required changes.

The remaining test actions regarding the monitoring tasks present in this protocol were not timed due to the fact that this was an isolated test protocol. It has no value the timing of these tasks, as the occurrence states were manually assigned for the user to verify if the page's behaviour was correct. In a real situation, the time it takes for the monitoring tasks to be performed depends entirely on the time taken by the technician's time to execute each state. After a technician triggers a state change, the application broadcasts it to the dispatchers through its monitoring page.

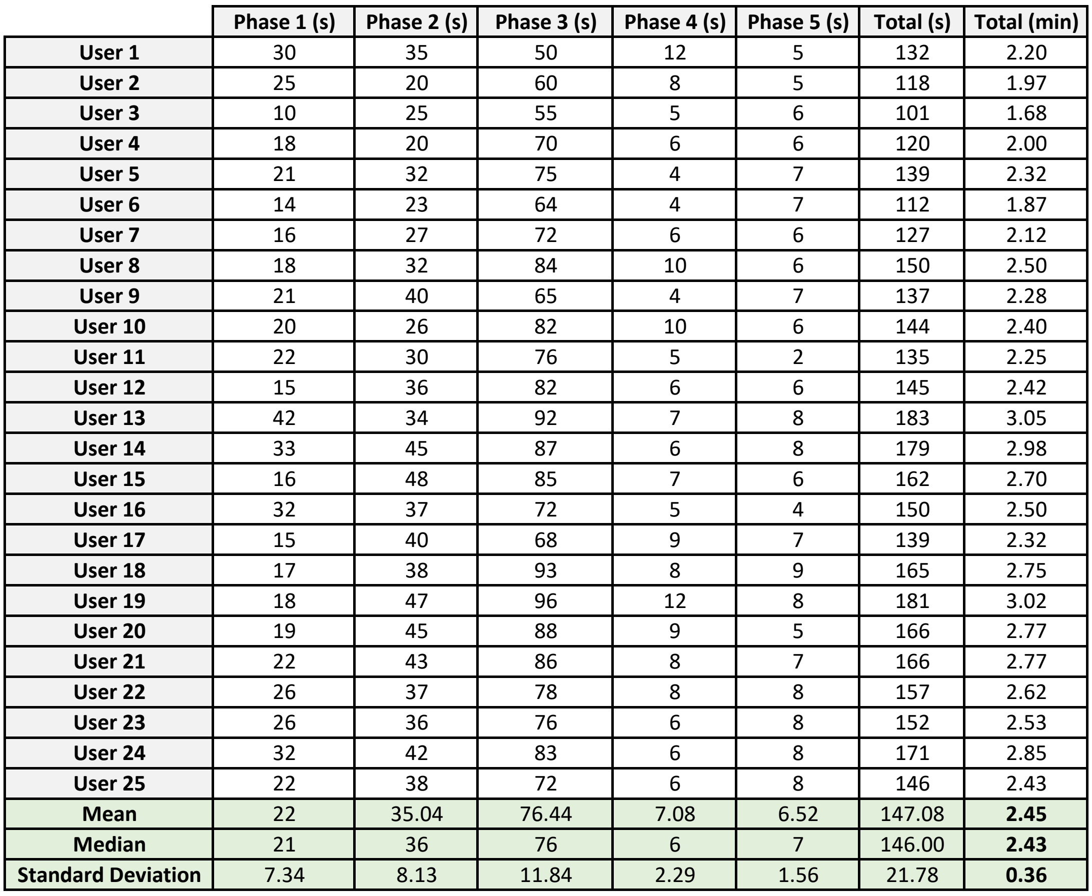
Regarding the hospital test protocol, as the majority of tasks were related to information consultation, only the total time taken to validate the patient information was timed.

The tables with each chronometric's results are as follows:

Figure 6.5 represents a table with the chronometric results of the Administrative Central test protocol.

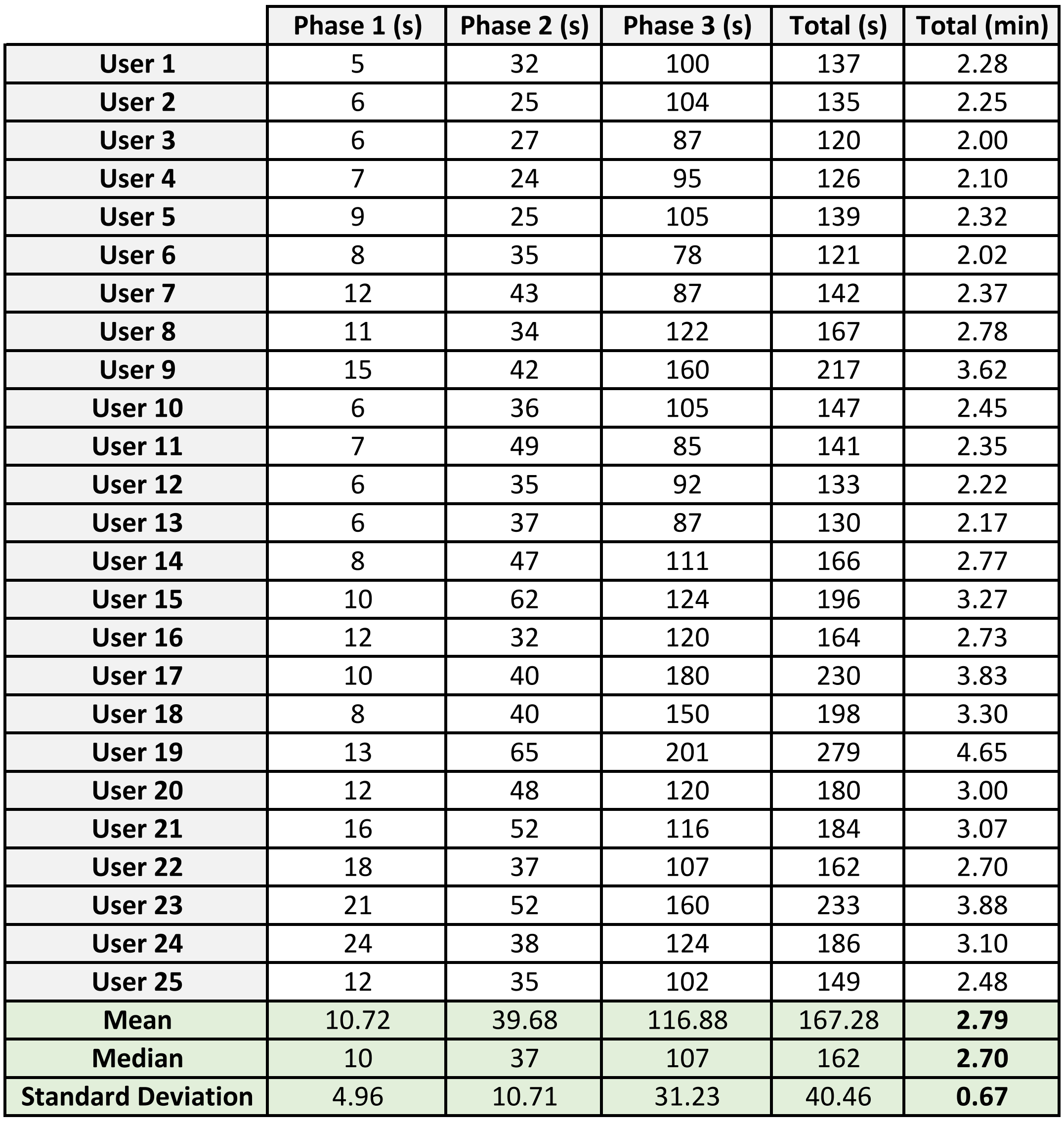
Figure 6.6 displays a table with the chronometric results of the Non-Administrative Central test protocol.

Figure 6.7 highlights a table with the chronometric results of the Hospital test protocol.



	Phase 1 (s)	Phase 2 (s)	Phase 3 (s)	Phase 4 (s)	Phase 5 (s)	Total (s)	Total (min)
User 1	30	35	50	12	5	132	2.20
User 2	25	20	60	8	5	118	1.97
User 3	10	25	55	5	6	101	1.68
User 4	18	20	70	6	6	120	2.00
User 5	21	32	75	4	7	139	2.32
User 6	14	23	64	4	7	112	1.87
User 7	16	27	72	6	6	127	2.12
User 8	18	32	84	10	6	150	2.50
User 9	21	40	65	4	7	137	2.28
User 10	20	26	82	10	6	144	2.40
User 11	22	30	76	5	2	135	2.25
User 12	15	36	82	6	6	145	2.42
User 13	42	34	92	7	8	183	3.05
User 14	33	45	87	6	8	179	2.98
User 15	16	48	85	7	6	162	2.70
User 16	32	37	72	5	4	150	2.50
User 17	15	40	68	9	7	139	2.32
User 18	17	38	93	8	9	165	2.75
User 19	18	47	96	12	8	181	3.02
User 20	19	45	88	9	5	166	2.77
User 21	22	43	86	8	7	166	2.77
User 22	26	37	78	8	8	157	2.62
User 23	26	36	76	6	8	152	2.53
User 24	32	42	83	6	8	171	2.85
User 25	22	38	72	6	8	146	2.43
Mean	22	35.04	76.44	7.08	6.52	147.08	**2.45**
Median	21	36	76	6	7	146.00	**2.43**
Standard Deviation	7.34	8.13	11.84	2.29	1.56	21.78	**0.36**

Figure 6.5: Administrative Emergency Central Test Protocol Chronometric Results



	Phase 1 (s)	Phase 2 (s)	Phase 3 (s)	Total (s)	Total (min)
User 1	5	32	100	137	2.28
User 2	6	25	104	135	2.25
User 3	6	27	87	120	2.00
User 4	7	24	95	126	2.10
User 5	9	25	105	139	2.32
User 6	8	35	78	121	2.02
User 7	12	43	87	142	2.37
User 8	11	34	122	167	2.78
User 9	15	42	160	217	3.62
User 10	6	36	105	147	2.45
User 11	7	49	85	141	2.35
User 12	6	35	92	133	2.22
User 13	6	37	87	130	2.17
User 14	8	47	111	166	2.77
User 15	10	62	124	196	3.27
User 16	12	32	120	164	2.73
User 17	10	40	180	230	3.83
User 18	8	40	150	198	3.30
User 19	13	65	201	279	4.65
User 20	12	48	120	180	3.00
User 21	16	52	116	184	3.07
User 22	18	37	107	162	2.70
User 23	21	52	160	233	3.88
User 24	24	38	124	186	3.10
User 25	12	35	102	149	2.48
Mean	10.72	39.68	116.88	167.28	**2.79**
Median	10	37	107	162	**2.70**
Standard Deviation	4.96	10.71	31.23	40.46	**0.67**

Figure 6.6: Non-Administrative Emergency Central Test Protocol Chronometric Results

	Total (min)
User 1	4.15
User 2	5.30
User 3	5.25
User 4	5.08
User 5	6.72
User 6	4.42
User 7	4.62
User 8	7.03
User 9	5.38
User 10	4.53
User 11	4.93
User 12	6.58
User 13	5.93
User 14	6.87
User 15	8.10
User 16	4.25
User 17	4.72
User 18	6.33
User 19	6.93
User 20	6.30
User 21	7.12
User 22	6.33
User 23	6.95
User 24	6.35
User 25	7.07
Mean	5.89
Median	6.30
Standard Deviation	1.11

Figure 6.7: Hospital Test Protocol Chronometric Results

6.3 Discussion

The project contained in this thesis proposed a system that addressed the pre-hospital emergency framework current limitations, mainly in the interaction between administrative and non-administrative emergency centrals and in the phase of patient transfer to the target health care unit. The results collected served as a basis to verify the suitability of the developed project in solving these issues, and were therefore discussed.

By analyzing the results contemplated on figure 6.2, it was possible to verify that:

– In section 2, the only negative results came from the question 2 (*The search bar is useful*), with 8% of the users (1 from the healthcare sector and the other from the education sector) choosing the Disagree option. The only question with neutral choices was the question 3 (*The table of active occurrences has the appropriate information*) with 8% of the users (both from the healthcare sector) choosing the No opinion option. The overwhelming majority of the results from this section were positive, coming from both the Agree and Completely Agree options, accounting for more than 92% of the user choices in all section questions.

– In section 3, no negative or neutral results were gathered. All the questions in this section were answered with positive choices, where the Completely Agree option exceeds the Agree option in all questions except for question 4 (*Navigation through the detail pages is intuitive*) with a 48%/52% ratio.

– In section 4, the positive options account in all questions for at least 96% of the total chosen answers. The Completely Agree option exceeded the Agree option in all questions with a difference of at least 20%, with the highest difference gathered being 40% in question 2 (*The map present in the geographical monitoring panel is useful*).

– In section 5, there was only 1 negative option accounted in question 3 (*The search bar is useful*), with the remaining being answers from positive options.

– In section 6, all the test users considered the application to be useful in a real-life context (40% Agree and 60% Completely Agree). 76% of the test users found the test protocols to be Easy and 24% found them Very Easy.

It is possible to verify that the test users responded positively to their experience in accessing an administrative emergency central. Regarding the administrative central questionnaire some users gathered observations or possible application improvements, of which are highlighted:

- In the details page, the states page should come after the victim information.
- The map present in the geographical monitoring shows in detail where the occurrence is.
- The active occurrences table, although already sorted through the most recent occurrences, should have the occurrence creation date displayed.

The suggestions given by the users were taken into consideration, leading to the creation date of the occurrence being added to the active occurrences table and the occurrence states page being placed next to the occurrence victim information.

Due to the fact that the non-administrative emergency central test protocol was fairly similar to the administrative emergency central test protocol and the user interface was the same (the only visible changes were the restrictions implemented on each emergency central type), the results gathered were almost identical. The analysis of the results contemplated on figure 6.3 proves this assertion, as every section results almost equates to the ones described in figure 6.2.

The only question result that must be discussed, as it is the only one that does not have a connection to the administrative emergency central test protocol is the question 7 from section 4 (*The monitoring page is an asset to ensure that the execution of the occurrence proceeds smoothly*). The answers to this question were 100% positive, divided into 36% of answers as Agree and 65% as Completely Agree. This shows that the users responded positively to the monitoring page feature as a whole.

As previous, the test users responded positively to their experience in accessing a non-administrative emergency central. Regarding the non-administrative emergency central questionnaire, the users also gathered some observations or possible application improvements, such as:

- The notifications table in the monitoring page should refresh automatically.
- In the occurrence details the victim should come before the states.
- Searching by occurrence number is difficult, because the numbers have no logical meaning.

The first observation, although taken into consideration, was not implemented due to shortage of time, being given priority to other unresolved matters. The second observation, as previously stated was corrected. The third observation was discarded, as the occurrence number was the search parameter deemed most useful, as it identifies uniquely each occurrence.

The hospital test protocol, although part of the same application, belonged to a different subsystem. So, the results gathered from the questionnaire were target of discussion. The analysis of figure 6.4 allows for the conclusion that:

– In section 2, there was no negative results. The only non-positive result came from question 2 (*The search bar is useful*), where a user working in the civil protection sector chose the No Opinion option, 4% of the total in that question. The remaining questions of this section were overwhelmingly positive, where the Completely Agree option exceeded the Agree option by at least 20% in all questions, and is most predominant in question 1 (*It is easy to understand which victims have not yet been checked-in by the hospital*), where 84% test users chose the Completely Agree option and 16% chose Agree, with a difference of 68%.

– In section 3, there was only 1 negative result from question 6 (*The information received on a patient's evaluation page is understandable*) from a test user working in the civil protection sector. There was also 1 No opinion option selected in question 7 (*Information regarding the procedures applied on the victim is easy to understand*), also from a civil protection test user. The remaining answers were positive with the Completely Agree option percentage exceeding the Agree option considerably, especially in question 1 (*The graphical interface of the details page is aesthetically appealing*), where the Completely Agree option gathered 80% of the submitted results. Question 8 was the only 1 that the Agree option (60%) exceeded the Completely Agree option (40%). Accounting the fact that most of the test users were not from the healthcare sector, the tooltips proved to be useful for the user's understanding of the information being presented to them, as can be seen by question 8 (*The tooltips present in the scale procedures are useful and sufficient for understanding the displayed values*) results (68% Completely Agree and 32% Agree).

– Section 4 followed the trend from the previous sections, with a overwhelming positive reception. Only 1 answer was non-positive.

– In section 6, all the test users considered the application to be useful in a real-life context, with 72% of the users voting Completely Agree and 28% voting Agree. In terms of test protocol difficulty, 4% considered it to be Difficult, 56% considered it to be Easy and 40% considered to be Very Easy.

As it can concluded, the hospital subsystem of the application received an overall positive reception, in terms of user satisfaction with its use. It was also gathered observations and improvement suggestions from the users, such as:

- All the procedures should have a tooltip with a description of each one.
- If implemented in hospitals it will be very useful for tracking everything that happened to the patient prior to his admission.
- Should allow insertion of contact and relationship of significant person to contact in case of need.

The first observation, although taken into consideration, was not implemented due to shortage of time, being given priority to other unresolved matters. The third observation was not considered, as of now, because its implementation required to change the structure of the data model, so it will be included in the future work plans.

The statistical measures used to analyse the chronometric results were the mean (arithmetic), median and standard deviation).

From the results highlighted in figure 6.5:

- Phase 1 had a mean of 22 seconds, a median of 21 seconds and a standard deviation of 7.34. The value of the standard deviation.
- Phase 2 had a mean of 35.04 seconds, a median of 36 seconds and a standard deviation of 8.13.
- Phase 3 had a mean of 76.44 seconds, a median of 76 seconds and a standard deviation of 11.84.

- Phase 4 had a mean of 7.08 seconds, a median of 6 seconds and a standard deviation of 2.29.
- Phase 5 had a mean of 6.52 seconds, a median of 7 seconds and a standard deviation of 1.56.
- The total had a mean of 147.08 seconds, a median of 146 seconds and a standard deviation of 21.78. In minutes, these values equated to a mean of 2.45, a median of 2.43 and a standard deviation of 0.36.

From the results highlighted in figure 6.6:

- Phase 1 had a mean of 10.72 seconds, a median of 10 seconds and a standard deviation of 4.96.
- Phase 2 had a mean of 39.68 seconds, a median of 37 seconds and a standard deviation of 10.71.
- Phase 3 had a mean of 116.88 seconds, a median of 107 seconds and a standard deviation of 31.23.
- The total had a mean of 167.28 seconds, a median of 162 seconds and a standard deviation of 40.46. In minutes, these values equated to a mean of 2.79, a median of 2.70 and a standard deviation of 0.67.

From the results highlighted in figure 6.7:

- In minutes, the total had a mean of 5.89, a median of 6.3 and a standard deviation of 1.11.

Chapter 7

Conclusions

This chapter presents a summary of the project's development, its main contributions, and draws conclusions about the accomplished work. Furthermore, it describes the main limitations of the project and proposes future work.

7.1 Summary

Due to the limitations encountered in the pre-hospital emergency framework in Portugal, we proposed an application aimed at automating and improving the processes currently in place. Currently, there is still a high degree of dependence on human processes that restricts the quality of service provided by the pre-hospital professionals. Occurrences that fall under the scope of INEM do not experience these problems, as their treatment is carried out through SIEM's purview, which has access to all INEM's resources. By working with ANTEPH, we asserted that the vast majority of occurrences still operate outside the scope of INEM, and in these cases the service is poor and its process flow is mostly human dependent, where the probability of committing blunders that affect the service is increased.

The proposed system relied on an API developed through Django REST framework, where its data model and resulting API methods were modeled after the INEM's *Verbete Nacional de Socorro*. The API was then integrated into a LCDP platform, denominated Outsystems, where the application logic was implemented (both business and UI logic). This application was implemented to meet its specified requirements. The application was divided into 2 subsystems: the emergency central subsystem and the hospital subsystem.

The emergency central subsystem is comprised of 2 types of institutions and their respective users, namely: administrative emergency centrals, that assume the same role as the CODU; non-administrative emergency centrals, that assume the same role as CVP stations; and firefighter stations, that are not in protocol with the INEM and therefore have to execute the occurrence's with their own resources.

The hospital subsystem is comprised of health care units, that assume the role of the destination health care units. This subsystem is only aimed at improving the transfer of victim information from the pre-hospital environment to the hospital environment. The scope of the system ends after the transfer is completed and the victim information is recorded into the hospital system.

To validate the suitability of the application and its user experience, we formulated a set of timed test protocols, designed to simulate plausible real-life situations, and capture user's feedback concerning the use of the several systems. The final questionnaire and chronometric results asserted the application's potential suitability to resolve the problem outlined for this thesis, as the questionnaire results were overwhelmingly positive (<90% of the total answers derived from positive answers, such as Agree and Completely Agree). This statement is further attested as all the 25 users employed over the 3 testing protocols stated that were willing to use these systems in a real-life context (100% of the submitted answers in the 3 questionnaires were derived from the positive answers).

We conclude that the project was successful in achieving its objectives. The following sections present a discussion about the project contributions, limitations, and the success of its management practices.

7.2 Contributions

This thesis main contributions are:

- Technical study of the Django framework and its toolkit Django REST framework.
- Comparative technical study of the Outsystems and Mendix platforms, both LCDP's considered for the project.
- Technical study of Nginx applied in the scope of the project.
- Technical study of Gunicorn applied in the scope of the project.
- Development of a REST API and a software application based on the document *Verbete Nacional de Socorro*, with aims of automatizing the information contained within it, namely at the pre-hospital emergency central and hospital level.
- Reporting of the usage times of the application subsystems during the execution of test protocols.

7.3 Project Management Performance

The project management was successful, since all the criteria outlined for its success were fulfilled, allowing for the project's success to be achieved.

At the project start, a considerable time was invested in properly scheduling all the stages of the project development and its deliverables. This time proved to be useful, as the project development evolved exactly as outlined and without major setbacks.

The criterion planned time estimation vs the actual execution was successful, as the project was concluded in its scheduled time.

The criterion related to the number of adjustments to the project schedule was successful, since the adjustments made during the development of the project were only noticeable during the phase in which they occurred, and even when they did occur, they never compromised the deliverables.

The criterion on-time task completion percentage was successful, as more than 90% of the tasks were completed during the stipulated time.

The criterion related to the number of change requests was successful as the only change requests were verified during the implementation stage and during the writing of this thesis. As the change requests made didn't hinder the stages completion time and the quality of the deliverables produced, they were considered a natural part of the project's development life cycle.

Given these factors, we reinforce that the project's management success criteria delineated in the planning stage were met.

7.4 Project Performance

As mentioned previously, the project was deemed successful in its implementation at both subsystem levels. We will conclude the reasons for this statement by cross-referencing the requirements specified in Chapter 4 with their implementation status and in which implemented pages their compliance was met.

The tables ranging from 7.1 to 7.10 pertain to the functional requirements of the **Emergency Central subsystem** and table 7.11 to 7.16 pertain to the functional requirements of the **hospital subsystem** .

Table 7.1: Requirement 4.1 Final Status

ECSFR1	**Login Page**	**Status**	Implemented
Only registered users in the SIREPH system are able to access the application. Details: The ECSFR1 requirement is fully implemented and its compliance is met in the login page. Through its authentication token, the system verifies if the user currently logging is registered or not.			

Table 7.2: Requirement 4.2 Final Status

ECSFR2	**Login Page**	**Status**	Implemented
Only users registered as dispatchers are able to access the application's emergency central subsystem. Details: The ECSFR2 requirement is fully implemented and its compliance is met in the login page, where the login action verifies if the user currently logging is a dispatcher in the system.			

Table 7.3: Requirement 4.3 Final Status

ECSFR3	**Homepage**	**Status**	Implemented
The occurrence information is only available to authorized users. Details: The ECSFR3 requirement is fully implemented and its compliance is met in the homepage, where dispatchers can only see occurrence's assigned to their emergency central. Administrative central users can see every occurrence registered in the system.			

Table 7.4: Requirement 4.4 Final Status

ECSFR4	**Occurrence Details Pages**	**Status**	Implemented
The application allows the visualization of all the information related to a specific occurrence. Details: The ECSFR4 requirement is fully implemented and its compliance is met in the Details pages, where dispatchers can view occurrence information, occurrence states and team information as well as the occurrence's victim(s) information.			

Table 7.5: Requirement 4.5 Final Status

ECSFR5	**Occurrence Details Pages**	**Status**	Implemented
The application allows the update of all the information related to a specific active occurrence. Details: The ECSFR5 requirement is fully implemented and its compliance is met in the occurrence details pages, where dispatchers can update information from an active occurrence.			

Table 7.6: Requirement 4.6 Final Status

ECSFR6	**Occurrence Information Detail Page**	**Status**	Implemented
Only administrative central users are able to assign an emergency central to an occurrence. Details: The ECSFR6 requirement is fully implemented and its compliance is met in the occurrence information detail page, where dispatchers from administrative centrals can assign occurrences to an emergency central.			

Table 7.7: Requirement 4.7 Final Status

ECSFR7	**Occurrence Information Detail Page**	**Status**	Implemented
Non-administrative central users are able to assign an occurrence to an active team from their central. Details: The ECSFR7 requirement is fully implemented and its compliance is met in the occurrence information detail page, where dispatchers from non-administrative centrals can assign occurrences to a team of technicians from their own central.			

Table 7.8: Requirement 4.8 Final Status

ECSFR8	**Monitorization Page**	**Status**	Implemented
The application allows for the geographical and temporal monitoring of occurrence state updates. Details: The ECSFR8 requirement is fully implemented and its compliance is met in the monitorization page, with the implementation of the geographical map interface and the occurrence notification table.			

Table 7.9: Requirement 4.9 Final Status

ECSFR9	**Monitorization Page**	**Status**	Implemented
The application displays the current occurrence state. Details: The ECSFR9 requirement is fully implemented and its compliance is met in the monitorization page. The marker present in the geographical panel displays the coordinates of the current state and the form wizard also displays the current state position within all possible states of an occurrence.			

Table 7.10: Requirement 4.10 Final Status

ECSFR10	**Occurrence History Page**	**Status**	Implemented
The application keeps record of the completed occurrences. Details: The ECSFR10 requirement is fully implemented and its compliance is met in the history page. Non-administrative users can see the list of completed occurrences executed by their central and administrative users can see the list of all completed occurrences in the system.			

Table 7.11: Requirement 4.11 Final Status

HFR1	**Login Page**	**Status**	Implemented
Only registered users in the SIREPH system are able to access the application's hospital subsystem. Details: The HFR1 requirement is fully implemented and its compliance is met in the login page. Through its authentication token, the system verifies if the user currently logging is registered or not.			

Table 7.12: Requirement 4.12 Final Status

HFR2	**Login Page**	**Status**	Implemented
Only users registered as hospital staff are able to access the application's hospital subsystem. Details: The HFR2 requirement is fully implemented and its compliance is met in the login page, where the login action verifies if the user currently logging is a hospital staff in the system.			

Table 7.13: Requirement 4.13 Final Status

HFR3	**Homepage**	**Status**	Implemented
The patient information is only available to authorized users. Details: The HFR3 requirement is fully implemented and its compliance is met in the homepage, where hospital staff users can only see the information of patients that are assigned to their hospital.			

Table 7.14: Requirement 4.14 Final Status

HFR4	**Patient Details Pages**	**Status**	Implemented
The application allows the visualization of all the information related to a specific patient Details: The HFR4 requirement is fully implemented and its compliance is met in the patient details pages, where hospital staff users can view all the information regarding a patient.			

Table 7.15: Requirement 4.15 Final Status

HFR5	**Patient Information Detail Page**	**Status**	Implemented
Hospital staff users are able to check patients into their hospital. Details: The HFR5 requirement is fully implemented and its compliance is met in the patient information detail pages, where hospital staff users can select the date and time of the patient check-in at the hospital.			

Table 7.16: Requirement 4.16 Final Status

HFR6	**Patient History Page**	**Status**	Implemented
The application keeps record of the patient's that were checked into the user's hospital. Details: The HFR6 requirement is fully implemented and its compliance is met in the patient history page, where hospital staff users can view all the patients that were checked into their hospital.			

Regarding the specified non-requirements the tables ranging from 7.17 to 7.26 verify their compliance status at the end of the project.

Table 7.17: Requirement 4.17 Final Status

NFR1	**Responsive**	**Status**	Implemented
The application is optimized for all devices, result of being built as an Outsystems Reactive Web Application.			

Table 7.18: Requirement 4.18 Final Status

NFR2	**Usability**	**Status**	Implemented
The application performed positively on the questionnaire usability questions.			

Table 7.19: Requirement 4.19 Final Status

NFR3	**Availability**	**Status**	Implemented
The application reported 100% uptime during the performed user validation tests and no decision made in its implementation presents constraints that could impact its availability.			

Table 7.20: Requirement 4.20 Final Status

NFR4	**Security**	**Status**	Implemented
The application is secured against unauthorized access to functionality and stored data through Django's authentication system, protecting the system by token authentication and permission to the API methods called in the application's usage is dependent on the user being authenticated. Furthermore, Outsystems possess built-in security features that provide an extra layer of security.			

Table 7.21: Requirement 4.21 Final Status

NFR5	**Integrability**	**Status**	Implemented
The application correctly persists information in the SIREPH's system			

Table 7.22: Requirement 4.22 Final Status

NFR6	**Integrity**	**Status**	Implemented
The application displayed the accurate information during the scope of all user validation tests.			

Table 7.23: Requirement 4.23 Final Status

NFR7	**Performance**	**Status**	Implemented
All the application's URL's reported web response times within the desired range of 100 milliseconds to 1 second, tested using the capabilities of the dotcom-tools [92] platform.			

Table 7.24: Requirement 4.24 Final Status

NFR8	**Reliability**	**Status**	Implemented
The application didn't report any system errors during the user validation tests and the tasks were performed without failure 100% of the time (excluding user error incidents.			

Table 7.25: Requirement 4.25 Final Status

NFR9	**Extensibility**	**Status**	Implemented
The system's architecture is built upon established technologies that are able to accommodate future updates seamlessly.			

Table 7.26: Requirement 4.26 Final Status

NFR10	**Scalability**	**Status**	Partially Implemented (Unverified at national level)
The application didn't report any scalability issues during the user validation tests and it doesn't possess in its implementation any constraints that restrict its scalability, however no validation was conducted to verify the application's capacity to accommodate the expected service load for medical emergency response at the national level.			

Throughout this section the project's success was corroborated, as all specified functional and non-functional requirements were successfully implemented. The results gathered from the user validation efforts further attest to the project's status as a successful proof of concept.

7.5 Limitations

The limitations encountered throughout the course of this thesis are outlined in the following items:

- There was limited access to information concerning pre-hospital emergency related procedures, mainly at the structural level, where information is scattered and often contradictory amongst entities.

- The project contained in this thesis was part of a larger shared system. Due to this constraint, the development of the API proved an even more challenging task as all the participants had to be in sync about all the ongoing developments, so as not to jeopardize the intended features of the other participant's application.

7.6 Future Work

The proposed future work is listed in the following items:

- Conduct further validation tests with a higher sample size and a fitter sample set for each of the application's subsystems, preferably in the context of an emergency scenario simulation.
- Further develop the monitoring page, with additional features such as automatic updates, real-time geographical assistance and real-time dashboards, through the integration of Django Channels [93] capabilities or similar technologies into the system.
- Implement further user-triggered tooltips in the application's hospital subsystem UI, to provide a clearer user experience, especially in the content of patient evaluations and procedures.
- Implement a fleet management feature, in order for the non-administrative emergency centrals to be able to oversight the use of their means of assistance.
- Implement another subsystem dedicated to the collection, analysis and generation of interactive visualizations of all the pre-hospital emergency related information captured within the SIREPH system, to equip the competent authorities with the knowledge needed for a more effective decision making.
- Implement a feature which lists the medical specialties available at the destination health unit, allowing the assignment of the one that fits the patient's situation, and thus improving the integration of the pre-hospital emergency framework with the hospital framework during the patient's handover phase.

References

[1] M. H. Wilson et al. "Pre-hospital emergency medicine". In: *The Lancet* 386.10012 (), pp. 2526–2534. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(15)00985-X. URL: https://www.sciencedirect.com/science/article/pii/S014067361500985X (cit. on p. 1).

[2] A. Mortaro et al. "The role of the emergency medical dispatch centre (EMDC) and prehospital emergency care safety: results from an incident report (IR) system". In: *CJEM* 17.4 (), pp. 411–419. DOI: 10.1017/cem.2014.74 (cit. on pp. 1, 9).

[3] D. da República Eletrónico. *Decreto-Lei n.º 234/81 dos Ministérios da Defesa Nacional. das Finanças e do Plano, dos Assuntos Sociais e da Reforma Administrativa.* (visited on 2022-02-16). 1981. URL: https://dre.pt/dre/detalhe/decreto-lei/234-1981-576950 (cit. on p. 2).

[4] INEM. *Sistema Integrado de Emergência Médica.* Version 2.0. 2013. URL: https://www.inem.pt/wp-content/uploads/2017/06/Sistema-Integrado-de-Emerg%C3%AAncia-M%C3%A9dica.pdf (visited on 02/20/2022) (cit. on p. 2).

[5] D. da República Eletrónico. *Decreto-Lei n.º 10319/2014 do Ministério da Saúde - Gabinete do Secretário de Estado Adjunto do Ministro da Saúde.* 2014. URL: https://dre.pt/dre/detalhe/despacho/10319-2014-55606457 (visited on 02/20/2022) (cit. on p. 3).

[6] J.-H. Kao et al. "A Web-based Medical Emergency Guiding System". In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.* 2012, pp. 739–744. DOI: 10.1109/ASONAM.2012.136 (cit. on p. 10).

[7] E. Kyriacou et al. "eEmergency System to Support Emergency call Evaluation and Ambulance dispatch Procedures". In: *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON).* 2020, pp. 354–357. DOI: 10.1109/MELECON48756.2020.9140544 (cit. on p. 11).

[8] B. Silva and J. C. Nascimento. "Key Issues of Portuguese Health Information Systems". In: *2021 16th Iberian Conference on Information Systems and Technologies (CISTI).* 2021, pp. 1–6. DOI: 10.23919/CISTI52073.2021.9476539 (cit. on p. 12).

[9] *Serviço Nacional de Saúde.* URL: https://www.spms.min-saude.pt/2019/01/product-sclinicohospitalar/ (visited on 10/03/2022) (cit. on p. 12).

[10] A. Murad et al. "Enabling Patient Information Handoff from Pre-hospital Transport Providers to Hospital Emergency Departments: Design-Science Approach to Field Testing". In: *2014 47th Hawaii International Conference on System Sciences.* 2014, pp. 2665–2674. DOI: 10.1109/HICSS.2014.336 (cit. on p. 13).

[11] S. Trzeciak and E. Rivers. “Emergency department overcrowding in the United States: An emerging threat to patient safety and public health”. In: *Emergency medicine journal : EMJ* 20 (Oct. 2003), pp. 402–5. DOI: 10.1136/emj.20.5.402 (cit. on p. 13).

[12] K. Ye et al. “Handover in the emergency department: deficiencies and adverse effects”. In: *Emergency Medicine Australasia* 19.5 (2007), pp. 433–441 (cit. on p. 13).

[13] G. He and Y. Yang. “The research and implementation of medical information collaborative service platform based on the Web Service”. In: *2012 International Conference on Systems and Informatics (ICSAI2012)*. 2012, pp. 2671–2674. DOI: 10.1109/ICSAI.2012.6223604 (cit. on pp. 14, 15).

[14] B. Shaw et al. *Web development with django: Learn to build modern web applications with a python-based framework*. Packt Publishing Ltd., 2021 (cit. on pp. 17–20, 22–25).

[15] I. Engineering. *Web service efficiency at Instagram with python*. June 2016. URL: https://instagram-engineering.com/web-service-efficiency-at-instagram-with-python-4976d078e366 (visited on 03/02/2022) (cit. on p. 17).

[16] N. George. *Mastering django core: The Complete Guide to django 1.8 LTS*. GNW Independent Publishing, 2016 (cit. on p. 18).

[17] M. Lorenz et al. “Object-Relational Mapping Revisited - A Quantitative Study on the Impact of Database Technology on O/R Mapping Strategies”. In: *HICSS*. 2017 (cit. on p. 18).

[18] D. Colley, C. Stanier, and M. Asaduzzaman. “The Impact of Object-Relational Mapping Frameworks on Relational Query Performance”. In: *2018 International Conference on Computing, Electronics Communications Engineering (iCCECE)*. 2018, pp. 47–52. DOI: 10.1109/iCCECOME.2018.8659222 (cit. on p. 19).

[19] T.-H. Chen et al. “Detecting Performance Anti-Patterns for Applications Developed Using Object-Relational Mapping”. In: *Proceedings of the 36th International Conference on Software Engineering*. ICSE 2014. Hyderabad, India: Association for Computing Machinery, 2014, pp. 1001–1012. ISBN: 9781450327565. DOI: 10.1145/2568225.2568259. URL: https://doi.org/10.1145/2568225.2568259 (cit. on p. 19).

[20] *SQL antipatterns: Avoiding the pitfalls of Database Programming*. Pragmatic Bookshelf, 2012 (cit. on p. 19).

[21] *Django Project*. URL: https://www.djangoproject.com/ (visited on 03/02/2022) (cit. on pp. 19–22, 24, 58).

[22] D. Greenfeld and A. Roy. *Two scoops of django: Best practices for django 1.8*. Two Scoops Press, 2015 (cit. on pp. 20, 22).

[23] J. C. S. Eric Enge Stephan Spencer. *The Art of SEO – Mastering Search Engine Optimization*. O’Reilly Media, 2015 (cit. on p. 21).

[24] *Cool uris don’t change*. URL: https://www.w3.org/Provider/Style/URI (visited on 04/03/2022) (cit. on p. 22).

[25] R. J. Anthony. *Systems Programming Designing and Developing Distributed Applications*. Massachusetts: Morgan Kaufmann, 2016 (cit. on p. 24).

[26] *TechTarget - The 5 essential HTTP methods in RESTful API development*. URL: https://www.techtarget.com/searchapparchitecture/tip/The-5-essential-HTTP-methods-in-RESTful-API-development (visited on 09/28/2022) (cit. on p. 26).

[27] *InfoQ - What is idempotent in REST*. URL: https://www.infoq.com/news/2013/04/idempotent/ (visited on 09/28/2022) (cit. on p. 26).

[28] T. Christie. *Django rest framework*. URL: https://www.django-rest-framework.org/ (visited on 07/03/2022) (cit. on pp. 26, 28).

[29] D. Hoffman and P. Strooper. "API documentation with executable examples". In: *Journal of Systems and Software* 66.2 (2003), pp. 143–156. ISSN: 0164-1212. DOI: https://doi.org/10.1016/S0164-1212(02)00055-9. URL: https://www.sciencedirect.com/science/article/pii/S0164121202000559 (cit. on p. 28).

[30] G. Uddin, F. Khomh, and C. K. Roy. "Towards Crowd-Sourced API Documentation". In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. 2019, pp. 310–311. DOI: 10.1109/ICSE-Companion.2019.00129 (cit. on p. 28).

[31] *OpenAPI Specification*. URL: https://swagger.io/specification/ (visited on 09/22/2022) (cit. on p. 28).

[32] S. Casas et al. "Uses and applications of the OpenAPI/Swagger specification: a systematic mapping of the literature". In: *2021 40th International Conference of the Chilean Computer Science Society (SCCC)*. 2021, pp. 1–8. DOI: 10.1109/SCCC54552.2021.9650408 (cit. on p. 28).

[33] *drf-yasg*. URL: https://drf-yasg.readthedocs.io/en/stable/ (visited on 09/22/2022) (cit. on p. 29).

[34] *Outsystems Low-Code Guide*. URL: https://www.outsystems.com/guide/low-code/ (visited on 09/22/2022) (cit. on p. 29).

[35] P. M. Gomes and M. A. Brito. "Low-Code Development Platforms: A Descriptive Study". In: *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*. 2022, pp. 1–4. DOI: 10.23919/CISTI54924.2022.9820354 (cit. on p. 29).

[36] *Gartner Magic Quadrant for Enterprise Low-Code Application Platforms*. URL: https://www.gartner.com/doc/reprints?id=1-295WSI86&ct=220217&st=sb (visited on 09/22/2022) (cit. on p. 29).

[37] *Cio Dive - Low-Code Implementation Coronavirus 2020*. URL: https://www.ciodive.com/news/low-code-implementation-coronavirus-2020/580024/ (visited on 09/22/2022) (cit. on p. 29).

[38] *Outsystems Documentation -Setting Up Outsystems*. URL: https://success.outsystems.com/Documentation/11/Setup_and_maintain_your_OutSystems_infrastructure/Setting_Up_OutSystems (visited on 09/23/2022) (cit. on p. 30).

[39] *Outsystems Documentation - Service Studio Overview*. URL: https://success.outsystems.com/Documentation/11/Getting_started/Service_Studio_Overview (visited on 09/23/2022) (cit. on p. 30).

[40] *Outsystems Documentation - Integration Studio.* URL: https://success.outsystems.com/Documentation/11/Reference/Integration_Studio (visited on 09/23/2022) (cit. on p. 31).

[41] *Outsystems Documentation - Monitor and Troubleshoot.* URL: https://success.outsystems.com/Documentation/11/Managing_the_Applications_Lifecycle/Monitor_and_Troubleshoot (visited on 09/23/2022) (cit. on p. 31).

[42] *Outsystems Documentation - LifeTime.* URL: https://success.outsystems.com/Documentation/11/Managing_the_Applications_Lifecycle/Manage_Your_OutSystems_Infrastructure (visited on 09/23/2022) (cit. on p. 31).

[43] *Outsystems Guided Path - Service Studio Walkthrough.* URL: https://www.outsystems.com/training/lesson/2185/service-studio-walkthrough?LearningPathId=18 (visited on 09/23/2022) (cit. on p. 32).

[44] *Outsystems Guided Path - Application Layers.* URL: https://www.outsystems.com/training/lesson/2186/application-layers?LearningPathId=18 (visited on 09/23/2022) (cit. on p. 33).

[45] *Outsystems Guided Path - 1-Click Publish.* URL: https://www.outsystems.com/training/lesson/2189/1-click-publish?LearningPathId=18 (visited on 09/23/2022) (cit. on p. 34).

[46] *Outsystems Guided Path - Modular Programming.* URL: https://www.outsystems.com/training/lesson/2159/modular-programming?LearningPathId=18 (visited on 09/27/2022) (cit. on p. 34).

[47] *Battery - Mendix Case Study.* URL: https://www.battery.com/blog/mendix-case-study/ (visited on 09/27/2022) (cit. on p. 34).

[48] *Mendix Documentation - General Info.* URL: https://docs.mendix.com/studio/general/ (visited on 09/27/2022) (cit. on p. 35).

[49] *Mendix Documentation - Workflows.* URL: https://docs.mendix.com/studio/workflows/ (visited on 09/27/2022) (cit. on p. 36).

[50] *Mendix Documentation - Microflows.* URL: https://docs.mendix.com/studio/microflows/ (visited on 09/27/2022) (cit. on p. 36).

[51] *Mendix Documentation - Domain Model.* URL: https://docs.mendix.com/studio/domain-models/ (visited on 09/27/2022) (cit. on p. 37).

[52] *Nginx.* URL: https://www.nginx.com/resources/glossary/nginx/ (visited on 10/12/2022) (cit. on p. 37).

[53] *Linux Web Server Performance Benchmark – 2016 Results.* URL: https://www.rootusers.com/linux-web-server-performance-benchmark-2016-results/ (visited on 10/12/2022) (cit. on p. 37).

[54] *vsupalov - Gunicorn and Nginx in a Nutshell.* URL: https://peps.python.org/pep-3333/ (visited on 10/12/2022) (cit. on pp. 37, 38).

[55] *PEP 3333 – Python Web Server Gateway Interface v1.0.1.* URL: https://peps.python.org/pep-3333/ (visited on 10/12/2022) (cit. on p. 38).

[56] *Gunicorn Design.* URL: https://docs.gunicorn.org/en/stable/design.html (visited on 10/12/2022) (cit. on p. 38).

[57] "Chapter 9 - PostgreSQL on Linux". In: *DBAs Guide to Databases Under Linux*. Ed. by D. Egan, P. Zikopoulos, and C. Rogers. Burlington: Syngress, 2000, pp. 359–418. ISBN: 978-1-928994-04-6. DOI: https://doi.org/10.1016/B978-192899404-6/50012-4. URL: https://www.sciencedirect.com/science/article/pii/B9781928994046500124 (cit. on p. 38).

[58] *Hetzner Docs - PostgreSQL*. URL: https://docs.hetzner.com/konsoleh/account-management/databases/postgresql (visited on 09/28/2022) (cit. on p. 39).

[59] R. Joslin and R. Müller. "Relationships between a project management methodology and project success in different project governance contexts". In: *International Journal of Project Management* 33 (Apr. 2015). DOI: 10.1016/j.ijproman.2015.03.005 (cit. on p. 41).

[60] K. Papke-Shields, C. Beise, and Q. Jing. "Do project managers practice what they preach, and does it matter to project success?" In: *International Journal of Project Management - INT J PROJ MANAG* 28 (Oct. 2010), pp. 650–662. DOI: 10.1016/j.ijproman.2009.11.002 (cit. on p. 41).

[61] H. Taherdoost and A. Keshavarzsaleh. "Critical Factors that Lead to Projects' Success/Failure in Global Marketplace". In: *Procedia Technology* 22 (2016). 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania, pp. 1066–1075. ISSN: 2212-0173. DOI: https://doi.org/10.1016/j.protcy.2016.01.151. URL: https://www.sciencedirect.com/science/article/pii/S2212017316001523 (cit. on pp. 41–43).

[62] L. A. Ika. "Project success as a topic in project management journals". In: *Project Management Journal* 40.4 (2009), pp. 6–19. DOI: https://doi.org/10.1002/pmj.20137. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pmj.20137. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/pmj.20137 (cit. on p. 41).

[63] J. Pinto and S. Mantel. "The Causes of Project Failure". English (US). In: *IEEE Transactions on Engineering Management* 37.4 (Nov. 1990), pp. 269–276. ISSN: 0018-9391. DOI: 10.1109/17.62322 (cit. on p. 41).

[64] A. de Wit. "Measurement of project success". In: *International Journal of Project Management* 6.3 (1988), pp. 164–170. ISSN: 0263-7863. DOI: https://doi.org/10.1016/0263-7863(88)90043-9. URL: https://www.sciencedirect.com/science/article/pii/0263786388900439 (cit. on p. 42).

[65] M. Radujković and M. Sjekavica. "Project Management Success Factors". In: *Procedia Engineering* 196 (2017). Creative Construction Conference 2017, CCC 2017, 19-22 June 2017, Primosten, Croatia, pp. 607–615. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2017.08.048. URL: https://www.sciencedirect.com/science/article/pii/S1877705817331740 (cit. on p. 42).

[66] *Project Management Success Factors: a Bibliometric Analysis*. URL: https://dialnet.unirioja.es/descarga/articulo/5115215.pdf (visited on 09/29/2022) (cit. on p. 42).

[67] J. Thomas and M. Mullaly. "Understanding the Value of Project Management: First Steps on an International Investigation in Search of Value". In: *Project Management Journal* 38 (Sept. 2007), pp. 74–89. DOI: 10.1002/pmj.20007 (cit. on p. 42).

[68] I. HYVÄRI. "Success of Projects in Different Organizational Conditions". In: *Project Management Journal* 37 (Sept. 2006). DOI: 10.1177/875697280603700404 (cit. on p. 43).

[69] A. N. dos Tecnicos de Emergência Pré-Hospitalar. *ANTEPH*. URL: https://www.anteph.pt/ (visited on 02/20/2022) (cit. on p. 45).

[70] R. Hartson and P. Pyla. "Chapter 4 - Agile Lifecycle Processes and the Funnel Model of Agile UX". In: *The UX Book (Second Edition)*. Ed. by R. Hartson and P. Pyla. Second Edition. Boston: Morgan Kaufmann, 2019, pp. 63–80. ISBN: 978-0-12-805342-3. DOI: https://doi.org/10.1016/B978-0-12-805342-3.00004-7. URL: https://www.sciencedirect.com/science/article/pii/B9780128053423000047 (cit. on p. 46).

[71] R. Sherman. "Chapter 18 - Project Management". In: *Business Intelligence Guidebook*. Ed. by R. Sherman. Boston: Morgan Kaufmann, 2015, pp. 449–492. ISBN: 978-0-12-411461-6. DOI: https://doi.org/10.1016/B978-0-12-411461-6.00018-6. URL: https://www.sciencedirect.com/science/article/pii/B9780124114616000186 (cit. on pp. 46, 47, 58).

[72] *Analysis of Sofware Development Methodologies*. URL: https://expert.taylors.edu.my/file/rems/publication/109566_6142_2.pdf (visited on 09/30/2022) (cit. on p. 47).

[73] P. S. Ganney, S. Pisharody, and E. Claridge. "Chapter 9 - Software engineering". In: *Clinical Engineering (Second Edition)*. Ed. by A. Taktak et al. Second Edition. Academic Press, 2020, pp. 131–168. ISBN: 978-0-08-102694-6. DOI: https://doi.org/10.1016/B978-0-08-102694-6.00009-7. URL: https://www.sciencedirect.com/science/article/pii/B9780081026946000097 (cit. on p. 47).

[74] E. Knauss and C. E. Boustani. "Assessing the Quality of Software Requirements Specifications". In: *2008 16th IEEE International Requirements Engineering Conference*. 2008, pp. 341–342. DOI: 10.1109/RE.2008.29 (cit. on p. 49).

[75] J. Mund et al. "Does Quality of Requirements Specifications Matter? Combined Results of Two Empirical Studies". In: *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 2015, pp. 1–10. DOI: 10.1109/ESEM.2015.7321195 (cit. on p. 49).

[76] E. Knauss, C. Boustani, and T. Flohr. "Investigating the Impact of Software Requirements Specification Quality on Project Success". In: vol. 32. Jan. 2009, pp. 28–42. ISBN: 978-3-642-02151-0. DOI: 10.1007/978-3-642-02152-7_4 (cit. on p. 49).

[77] "ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering". In: *ISO/IEC/IEEE 29148:2018(E)* (2018), pp. 1–104. DOI: 10.1109/IEEESTD.2018.8559686 (cit. on pp. 50–52, 55).

[78] "IEEE Standard Glossary of Software Engineering Terminology". In: *IEEE Std 610.12-1990* (1990), pp. 1–84. DOI: 10.1109/IEEESTD.1990.101064 (cit. on p. 52).

[79] J. Nielsen. *Nielsen Norman Group - Usability 101: Introduction to Usability.* URL: https://www.nngroup.com/articles/usability-101-introduction-to-usability/ (visited on 10/14/2022) (cit. on p. 55).

[80] J. Nielsen. *Nielsen Norman Group - Response Times: The 3 Important Limits.* URL: https://www.nngroup.com/articles/response-times-3-important-limits/ (visited on 10/14/2022) (cit. on p. 56).

[81] *Nginx.* URL: https://www.nginx.com/ (visited on 10/12/2022) (cit. on p. 56).

[82] *Gunicorn.* URL: https://gunicorn.org/ (visited on 10/12/2022) (cit. on p. 56).

[83] INEM. *Carteira de Serviços do INEM.* (visited on 2022-02-22). URL: %5Curl%7Bhttps://www.inem.pt/wp-content/uploads/2019/12/Carteira-de-Servi%C3%A7os-do-INEM.pdf%7D (cit. on p. 60).

[84] *Balsamiq.* URL: https://balsamiq.com/ (visited on 10/01/2022) (cit. on p. 62).

[85] *Django REST framework - Authentication.* URL: https://www.django-rest-framework.org/api-guide/authentication/ (visited on 10/18/2022) (cit. on p. 68).

[86] *Django REST framework - Permissions.* URL: https://www.django-rest-framework.org/api-guide/permissions/ (visited on 10/18/2022) (cit. on p. 68).

[87] *HubSpot - What Is an API Endpoint?* URL: https://blog.hubspot.com/website/api-endpoint (visited on 10/15/2022) (cit. on p. 69).

[88] U. N. H. Rights. *Istanbul Protocol Professional Training Series No. 8/Rev. 2 - Manual on the Effective Investigation and Documentation of Torture and Other Cruel, Inhuman or Degrading Treatment or Punishment.* Version 2.0. 2022. URL: https://www.ohchr.org/sites/default/files/documents/publications/2022-06-29/Istanbul-Protocol_Rev2_EN.pdf (visited on 10/20/2022) (cit. on p. 77).

[89] D. G. de Saúde. *Via Verde Sepsis no Adulto.* Version 2.0. 2017. URL: https://normas.dgs.min-saude.pt/wp-content/uploads/2019/09/Via-Verde-Sepsis-no-Adulto.pdf (visited on 10/20/2022) (cit. on p. 78).

[90] M. Mayeda and A. Andrews. "Chapter Two - Evaluating software testing techniques: A systematic mapping study". In: ed. by A. R. Hurson. Vol. 123. Advances in Computers. Elsevier, 2021, pp. 41–114. DOI: https://doi.org/10.1016/bs.adcom.2021.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0065245821000279 (cit. on p. 79).

[91] K. Herzig. "There's never enough time to do all the testing you want". In: *Perspectives on Data Science for Software Engineering.* Ed. by T. Menzies, L. Williams, and T. Zimmermann. Boston: Morgan Kaufmann, 2016, pp. 91–95. ISBN: 978-0-12-804206-9. DOI: https://doi.org/10.1016/B978-0-12-804206-9.00018-0. URL: https://www.sciencedirect.com/science/article/pii/B9780128042069000180 (cit. on p. 79).

[92] *dotcom-tools - Web Server Test.* URL: https://www.dotcom-tools.com/web-servers-test (visited on 11/02/2022) (cit. on p. 105).

[93] *Django Channels.* URL: https://channels.readthedocs.io/en/stable/ (visited on 10/25/2022) (cit. on p. 106).

Appendix A

SIREPH Data Model Diagram

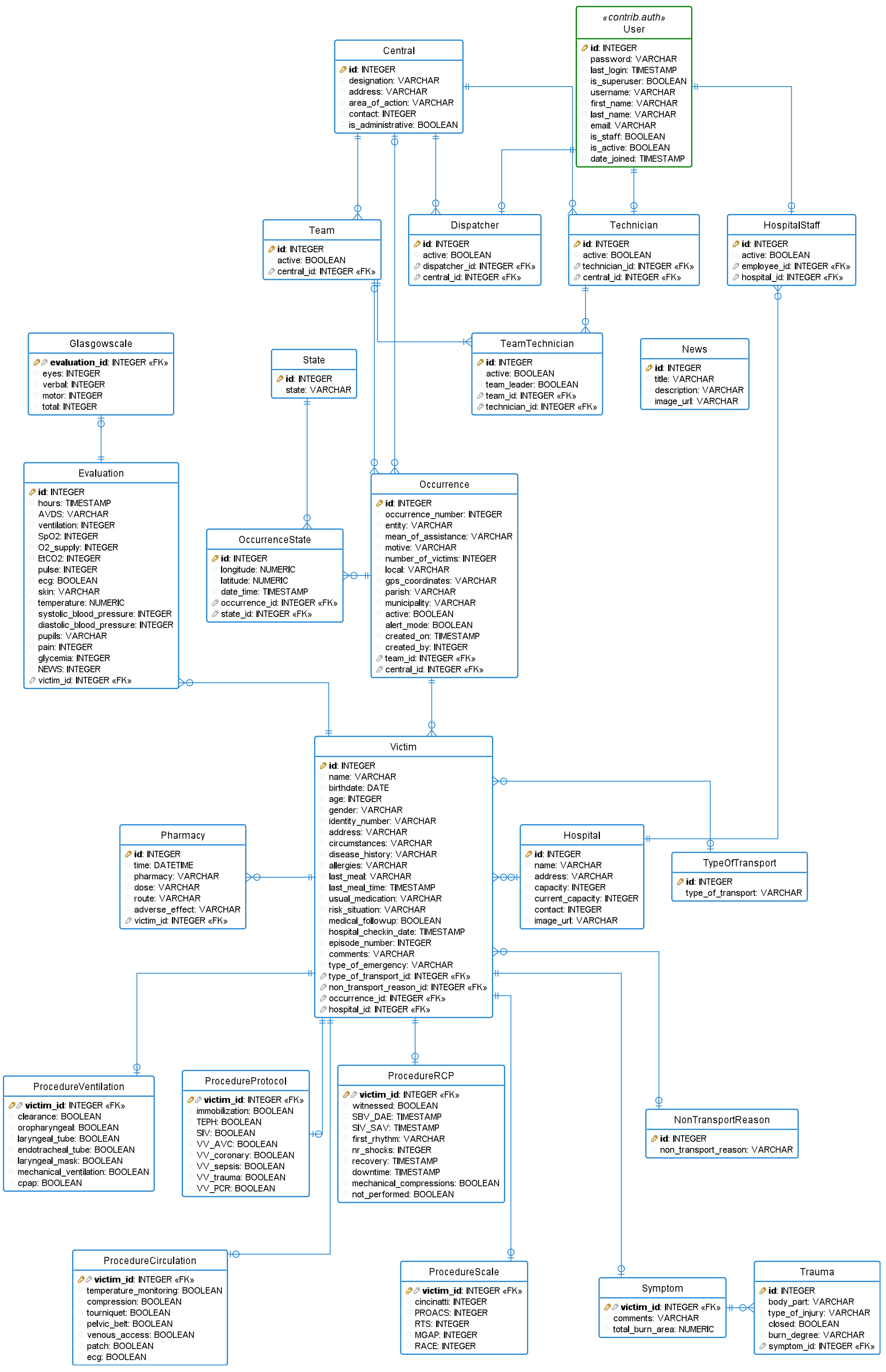
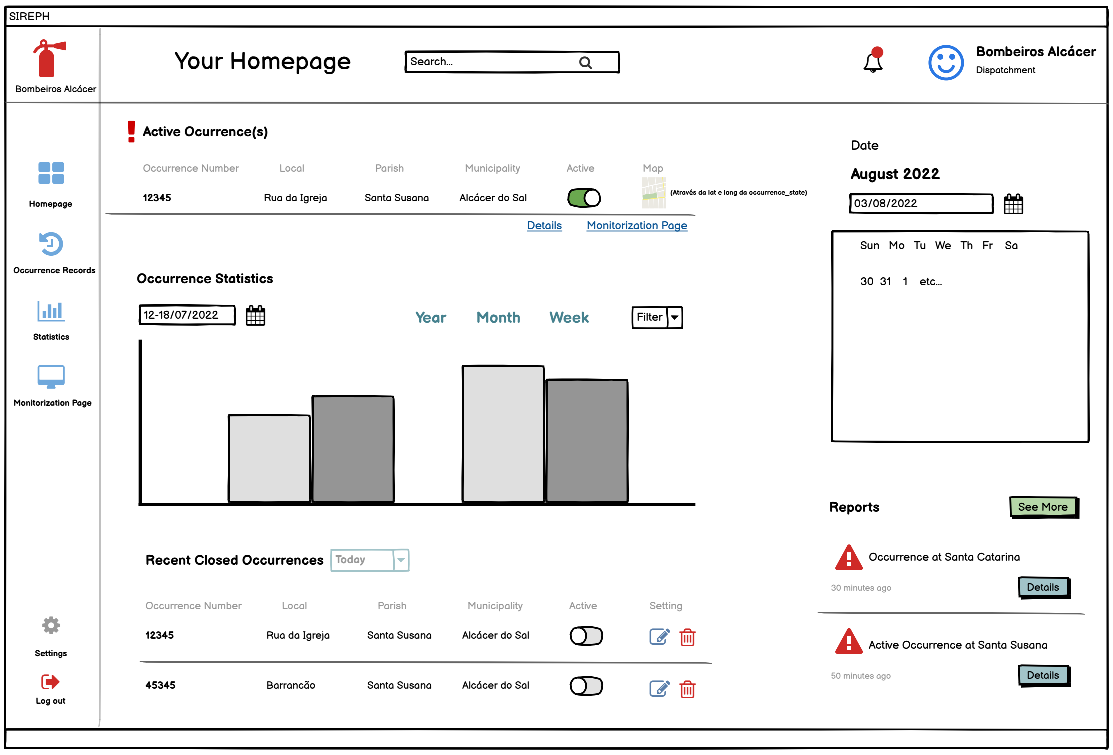


Figure A.1: SIREPH Data Model Diagram

Appendix B

Emergency Central Subsystem User Interface Mockups



Active Ocurrence(s)

Occurrence Number	Local	Parish	Municipality	Active	Map
12345	Rua da Igreja	Santa Susana	Alcácer do Sal		(Através da lat e long da occurrence_state)

Occurrence Statistics

12-18/07/2022

Date

August 2022

03/08/2022

Sun	Mo	Tu	We	Th	Fr	Sa
30	31	1	etc...			

Recent Closed Occurrences Today

Occurrence Number	Local	Parish	Municipality	Active	Setting
12345	Rua da Igreja	Santa Susana	Alcácer do Sal		
45345	Barrancão	Santa Susana	Alcácer do Sal		

Reports

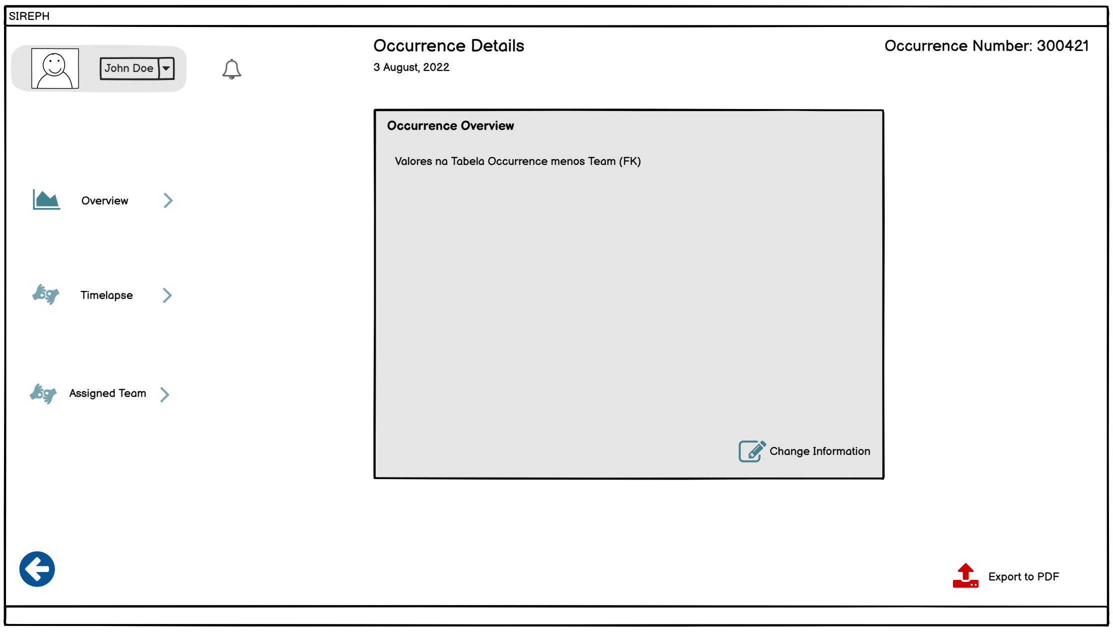
Occurrence at Santa Catarina

30 minutes ago

Active Occurrence at Santa Susana

50 minutes ago

Figure B.1: Emergency Central Subsystem Homepage Mockup



Occurrence Details

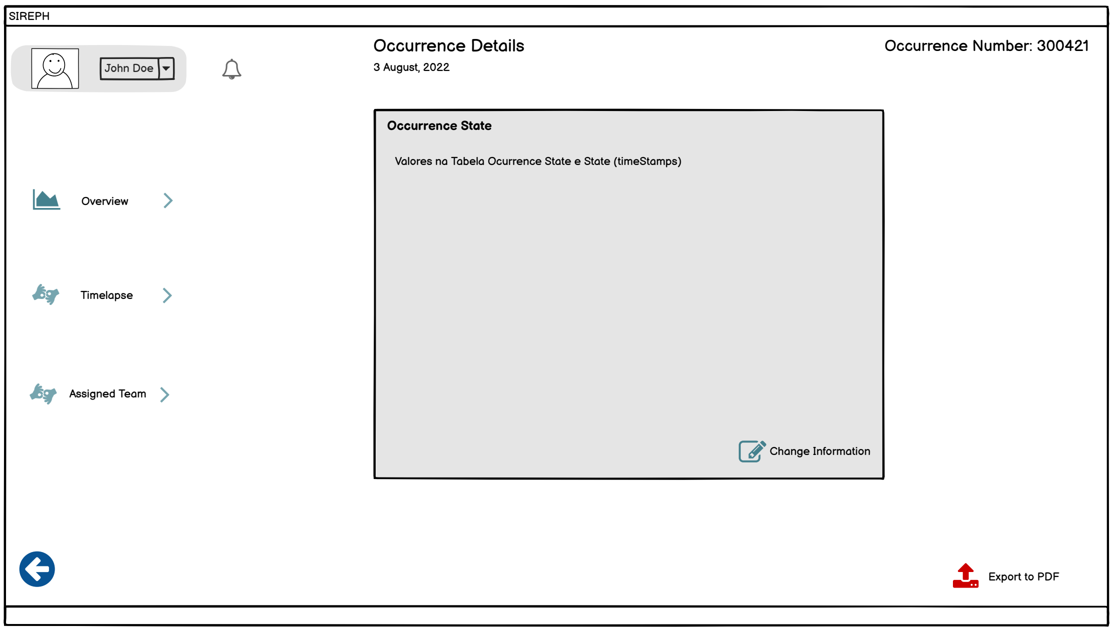
3 August, 2022

Occurrence Number: 300421

Occurrence Overview

Valores na Tabela Occurrence menos Team (FK)

Figure B.2: Emergency Central Subsystem Occurrence Detail Overview Page Mockup



Occurrence Details

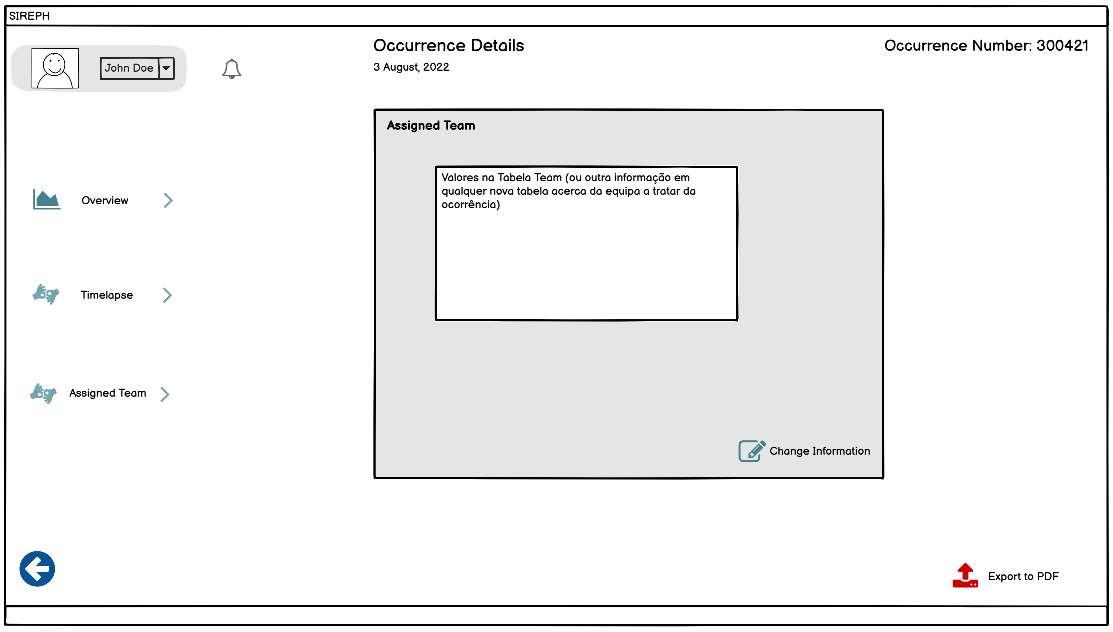
3 August, 2022

Occurrence Number: 300421

Occurrence State

Valores na Tabela Ocurrence State e State (timeStamps)

Figure B.3: Emergency Central Subsystem Occurrence Detail State Page Mockup



Occurrence Details

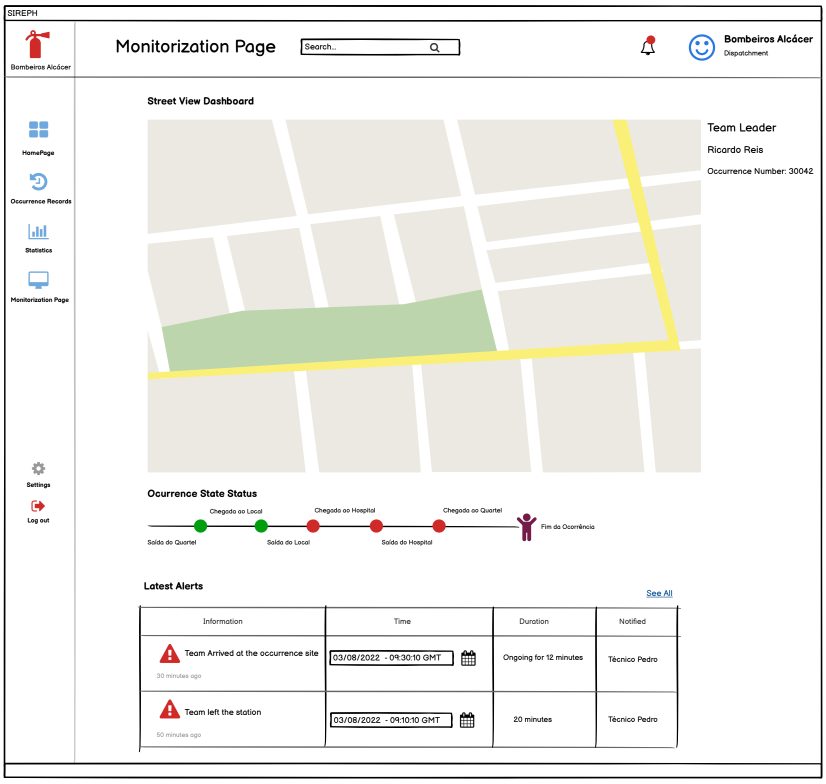
3 August, 2022

Occurrence Number: 300421

Assigned Team

Valores na Tabela Team (ou outra informação em qualquer nova tabela acerca da equipa a tratar da ocorrência)

Figure B.4: Emergency Central Subsystem Occurrence Detail Team Page Mockup



Monitorization Page

Bombeiros Alcácer
Dispatchment

Street View Dashboard

Team Leader

Ricardo Reis

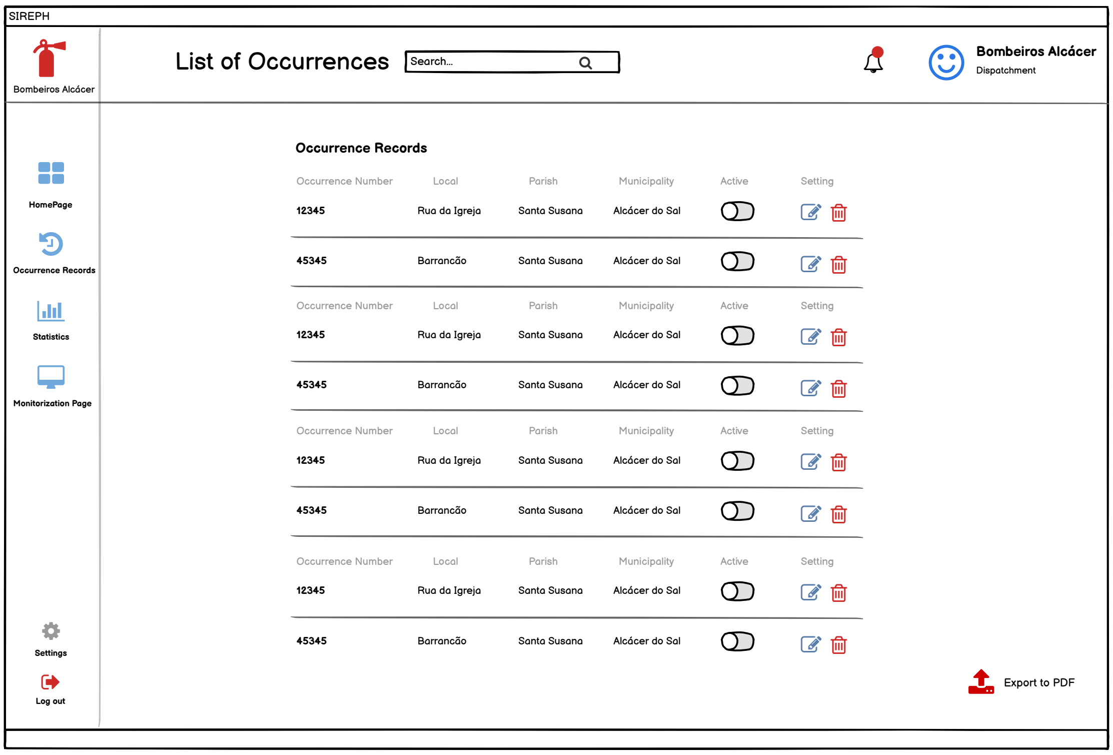
Occurrence Number: 30042

Ocurrence State Status

Latest Alerts

Information	Time	Duration	Notified
Team Arrived at the occurrence site 30 minutes ago	03/08/2022 - 09:30:10 GMT	Ongoing for 12 minutes	Técnico Pedro
Team left the station 50 minutes ago	03/08/2022 - 09:10:10 GMT	20 minutes	Técnico Pedro

Figure B.5: Emergency Central Subsystem Monitorization Page Mockup



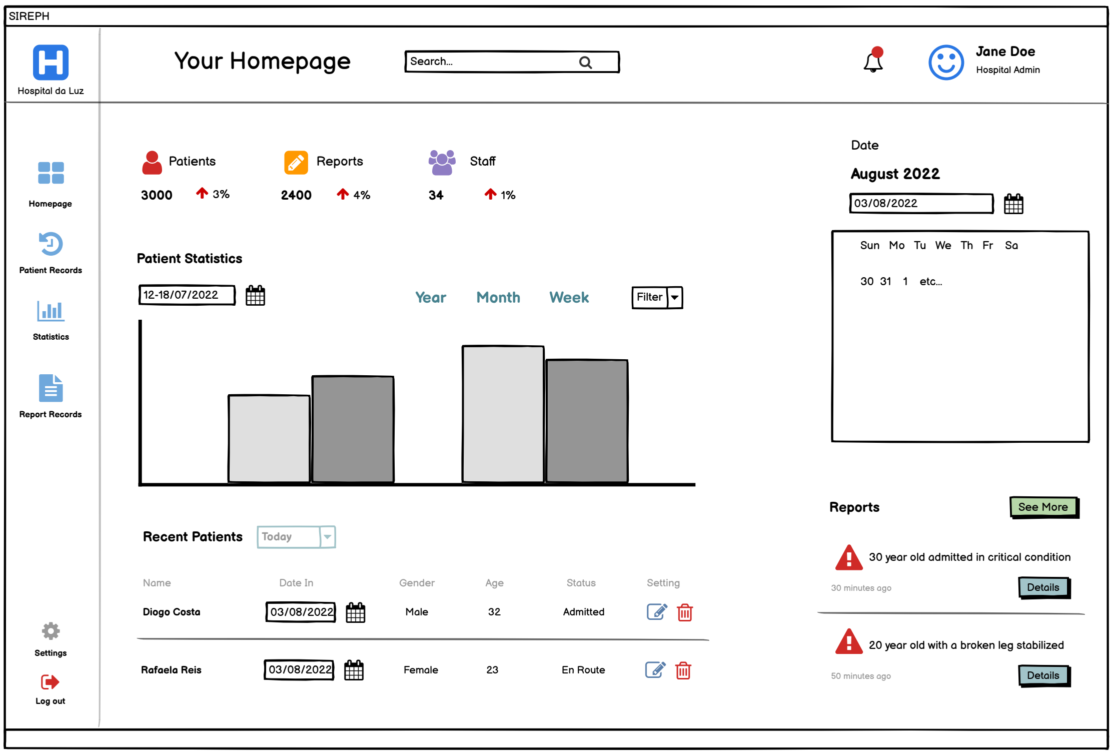
Occurrence Records

Occurrence Number	Local	Parish	Municipality	Active	Setting
12345	Rua da Igreja	Santa Susana	Alcácer do Sal		
45345	Barrancão	Santa Susana	Alcácer do Sal		
Occurrence Number	Local	Parish	Municipality	Active	Setting
12345	Rua da Igreja	Santa Susana	Alcácer do Sal		
45345	Barrancão	Santa Susana	Alcácer do Sal		
Occurrence Number	Local	Parish	Municipality	Active	Setting
12345	Rua da Igreja	Santa Susana	Alcácer do Sal		
45345	Barrancão	Santa Susana	Alcácer do Sal		
Occurrence Number	Local	Parish	Municipality	Active	Setting
12345	Rua da Igreja	Santa Susana	Alcácer do Sal		
45345	Barrancão	Santa Susana	Alcácer do Sal		

Figure B.6: Emergency Central Subsystem Occurrence History Page Mockup

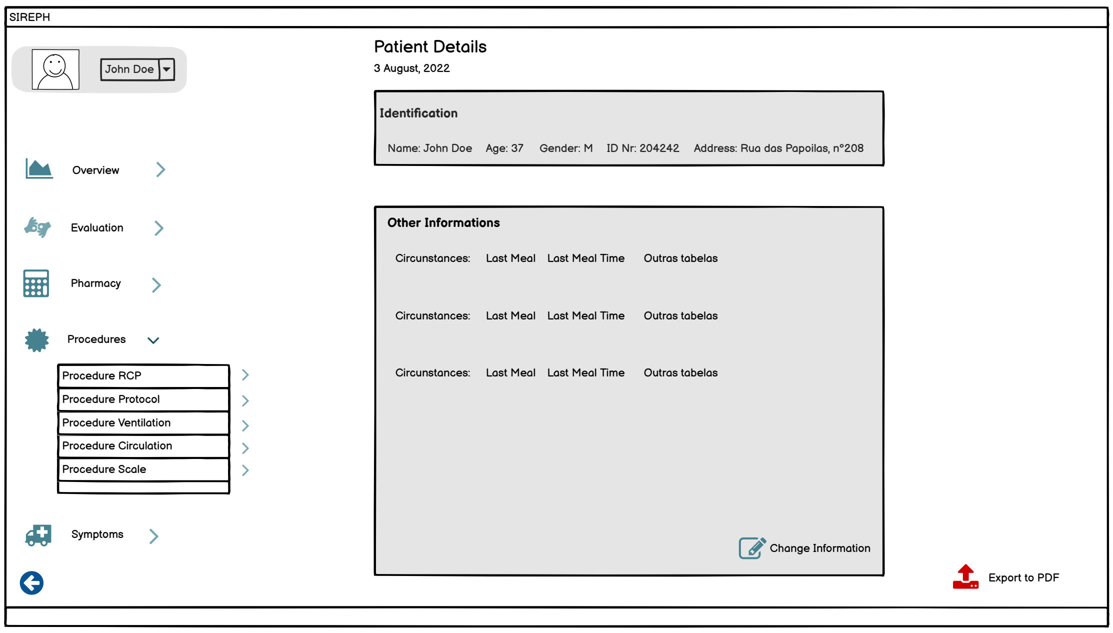
Appendix C

Hospital Subsystem User Interface Mockups



Name	Date In	Gender	Age	Status	Setting
Diogo Costa	03/08/2022	Male	32	Admitted	
Rafaela Reis	03/08/2022	Female	23	En Route	

Figure C.1: Hospital Subsystem Homepage Mockup



Patient Details

3 August, 2022

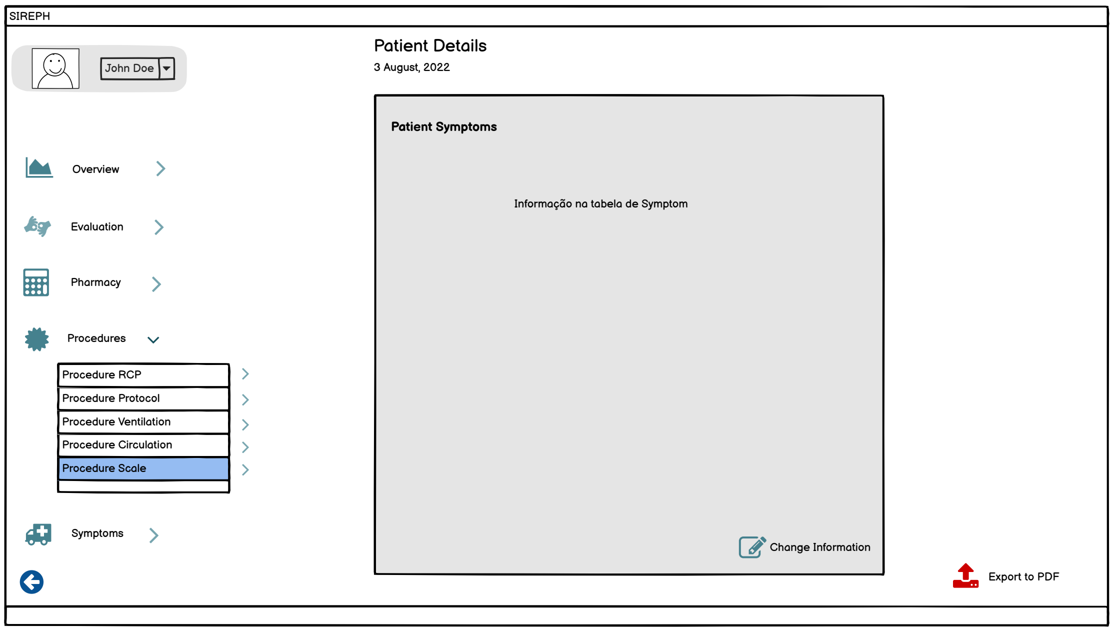
Identification

Name: John Doe Age: 37 Gender: M ID Nr: 204242 Address: Rua das Papoilas, nº208

Other Informations

Circunstances:	Last Meal	Last Meal Time	Outras tabelas
Circunstances:	Last Meal	Last Meal Time	Outras tabelas
Circunstances:	Last Meal	Last Meal Time	Outras tabelas

Figure C.2: Hospital Subsystem Patient Detail Information Page Mockup



Patient Details

3 August, 2022

Patient Symptoms

Informação na tabela de Symptom

Figure C.3: Hospital Subsystem Patient Detail Symptoms Page Mockup

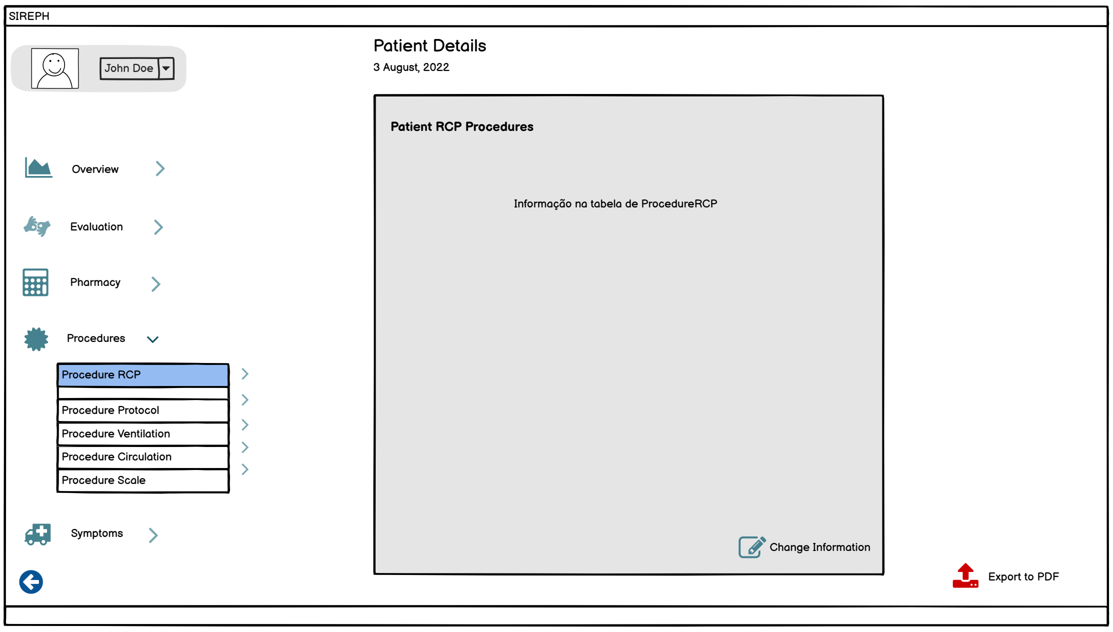


Figure C.4: Hospital Subsystem Patient Detail RCP Procedures Page Mockup

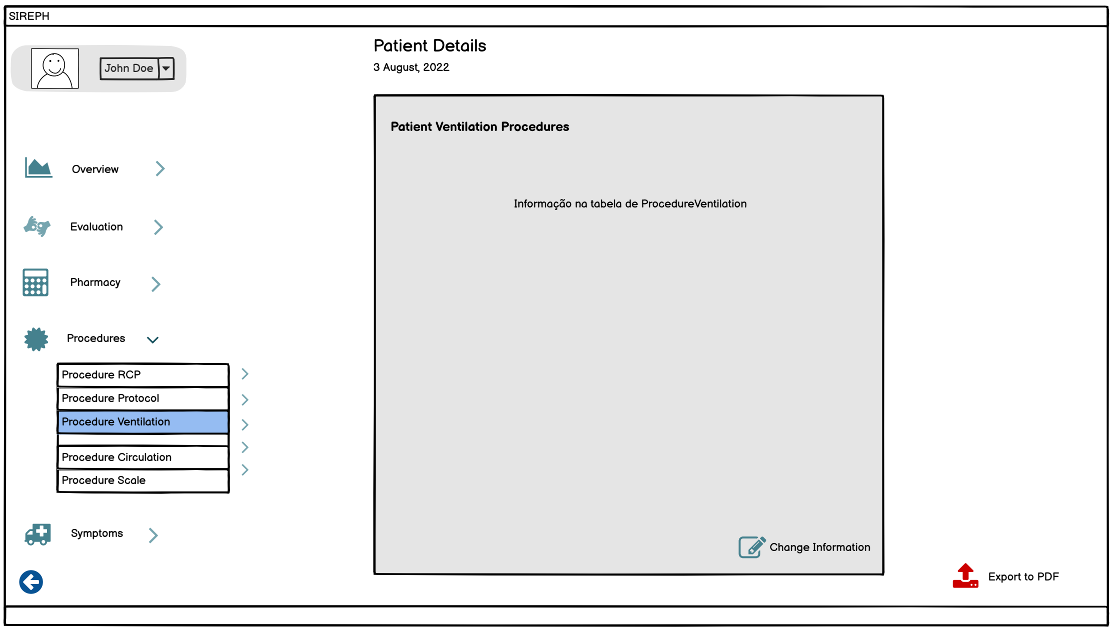
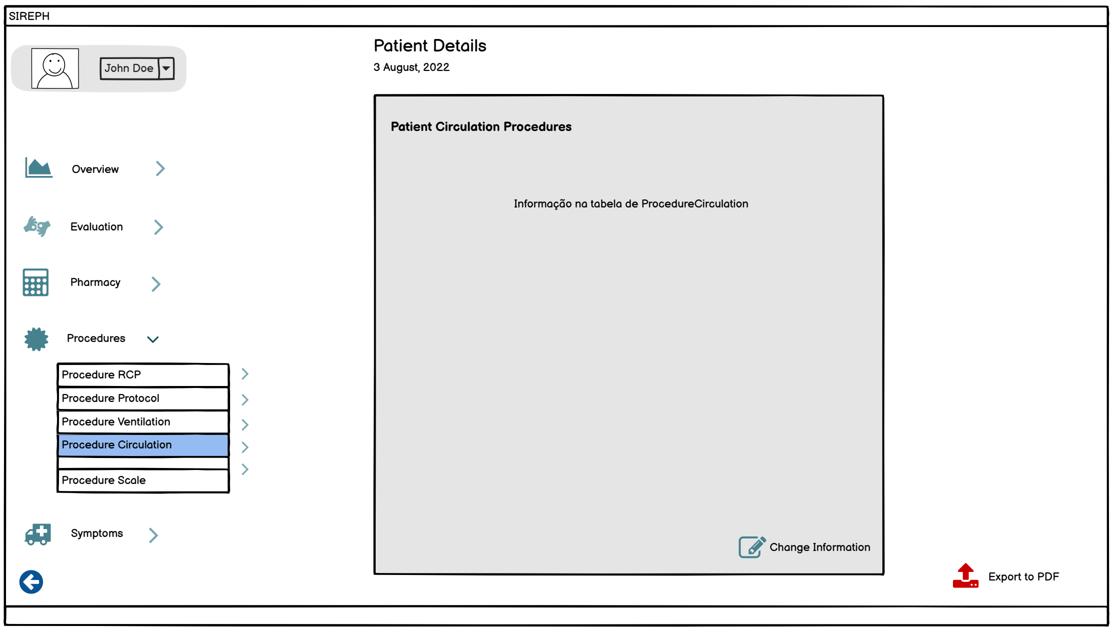


Figure C.5: Hospital Subsystem Patient Detail Ventilation Procedures Page Mockup



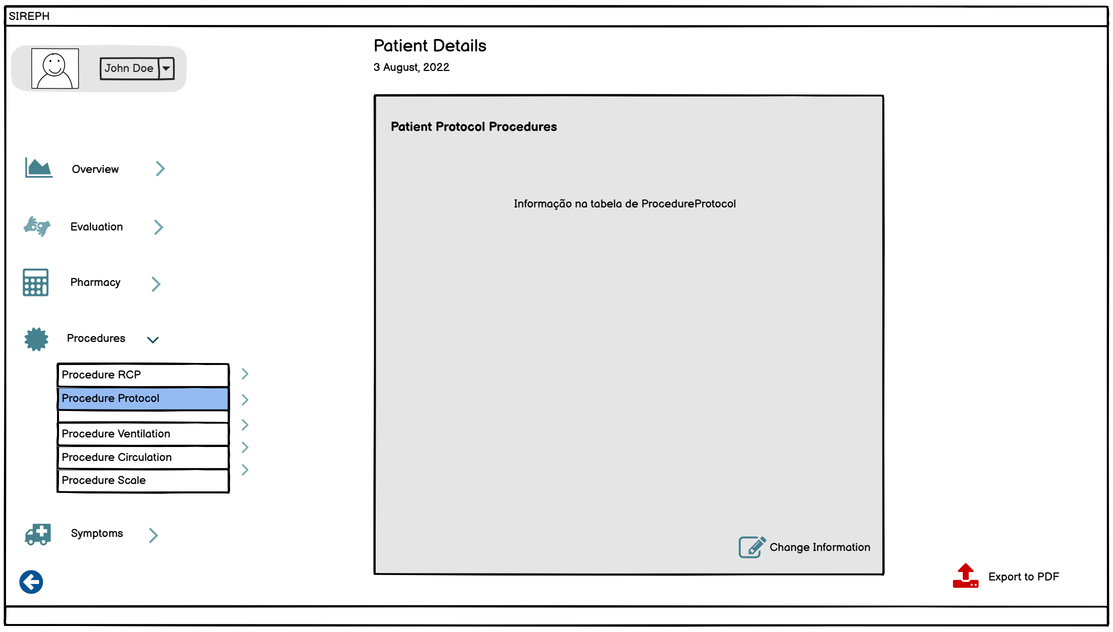
Patient Details

3 August, 2022

Patient Circulation Procedures

Informação na tabela de ProcedureCirculation

Figure C.6: Hospital Subsystem Patient Detail Circulation Procedures Page Mockup



Patient Details

3 August, 2022

Patient Protocol Procedures

Informação na tabela de ProcedureProtocol

Figure C.7: Hospital Subsystem Patient Detail Protocol Procedures Page Mockup

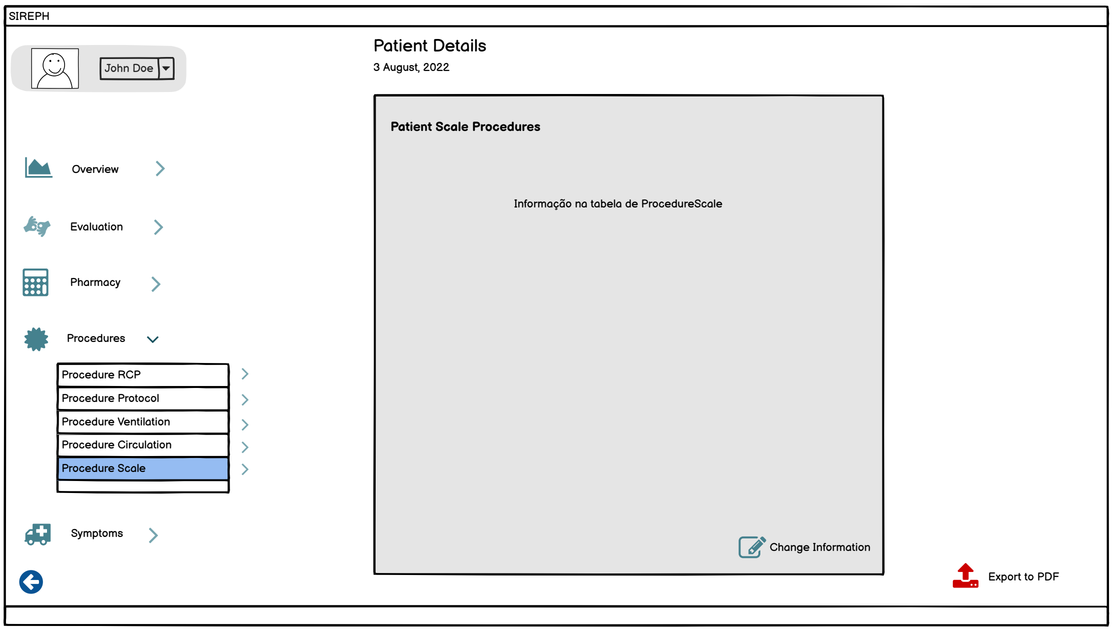


Figure C.8: Hospital Subsystem Patient Detail Scale Procedures Page Mockup

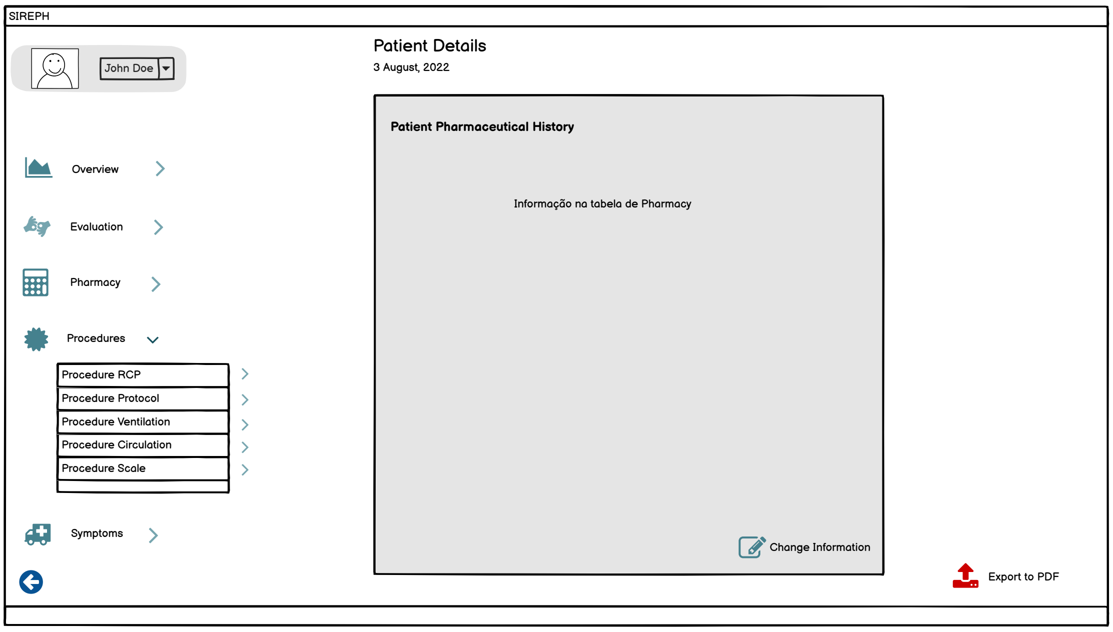
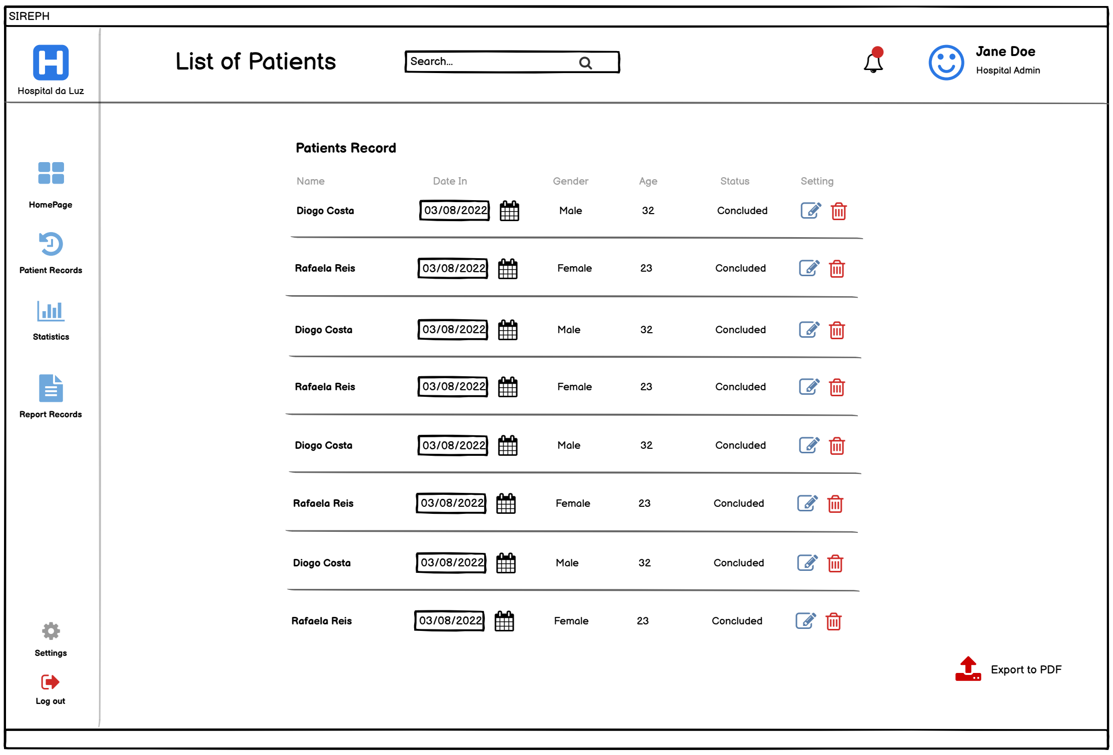


Figure C.9: Hospital Subsystem Patient Detail Pharmacy Page Mockup



Patients Record

Name	Date In	Gender	Age	Status	Setting
Diogo Costa	03/08/2022	Male	32	Concluded	
Rafaela Reis	03/08/2022	Female	23	Concluded	
Diogo Costa	03/08/2022	Male	32	Concluded	
Rafaela Reis	03/08/2022	Female	23	Concluded	
Diogo Costa	03/08/2022	Male	32	Concluded	
Rafaela Reis	03/08/2022	Female	23	Concluded	
Diogo Costa	03/08/2022	Male	32	Concluded	
Rafaela Reis	03/08/2022	Female	23	Concluded	

Figure C.10: Hospital Subsystem Patient History Page Mockup

Appendix D

Emergency Central Subsystem Definitive User Interface Pages

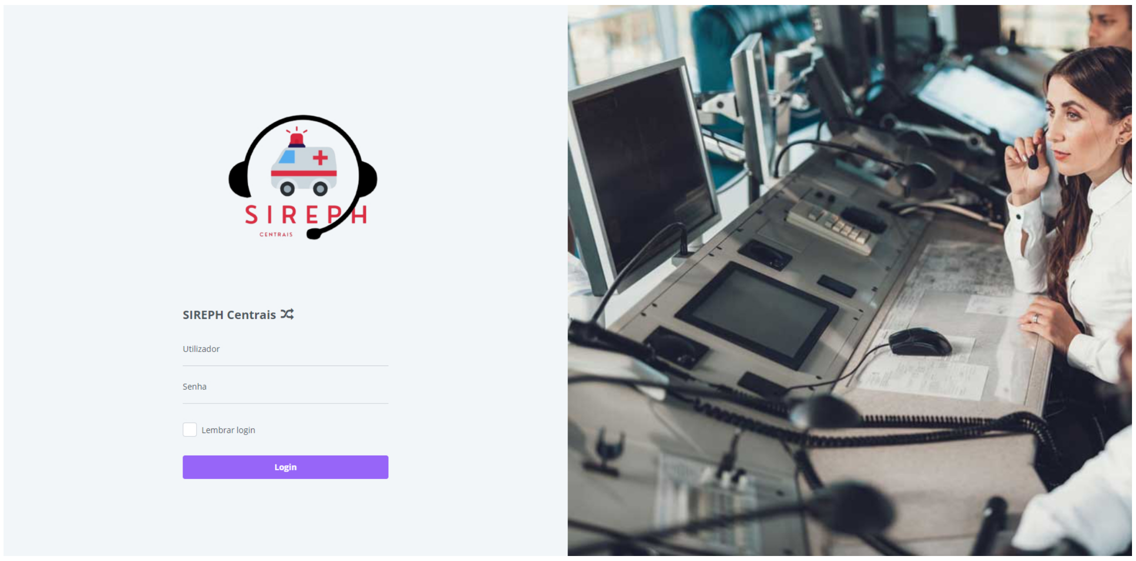
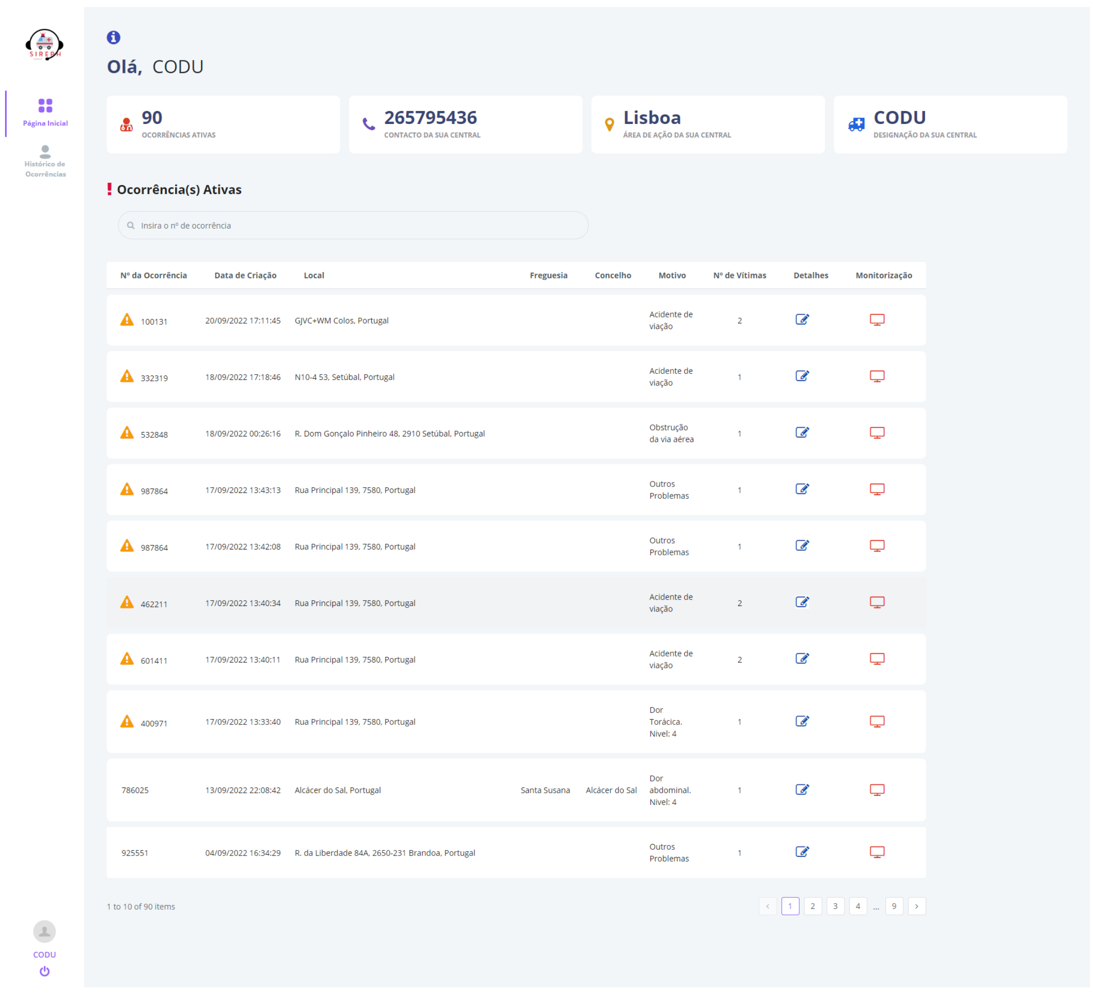


Figure D.1: Emergency Central Subsystem Login Page



Olá, CODU

90 OCORRÊNCIAS ATIVAS

265795436 CONTACTO DA SUA CENTRAL

Lisboa ÁREA DE AÇÃO DA SUA CENTRAL

CODU DESIGNAÇÃO DA SUA CENTRAL

! Ocorrência(s) Ativas

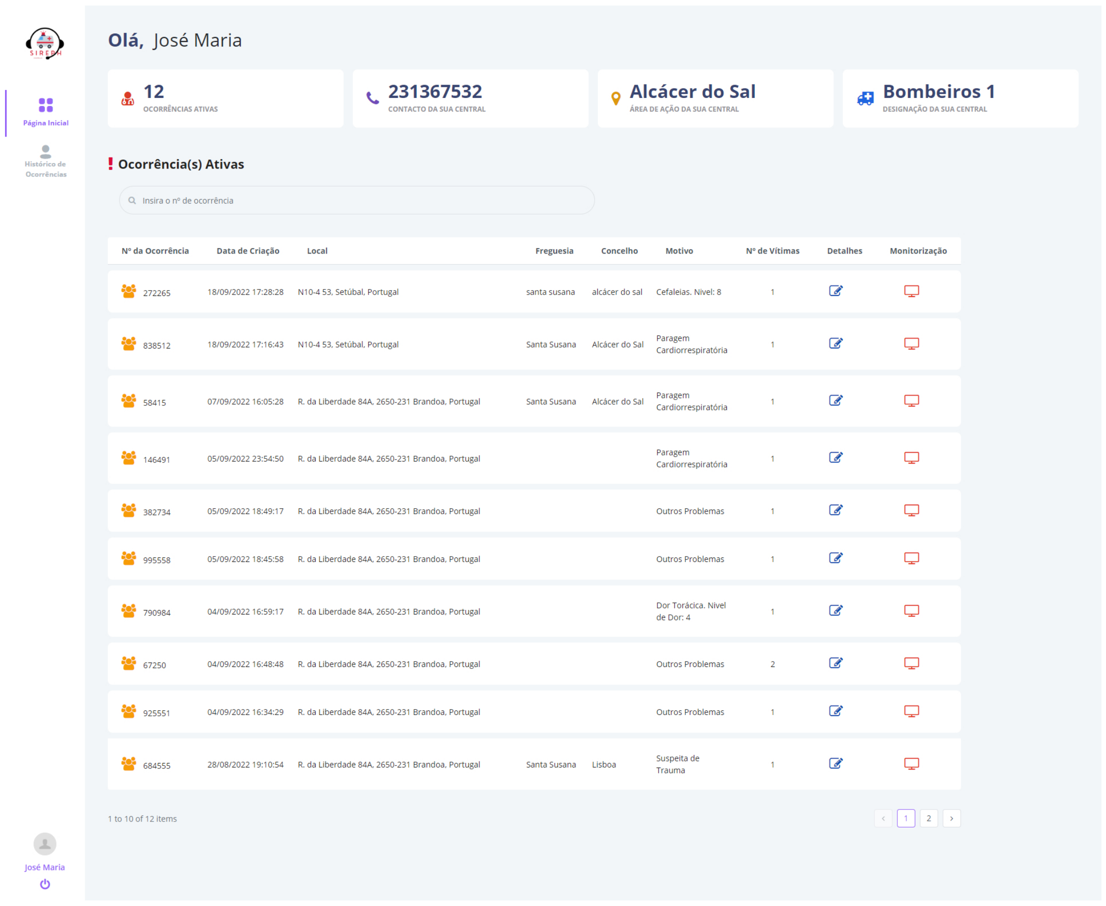
Insira o nº de ocorrência

Nº da Ocorrência	Data de Criação	Local	Freguesia	Concelho	Motivo	Nº de Vítimas	Detalhes	Monitorização
⚠ 100131	20/09/2022 17:11:45	GJVC+WM Colos, Portugal			Acidente de viação	2		
⚠ 332319	18/09/2022 17:18:46	N10-4 53, Setúbal, Portugal			Acidente de viação	1		
⚠ 532848	18/09/2022 00:26:16	R. Dom Gonçalo Pinheiro 48, 2910 Setúbal, Portugal			Obstrução da via aérea	1		
⚠ 987864	17/09/2022 13:43:13	Rua Principal 139, 7580, Portugal			Outros Problemas	1		
⚠ 987864	17/09/2022 13:42:08	Rua Principal 139, 7580, Portugal			Outros Problemas	1		
⚠ 462211	17/09/2022 13:40:34	Rua Principal 139, 7580, Portugal			Acidente de viação	2		
⚠ 601411	17/09/2022 13:40:11	Rua Principal 139, 7580, Portugal			Acidente de viação	2		
⚠ 400971	17/09/2022 13:33:40	Rua Principal 139, 7580, Portugal			Dor Torácica. Nivel: 4	1		
786025	13/09/2022 22:08:42	Alcácer do Sal, Portugal	Santa Susana	Alcácer do Sal	Dor abdominal. Nivel: 4	1		
925551	04/09/2022 16:34:29	R. da Liberdade 84A, 2650-231 Brandoa, Portugal			Outros Problemas	1		

1 to 10 of 90 items

‹ 1 2 3 4 ... 9 ›

Figure D.2: Emergency Central Subsystem Administrative Central Homepage



Olá, José Maria

12 OCORRÊNCIAS ATIVAS	231367532 CONTACTO DA SUA CENTRAL	Alcácer do Sal ÁREA DE AÇÃO DA SUA CENTRAL	Bombeiros 1 DESIGNAÇÃO DA SUA CENTRAL

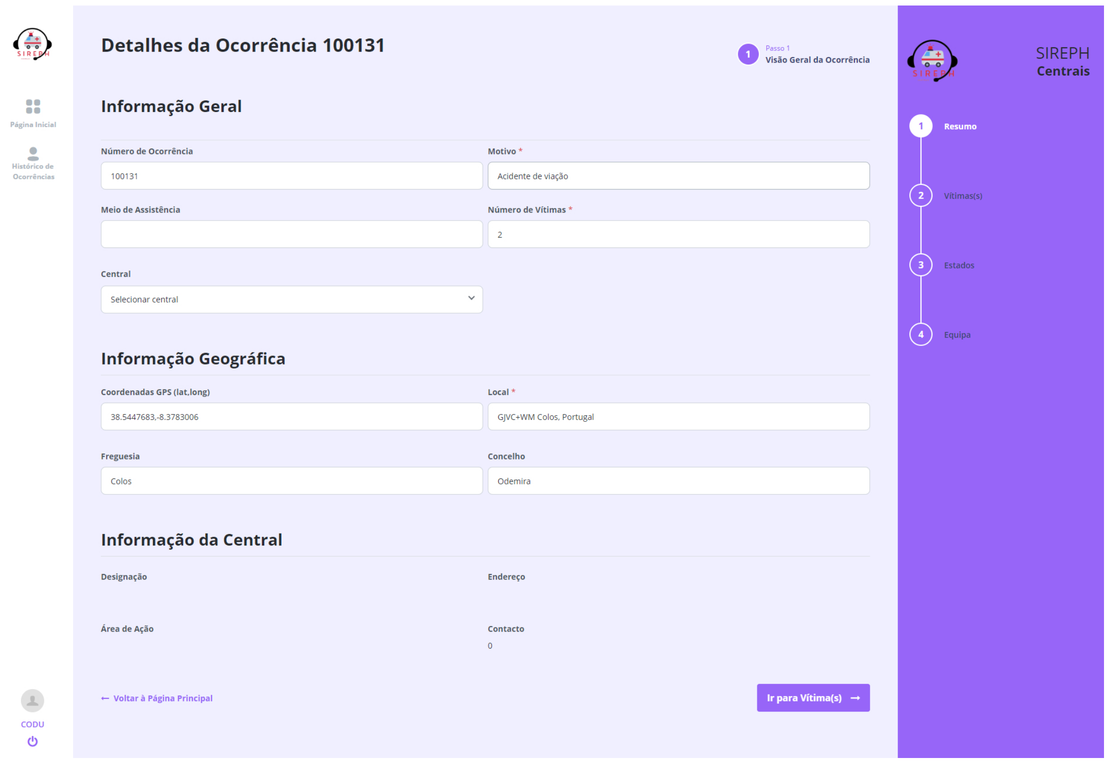
! Ocorrência(s) Ativas

Insira o nº de ocorrência

Nº da Ocorrência	Data de Criação	Local	Freguesia	Concelho	Motivo	Nº de Vítimas	Detalhes	Monitorização
272265	18/09/2022 17:28:28	N10-4 53, Setúbal, Portugal	santa susana	alcácer do sal	Cefaleias. Nivel: 8	1	✎	🖥
838512	18/09/2022 17:16:43	N10-4 53, Setúbal, Portugal	Santa Susana	Alcácer do Sal	Paragem Cardiorrespiratória	1	✎	🖥
58415	07/09/2022 16:05:28	R. da Liberdade 84A, 2650-231 Brandoa, Portugal	Santa Susana	Alcácer do Sal	Paragem Cardiorrespiratória	1	✎	🖥
146491	05/09/2022 23:54:50	R. da Liberdade 84A, 2650-231 Brandoa, Portugal			Paragem Cardiorrespiratória	1	✎	🖥
382734	05/09/2022 18:49:17	R. da Liberdade 84A, 2650-231 Brandoa, Portugal			Outros Problemas	1	✎	🖥
995558	05/09/2022 18:45:58	R. da Liberdade 84A, 2650-231 Brandoa, Portugal			Outros Problemas	1	✎	🖥
790984	04/09/2022 16:59:17	R. da Liberdade 84A, 2650-231 Brandoa, Portugal			Dor Torácica. Nivel de Dor: 4	1	✎	🖥
67250	04/09/2022 16:48:48	R. da Liberdade 84A, 2650-231 Brandoa, Portugal			Outros Problemas	2	✎	🖥
925551	04/09/2022 16:34:29	R. da Liberdade 84A, 2650-231 Brandoa, Portugal			Outros Problemas	1	✎	🖥
684555	28/08/2022 19:10:54	R. da Liberdade 84A, 2650-231 Brandoa, Portugal	Santa Susana	Lisboa	Suspeita de Trauma	1	✎	🖥

1 to 10 of 12 items

Figure D.3: Emergency Central Subsystem Non-Administrative Central Homepage



Detalhes da Ocorrência 100131

Informação Geral

Número de Ocorrência: 100131	Motivo *: Acidente de viação
Meio de Assistência:	Número de Vítimas *: 2
Central: Selecionar central	

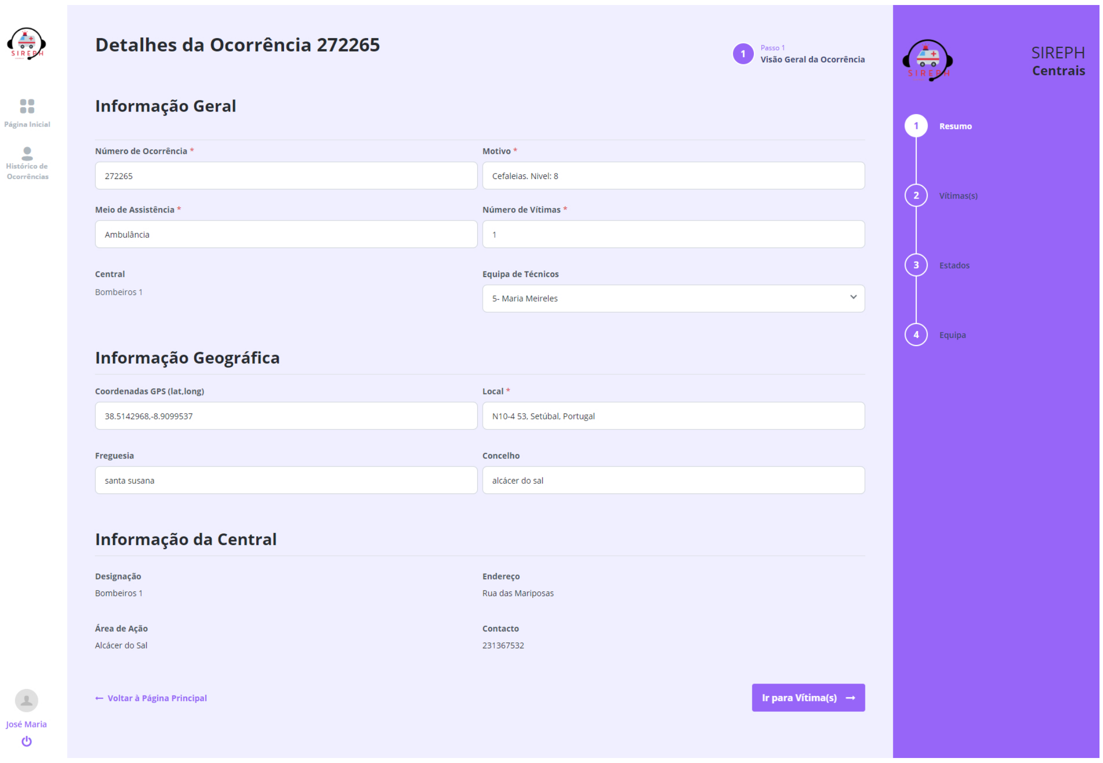
Informação Geográfica

Coordenadas GPS (lat,long): 38.5447683,-8.3783006	Local *: GJVC+WM Colos, Portugal
Freguesia: Colos	Concelho: Odemira

Informação da Central

Designação:	Endereço:
Área de Ação:	Contacto: 0

Figure D.4: Emergency Central Subsystem Administrative Central Information Detail Page



Detalhes da Ocorrência 272265

Informação Geral

Número de Ocorrência *	Motivo *
272265	Cefaleias, Nível: 8
Meio de Assistência *	Número de Vítimas *
Ambulância	1
Central	Equipa de Técnicos
Bombeiros 1	5- Maria Meireles

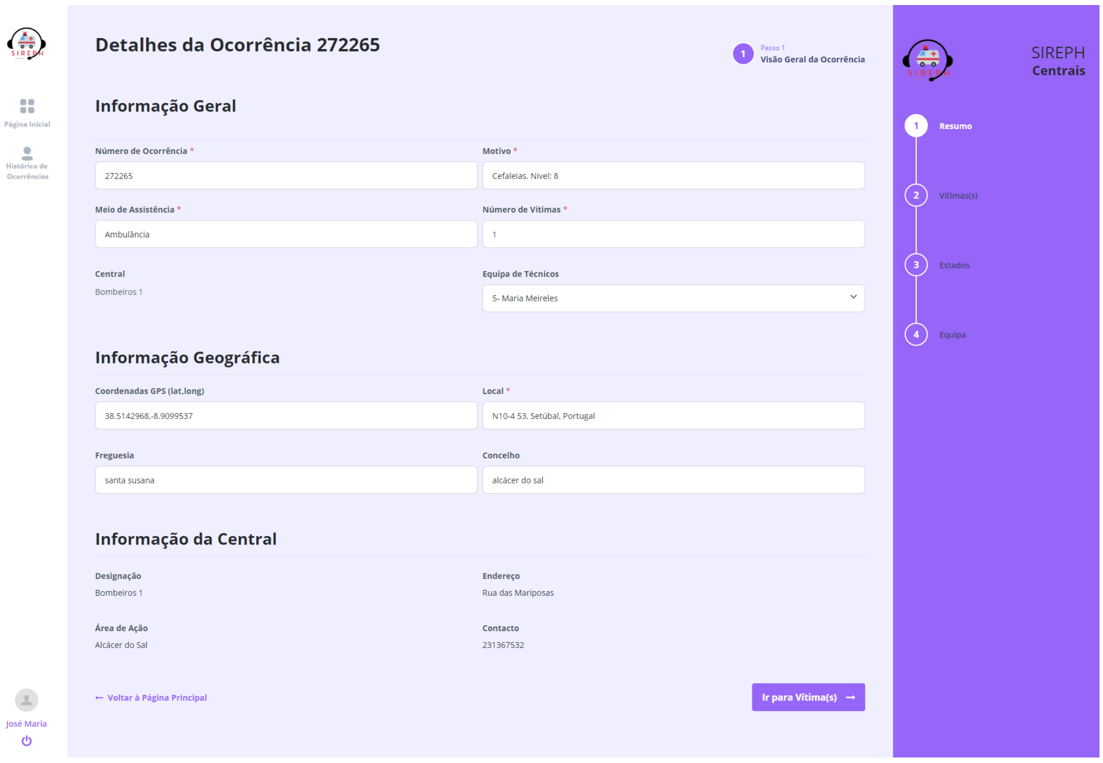
Informação Geográfica

Coordenadas GPS (lat,long)	Local *
38.5142968,-8.9099537	N10-4 53, Setúbal, Portugal
Freguesia	Concelho
santa susana	alcácer do sal

Informação da Central

Designação	Endereço
Bombeiros 1	Rua das Mariposas
Área de Ação	Contacto
Alcácer do Sal	231367532

Figure D.5: Emergency Central Subsystem Non-Administrative Central Information Detail Page



Detalhes da Ocorrência 272265

Informação Geral

Número de Ocorrência *	Motivo *
272265	Cefaleias, Nível: 8
Meio de Assistência *	Número de Vítimas *
Ambulância	1
Central	Equipa de Técnicos
Bombeiros 1	5- Maria Meireles

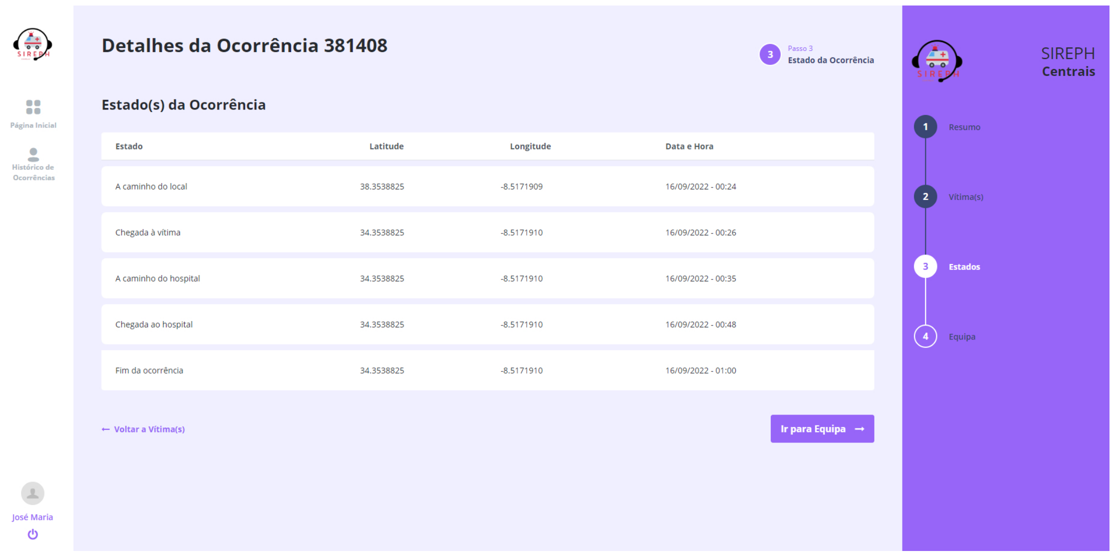
Informação Geográfica

Coordenadas GPS (lat,long)	Local *
38.5142968,-8.9099537	N10-4 53, Setúbal, Portugal
Freguesia	Concelho
santa susana	alcácer do sal

Informação da Central

Designação	Endereço
Bombeiros 1	Rua das Mariposas
Área de Ação	Contacto
Alcácer do Sal	231367532

Figure D.6: Emergency Central Subsystem Occurrence Victim Detail Page

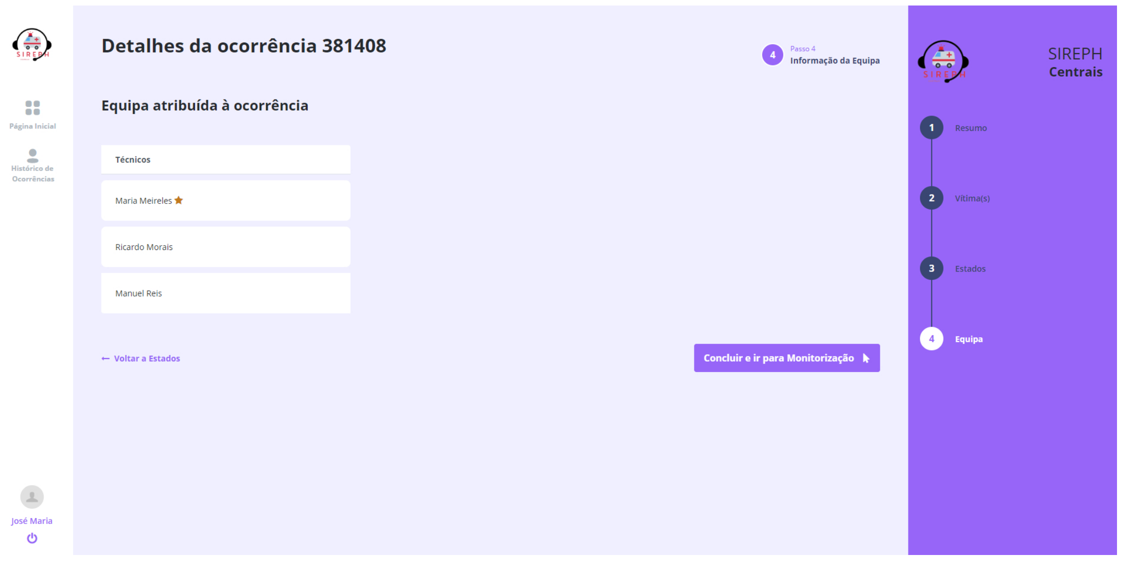


Detalhes da Ocorrência 381408

Estado(s) da Ocorrência

Estado	Latitude	Longitude	Data e Hora
A caminho do local	38.3538825	-8.5171909	16/09/2022 - 00:24
Chegada à vítima	34.3538825	-8.5171910	16/09/2022 - 00:26
A caminho do hospital	34.3538825	-8.5171910	16/09/2022 - 00:35
Chegada ao hospital	34.3538825	-8.5171910	16/09/2022 - 00:48
Fim da ocorrência	34.3538825	-8.5171910	16/09/2022 - 01:00

Figure D.7: Emergency Central Subsystem Occurrence States Detail Page

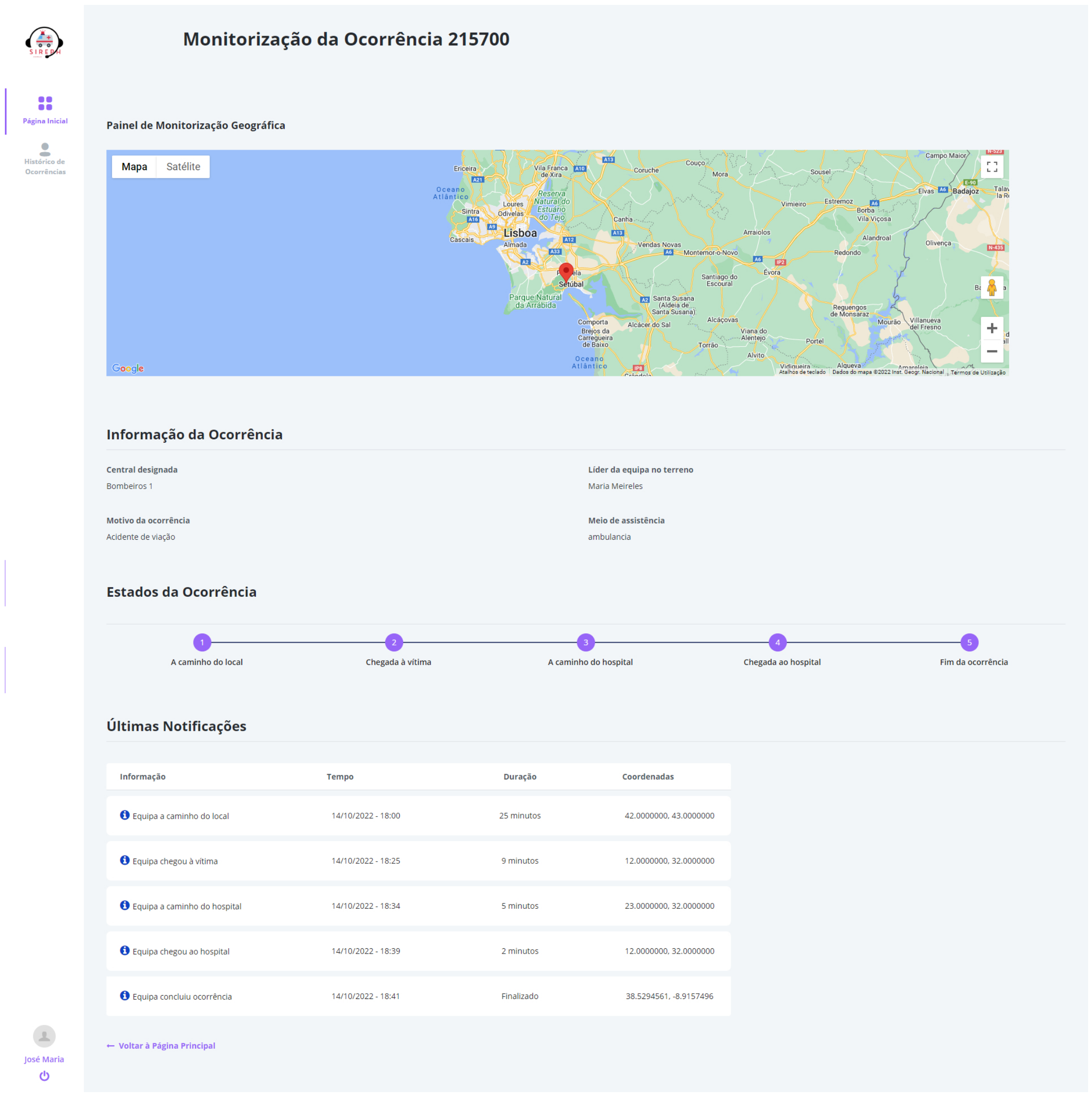


Detalhes da ocorrência 381408

Equipa atribuída à ocorrência

Técnicos
Maria Meireles ★
Ricardo Morais
Manuel Reis

Figure D.8: Emergency Central Subsystem Occurrence Team Detail Page



Monitorização da Ocorrência 215700

Painel de Monitorização Geográfica

Informação da Ocorrência

Central designada	**Líder da equipa no terreno**
Bombeiros 1	Maria Meireles
Motivo da ocorrência	**Meio de assistência**
Acidente de viação	ambulancia

Estados da Ocorrência

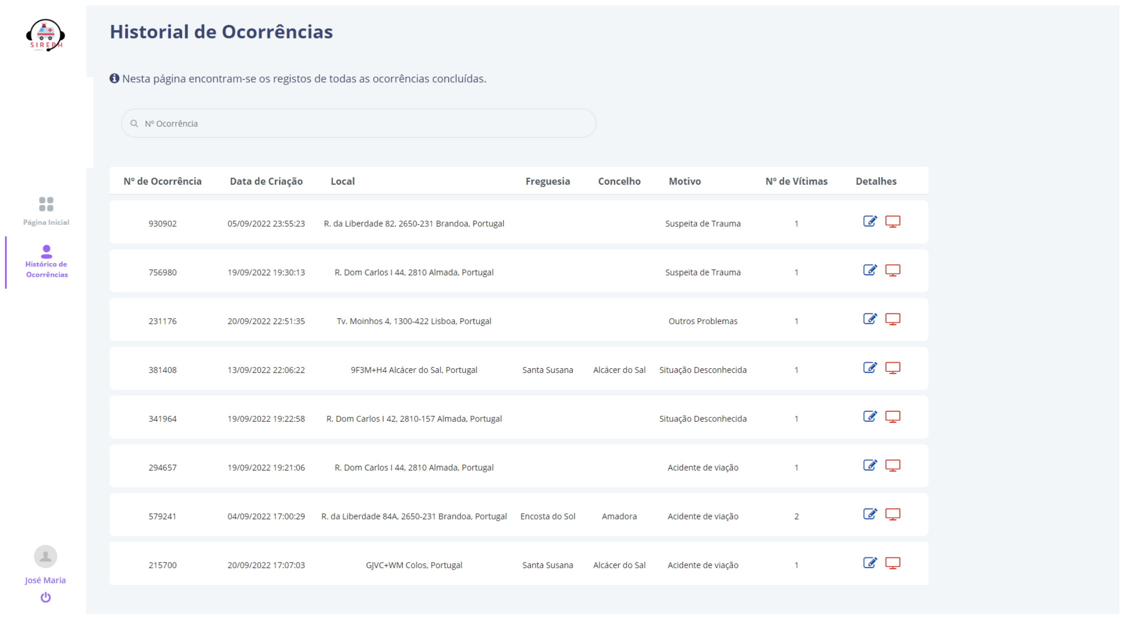
1 A caminho do local — 2 Chegada à vítima — 3 A caminho do hospital — 4 Chegada ao hospital — 5 Fim da ocorrência

Últimas Notificações

Informação	Tempo	Duração	Coordenadas
Equipa a caminho do local	14/10/2022 - 18:00	25 minutos	42.0000000, 43.0000000
Equipa chegou à vítima	14/10/2022 - 18:25	9 minutos	12.0000000, 32.0000000
Equipa a caminho do hospital	14/10/2022 - 18:34	5 minutos	23.0000000, 32.0000000
Equipa chegou ao hospital	14/10/2022 - 18:39	2 minutos	12.0000000, 32.0000000
Equipa concluiu ocorrência	14/10/2022 - 18:41	Finalizado	38.5294561, -8.9157496

← Voltar à Página Principal

Figure D.9: Emergency Central Subsystem Monitorization Page



Historial de Ocorrências

Nesta página encontram-se os registos de todas as ocorrências concluídas.

Nº de Ocorrência	Data de Criação	Local	Freguesia	Concelho	Motivo	Nº de Vítimas	Detalhes
930902	05/09/2022 23:55:23	R. da Liberdade 82, 2650-231 Brandoa, Portugal			Suspeita de Trauma	1	
756980	19/09/2022 19:30:13	R. Dom Carlos I 44, 2810 Almada, Portugal			Suspeita de Trauma	1	
231176	20/09/2022 22:51:35	Tv. Moinhos 4, 1300-422 Lisboa, Portugal			Outros Problemas	1	
381408	13/09/2022 22:06:22	9F3M+H4 Alcácer do Sal, Portugal	Santa Susana	Alcácer do Sal	Situação Desconhecida	1	
341964	19/09/2022 19:22:58	R. Dom Carlos I 42, 2810-157 Almada, Portugal			Situação Desconhecida	1	
294657	19/09/2022 19:21:06	R. Dom Carlos I 44, 2810 Almada, Portugal			Acidente de viação	1	
579241	04/09/2022 17:00:29	R. da Liberdade 84A, 2650-231 Brandoa, Portugal	Encosta do Sol	Amadora	Acidente de viação	2	
215700	20/09/2022 17:07:03	GJVC+WM Colos, Portugal	Santa Susana	Alcácer do Sal	Acidente de viação	1	

Figure D.10: Emergency Central Subsystem History Page

Appendix E

Hospital Subsystem Definitive User Interface Pages

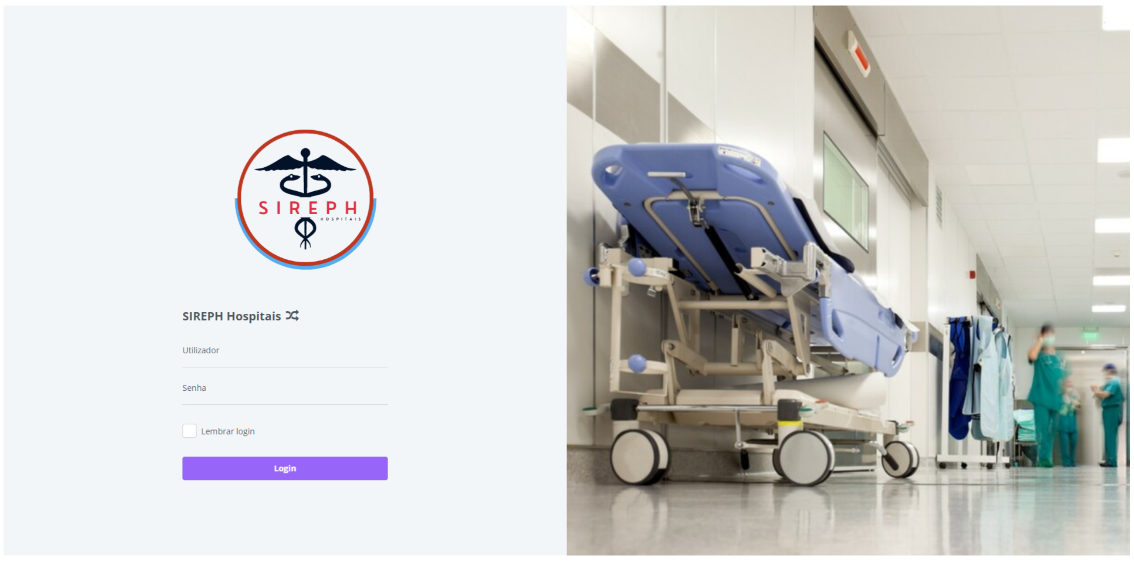
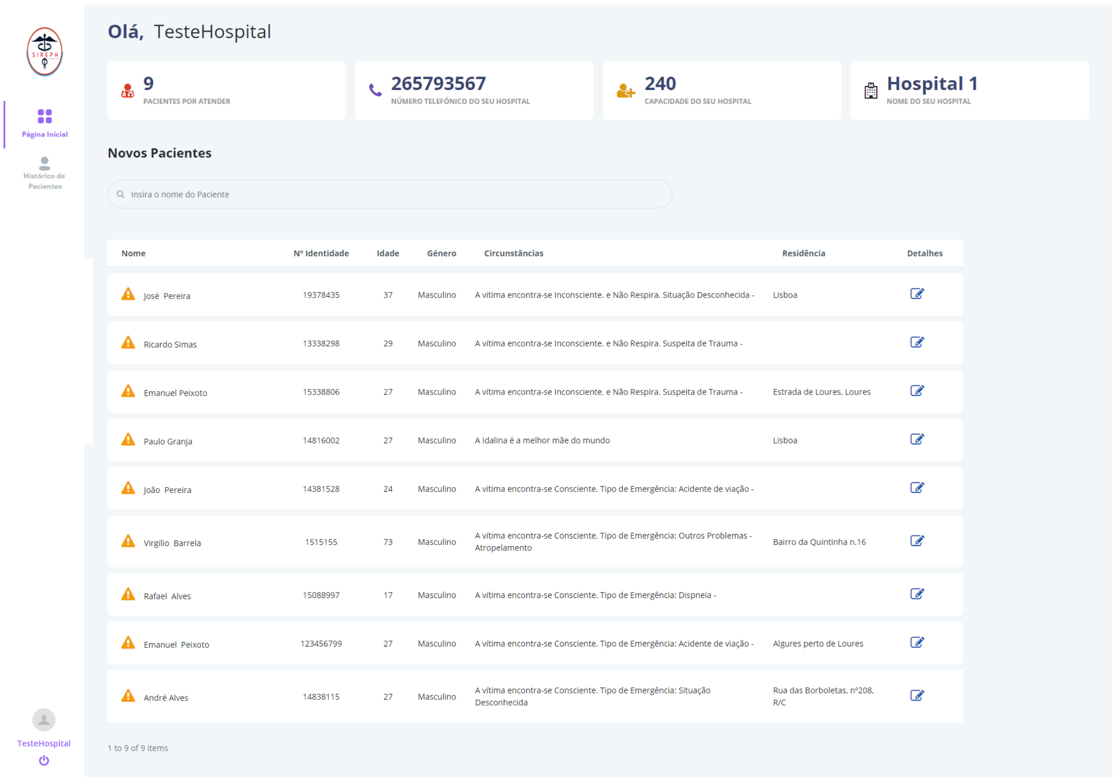


Figure E.1: Hospital Subsystem Login Page



Olá, TesteHospital

9 PACIENTES POR ATENDER

265793567 NÚMERO TELEFÓNICO DO SEU HOSPITAL

240 CAPACIDADE DO SEU HOSPITAL

Hospital 1 NOME DO SEU HOSPITAL

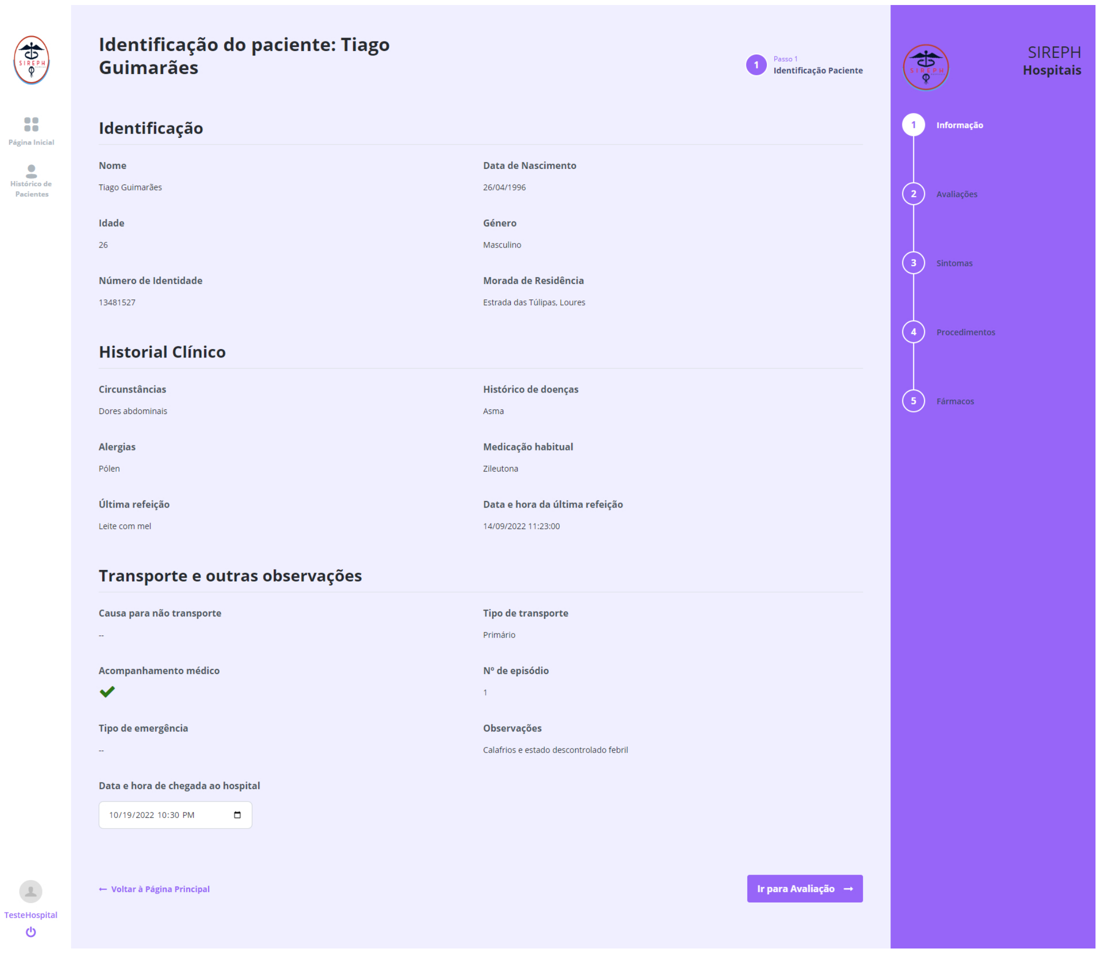
Novos Pacientes

Insira o nome do Paciente

Nome	Nº Identidade	Idade	Género	Circunstâncias	Residência	Detalhes
José Pereira	19378435	37	Masculino	A vítima encontra-se Inconsciente. e Não Respira. Situação Desconhecida -	Lisboa	
Ricardo Simas	13338298	29	Masculino	A vítima encontra-se Inconsciente. e Não Respira. Suspeita de Trauma -		
Emanuel Peixoto	15338806	27	Masculino	A vítima encontra-se Inconsciente. e Não Respira. Suspeita de Trauma -	Estrada de Loures, Loures	
Paulo Granja	14816002	27	Masculino	A Idalina é a melhor mãe do mundo	Lisboa	
João Pereira	14381528	24	Masculino	A vítima encontra-se Consciente. Tipo de Emergência: Acidente de viação -		
Virgílio Barrela	1515155	73	Masculino	A vítima encontra-se Consciente. Tipo de Emergência: Outros Problemas - Atropelamento	Bairro da Quintinha n.16	
Rafael Alves	15088997	17	Masculino	A vítima encontra-se Consciente. Tipo de Emergência: Dispneia -		
Emanuel Peixoto	123456799	27	Masculino	A vítima encontra-se Consciente. Tipo de Emergência: Acidente de viação -	Algures perto de Loures	
André Alves	14838115	27	Masculino	A vítima encontra-se Consciente. Tipo de Emergência: Situação Desconhecida	Rua das Borboletas, nº208, R/C	

1 to 9 of 9 items

Figure E.2: Hospital Subsystem Homepage



Identificação do paciente: Tiago Guimarães

Identificação

Nome	**Data de Nascimento**
Tiago Guimarães	26/04/1996
Idade	**Género**
26	Masculino
Número de Identidade	**Morada de Residência**
13481527	Estrada das Túlipas, Loures

Historial Clínico

Circunstâncias	**Histórico de doenças**
Dores abdominais	Asma
Alergias	**Medicação habitual**
Pólen	Zileutona
Última refeição	**Data e hora da última refeição**
Leite com mel	14/09/2022 11:23:00

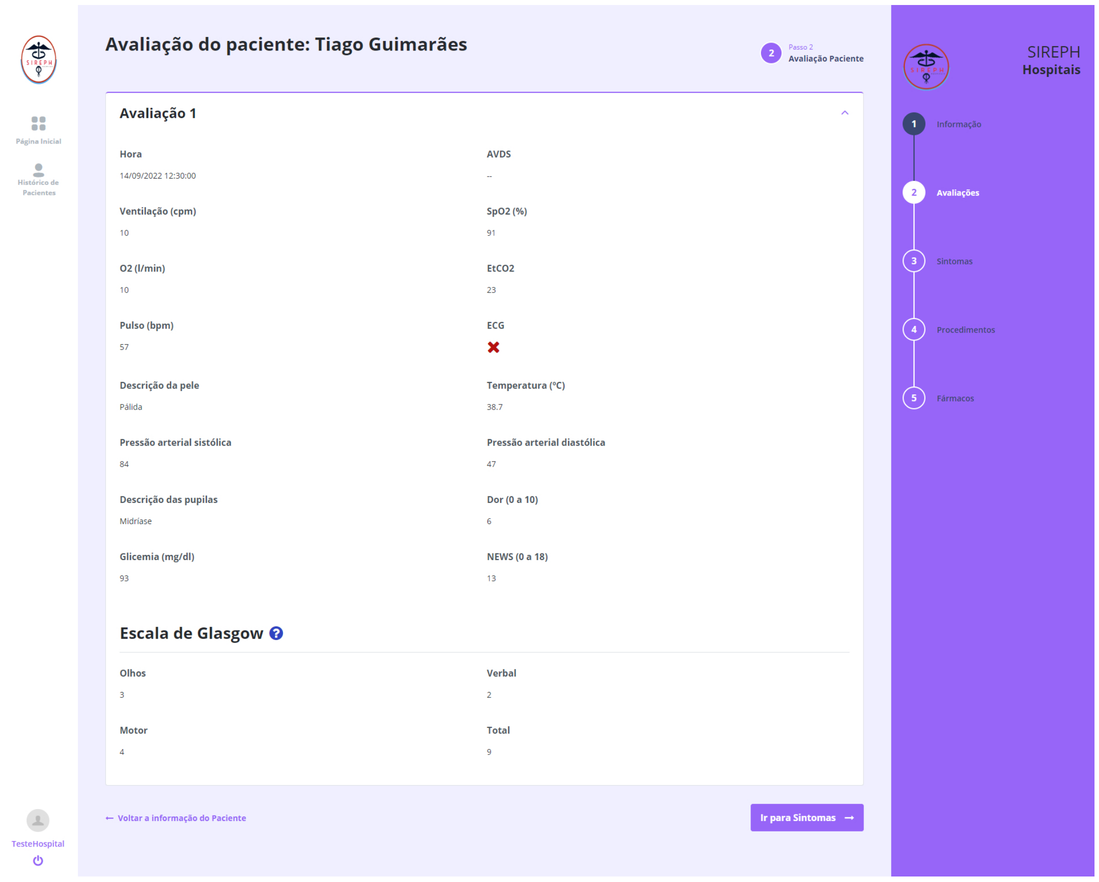
Transporte e outras observações

Causa para não transporte	**Tipo de transporte**
--	Primário
Acompanhamento médico	**Nº de episódio**
✓	1
Tipo de emergência	**Observações**
--	Calafrios e estado descontrolado febril
Data e hora de chegada ao hospital	
10/19/2022 10:30 PM	

← Voltar à Página Principal

Ir para Avaliação →

Figure E.3: Hospital Subsystem Patient Information Detail Page



Avaliação do paciente: Tiago Guimarães

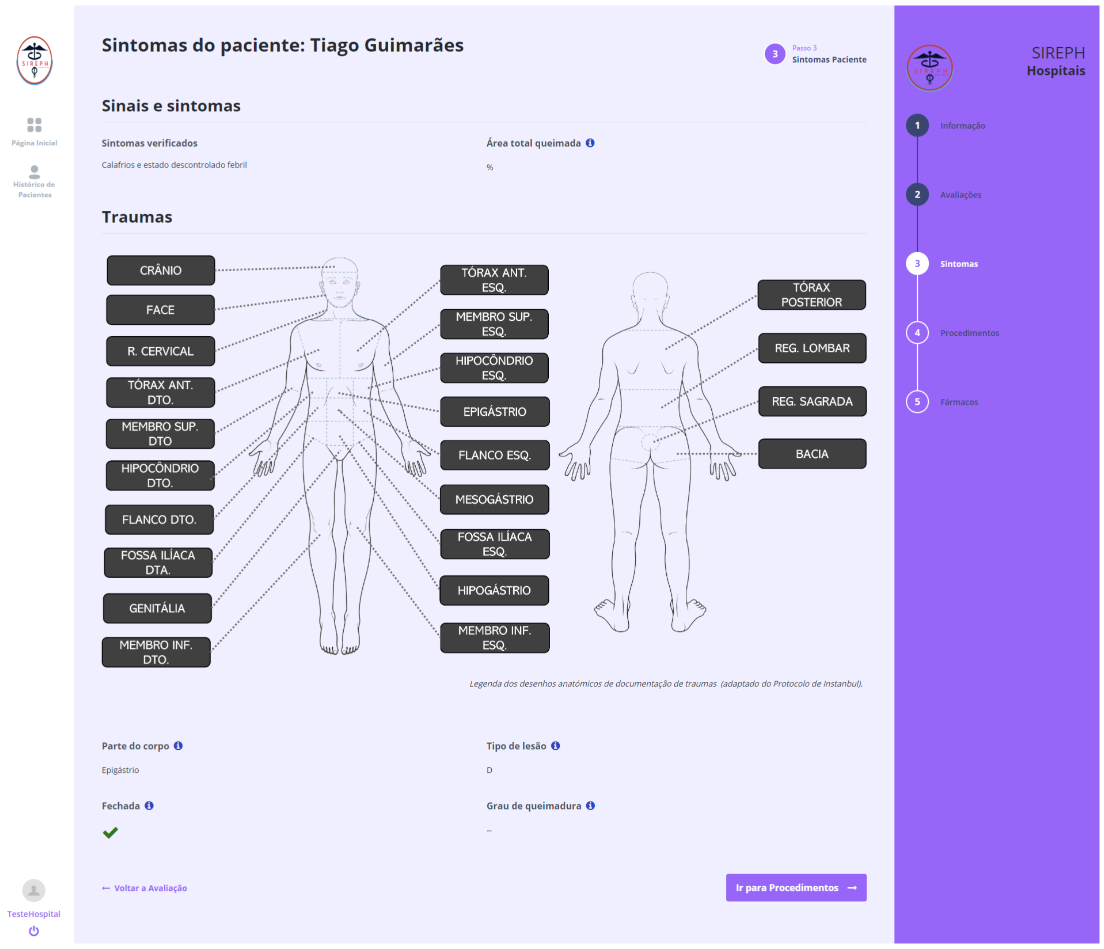
Avaliação 1

Hora	**AVDS**
14/09/2022 12:30:00	--
Ventilação (cpm)	**SpO2 (%)**
10	91
O2 (l/min)	**EtCO2**
10	23
Pulso (bpm)	**ECG**
57	✗
Descrição da pele	**Temperatura (ºC)**
Pálida	38.7
Pressão arterial sistólica	**Pressão arterial diastólica**
84	47
Descrição das pupilas	**Dor (0 a 10)**
Midríase	6
Glicemia (mg/dl)	**NEWS (0 a 18)**
93	13

Escala de Glasgow ?

Olhos	**Verbal**
3	2
Motor	**Total**
4	9

Figure E.4: Hospital Subsystem Patient Evaluation Detail Page



Sintomas do paciente: Tiago Guimarães

3 Passo 3 Sintomas Paciente

Sinais e sintomas

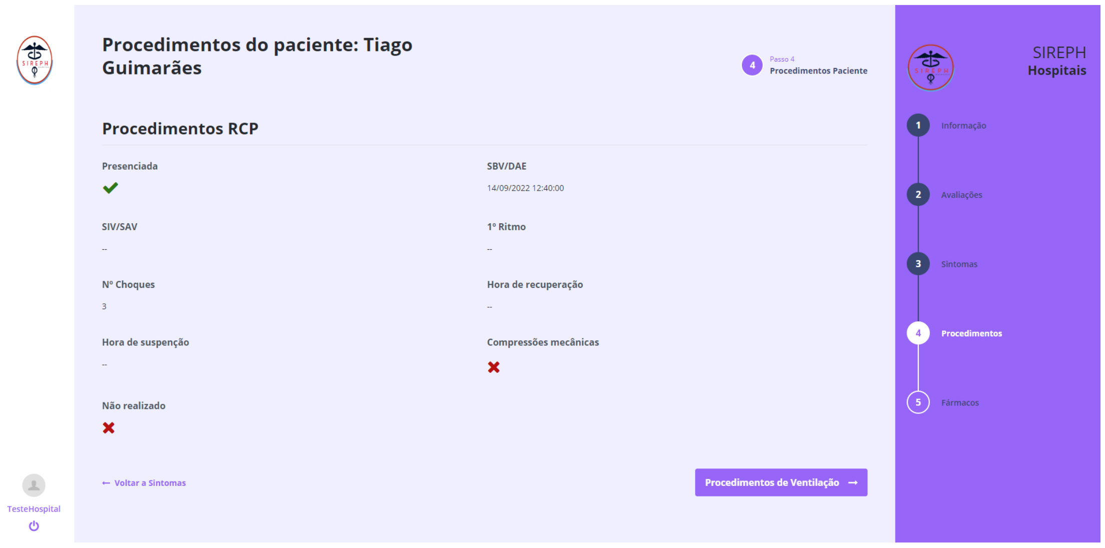
Sintomas verificados	Área total queimada
Calafrios e estado descontrolado febril	%

Traumas

Legenda dos desenhos anatómicos de documentação de traumas (adaptado do Protocolo de Instanbul).

Parte do corpo	Tipo de lesão
Epigástrio	D
Fechada	Grau de queimadura
✓	--

Figure E.5: Hospital Subsystem Patient Symptoms Detail Page



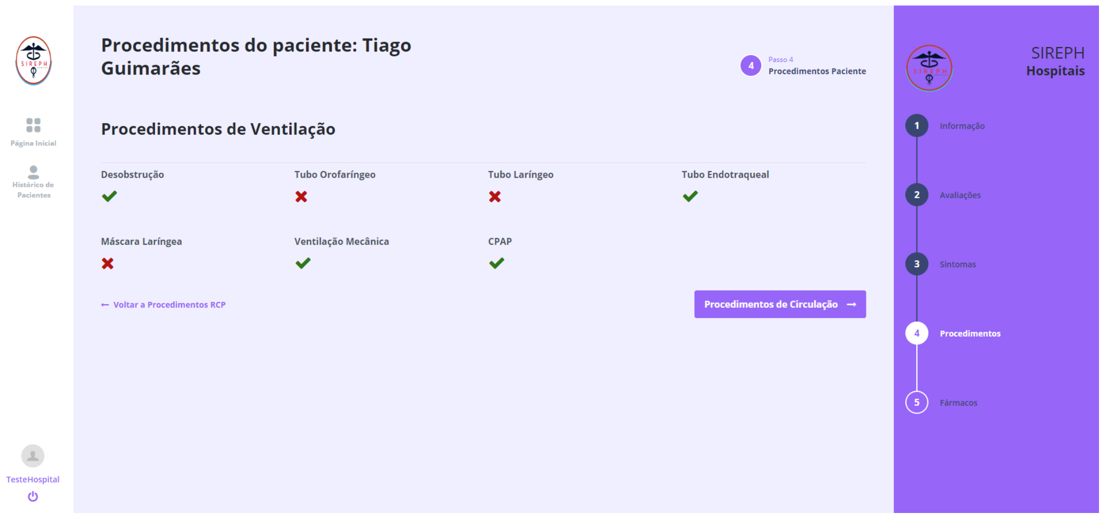
Procedimentos do paciente: Tiago Guimarães

4 Passo 4 Procedimentos Paciente

Procedimentos RCP

Presenciada	SBV/DAE
✓	14/09/2022 12:40:00
SIV/SAV	1º Ritmo
--	--
Nº Choques	Hora de recuperação
3	--
Hora de suspenção	Compressões mecânicas
--	✗
Não realizado	
✗	

Figure E.6: Hospital Subsystem Patient RCP Procedures Detail Page


Página Inicial

Histórico de Pacientes

TesteHospital

Procedimentos do paciente: Tiago Guimarães

4 Passo 4 Procedimentos Paciente

Procedimentos de Ventilação

Desobstrução	Tubo Orofaríngeo	Tubo Laríngeo	Tubo Endotraqueal
✓	✗	✗	✓
Máscara Laríngea	Ventilação Mecânica	CPAP	
✗	✓	✓	

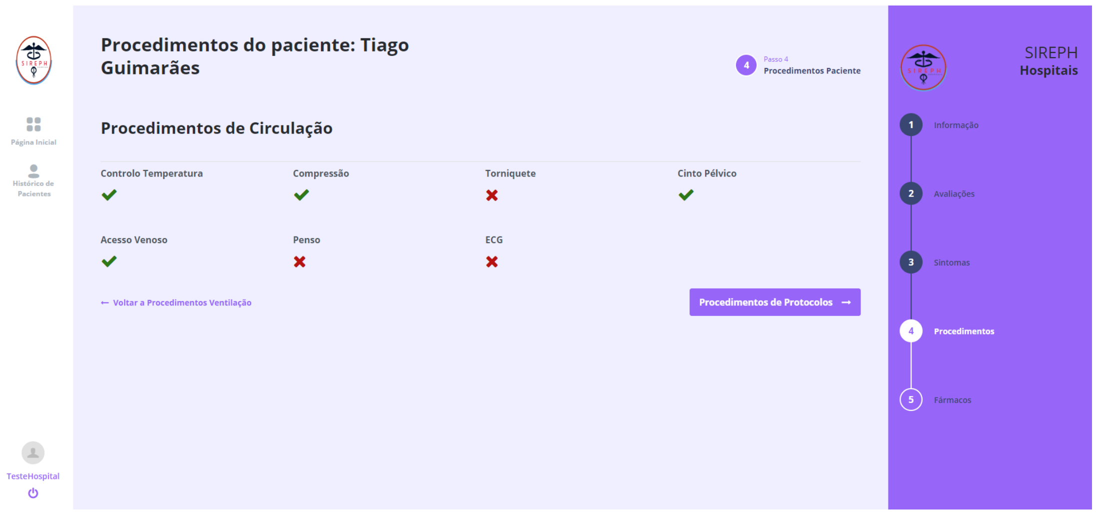
← Voltar a Procedimentos RCP

Procedimentos de Circulação →

SIREPH Hospitais

1 Informação
2 Avaliações
3 Sintomas
4 Procedimentos
5 Fármacos

Figure E.7: Hospital Subsystem Patient Ventilation Procedures Detail Page


Página Inicial

Histórico de Pacientes

TesteHospital

Procedimentos do paciente: Tiago Guimarães

4 Passo 4 Procedimentos Paciente

Procedimentos de Circulação

Controlo Temperatura	Compressão	Torniquete	Cinto Pélvico
✓	✓	✗	✓
Acesso Venoso	Penso	ECG	
✓	✗	✗	

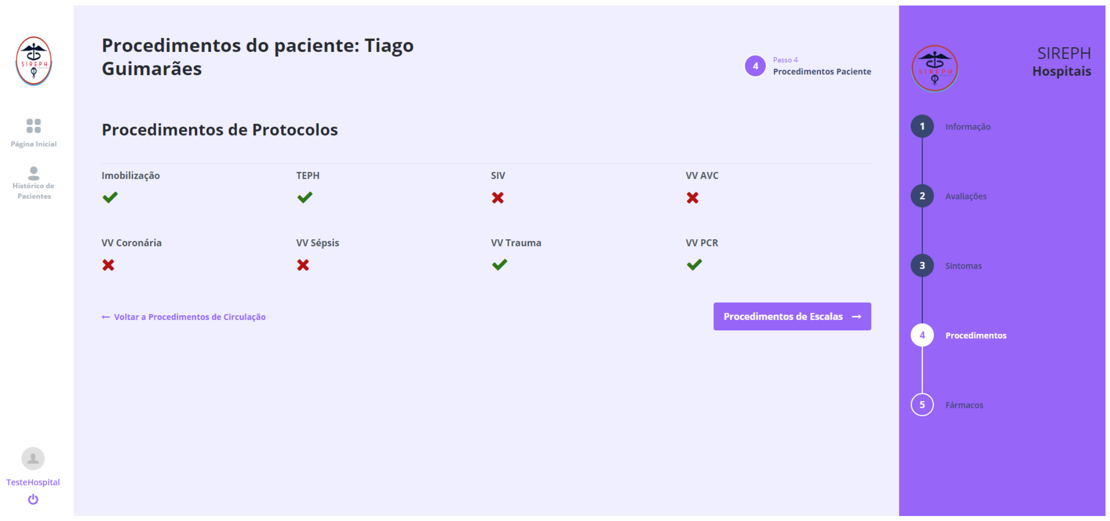
← Voltar a Procedimentos Ventilação

Procedimentos de Protocolos →

SIREPH Hospitais

1 Informação
2 Avaliações
3 Sintomas
4 Procedimentos
5 Fármacos

Figure E.8: Hospital Subsystem Patient Circulation Procedures Detail Page


Página Inicial

Histórico de Pacientes

TesteHospital

Procedimentos do paciente: Tiago Guimarães

4 Passo 4 Procedimentos Paciente

Procedimentos de Protocolos

Imobilização	TEPH	SIV	VV AVC
✓	✓	✗	✗
VV Coronária	VV Sépsis	VV Trauma	VV PCR
✗	✗	✓	✓

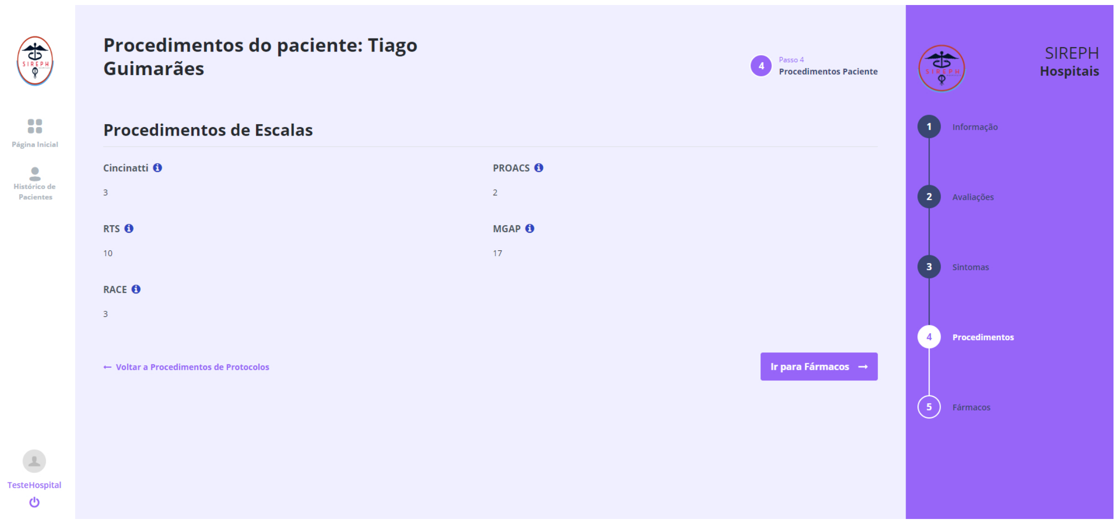
← Voltar a Procedimentos de Circulação

Procedimentos de Escalas →

SIREPH Hospitais

1 Informação
2 Avaliações
3 Sintomas
4 Procedimentos
5 Fármacos

Figure E.9: Hospital Subsystem Patient Protocol Procedures Detail Page

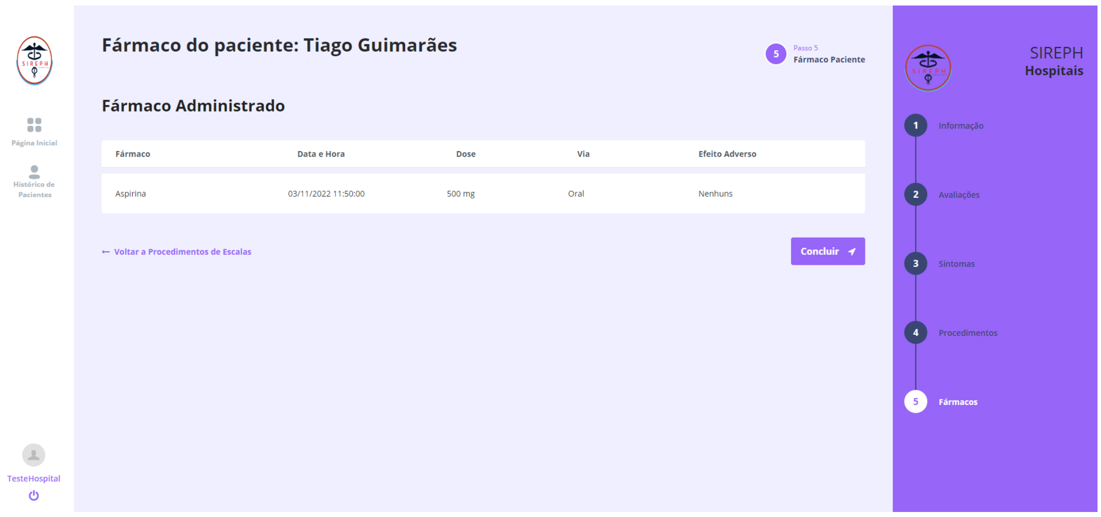


Procedimentos do paciente: Tiago Guimarães

Procedimentos de Escalas

Cincinatti	PROACS
3	2
RTS	MGAP
10	17
RACE	
3	

Figure E.10: Hospital Subsystem Patient Scale Procedures Detail Page

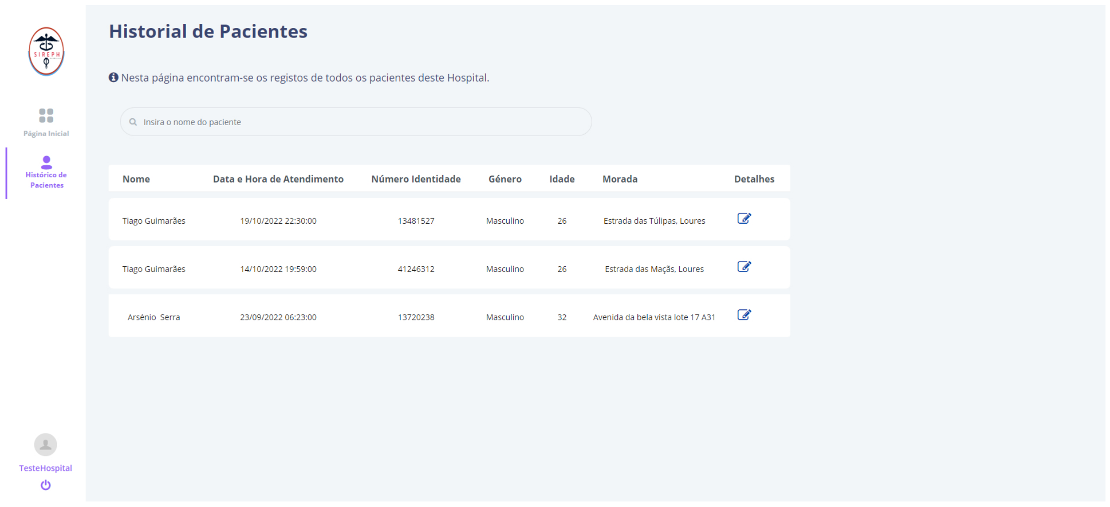


Fármaco do paciente: Tiago Guimarães

Fármaco Administrado

Fármaco	Data e Hora	Dose	Via	Efeito Adverso
Aspirina	03/11/2022 11:50:00	500 mg	Oral	Nenhuns

Figure E.11: Hospital Subsystem Patient Pharmacy Detail Page



Historial de Pacientes

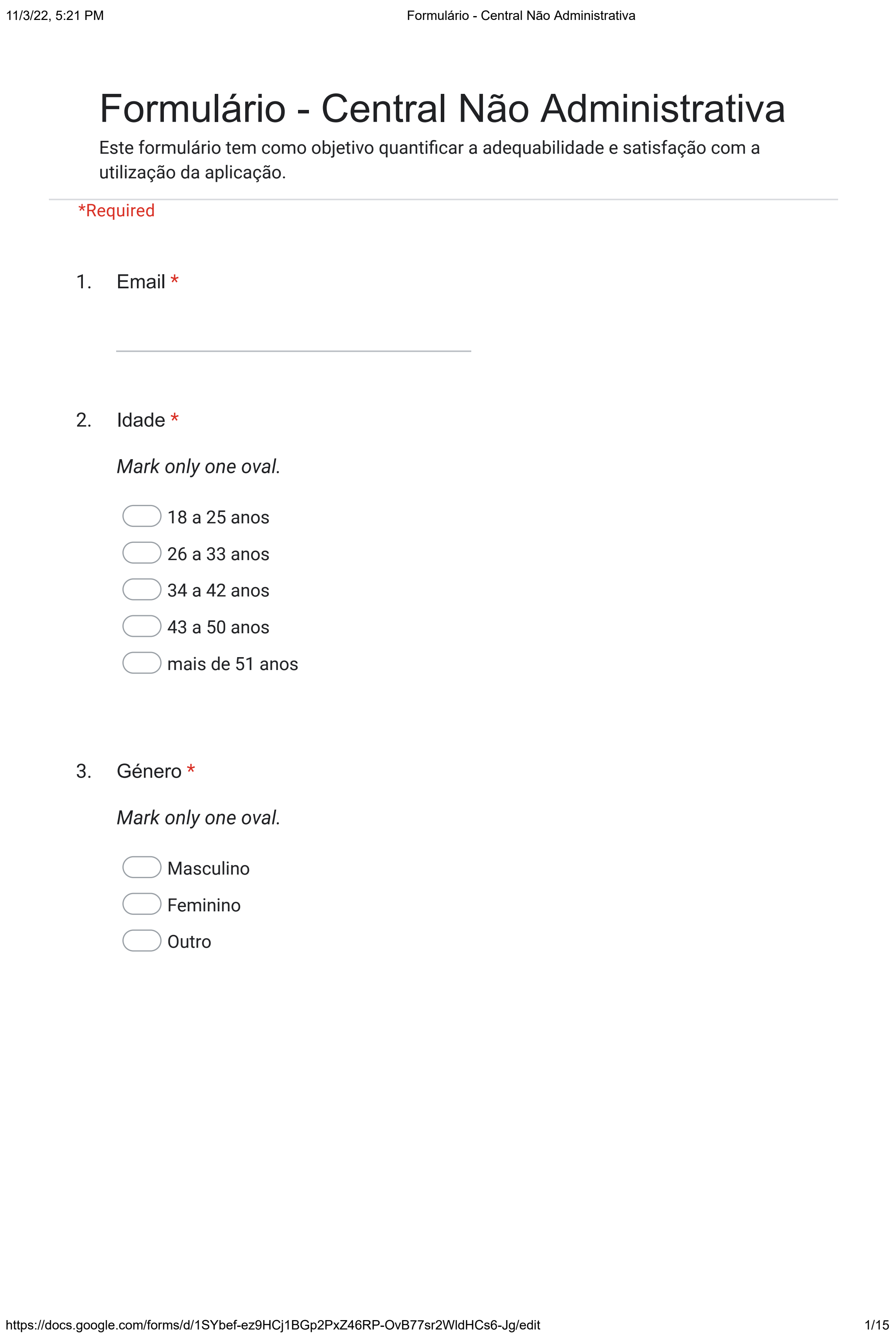
Nesta página encontram-se os registos de todos os pacientes deste Hospital.

Nome	Data e Hora de Atendimento	Número Identidade	Género	Idade	Morada	Detalhes
Tiago Guimarães	19/10/2022 22:30:00	13481527	Masculino	26	Estrada das Túlipas, Loures	
Tiago Guimarães	14/10/2022 19:59:00	41246312	Masculino	26	Estrada das Maçãs, Loures	
Arsénio Serra	23/09/2022 06:23:00	13720238	Masculino	32	Avenida da bela vista lote 17 A31	

Figure E.12: Hospital Subsystem History Page

Appendix F

Non-Administrative Central Form



11/3/22, 5:21 PM Formulário - Central Não Administrativa

Formulário - Central Não Administrativa

Este formulário tem como objetivo quantificar a adequabilidade e satisfação com a utilização da aplicação.

*Required

1. Email *

   ____

2. Idade *

   Mark only one oval.

   - ◯ 18 a 25 anos
   - ◯ 26 a 33 anos
   - ◯ 34 a 42 anos
   - ◯ 43 a 50 anos
   - ◯ mais de 51 anos

3. Género *

   Mark only one oval.

   - ◯ Masculino
   - ◯ Feminino
   - ◯ Outro

https://docs.google.com/forms/d/1SYbef-ez9HCj1BGp2PxZ46RP-OvB77sr2WldHCs6-Jg/edit 1/15

4. Formação Académica *

   Mark only one oval.

   - ◯ Ensino Básico
   - ◯ Ensino Secundário
   - ◯ Bacharelato
   - ◯ Licenciatura
   - ◯ Mestrado
   - ◯ Doutoramento

5. Profissão *

   ____

6. Anos de atividade da Profissão anteriormente introduzida *

   Mark only one oval.

   - ◯ Até 3 anos
   - ◯ 4 a 7 anos
   - ◯ 8 a 11 anos
   - ◯ Mais de 11 anos

Página Inicial

As perguntas nesta secção encontram-se relacionadas com a página principal da aplicação para os utilizadores das centrais administrativas.

7. É fácil compreender quais as ocorrências que ainda não possuem equipa atribuída *

Mark only one oval.

- ⬭ Discordo Totalmente
- ⬭ Discordo
- ⬭ Sem Opinião
- ⬭ Concordo
- ⬭ Concordo Totalmente

8. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

9. A barra de pesquisa é útil *

Mark only one oval.

- ⬭ Discordo Totalmente
- ⬭ Discordo
- ⬭ Sem Opinião
- ⬭ Concordo
- ⬭ Concordo Totalmente

10. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

11. A tabela das ocorrências ativas da sua central tem a informação adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

12. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

13. A interface gráfica é esteticamente apelativa *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

14. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

15. É fácil encontrar o botão para os detalhes de uma ocorrência ativa da sua central *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

16. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

17. É fácil encontrar a página de monitorização *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

18. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

Página de Detalhes (Resumo, Estado(s), Vítima(s), Equipa)

As perguntas nesta secção encontram-se relacionadas com a página de detalhes da aplicação para os utilizadores das centrais administrativas.

19. A interface gráfica das páginas de detalhes é esteticamente apelativa *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

20. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

21. É fácil ler e alterar a informação recebida *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

22. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

23. É fácil atribuir uma ocorrência a uma equipa de técnicos *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

24. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

25. A navegação pelas páginas de detalhes é intuitiva *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

26. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

27. A informação apresentada é adequada *

Mark only one oval.

- ⬭ Discordo Totalmente
- ⬭ Discordo
- ⬭ Sem Opinião
- ⬭ Concordo
- ⬭ Concordo Totalmente

28. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

Página de Monitorização

As perguntas nesta secção encontram-se relacionadas com a página de monitorização da aplicação para os utilizadores das centrais administrativas.

29. A interface gráfica da página de monitorização é esteticamente apelativa *

Mark only one oval.

- ⬭ Discordo Totalmente
- ⬭ Discordo
- ⬭ Sem Opinião
- ⬭ Concordo
- ⬭ Concordo Totalmente

30. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

31. O mapa presente no painel de monitorização geográfica é útil *

Mark only one oval.

- ⬭ Discordo Totalmente
- ⬭ Discordo
- ⬭ Sem Opinião
- ⬭ Concordo
- ⬭ Concordo Totalmente

32. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

33. O marcador presente no mapa, relativo à localização do estado atual da ocorrência é útil *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

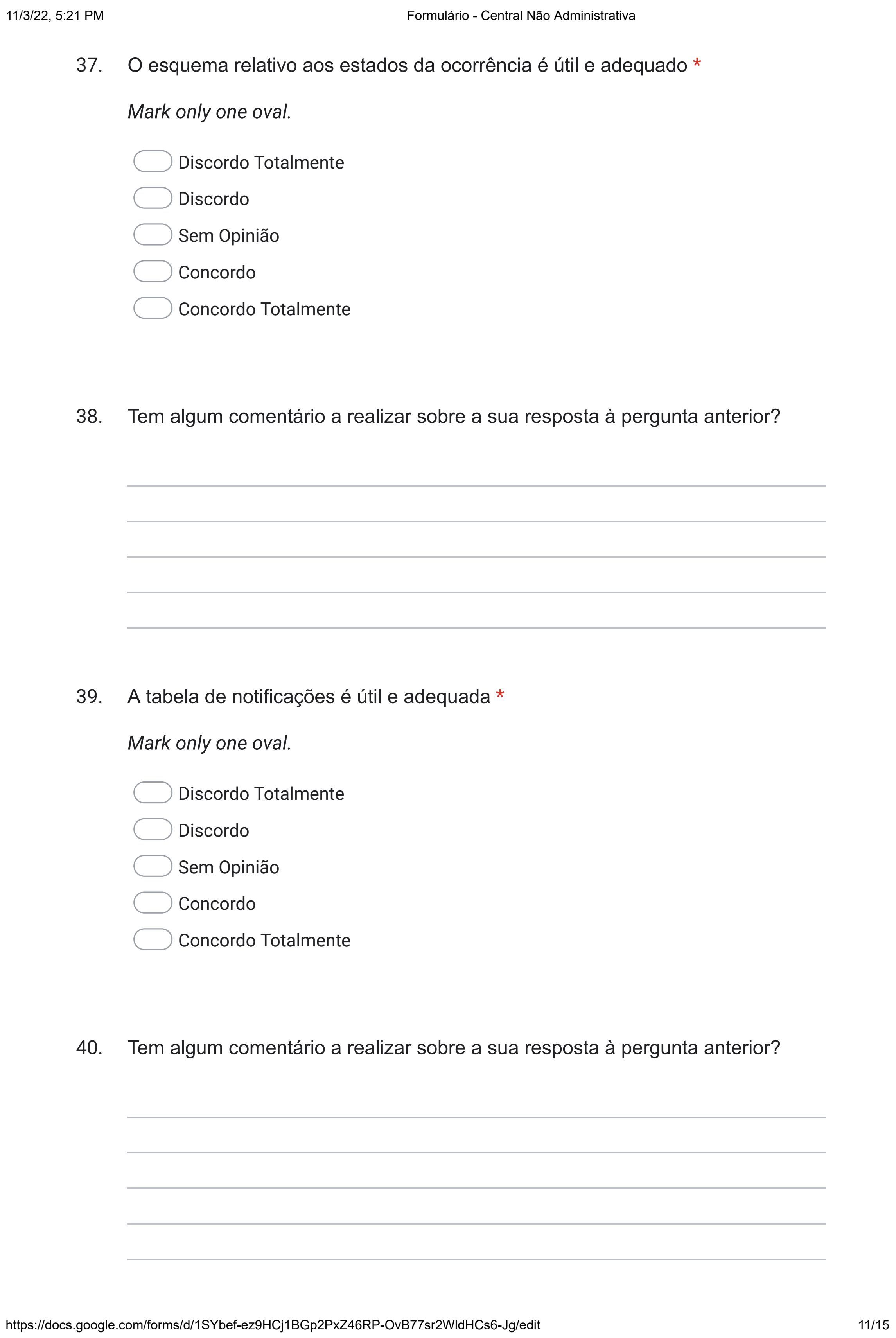
34. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

35. A informação sobre a ocorrência a monitorizar-se é adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

36. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?



11/3/22, 5:21 PM Formulário - Central Não Administrativa

37. O esquema relativo aos estados da ocorrência é útil e adequado *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

38. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

39. A tabela de notificações é útil e adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

40. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

https://docs.google.com/forms/d/1SYbef-ez9HCj1BGp2PxZ46RP-OvB77sr2WldHCs6-Jg/edit 11/15

41. Considero a página de monitorização de um modo geral uma mais valia para assegurar que a realização da ocorrência ocorre sem problemas

Mark only one oval.

- ⬭ Discordo Totalmente
- ⬭ Discordo
- ⬭ Concordo
- ⬭ Concordo Totalmente

42. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

____
____
____
____
____

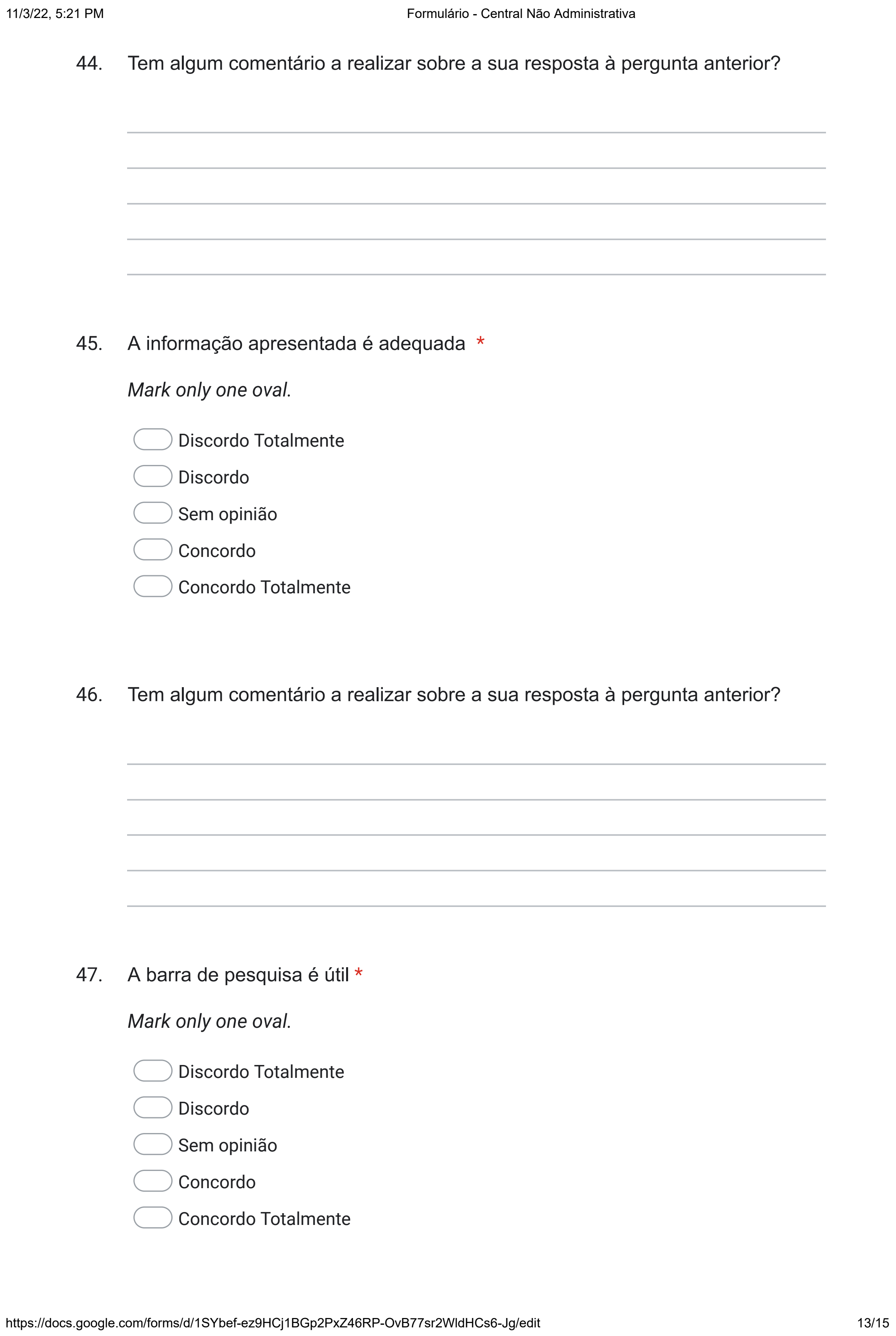
Página de Histórico

As perguntas nesta secção encontram-se relacionadas com a página de histórico da aplicação para os utilizadores das centrais administrativas.

43. É fácil encontrar a página de histórico *

Mark only one oval.

- ⬭ Discordo Totalmente
- ⬭ Discordo
- ⬭ Sem opinião
- ⬭ Concordo
- ⬭ Concordo Totalmente



11/3/22, 5:21 PM Formulário - Central Não Administrativa

44. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

45. A informação apresentada é adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem opinião
- Concordo
- Concordo Totalmente

46. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

47. A barra de pesquisa é útil *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem opinião
- Concordo
- Concordo Totalmente

https://docs.google.com/forms/d/1SYbef-ez9HCj1BGp2PxZ46RP-OvB77sr2WldHCs6-Jg/edit 13/15

48. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

Opiniões

49. Considera que a aplicação poderá ser útil num contexto real? *

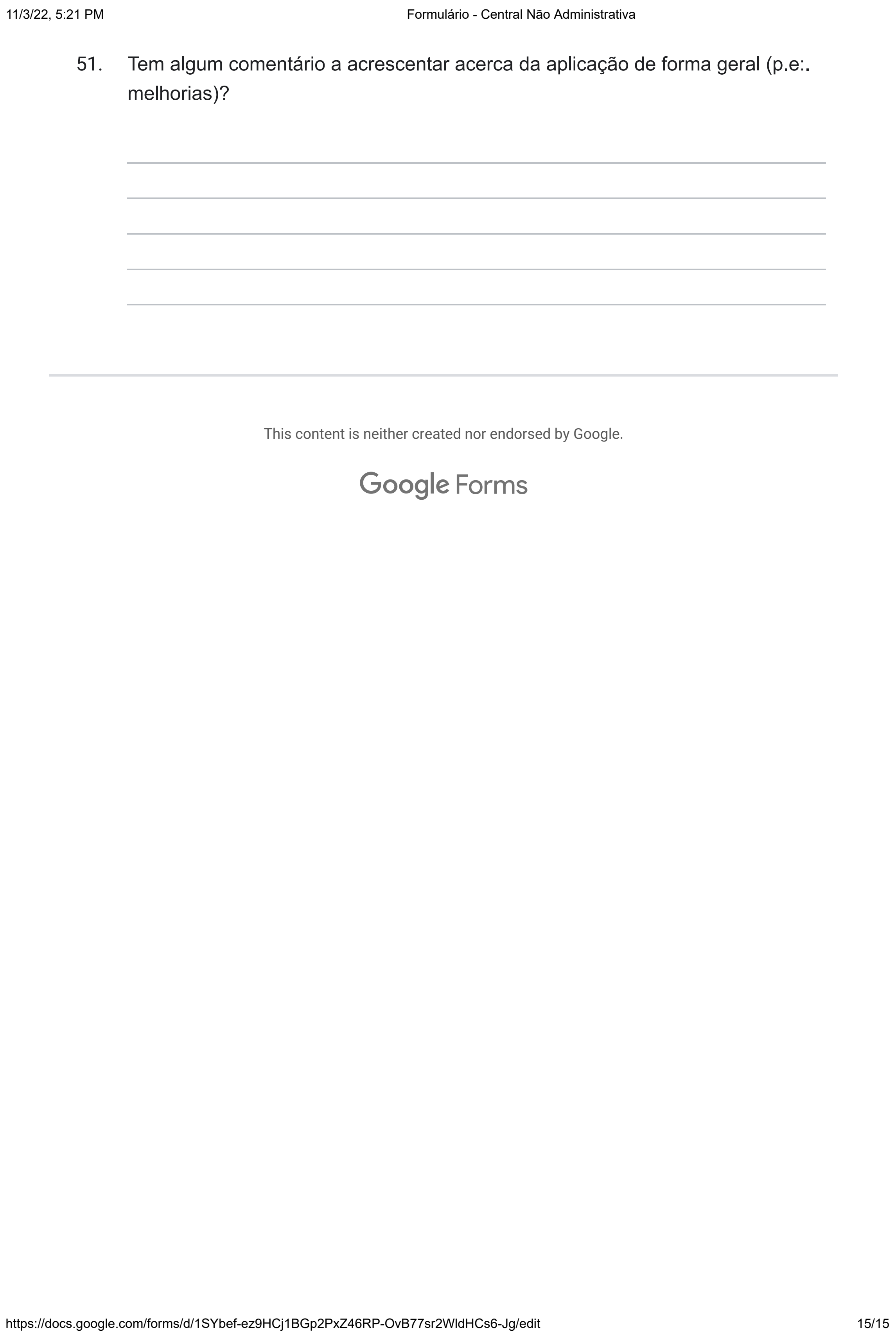
Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

50. Considera que a realização do cenário foi

Mark only one oval.

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil



11/3/22, 5:21 PM Formulário - Central Não Administrativa

51. Tem algum comentário a acrescentar acerca da aplicação de forma geral (p.e:. melhorias)?

This content is neither created nor endorsed by Google.

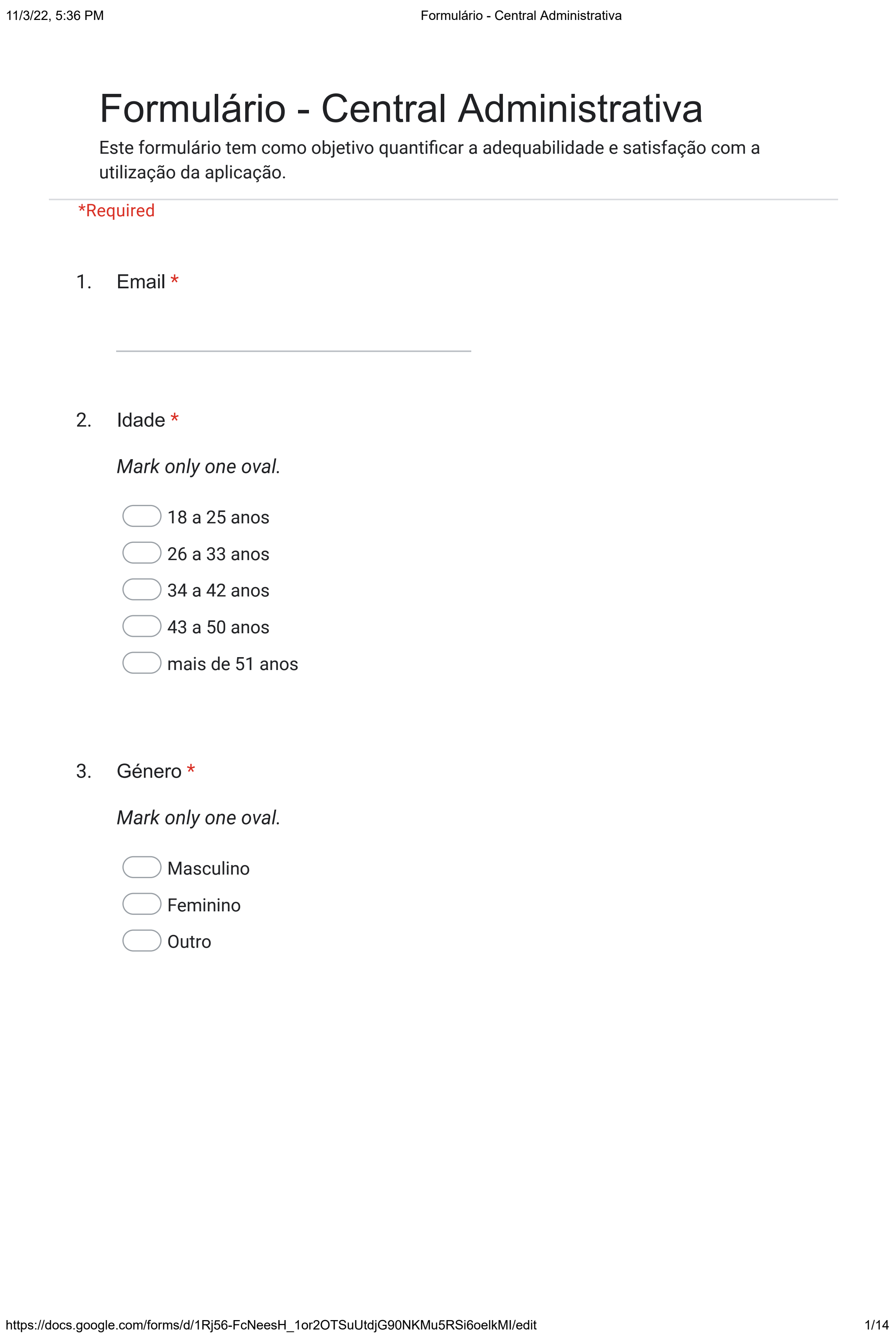
Google Forms

https://docs.google.com/forms/d/1SYbef-ez9HCj1BGp2PxZ46RP-OvB77sr2WldHCs6-Jg/edit 15/15

Figure F.1: Non-Administrative Emergency Central Form Template

Appendix G

Administrative Central Form



11/3/22, 5:36 PM Formulário - Central Administrativa

Formulário - Central Administrativa

Este formulário tem como objetivo quantificar a adequabilidade e satisfação com a utilização da aplicação.

*Required

1. Email *

2. Idade *

Mark only one oval.

- ◯ 18 a 25 anos
- ◯ 26 a 33 anos
- ◯ 34 a 42 anos
- ◯ 43 a 50 anos
- ◯ mais de 51 anos

3. Género *

Mark only one oval.

- ◯ Masculino
- ◯ Feminino
- ◯ Outro

https://docs.google.com/forms/d/1Rj56-FcNeesH_1or2OTSuUtdjG90NKMu5RSi6oelkMI/edit 1/14

4. Formação Académica *

Mark only one oval.

- ◯ Ensino Básico
- ◯ Ensino Secundário
- ◯ Bacharelato
- ◯ Licenciatura
- ◯ Mestrado
- ◯ Doutoramento

5. Profissão *

____

6. Anos de atividade da Profissão anteriormente introduzida *

Mark only one oval.

- ◯ Até 3 anos
- ◯ 4 a 7 anos
- ◯ 8 a 11 anos
- ◯ Mais de 11 anos

Página Inicial

As perguntas nesta secção encontram-se relacionadas com a página principal da aplicação para os utilizadores das centrais administrativas.

7. É fácil compreender quais as ocorrências que ainda não possuem central atribuída *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

8. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

9. A barra de pesquisa é útil *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

10. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

11. A tabela das ocorrências ativas tem a informação adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

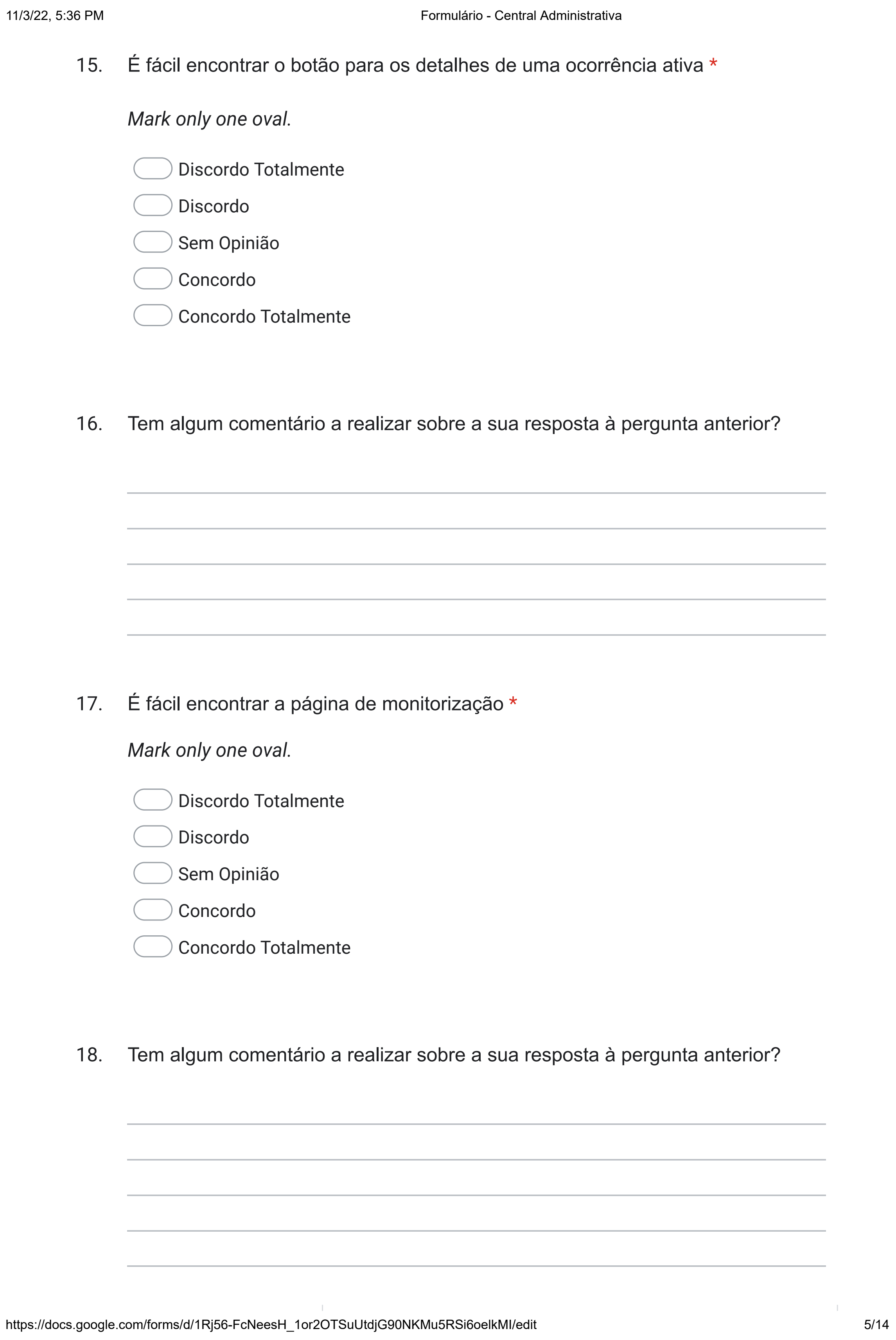
12. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

13. A interface gráfica é esteticamente apelativa *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

14. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?



11/3/22, 5:36 PM Formulário - Central Administrativa

15. É fácil encontrar o botão para os detalhes de uma ocorrência ativa *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

16. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

17. É fácil encontrar a página de monitorização *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

18. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

https://docs.google.com/forms/d/1Rj56-FcNeesH_1or2OTSuUtdjG90NKMu5RSi6oelkMI/edit 5/14

Página de Detalhes (Resumo, Estado(s), Vítima(s), Equipa)

As perguntas nesta secção encontram-se relacionadas com a página de detalhes da aplicação para os utilizadores das centrais administrativas.

19. A interface gráfica das páginas de detalhes é esteticamente apelativa *

Mark only one oval.

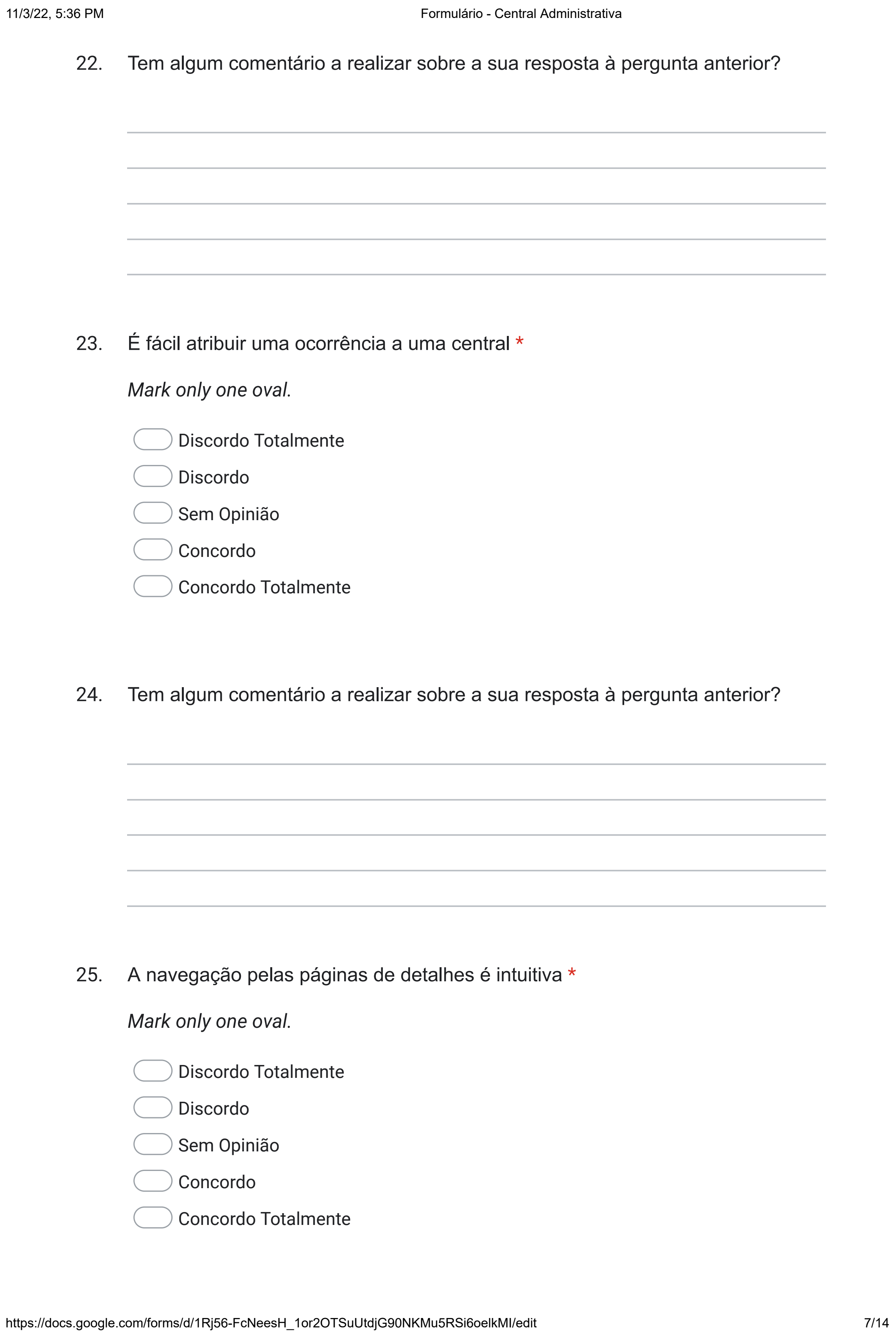
- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

20. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

21. É fácil ler e alterar a informação recebida *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente



22. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

23. É fácil atribuir uma ocorrência a uma central *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

24. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

25. A navegação pelas páginas de detalhes é intuitiva *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

26. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

27. A informação apresentada é adequada *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

28. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

Página de Monitorização

As perguntas nesta secção encontram-se relacionadas com a página de monitorização da aplicação para os utilizadores das centrais administrativas.

29. A interface gráfica da página de monitorização é esteticamente apelativa *

Mark only one oval.

◯ Discordo Totalmente
◯ Discordo
◯ Sem Opinião
◯ Concordo
◯ Concordo Totalmente

30. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

31. O mapa presente no painel de monitorização geográfica é útil *

Mark only one oval.

◯ Discordo Totalmente
◯ Discordo
◯ Sem Opinião
◯ Concordo
◯ Concordo Totalmente

32. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

33. O marcador presente no mapa, relativo à localização do estado atual da ocorrência é útil *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

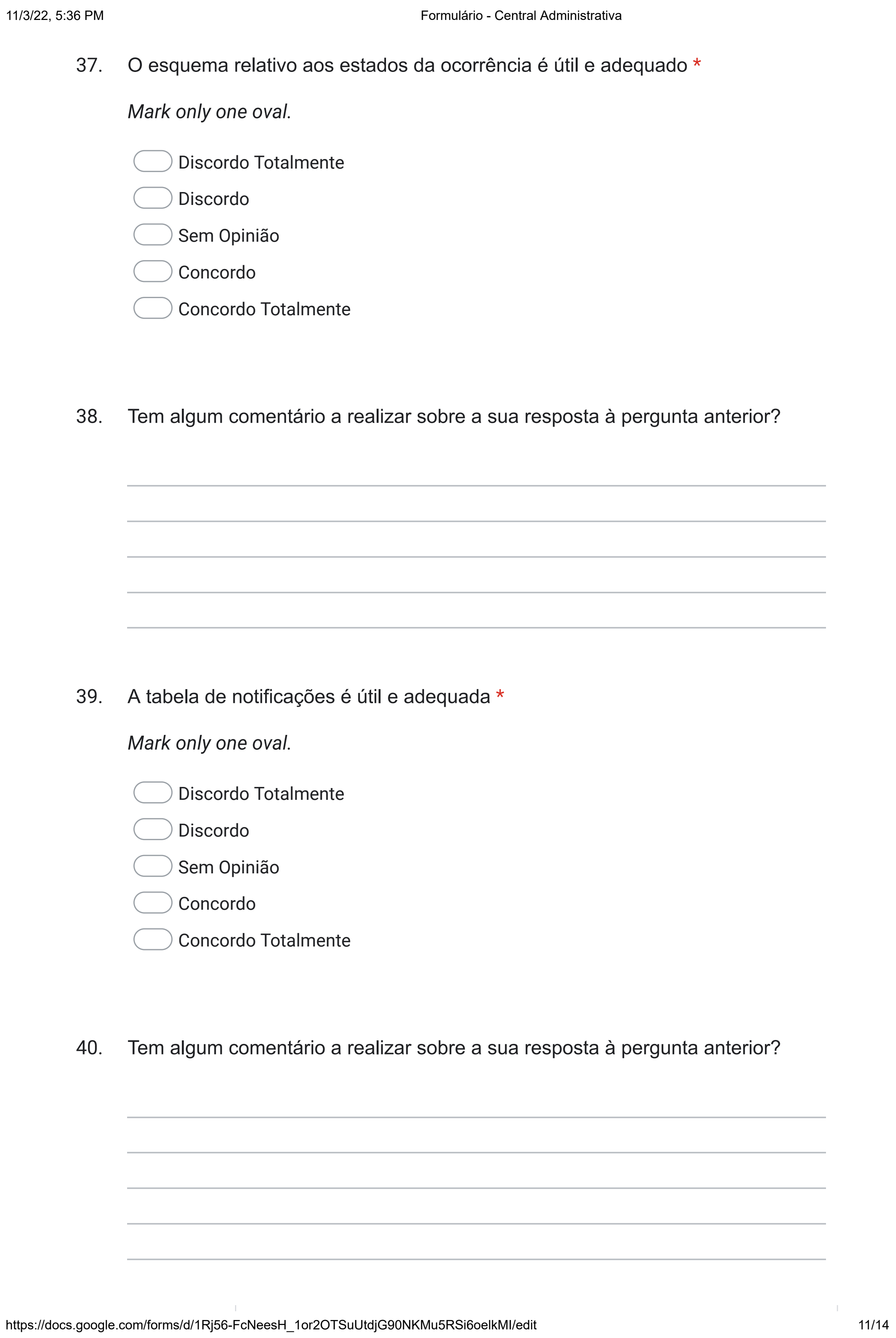
34. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

35. A informação sobre a ocorrência a monitorizar-se é adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

36. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?



11/3/22, 5:36 PM Formulário - Central Administrativa

37. O esquema relativo aos estados da ocorrência é útil e adequado *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

38. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

39. A tabela de notificações é útil e adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

40. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

https://docs.google.com/forms/d/1Rj56-FcNeesH_1or2OTSuUtdjG90NKMu5RSi6oelkMI/edit 11/14

Página de Histórico

As perguntas nesta secção encontram-se relacionadas com a página de histórico da aplicação para os utilizadores das centrais administrativas.

41. É fácil encontrar a página de histórico *

Mark only one oval.

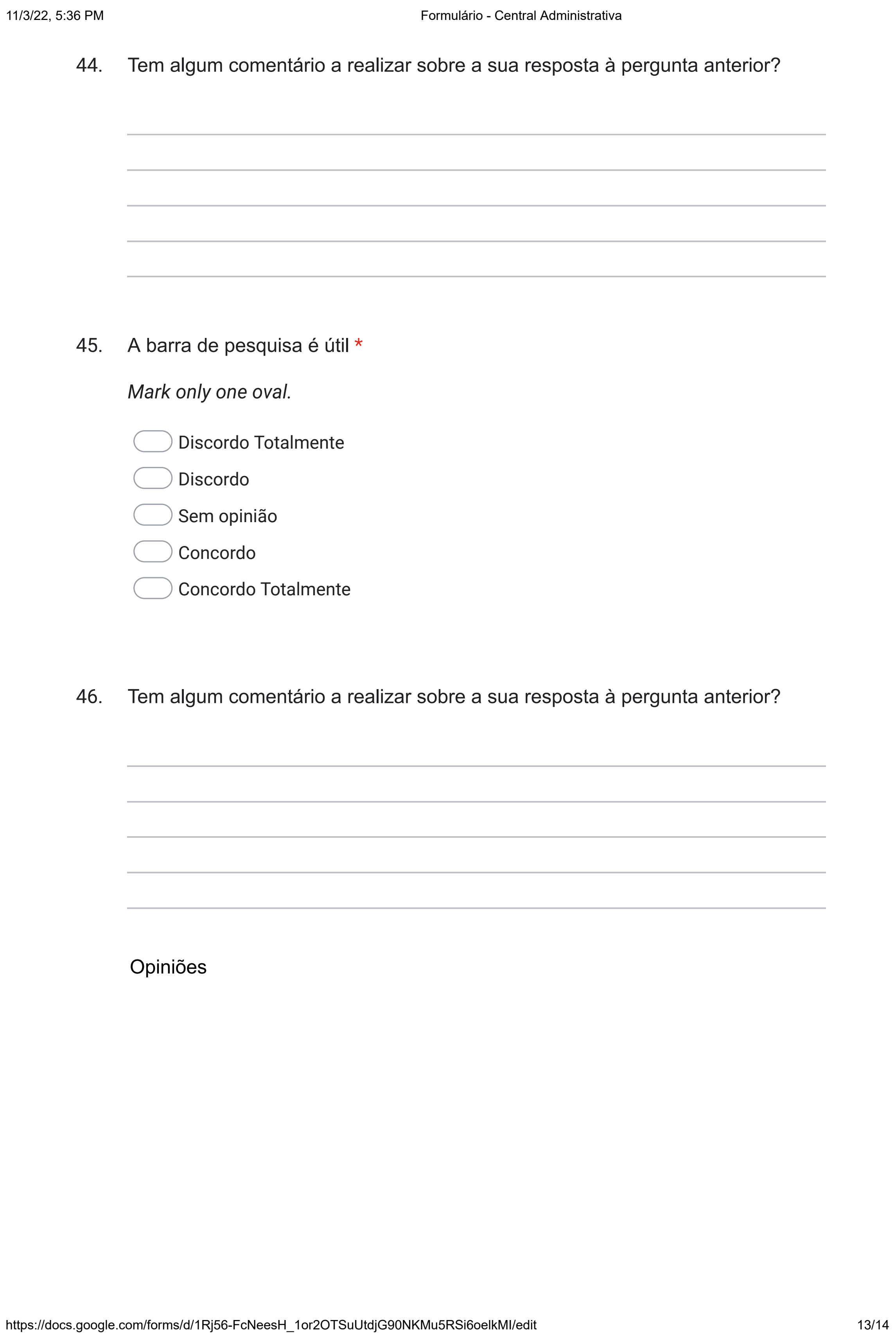
- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem opinião
- ◯ Concordo
- ◯ Concordo Totalmente

42. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

43. A informação apresentada é adequada *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem opinião
- ◯ Concordo
- ◯ Concordo Totalmente



11/3/22, 5:36 PM Formulário - Central Administrativa

44. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

45. A barra de pesquisa é útil *

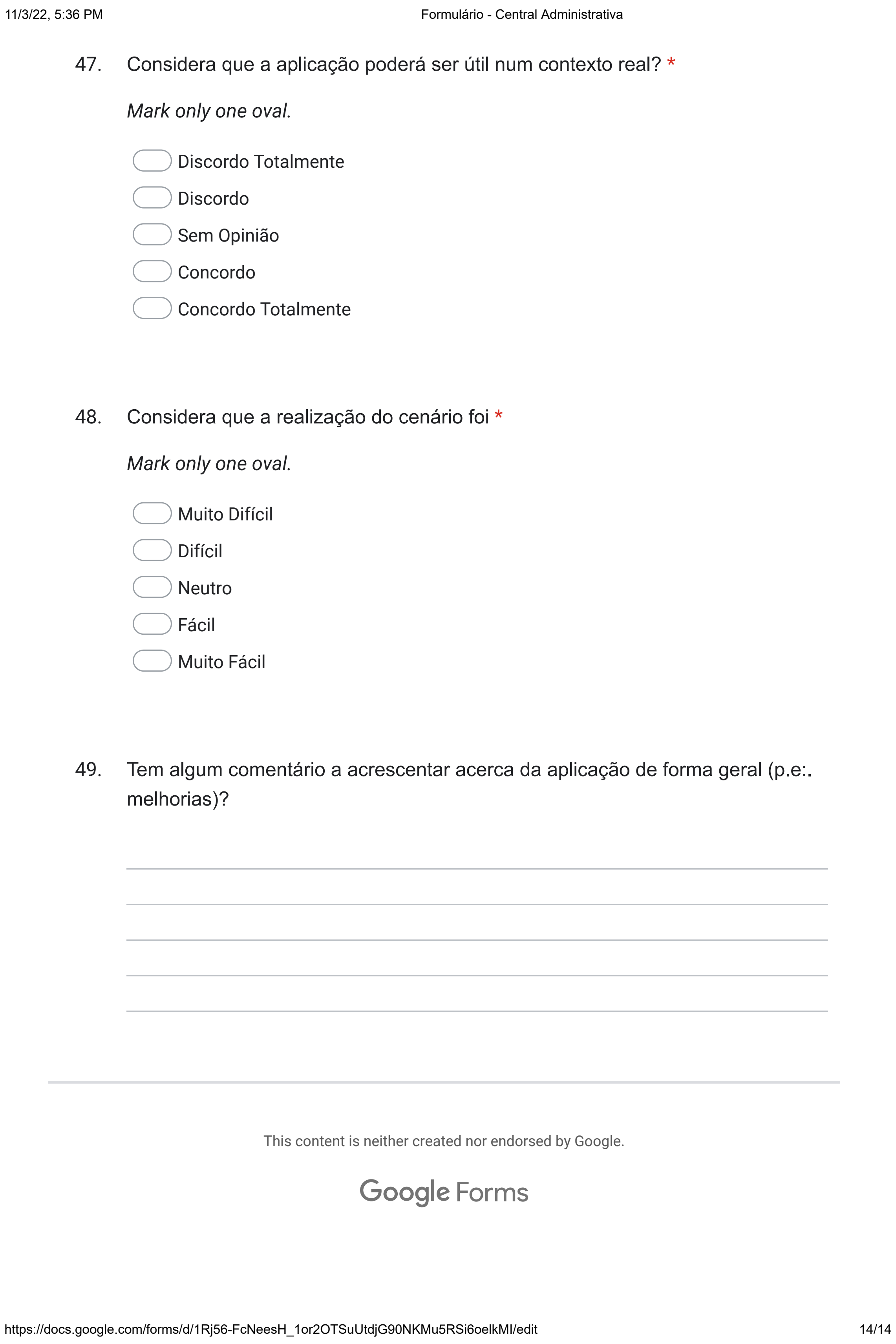
Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem opinião
- Concordo
- Concordo Totalmente

46. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

Opiniões

https://docs.google.com/forms/d/1Rj56-FcNeesH_1or2OTSuUtdjG90NKMu5RSi6oelkMI/edit 13/14



11/3/22, 5:36 PM Formulário - Central Administrativa

47. Considera que a aplicação poderá ser útil num contexto real? *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

48. Considera que a realização do cenário foi *

Mark only one oval.

- ◯ Muito Difícil
- ◯ Difícil
- ◯ Neutro
- ◯ Fácil
- ◯ Muito Fácil

49. Tem algum comentário a acrescentar acerca da aplicação de forma geral (p.e:. melhorias)?

This content is neither created nor endorsed by Google.

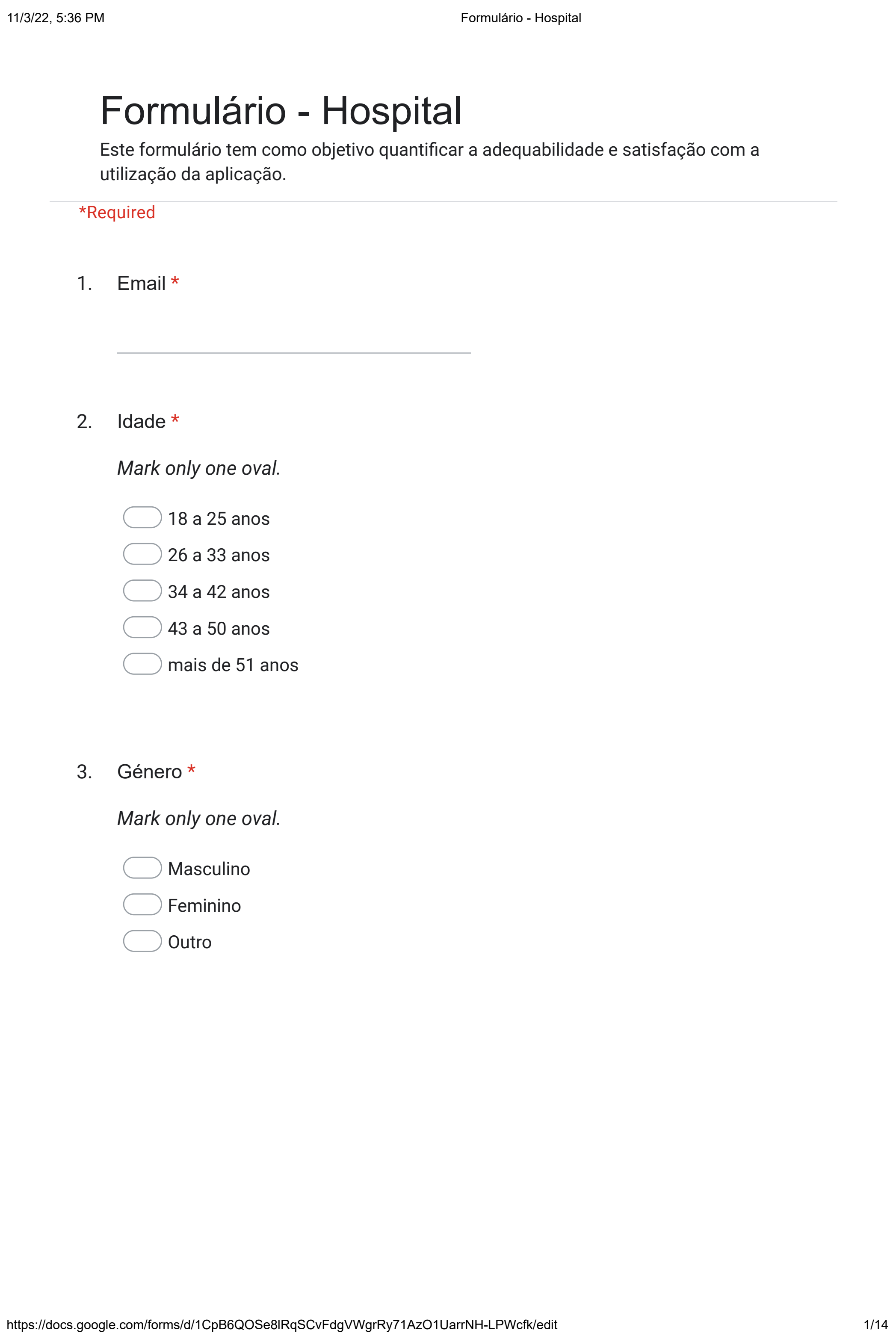
Google Forms

https://docs.google.com/forms/d/1Rj56-FcNeesH_1or2OTSuUtdjG90NKMu5RSi6oelkMI/edit 14/14

Figure G.1: Administrative Emergency Central Form Template

Appendix H

Hospital Form



Formulário - Hospital

Este formulário tem como objetivo quantificar a adequabilidade e satisfação com a utilização da aplicação.

*Required

1. Email *

   ____

2. Idade *

   Mark only one oval.

   - ◯ 18 a 25 anos
   - ◯ 26 a 33 anos
   - ◯ 34 a 42 anos
   - ◯ 43 a 50 anos
   - ◯ mais de 51 anos

3. Género *

   Mark only one oval.

   - ◯ Masculino
   - ◯ Feminino
   - ◯ Outro

4. Formação Académica *

Mark only one oval.

- ◯ Ensino Básico
- ◯ Ensino Secundário
- ◯ Bacharelato
- ◯ Licenciatura
- ◯ Mestrado
- ◯ Doutoramento

5. Profissão *

____

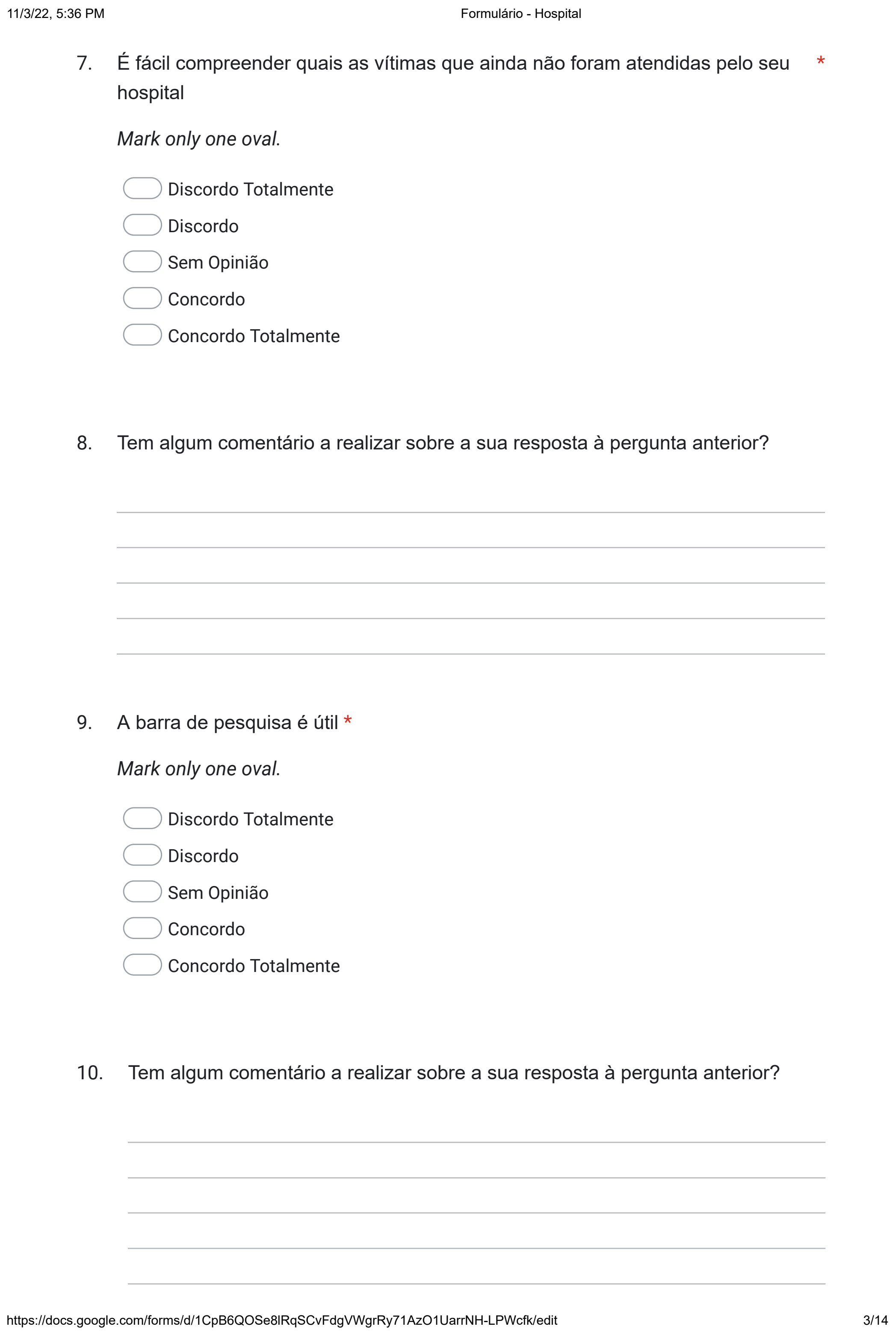
6. Anos de atividade da Profissão anteriormente introduzida *

Mark only one oval.

- ◯ Até 3 anos
- ◯ 4 a 7 anos
- ◯ 8 a 11 anos
- ◯ Mais de 11 anos

Página Inicial

As perguntas nesta secção encontram-se relacionadas com a página principal da aplicação para os utilizadores das centrais administrativas.



11/3/22, 5:36 PM Formulário - Hospital

7. É fácil compreender quais as vítimas que ainda não foram atendidas pelo seu hospital *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

8. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

9. A barra de pesquisa é útil *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

10. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

https://docs.google.com/forms/d/1CpB6QOSe8lRqSCvFdgVWgrRy71AzO1UarrNH-LPWcfk/edit 3/14

11. A tabela de pacientes do seu hospital tem a informação adequada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

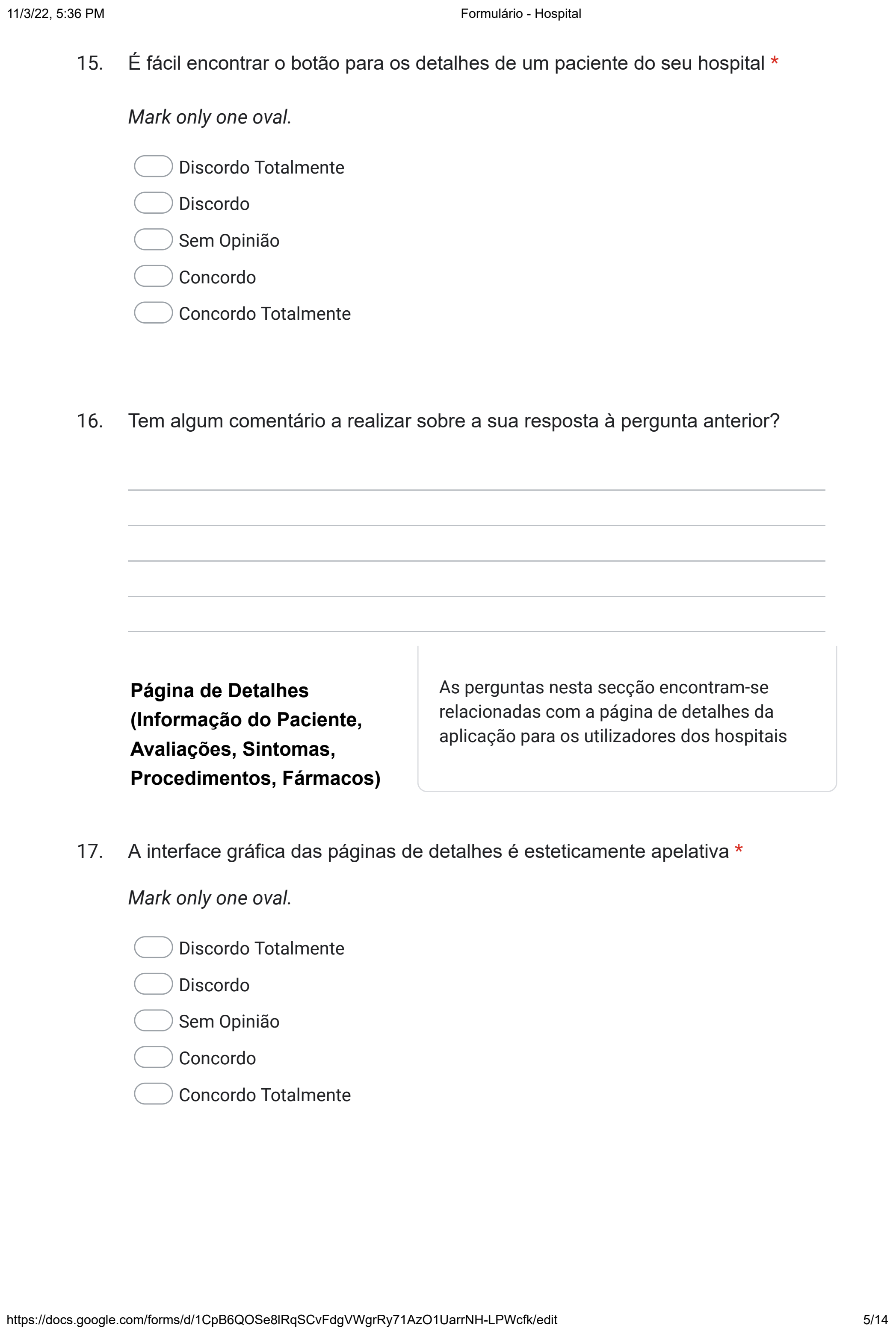
12. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

13. A interface gráfica é esteticamente apelativa *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

14. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?



11/3/22, 5:36 PM Formulário - Hospital

15. É fácil encontrar o botão para os detalhes de um paciente do seu hospital *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

16. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

Página de Detalhes (Informação do Paciente, Avaliações, Sintomas, Procedimentos, Fármacos)

As perguntas nesta secção encontram-se relacionadas com a página de detalhes da aplicação para os utilizadores dos hospitais

17. A interface gráfica das páginas de detalhes é esteticamente apelativa *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

https://docs.google.com/forms/d/1CpB6QOSe8lRqSCvFdgVWgrRy71AzO1UarrNH-LPWcfk/edit 5/14

18. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

19. A informação recebida sobre o paciente está adequada e bem estruturada *

Mark only one oval.

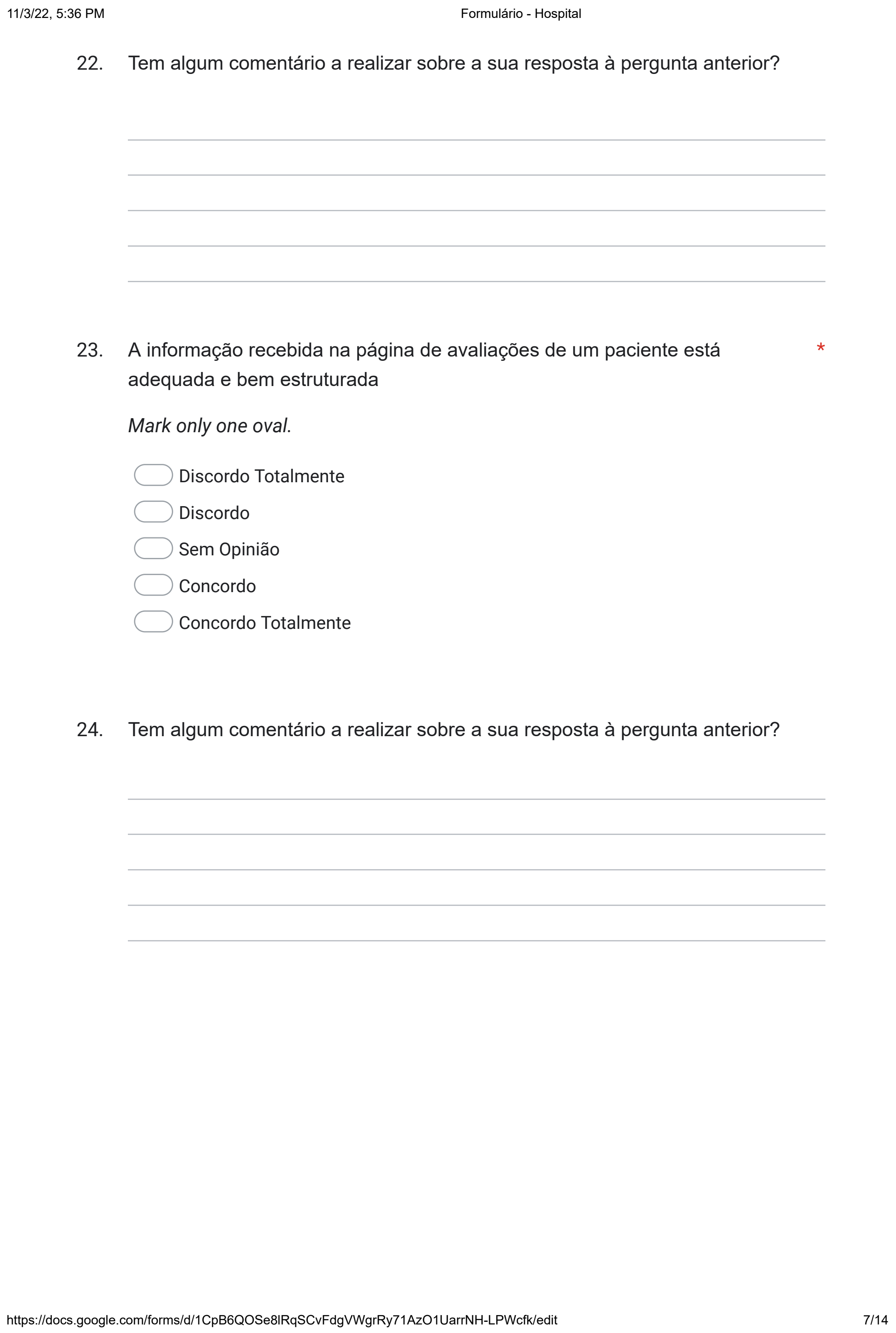
- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

20. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

21. É fácil atribuir a data e hora que o paciente foi atendido *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente



11/3/22, 5:36 PM Formulário - Hospital

22. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

23. A informação recebida na página de avaliações de um paciente está adequada e bem estruturada *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

24. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

https://docs.google.com/forms/d/1CpB6QOSe8lRqSCvFdgVWgrRy71AzO1UarrNH-LPWcfk/edit 7/14

25. Toda a informação recebida na página de avaliações de um paciente é compreensível *

Mark only one oval.

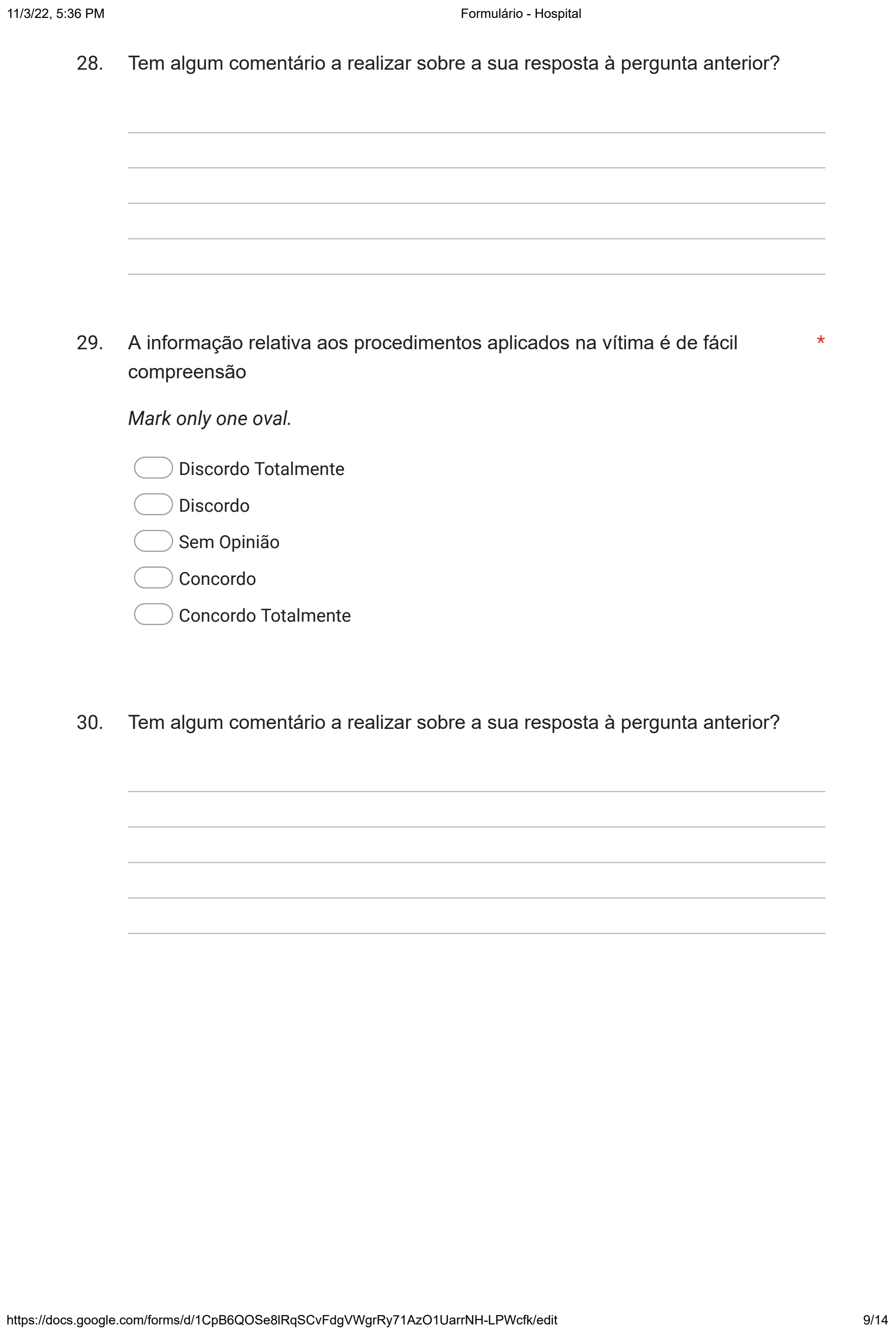
- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente

26. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

27. A imagem humana presente na página de sintomas é útil para a compreensão dos traumas de um paciente *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem Opinião
- ◯ Concordo
- ◯ Concordo Totalmente



28. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

29. A informação relativa aos procedimentos aplicados na vítima é de fácil compreensão *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

30. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

31. As tooltips presentes nos procedimentos de escala são úteis e suficientes à compreensão dos valores exibidos *

Mark only one oval.

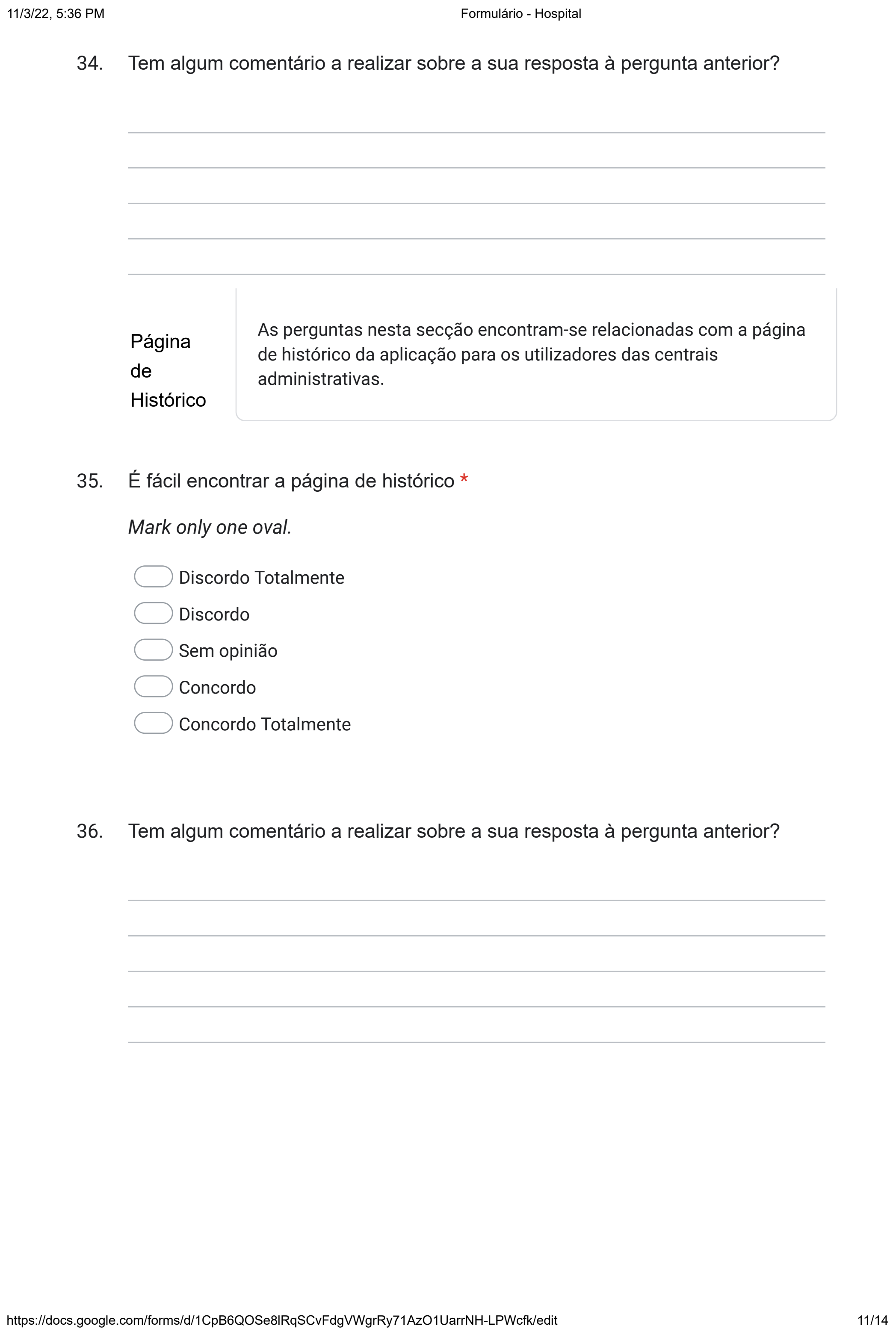
- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

32. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

33. A informação presente na tabela dos fármacos administrados é adequada e de fácil leitura *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente



11/3/22, 5:36 PM Formulário - Hospital

34. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

Página de Histórico	As perguntas nesta secção encontram-se relacionadas com a página de histórico da aplicação para os utilizadores das centrais administrativas.

35. É fácil encontrar a página de histórico *

Mark only one oval.

◯ Discordo Totalmente
◯ Discordo
◯ Sem opinião
◯ Concordo
◯ Concordo Totalmente

36. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

https://docs.google.com/forms/d/1CpB6QOSe8lRqSCvFdgVWgrRy71AzO1UarrNH-LPWcfk/edit 11/14

37. A informação apresentada é adequada *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem opinião
- ◯ Concordo
- ◯ Concordo Totalmente

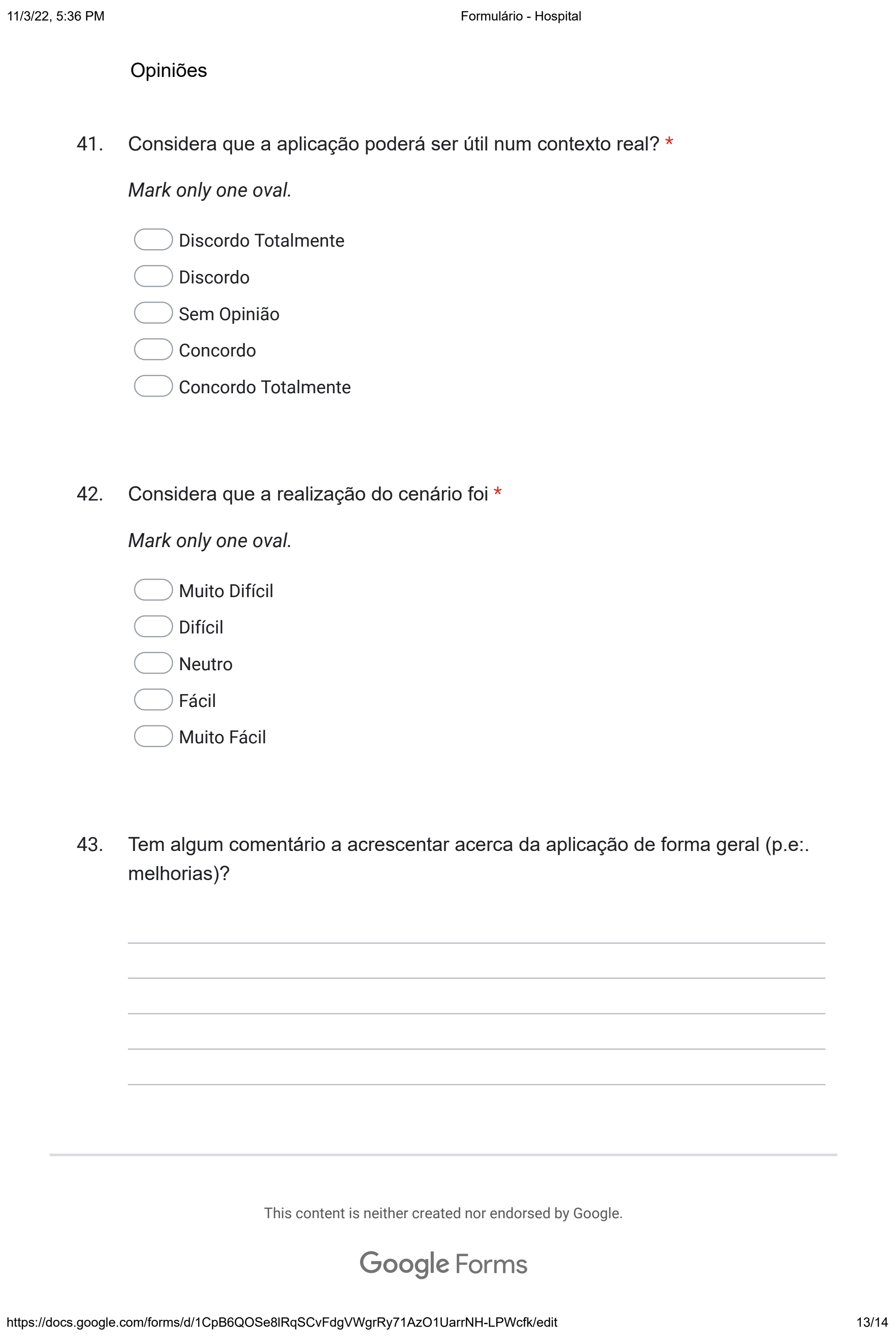
38. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?

39. A barra de pesquisa é útil *

Mark only one oval.

- ◯ Discordo Totalmente
- ◯ Discordo
- ◯ Sem opinião
- ◯ Concordo
- ◯ Concordo Totalmente

40. Tem algum comentário a realizar sobre a sua resposta à pergunta anterior?



11/3/22, 5:36 PM Formulário - Hospital

Opiniões

41. Considera que a aplicação poderá ser útil num contexto real? *

Mark only one oval.

- Discordo Totalmente
- Discordo
- Sem Opinião
- Concordo
- Concordo Totalmente

42. Considera que a realização do cenário foi *

Mark only one oval.

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

43. Tem algum comentário a acrescentar acerca da aplicação de forma geral (p.e:. melhorias)?

This content is neither created nor endorsed by Google.

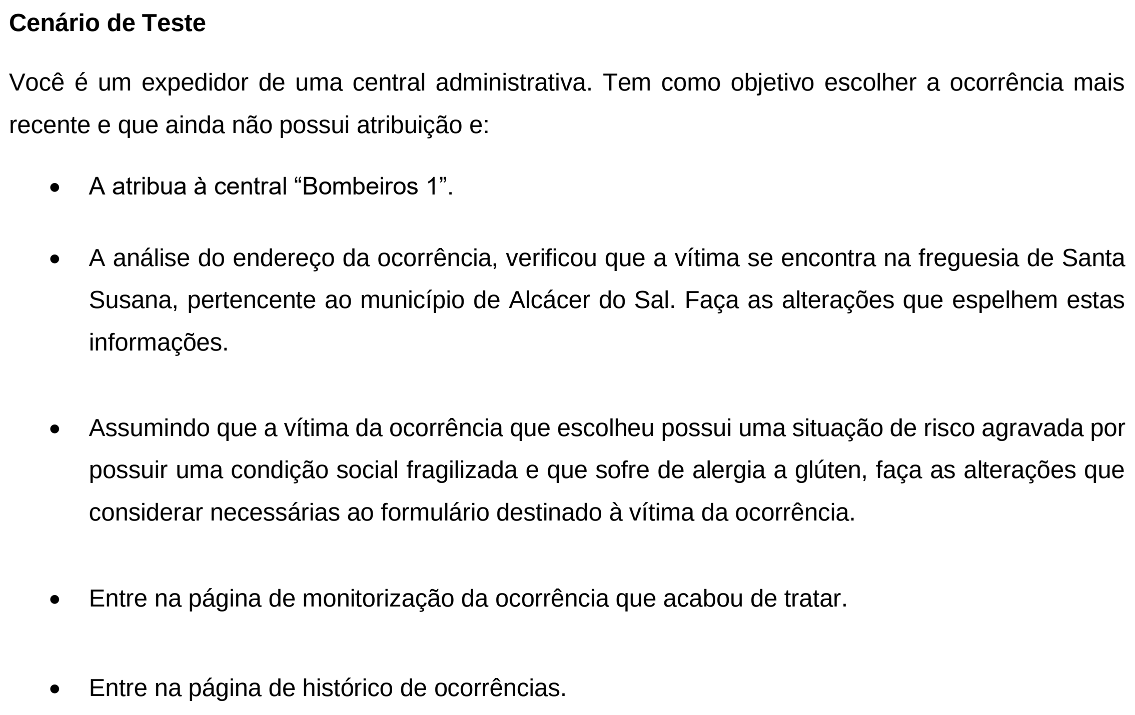
Google Forms

https://docs.google.com/forms/d/1CpB6QOSe8lRqSCvFdgVWgrRy71AzO1UarrNH-LPWcfk/edit 13/14

Figure H.1: Hospital Form Template

Appendix I

Test Protocols



Cenário de Teste

Você é um expedidor de uma central administrativa. Tem como objetivo escolher a ocorrência mais recente e que ainda não possui atribuição e:

- A atribua à central “Bombeiros 1”.
- A análise do endereço da ocorrência, verificou que a vítima se encontra na freguesia de Santa Susana, pertencente ao município de Alcácer do Sal. Faça as alterações que espelhem estas informações.
- Assumindo que a vítima da ocorrência que escolheu possui uma situação de risco agravada por possuir uma condição social fragilizada e que sofre de alergia a glúten, faça as alterações que considerar necessárias ao formulário destinado à vítima da ocorrência.
- Entre na página de monitorização da ocorrência que acabou de tratar.
- Entre na página de histórico de ocorrências.

Figure I.1: Administrative Emergency Central Test Protocol

Ações de Teste

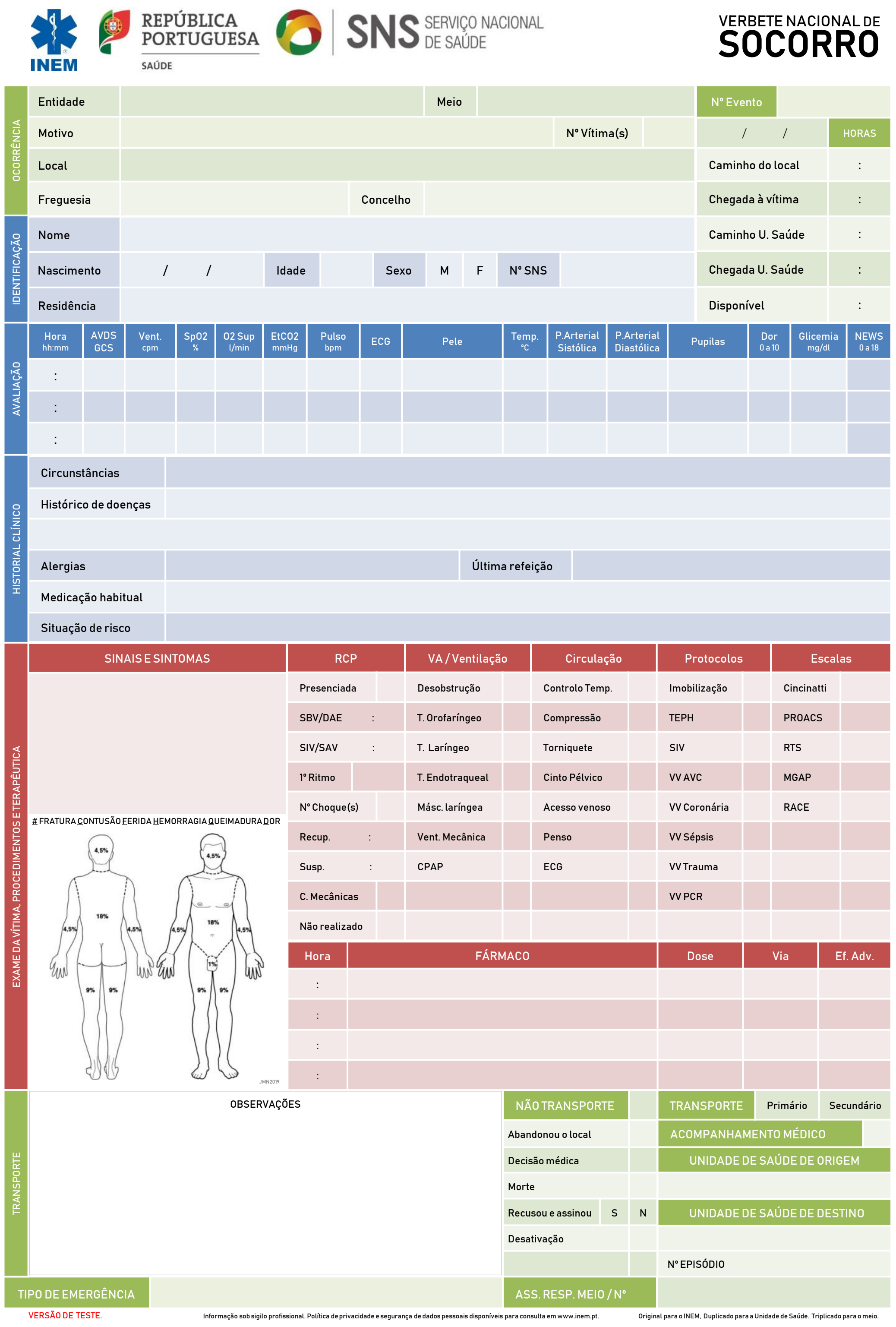
Você é um trabalhador no hospital da Luz. Tem como objetivo assegurar que recebeu todas as informações relevantes das vítimas chegadas ao seu hospital.

- Escolha a vítima com o nome Tiago Guimarães que chegou ao seu hospital e que ainda não foi dado como atendido.
- Preencha a sua data e hora de atendimento com a data e hora atual a que está a realizar este teste.
- A vítima que acabou de chegar possui informações relevantes a serem recebidas. Verifique que todas as informações mencionadas possuem os valores corretos na sua aplicação.
  - O paciente de nome Tiago Guimarães, reside em Loures e possui 26 anos. Nasceu a 26/04/1996. O seu historial clínico é marcado pela sua alergia a pólen, com histórico de problemas relacionados com asma e que a sua última refeição foi leite com mel no dia 14/09/2022 às 11:23:00 da manhã. Ele é acompanhado por um médico, tomando Zileutona recorrentemente e o seu tipo de transporte foi primário.
  - O paciente foi alvo de uma avaliação no dia 14/09/2022 às 12:30:00. A sua ventilação estava nos 10 *cpm*, o seu SpO2 estava nos 91%, o valor de O2 Sup estava nos 10 *l/min* e o seu pulso nos 57 *bpm*. Não foi realizado ECG e a sua pele encontrava-se pálida. A sua temperatura estava nos 38.7 ºC e a sua pressão arterial sistólica nos 84 e diastólica nos 47. As pupilas encontravam-se em midríase e glicemia a 93 *mg/dl*. A sua escala de Glasgow apontava com 3 para os Olhos, 2 para Verbal e 4 para Motor com um total de 9.
  - Os sintomas da vítima descrevem-se como calafrios e estado descontrolado febril. A nível de traumas sofria um trauma no epigástrio, com um tipo de lesão D, e o trauma encontrava-se fechado.
  - A nível de procedimentos RCP, estes foram presenciados, foram realizados 3 choques e SBV/DAE às 12:40 do dia 14/09/2022.
  - A nível de procedimentos de ventilação foi realizado desobstrução, tubo endotraqueal, ventilação mecânica e CPAP.
  - A nível de procedimentos de circulação foi realizado controlo de temperatura, compressão, cinto pélvico e acesso venoso.
  - A nível de procedimentos de protocolos, foi realizada imobilização, TEPH, VV Trauma e VV PCR.
  - A nível de procedimentos de escalas, cincinatti possuía o valor 3, PROACS o valor 2, RTS o valor 10, MGAP o valor 17 e RACE o valor 3.
  - A nível de fármacos, foi administrado aspirina no dia 14/09/2022 às 12:50:00, dose de 500 mg, por via oral e não possuiu efeitos adversos.
- Encontre a vítima que acabou de consultar no histórico de pacientes do seu hospital.

Figure I.2: Hospital Test Protocol

Annex I

Verbete INEM



INEM REPÚBLICA PORTUGUESA SAÚDE SNS SERVIÇO NACIONAL DE SAÚDE

VERBETE NACIONAL DE
SOCORRO

OCORRÊNCIA

Entidade		Meio		Nº Evento	
Motivo		Nº Vítima(s)		/ /	HORAS
Local				Caminho do local	:
Freguesia		Concelho		Chegada à vítima	:

IDENTIFICAÇÃO

Nome								Caminho U. Saúde	:
Nascimento	/ /	Idade		Sexo	M	F	Nº SNS	Chegada U. Saúde	:
Residência								Disponível	:

AVALIAÇÃO

Hora hh:mm	AVDS GCS	Vent. cpm	SpO2 %	O2 Sup l/min	EtCO2 mmHg	Pulso bpm	ECG	Pele	Temp. °C	P.Arterial Sistólica	P.Arterial Diastólica	Pupilas	Dor 0 a 10	Glicemia mg/dl	NEWS 0 a 18
:															
:															
:															

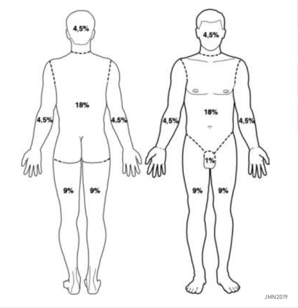
HISTORIAL CLÍNICO

Circunstâncias			
Histórico de doenças			
Alergias		Última refeição	
Medicação habitual			
Situação de risco			

EXAME DA VÍTIMA, PROCEDIMENTOS E TERAPÊUTICA

SINAIS E SINTOMAS

# FRATURA CONTUSÃO FERIDA HEMORRAGIA QUEIMADURA DOR



RCP		VA / Ventilação		Circulação		Protocolos		Escalas	
Presenciada		Desobstrução		Controlo Temp.		Imobilização		Cincinatti	
SBV/DAE :		T. Orofaríngeo		Compressão		TEPH		PROACS	
SIV/SAV :		T. Laríngeo		Torniquete		SIV		RTS	
1º Ritmo		T. Endotraqueal		Cinto Pélvico		VV AVC		MGAP	
Nº Choque(s)		Másc. laríngea		Acesso venoso		VV Coronária		RACE	
Recup. :		Vent. Mecânica		Penso		VV Sépsis			
Susp. :		CPAP		ECG		VV Trauma			
C. Mecânicas						VV PCR			
Não realizado									

Hora	FÁRMACO	Dose	Via	Ef. Adv.
:				
:				
:				
:				

TRANSPORTE

OBSERVAÇÕES	NÃO TRANSPORTE		TRANSPORTE	Primário	Secundário
	Abandonou o local		ACOMPANHAMENTO MÉDICO		
	Decisão médica		UNIDADE DE SAÚDE DE ORIGEM		
	Morte				
	Recusou e assinou	S N	UNIDADE DE SAÚDE DE DESTINO		
	Desativação				
			Nº EPISÓDIO		
TIPO DE EMERGÊNCIA	ASS. RESP. MEIO / Nº				

VERSÃO DE TESTE.

Informação sob sigilo profissional. Política de privacidade e segurança de dados pessoais disponíveis para consulta em www.inem.pt.

Original para o INEM. Duplicado para a Unidade de Saúde. Triplicado para o meio.

RACE (eventual necessidade de trombectomia se ≥5)

	ESQUERDA	DIREITA	VALOR
Paresia facial	Ausente	Ausente	0
	ligeira	ligeira	1
	Moderada/severa	Moderada/severa	2
Paresia MS	Ausente/Ligeiro (>10seg)	Ausente/Lig (>10seg)	0
	Moderada (<10seg)	Moderada (<10seg)	1
	Severa (não levanta)	Severa (não levanta)	2
Paresia MI	Ausente/Ligeiro (>5seg)	Ausente/Ligeiro (>5seg)	0
	Moderada (<5seg)	Moderada (<5seg)	1
	Severa (não levanta)	Severa (não levanta)	2
Desvio oculocefálico	Direto ausente	Esquerdo ausente	0
	Direito presente	Esquerdo presente	1
Agnosia Afasia	Reconhece o braço E o défice	Afasia obedece a 2 ordens	0
	Não reconhece o braço OU o défice	Afasia obedece a 1 ordens	1
	Não reconhece NEM braço NEM o défice	Não executa ordens	2

NEWS (risco elevado de mortalidade se ≥3)

PARÂMETRO FISIOLÓGICO	3	2	1	0	1	2	3
FREQUÊNCIA RESPIRATÓRIA	≤8		9-11	12-20		21-24	≥25
SPO2	≤91	92-93	94-95	≥96			
O2 SUPLEMENTAR		Sim		Não			
TEMPERATURA	≤35		35.1-36.0	36.1-38.0	38.1-39.0	≥39.1	
PRESSÃO ARTERIAL SISTÓLICA	≤90	19-100	101-110	111-219			≥220
FREQUÊNCIA CARDÍACA	≤40		41-50	51-90	91-110	111-130	≥131
NÍVEL DE CONSCIÊNCIA				A			V,D ou S

GRAVIDADE	0	Nula	1, 2 ou 4	Baixa	≥5 ou 3	Moderada	≥7	Elevada

VIA VERDE SEPSIS (suspeita de VVS se ≥3 com positivo nos três campos)

TEMPERATURA		CLÍNICA	
< 35	1	Cefaleia	1
> 38	1	Alteração do estado de consciência	1
		Dispneia / Tosse	1
		Dor abdominal	1
GRAVIDADE		Icterícia	1
Lactato > 2mmol/L	1	Disúria / Polaquiúria	1
TAS < 90mmHg	1	Dor lombar	1
PaO2 < 60mmHg	1	Sinais inflamatórios cutâneos	1
SatO2 < 90%	1	Critério clínico responsável	1

MGAP (referenciação a centro de trauma se <18)

MECANISMO LESÃO		IDADE		GCS	PAS		TOTAL	
Penetrante	0	<60	5	3 a 15	>120	5	23-29	baixo
Fechado	4	>60	0		60-123	3	18-22	médio
					<60	0	<18	alto

ESCALA DE COMA DE GLASGOW ATUALIZADA

OLHOS		VERBAL		MOTOR	
Espontânea	4	Orientada	5	A ordens	6
Ao som	3	Confusa	4	Localizadora	5
À Pressão	2	Palavras	3	Flexão normal	4
Ausente	1	Sons	2	Flexão anormal	3
Não testável	NT	Ausente	1	Extensão	2
		Não testável	NT	Ausente	1
				Não testável	NT

ESCALA DE CINCINNATI (positivo de 1 a 3)

ALTERAÇÃO	SIM	NÃO
Paresia facial	1	0
Queda de membro superior	1	0
Alteração na fala	1	0

ISBAR (transição de informação na transmissão de cuidados)

I DENTIFICAÇÃO	Profissional de saúde Nome e idade da vítima Situação que motivou a ocorrência
S ITUAÇÃO ATUAL	Descrição da condição clínica Principais alterações Sinais e sintomas
B ACKGROUND	Histórico de doenças Medicação habitual Alergias
A VALIAÇÃO	Tipo de emergência identificada Procedimentos realizados Protocolos instituídos
R ECOMENDAÇÕES	Recomendações Estudos ou avaliações indicadas Proposta de tratamento

RTS (referenciação a centro de trauma se ≥10)

GCS		FR		PAS	
13 a 15	4	10 a 29	4	> 89	4
9 a 12	3	> 29	3	76 a 89	3
6 a 8	2	6 a 9	2	50 a 75	2
4 a 5	1	1 a 5	1	1 a 49	1
3	0	0	0	0	0

ESCALA PROACS (risco elevado de mortalidade se ≥3)

IDADE	<72	0
	> ou = a 72	1
TAS	<116mmHg	1
	> ou = a 116mmHg	0
CLASSE DE KILLIP	1	0
	2	1
	3	1
	4	3
ELEVAÇÃO DE ST	sim	1
	Não	0

DECLARAÇÃO DE RECUSA | DECLARATION OF REFUSAL

(Quando sublinhado, risque o que não interessa)

Para os devidos efeitos, declaro ter sido informado(a) dos riscos inerentes à minha decisão de recusar o(s) procedimento(s) recomendados/transporte de ambulância e assumir toda a responsabilidade pelas eventuais consequências.

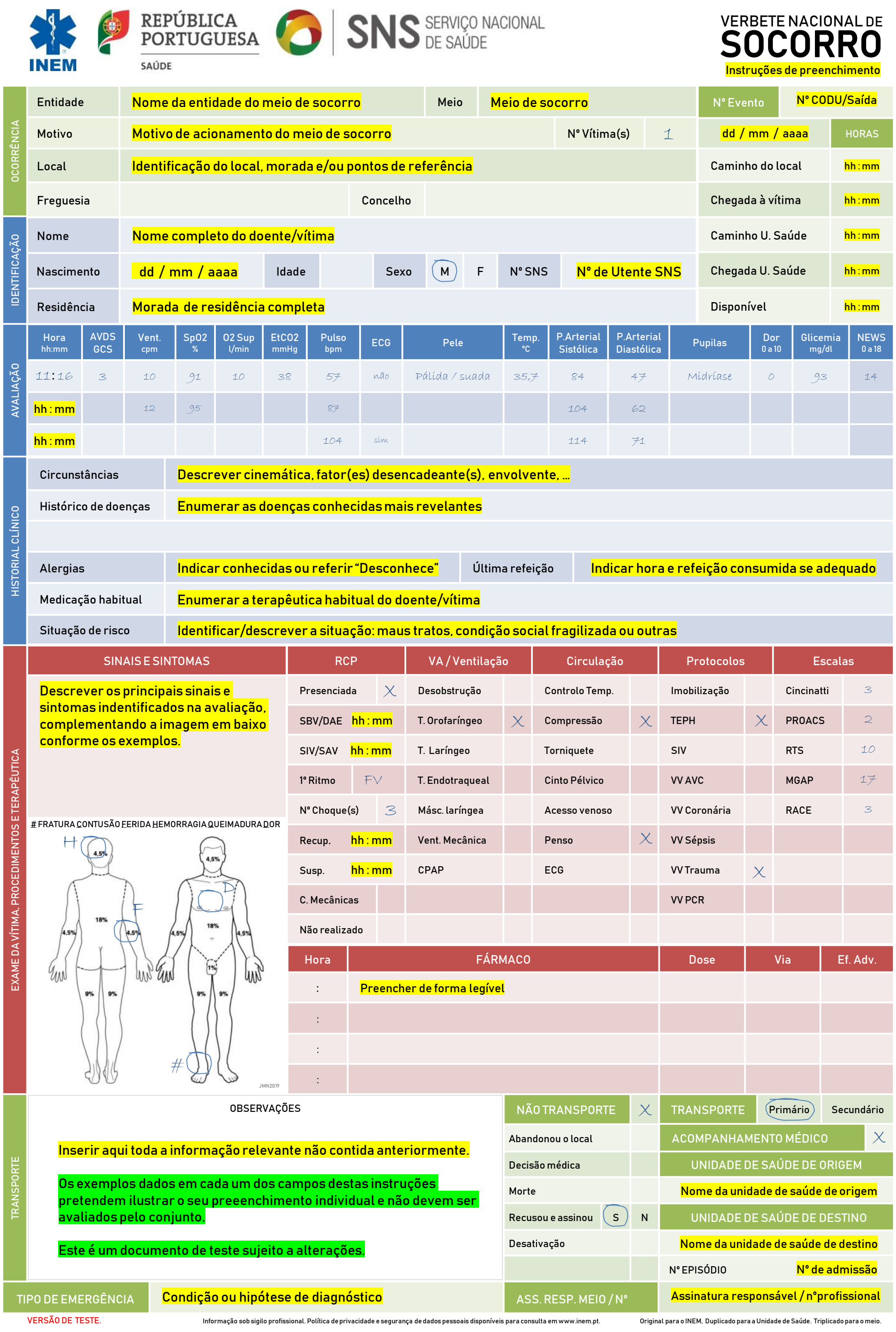
For the proper effects, I declare that I was informed of the riskiness of my decision of refusing the medical procedures/transportation by ambulance and I take full responsibility for possible consequences.

Nome do doente/representante legal | Patient/legal representative of the patient ____________________

Documento de identificação/n.º | ID Document/n.º ____________________

Assinatura | Signature ____________________ / /

Informação sob sigilo profissional. Política de privacidade e segurança de dados pessoais disponíveis para consulta em www.inem.pt.



INEM — REPÚBLICA PORTUGUESA SAÚDE — SNS SERVIÇO NACIONAL DE SAÚDE

VERBETE NACIONAL DE
SOCORRO
Instruções de preenchimento

OCORRÊNCIA

Entidade	Nome da entidade do meio de socorro	Meio	Meio de socorro	Nº Evento	Nº CODU/Saída
Motivo	Motivo de acionamento do meio de socorro	Nº Vítima(s)	1	dd / mm / aaaa	HORAS
Local	Identificação do local, morada e/ou pontos de referência			Caminho do local	hh : mm
Freguesia		Concelho		Chegada à vítima	hh : mm

IDENTIFICAÇÃO

Nome	Nome completo do doente/vítima							Caminho U. Saúde	hh : mm
Nascimento	dd / mm / aaaa	Idade		Sexo	[✓ M] F	Nº SNS	Nº de Utente SNS	Chegada U. Saúde	hh : mm
Residência	Morada de residência completa							Disponível	hh : mm

AVALIAÇÃO

Hora hh:mm	AVDS GCS	Vent. cpm	SpO2 %	O2 Sup l/min	EtCO2 mmHg	Pulso bpm	ECG	Pele	Temp. °C	P.Arterial Sistólica	P.Arterial Diastólica	Pupilas	Dor 0 a 10	Glicemia mg/dl	NEWS 0 a 18
11:16	3	10	91	10	38	57	não	Pálida / suada	35,7	84	47	Midríase	0	93	14
hh : mm		12	95			87				104	62				
hh : mm						104	sim			114	71				

HISTORIAL CLÍNICO

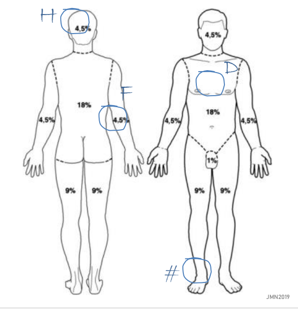
Circunstâncias	Descrever cinemática, fator(es) desencadeante(s), envolvente, ...		
Histórico de doenças	Enumerar as doenças conhecidas mais revelantes		
Alergias	Indicar conhecidas ou referir "Desconhece"	Última refeição	Indicar hora e refeição consumida se adequado
Medicação habitual	Enumerar a terapêutica habitual do doente/vítima		
Situação de risco	Identificar/descrever a situação: maus tratos, condição social fragilizada ou outras		

EXAME DA VÍTIMA, PROCEDIMENTOS E TERAPÊUTICA

SINAIS E SINTOMAS

Descrever os principais sinais e sintomas indentificados na avaliação, complementando a imagem em baixo conforme os exemplos.

FRATURA C CONTUSÃO F FERIDA H HEMORRAGIA Q QUEIMADURA D DOR



RCP		VA / Ventilação		Circulação		Protocolos		Escalas	
Presenciada	X	Desobstrução		Controlo Temp.		Imobilização		Cincinatti	3
SBV/DAE	hh : mm	T. Orofaríngeo	X	Compressão	X	TEPH	X	PROACS	2
SIV/SAV	hh : mm	T. Laríngeo		Torniquete		SIV		RTS	10
1º Ritmo	FV	T. Endotraqueal		Cinto Pélvico		VV AVC		MGAP	17
Nº Choque(s)	3	Másc. laríngea		Acesso venoso		VV Coronária		RACE	3
Recup.	hh : mm	Vent. Mecânica		Penso	X	VV Sépsis			
Susp.	hh : mm	CPAP		ECG		VV Trauma	X		
C. Mecânicas						VV PCR			
Não realizado									

Hora	FÁRMACO	Dose	Via	Ef. Adv.
:	Preencher de forma legível			
:				
:				
:				

TRANSPORTE

OBSERVAÇÕES

Inserir aqui toda a informação relevante não contida anteriormente.

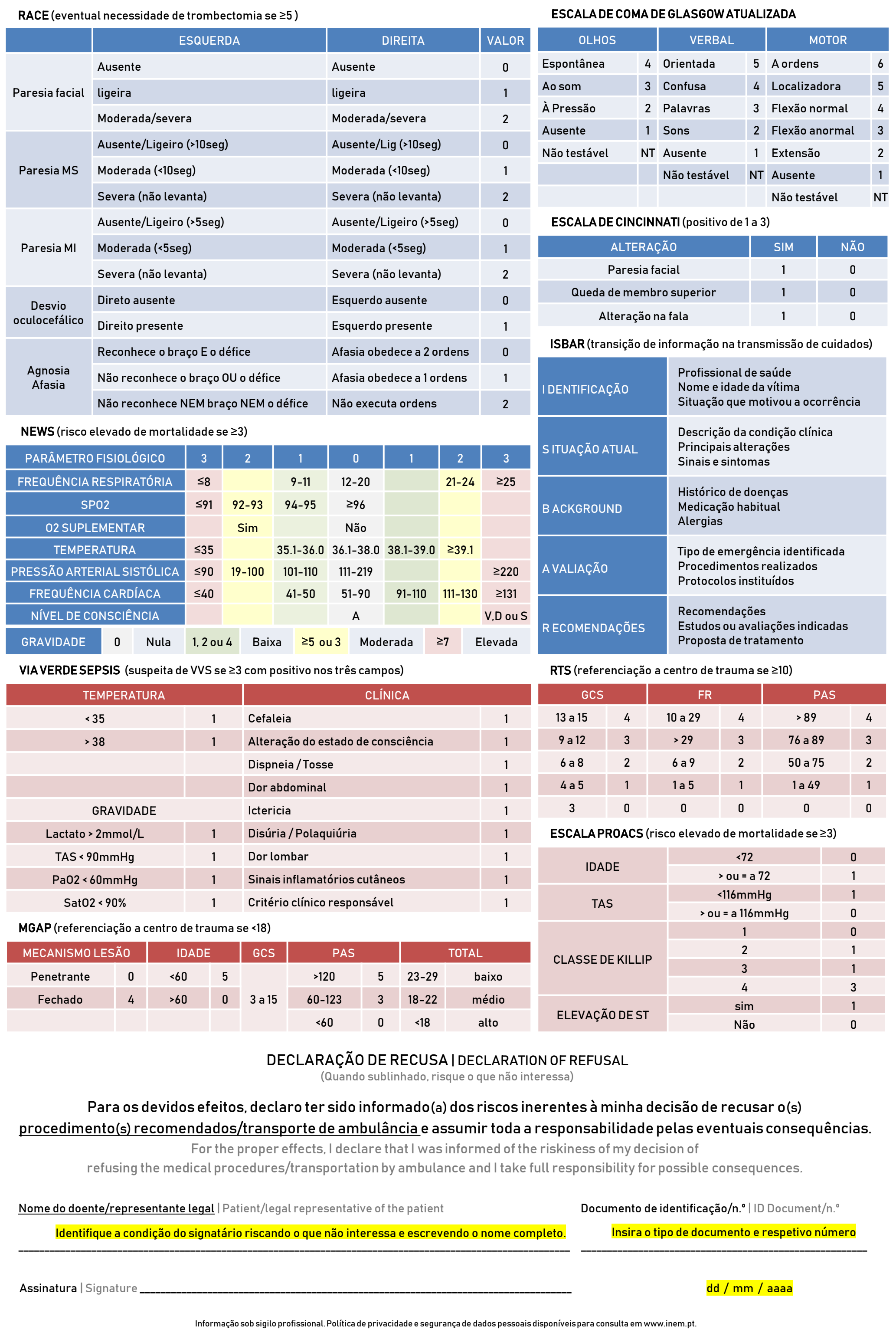
Os exemplos dados em cada um dos campos destas instruções pretendem ilustrar o seu preeenchimento individual e não devem ser avaliados pelo conjunto.

Este é um documento de teste sujeito a alterações.

NÃO TRANSPORTE	X	TRANSPORTE	[✓ Primário] Secundário
Abandonou o local		ACOMPANHAMENTO MÉDICO	X
Decisão médica		UNIDADE DE SAÚDE DE ORIGEM	
Morte		Nome da unidade de saúde de origem	
Recusou e assinou	[✓ S] N	UNIDADE DE SAÚDE DE DESTINO	
Desativação		Nome da unidade de saúde de destino	
		Nº EPISÓDIO	Nº de admissão

TIPO DE EMERGÊNCIA	Condição ou hipótese de diagnóstico	ASS. RESP. MEIO / Nº	Assinatura responsável / nºprofissional

VERSÃO DE TESTE. Informação sob sigilo profissional. Política de privacidade e segurança de dados pessoais disponíveis para consulta em www.inem.pt. Original para o INEM. Duplicado para a Unidade de Saúde. Triplicado para o meio.



RACE (eventual necessidade de trombectomia se ≥5)

	ESQUERDA	DIREITA	VALOR
Paresia facial	Ausente	Ausente	0
	ligeira	ligeira	1
	Moderada/severa	Moderada/severa	2
Paresia MS	Ausente/Ligeiro (>10seg)	Ausente/Lig (>10seg)	0
	Moderada (<10seg)	Moderada (<10seg)	1
	Severa (não levanta)	Severa (não levanta)	2
Paresia MI	Ausente/Ligeiro (>5seg)	Ausente/Ligeiro (>5seg)	0
	Moderada (<5seg)	Moderada (<5seg)	1
	Severa (não levanta)	Severa (não levanta)	2
Desvio oculocefálico	Direto ausente	Esquerdo ausente	0
	Direito presente	Esquerdo presente	1
Agnosia Afasia	Reconhece o braço E o défice	Afasia obedece a 2 ordens	0
	Não reconhece o braço OU o défice	Afasia obedece a 1 ordens	1
	Não reconhece NEM braço NEM o défice	Não executa ordens	2

ESCALA DE COMA DE GLASGOW ATUALIZADA

OLHOS		VERBAL		MOTOR	
Espontânea	4	Orientada	5	A ordens	6
Ao som	3	Confusa	4	Localizadora	5
À Pressão	2	Palavras	3	Flexão normal	4
Ausente	1	Sons	2	Flexão anormal	3
Não testável	NT	Ausente	1	Extensão	2
		Não testável	NT	Ausente	1
				Não testável	NT

ESCALA DE CINCINNATI (positivo de 1 a 3)

ALTERAÇÃO	SIM	NÃO
Paresia facial	1	0
Queda de membro superior	1	0
Alteração na fala	1	0

ISBAR (transição de informação na transmissão de cuidados)

I DENTIFICAÇÃO	Profissional de saúde Nome e idade da vítima Situação que motivou a ocorrência
S ITUAÇÃO ATUAL	Descrição da condição clínica Principais alterações Sinais e sintomas
B ACKGROUND	Histórico de doenças Medicação habitual Alergias
A VALIAÇÃO	Tipo de emergência identificada Procedimentos realizados Protocolos instituídos
R ECOMENDAÇÕES	Recomendações Estudos ou avaliações indicadas Proposta de tratamento

NEWS (risco elevado de mortalidade se ≥3)

PARÂMETRO FISIOLÓGICO	3	2	1	0	1	2	3
FREQUÊNCIA RESPIRATÓRIA	≤8		9-11	12-20		21-24	≥25
SPO2	≤91	92-93	94-95	≥96			
O2 SUPLEMENTAR		Sim		Não			
TEMPERATURA	≤35		35.1-36.0	36.1-38.0	38.1-39.0	≥39.1	
PRESSÃO ARTERIAL SISTÓLICA	≤90	19-100	101-110	111-219			≥220
FREQUÊNCIA CARDÍACA	≤40		41-50	51-90	91-110	111-130	≥131
NÍVEL DE CONSCIÊNCIA				A			V,D ou S

GRAVIDADE	0	Nula	1, 2 ou 4	Baixa	≥5 ou 3	Moderada	≥7	Elevada

VIA VERDE SEPSIS (suspeita de VVS se ≥3 com positivo nos três campos)

TEMPERATURA		CLÍNICA	
< 35	1	Cefaleia	1
> 38	1	Alteração do estado de consciência	1
		Dispneia / Tosse	1
		Dor abdominal	1
GRAVIDADE		Icterícia	1
Lactato > 2mmol/L	1	Disúria / Polaquiúria	1
TAS < 90mmHg	1	Dor lombar	1
PaO2 < 60mmHg	1	Sinais inflamatórios cutâneos	1
SatO2 < 90%	1	Critério clínico responsável	1

RTS (referenciação a centro de trauma se ≥10)

GCS		FR		PAS	
13 a 15	4	10 a 29	4	> 89	4
9 a 12	3	> 29	3	76 a 89	3
6 a 8	2	6 a 9	2	50 a 75	2
4 a 5	1	1 a 5	1	1 a 49	1
3	0	0	0	0	0

MGAP (referenciação a centro de trauma se <18)

MECANISMO LESÃO		IDADE		GCS	PAS		TOTAL	
Penetrante	0	<60	5	3 a 15	>120	5	23-29	baixo
Fechado	4	>60	0		60-123	3	18-22	médio
					<60	0	<18	alto

ESCALA PROACS (risco elevado de mortalidade se ≥3)

IDADE	<72	0
	> ou = a 72	1
TAS	<116mmHg	1
	> ou = a 116mmHg	0
CLASSE DE KILLIP	1	0
	2	1
	3	1
	4	3
ELEVAÇÃO DE ST	sim	1
	Não	0

DECLARAÇÃO DE RECUSA | DECLARATION OF REFUSAL
(Quando sublinhado, risque o que não interessa)

Para os devidos efeitos, declaro ter sido informado(a) dos riscos inerentes à minha decisão de recusar o(s) procedimento(s) recomendados/transporte de ambulância e assumir toda a responsabilidade pelas eventuais consequências.
For the proper effects, I declare that I was informed of the riskiness of my decision of refusing the medical procedures/transportation by ambulance and I take full responsibility for possible consequences.

Nome do doente/representante legal | Patient/legal representative of the patient
Identifique a condição do signatário riscando o que não interessa e escrevendo o nome completo.
____

Documento de identificação/n.º | ID Document/n.º
Insira o tipo de documento e respetivo número
____

Assinatura | Signature ____ dd / mm / aaaa

Informação sob sigilo profissional. Política de privacidade e segurança de dados pessoais disponíveis para consulta em www.inem.pt.

Figure I.1: *Verbete INEM* Template

2022 SIREPH: An Integrated Approach to the Pre-Hospital Emergency Framework JOÃO PEREIRA