DM

Psyment
Security design and implementation of a psychological assessment supportive web platform

MASTER DISSERTATION

Eduardo Alexandre Romão Coelho
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

January | 2024

Psyment
Security design and implementation of a psychological assessment supportive web platform

MASTER DISSERTATION

Eduardo Alexandre Romão Coelho
MASTER IN INFORMATICS ENGINEERING

SUPERVISION
Luís Duarte Andrade Ferreira

Faculdade de Ciências Exatas e da Engenharia

Mestrado em Engenharia Informática

2022/2023

Psyment: Security design and implementation of a psychological assessment supportive web platform

Projeto de dissertação de mestrado

Eduardo Alexandre Romão Coelho nº 2047518

Orientador: Luís Duarte Andrade Ferreira

(Blank page)

Resumo

Problemas de saúde mental estão cada vez mais presentes na nossa sociedade, onde é possível testemunhar cada vez mais pessoas a apresentar sintomas deste tipo de distúrbio. Deste modo, os profissionais relacionados à saúde mental usam um conjunto de ferramentas e testes para avaliar os seus pacientes. No entanto, muitas destas ferramentas ainda são baseadas em papel e os métodos usados para a sua avaliação muito ineficientes e arcaicos. O presente trabalho propõe o desenvolvimento de uma aplicação web que suporta os profissionais de saúde mental no processo de criação, avaliação e gestão de ferramentas e testes de avaliação de saúde mental. Este trabalho está dividido em três partes: 1) desenvolvimento do front-end, 2) desenvolvimento do back-end e 3) desenvolvimento de medidas de segurança informática, de modo a salvaguardar a informação sensível dos utilizadores da plataforma. Deste modo, o presente trabalho é dedicado à terceira parte – segurança da aplicação. Tecnologias associadas à saúde costumam lidar com dados sensíveis como dados pessoais e registos médicos dos pacientes, o que reforça a necessidade de segurança que uma aplicação deste tipo deve oferecer. Além disso, tem-se assistido a muitos ataques informáticos nos últimos tempos. Tendo estes pontos em vista, a aplicação proposta irá ser desenhada respeitando as normas e boas práticas da indústria de segurança, além de ter em conta os riscos e vulnerabilidades mais comuns neste tipo de tecnologia, de modo a ser uma solução válida e robusta para os psicólogos. Adicionalmente, entrevistas com profissionais de saúde foram conduzidas de forma a melhor entender como é que informação sensível é guardada e partilhada entre profissionais de saúde.

Palavras-chave – *saúde mental, profissionais de saúde, segurança informática, ataques informáticos, vulnerabilidades*

Abstract

Mental health issues are becoming more prevalent in our society, with an increasing number of people exhibiting symptoms of this type of disorder. As a result, mental health professionals assess their patients using a variety of tools and tests. However, many of these tools are still paper-based, and the methods for evaluating them could be more efficient and modern. The current work proposes the creation of a web application to assist mental health professionals in the creation, evaluation, and management of mental health assessment tools and tests. This work is divided into three sections: 1) front-end development, 2) back-end development, and 3) development of IT security measures to secure platform user's sensitive information. Thus, the present work is dedicated to the third section - security of the application. Health-related technologies typically deal with sensitive data such as personal information and patient medical records, reinforcing the need for security that an application of this type must provide. Furthermore, there have been numerous computer attacks in recent years. With these considerations in mind, the proposed application will be built under relevant security industry standards and norms, as well as considering the most common risks and vulnerabilities in this type of technology, to be a valid and robust solution for psychologists. In addition, multiple interviews with healthcare professionals were conducted to better understand how sensitive information is being safeguarded and shared among other healthcare professionals.

Keywords – *mental health, health professionals, cybersecurity, cyber-attacks, vulnerabilities*

Acknowledgements

I would like to thank my academic advisor Professor Doctor Luís Duarte Andrade Ferreira for accepting taking part of this journey with me, believing in my ability to succeed with the work proposed, and for the help and advice given throughout the project.

I would like to thank my dear friends and colleagues Bruno Rodrigues and Eva Freitas for supporting me in the most difficult times, and for the effort made in developing this project.

I would like to thank my father, Eduardo Coelho, for the support, advice and for always motivating me to be better and to achieve success in everything I do.

I would like to thank my mother, Ana Romão, for all the motivation, patience and for always believing in me and my abilities.

I would like to thank my brother, Afonso Coelho, for all the motivation, support and for making me want to be an example, a better brother and a better person.

I also want to thank my grandmother, Avó Rita, for always being with me wherever I am, motivating me to accomplish my goals. I will forever miss you.

Finally, I would like to thank the professionals at Casa de Saúde São João de Deus for their availability, and to all the psychologists and health professionals who were involved in some way in the work developed.

Thank you.

(Blank page)

Index

Index of figures

List of abbreviations

WHO – World Health Organization

IT – Information Technology

DOM – Document Object Model

CVE – Common Vulnerabilities and Exposures

XSS – Cross-Site Scripting

BSI – Brief Symptom Inventory

MMPI – Minnesota Multiphasic Personality Inventory

PII – Personally Identifiable Information

CAS – Covid Anxiety Scale

GUI – Graphical User Interface

HIPAA – Health Insurance Portability and Accountability Act

PHI – Personal Health Information

ISO – International Organization for Standardization

GDPR – General Data Protection Rules

CIA – Confidentiality, Integrity, Availability

ISMS – Information Security Management System

OWASP – Open Web Application Security Project

BAC – Broken Access Control

URL – Uniform Resource Locator

HTTP/HTTPS – Hypertext Transfer Protocol (Secure)

SSRF – Server-Site Request Forgery

WMS – Wechsler Memory Scale

WAIS – Wechsler Adult Intelligence Scale

EADS – Escala Ansiedade Depressão Stress

MoCA – Montreal Cognitive Assessment

SUS – System Usability Scale

IMI – Intrinsic Motivation Inventory

NasaTLX – Nasa Task Load Index

API – Application Programming Interface

RBAC – Role-based Access Control

DoS – Denial of Service

EJS – Embedded JavaScript

MVC – Model View Controller

MVVM – Model View ViewModel

RCE – Remote Code Execution

AES – Advanced Encryption Standard

IV – Initialization Vector

CBC – Cipher Block Chaining

PoC – Proof of Concept

1. Introduction

1.1. Motivation

Mental health is an important factor that is related to the quality and quantity of individual and collective interaction. It is this that allows for the necessary balance to deal with positive and negative emotions and is an inseparable part of what is considered "being healthy". According to the *World Health Organization* (WHO), mental health is a state of well-being in which an individual is able to use their own abilities, recover from daily stress, be productive, and contribute to their community [1]. It is estimated that out of every 100 people, 30 suffer from or will suffer from mental health problems, and that about 12 have a severe mental illness [2]. Despite being a well-known issue in our society, mental illnesses and disorders in certain variations can be difficult to detect, where healthcare professionals such as psychologists have a fundamental role.

In Portugal, mental health is also a growing topic, given the high and constant numbers of people who have suffered or suffer from mental health problems [3]. Portugal is still one of the countries in Europe with one of the highest rates of mental health disorders, where 22.9% of adults have already suffered from a mental health problem [4].

The study conducted by Silva et al. [3] used a sample of data considered representative of the use of mental health services in the Portuguese population to evaluate possible gaps in the area of mental health and identify possible improvements. It was concluded that, despite access to healthcare, the Portuguese population does not seek this help to treat these types of issues, making it an even more significant problem in the lives of the Portuguese population [3]. In addition, it is mentioned that the creation of more services, initiatives, and methods is necessary to increase the quality and quantity of mental health services and treatment options.

Health professionals related to mental illnesses or disorders use a variety of tools and instruments to assess the mental state and well-being of their patients. As mentioned earlier, some of their most common purposes in this type of evaluation are checking for the presence or absence of conditions considered healthy, preventing a psychological condition, analysing the change in the severity of symptoms, and monitoring patient result over therapies. However, many of these tools and techniques are based on analogously answered questionnaires - paper and pencil whose results have to be calculated by the psychologists themselves [5]–[8].

Although these are methods that have been used for many years and have proven results, and to some extent, effective, this process, in addition to being costly in terms of paper, also takes a lot of time and is not efficient. Additionally, these assessment methods are not universal and standardized, and these tools can be adapted according to the country or region, with each

psychologist having their own version of the same tool and/or method, which makes this assessment phase even more complex (see Appendices - Interviews).

Currently, we are in the digital age, where technology is increasingly present in our daily lives and facilitates numerous daily tasks of various natures. Technology is increasingly appearing alongside various areas, and health is one of them [9]–[11]. In this context, it provides several advances in the provision of services, new equipment, education, communication, and management [12]–[15]**.** Due to these advances brought by technology, more and more health professionals and patients have their lives facilitated and supported by the functionalities that technological evolution offers [16].

However, currently, one of the biggest focuses and concerns when thinking about technology applied to human needs is its security, given the number of cyberattacks that have been witnessed in recent times in the world of organizations [17]–[21]**.** Cyberattacks in general have been increasing and evolving over the years, becoming more sophisticated, to cause as much impact as possible. These attacks can use various methods and techniques whose goal can be, for example, *Denial of Service* - the system failing and resulting in service problems [22], and *Data Breach* – which represents the theft, loss, alteration, disclosure, or compromise of patient's data, violating their privacy [23]–[25]**.** In addition, computer systems used by health organizations deal with sensitive and confidential information, which makes this matter of security even more essential. In Portugal, computer attacks have more than doubled in the last two years, which shows the growth of this type of crime [21]**.** Cybercrime costs millions to the companies involved, in addition to creating a bad image for them and putting values such as reliability at stake [24]–[26]**.**

Thus, the present work aims to develop a system capable of creating, managing, and storing this type of evaluations in an intuitive, personalized, and secure manner, combining technology with health. This system is a web platform that aggregates various benefits in terms of accessibility, interoperability, and maintenance that ensures the security of all users involved, healthcare professionals, patients, and their respective data.

1.2. Research goals

The goal of this project is the development of an interactive web platform called *Psyment* that allows healthcare professionals, such as psychologists, to create, manage, and store assessment tools for their patients. This platform aims to combat the exhaustive and inefficient method of manual assessment with paper records [7], [8], [27], [28] and to facilitate the work of professionals who perform this type of assessment, both in the process of completion and in the analysis of results.

The main contribution of this thesis is on the development of *Psyment* and IT security measures to secure platform users' sensitive information. Furthermore, the entire infrastructure of the web application is designed to meet the existing standards for this type of software [29]–[31] and to ensure the maximum possible security at all levels of the architecture. To achieve the proposed objectives mentioned earlier, research was conducted on the types of tests that exist and the most used by healthcare professionals, methodology, daily professional routine, and the technologies that were considered most suitable for the intended purpose were chosen, in addition to common practices for developing projects of this type.

Regarding the security of the application, research is conducted on the most relevant security standards and protocols for this type of application, in addition to the most common types of vulnerabilities in applications of this nature, to mitigate any potential vulnerabilities. Then, a vulnerability assessment scan will be performed to test all components and their functionalities built on the platform.

1.3. Research contributions

In this research, are presented the following contributions:

- **An in-depth examination of the state-of-the-art in web application security**. We conducted extensive research on common security standards and vulnerabilities prevalent in web applications. This comprehensive exploration informed the development process, ensuring that the developed platform not only meets the functional needs of psychologists but also adheres high security standards in the field.
- **A testing phase of the web application, where we conducted a rigorous security scan and identified potential vulnerabilities**. Notably, the research revealed a medium-severity vulnerability, specifically a *DOM-based Cross-Site Scripting* (*XSS*) issue. Additionally, the use of an outdated *jQuery* library with known *Common Vulnerabilities and Exposures* (CVEs) was detected, along with other informational findings. These findings underscore the critical importance of ongoing security practices in the realm of web applications, where is emphasized the necessity of regular scanning, testing, and patching to safeguard against potential vulnerabilities.
- **A contribution to the broader knowledge base of web vulnerabilities**. The thorough review conducted during this project provides insights into common vulnerabilities and serves as a valuable resource for future research projects and industry best practices.
- **An innovative web-based platform tailored for psychologists**, emphasizing security at every stage of development, and expanding the understanding of web vulnerabilities. The integration of novel features, adherence to security best practices, and the identification

of vulnerabilities underscore the commitment to advancing both the practical and theoretical aspects of development in the context of psychological assessments.

Moreover, **the present work's abstract was accepted as part of a call for papers project named Call for Papers – Special Issue (Digital Creativity for Developing Digital Maturity Future Skills)** [32]. Following a rigorous review process, the abstract received approval, leading to an invitation to publish a full paper at Journal of Entrepreneurial Researchers [33].

2. State of the Art

Here, we present the current state-of-the-art regarding the type of assessment methods applied by psychologists in their daily work, the usage of technology in the healthcare industry, specifically for mental health assessments, the security related standards and protocols conventionally defined for appliance in this type of technology to make them more secure and steady. In subsection 2.1., we will mention the most common risks and vulnerabilities in web applications. In section 3 (Methods), we present interviews conducted with psychologists to understand their daily work better and share the results.

2.1. *Psychological Assessment and Testing instruments*

As mentioned before, mental health professionals use a variety of instruments and tools to assess mental health and well-being. These assessments, are a set of items that are put together to measure characteristics and tendencies of human beings and are correlated to a certain behaviour or condition [34]**.** Past or current behaviours can be measured by psychologists, depending on the type of assessment tool used. Nonetheless, most of the testing has some sort of scale associated to avoid problems of interpretation that can be triggered by psychologist's reasoning and knowledge. The scales associated to each test relate raw scores on tests to previously defined theoretical or empirical distributions approved by the scientific community [34]**.**

Brief Symptom Inventory (BSI) [35] is a commonly used psychological test that helps identify psychological symptoms in adolescents and adults. This test is the short form of *Symptom Check List (SCL-90R)* which is a self-report questionnaire designed to measure symptoms severity and can be used as a psychiatric case-finding instrument [36]**.**

BSI consists of 53 items that cover nine psychological dimensions from different spectrums: Somatization (tendency to experience and communicate psychological distress with a significant focus on physical symptoms such pain, weakness, etc.) [37], [38], Obsession-Compulsion (inability to resist maladaptive actions or thoughts, presence of obsessions and/or compulsions, unwanted and intrusive thoughts, and impulses) [39], [40]**,** Interpersonal Sensitivity (inability to accurately assess others' state from nonverbal signal, related to accurate judgments of others' interpersonal sensitivity) [32], Depression, Anxiety, Hostility, Phobic anxiety, Paranoid ideation and Psychoticism and also include three global indices of distress: Global Severity Index, Positive Symptom Distress Index and Positive Symptom Total [28]**.** Patients are asked to rate each of the 53 items on a 5-point Likert scale [41] of distress (0-4), ranging from not at all (0) to extremely (4), as they respond to this test. When responding, respondents are instructed to take into account how each item relates to their experiences over the previous seven days, including the current day [35]**.** Each of these global indices is calculated using different criteria which makes

the scoring process more complex. As so, if considering a psychologist that realizes this specific type of test ten times per day, it would take a lot of time to evaluate them and obtain the scoring results that need to be analysed further.

Another popular psychological test is the *Minnesota Multiphasic Personality Inventory (MMPI)* which was developed in the 1940s to assess mental health problems [27]**.** It is a standardized questionnaire that conjures up a variety of self-descriptions and scores them to provide a numerical measurement of a person's emotional adjustment and test-taking stance [42]**.** The original test had 504 affirmative statements that could be answered with "True" or "False". Then, all responses from both categories are counted giving a numerical result, as mentioned before. The earlier version of this test, the *MMPI-2*, has 567 true/false items. It is designed with ten clinical scales, which assess ten major dimensions of mental disorders: Hypochondriasis, Depression, Hysteria, Psychopatic Deviate, Masculinity/Feminity, Paranoia, Psychasthenia, Schizophrenia, Hypomania and Social Introversion [43]**.**

However, the *MMPI* has some limitations. The *MMPI* is also an extensive test (504 questions originally), which makes the process of analysing the user's responses somewhat time-consuming and complex [42]**.** Beyond that, the correlations between some scales of the *MMPI/MMPI2* are high, due to the numerous item overlap. In some topics, the same item will be simultaneously used for the scoring of several different scales, and most of the scales have a relatively high set of topics common to other scales [42]**.** This circumstance may give rise to confusion, potentially causing healthcare professionals to miscalculate scores for each dimension evaluated by the specified test. This risk is exacerbated by the fact that *"In all versions of the MMPI, the scale labels can be misleading (…),*" introducing an additional layer of complexity to the accurate interpretation of test results. [42]. The same source also claims that *"(…) practitioners need to carefully evaluate the meanings of scale evaluations (…)*" [42]. According to Helmes et al. [39], among several issues present in this specific test, one of the major structural problems of the *MMPI* is the "*Overlap among the scales*" – "*Overlap complicates the interpretation of a test profile by requiring the interpreter to take into account the spurious correlation between some scales and not others (…)"* [39]**.**

2.2. *Supportive technologies for mental health assessment*

According to Thimbleby, *"Technology drives healthcare more than any other force, and in the future it will continue to develop in dramatic ways."* [10]**.** It is unquestionable how much technology helps and drives the evolution of areas such as healthcare [10], [12]**.**

For example, *PsyPack* [44] is an online psychometric testing software for behavioural health professionals that has a variety of standardized psychological assessments. It offers in-clinic and remote management, automatic scoring, graphing, and reporting and works across

multiple devices [44]**.** The system is designed for patients with a basic understanding of technology at least, as they are the ones who respond to the tests by receiving it via e-mail. Then, the psychologist gets the exam results in his account within the platform and can create a preset-structured report with all available information. This information consists of some personal information that act as *PII (Personally Identifiable Information)* about the user being tested, a brief disclaimer about the specific test – definition and procedure, description, *Results of the evaluation*, *Response sheet*, recommendations, and Psychologist's notes in case he wants to provide some additional information. Figures 1 and 2, illustrated below, represent the result of a test from the psychologist's point of view, showing the *Results of the evaluation* and the *Response Sheet*, in the respective order.

Note: The psychological test used in the given example was the *Covid Anxiety Scale (CAS)* for convenience. No study or user in this document is associated with the mentioned assessment instrument.

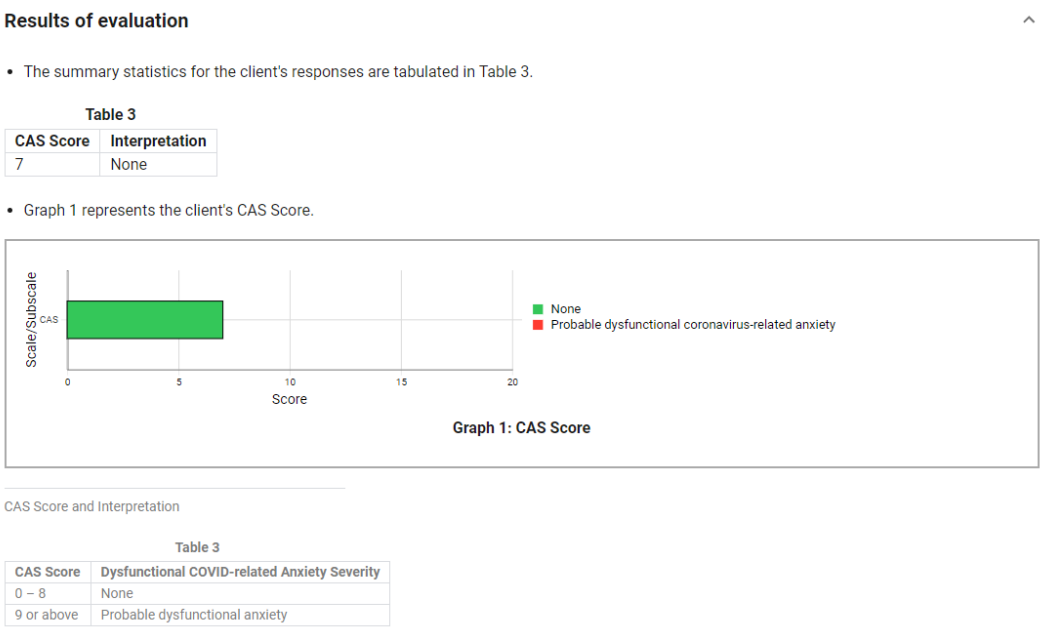


Figure 1. Results of evaluation of the Corona Anxiety Scale test.

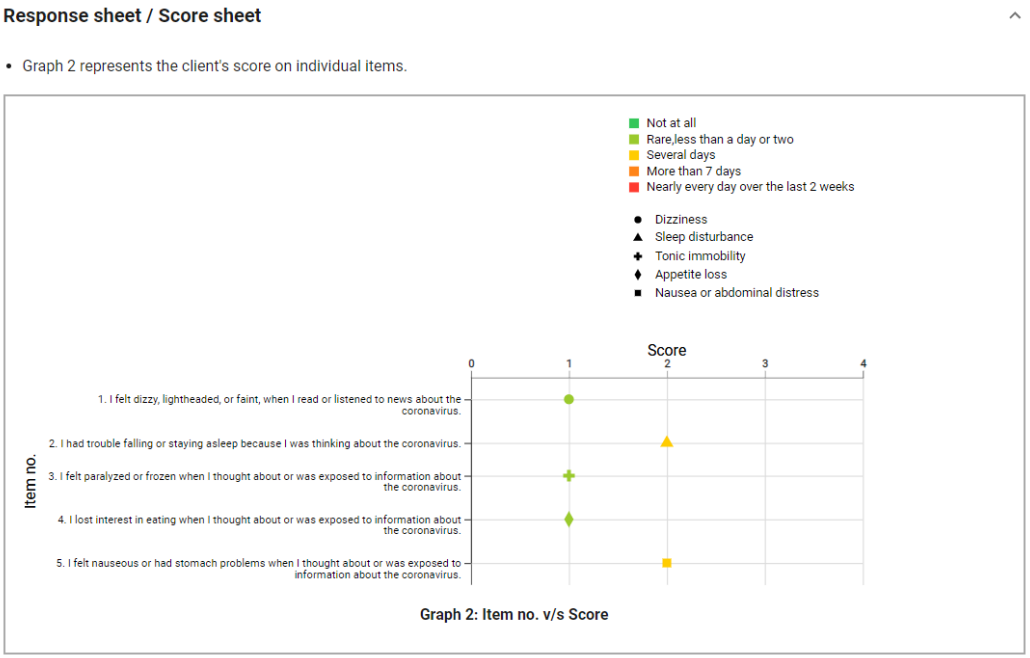


Figure 2. Response sheet of the Corona Anxiety Scale test.

After reviewing the system of *PsyPack*, it can be pointed out that the fact of not being able to create a test and only being able to use the tests available on the platform is highly limiting for health professionals, as they are constrained to using only the tests pre-existing on the platform. Moreover, the assessments are sent to patients via e-mail, making the system dependent on an external service (e-mail). The same system requires that the patient has an e-mail address and, of course, technological knowledge to access it and take the test. This can be difficult to achieve in the case of older patients or more severe diagnoses (e.g., dementia).

Another tool found within the same scope of functionalities and goals is the *PsyToolkit* [45] *(psytoolkit.org)* which consists of a free-to-use toolkit for creating and running cognitive-psychological experiments and surveys, including personality tests. It allows users to set up, run and analyse online questionnaires and experiments [45], [46]**.** According to its site, it was made by academics with a focus on research and teaching. It has several features such as multi-language support in online surveys, a library with more than one hundred prebuilt psychological surveys and experiments that can be edited and used. When a survey is created and posted online, a custom link for the test is generated and anyone can respond to it. To create or edit a question, there are two ways: using the *markup* language requested by the platform, which requires additional knowledge, and using the "*easy mode*" that allows to add questions through a *GUI* menu. It is also possible to define several extra configurations for the online surveys such as exclude unwanted browsers (e.g., *Internet Explorer* and/or *Safari*) to access the online page or add a "Consent and ethics information" section in the questionnaire. Regarding the design of the platform, it is composed by a layout with multiple colours. The following figures, Figure 3 and Figure 4 represent the menu of creation of an online survey and the survey created, respectively.

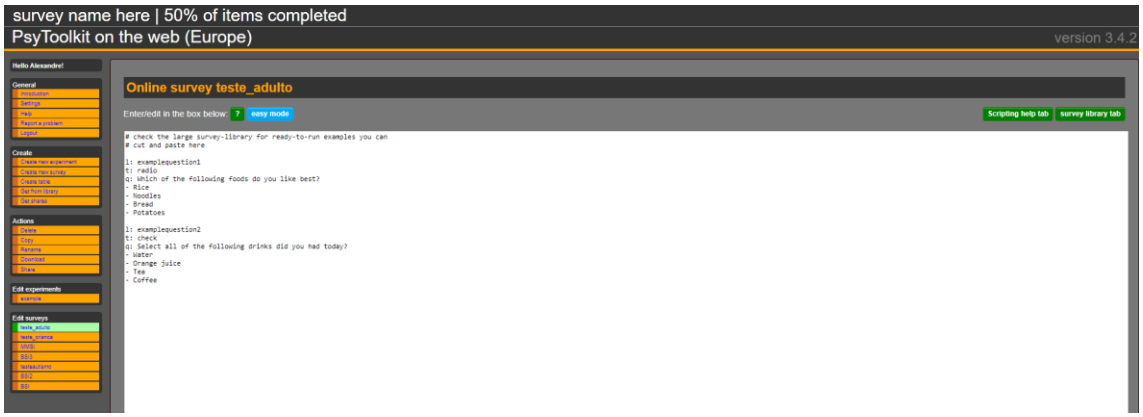


Figure 4. Arbitrary survey created with Psytoolkit.

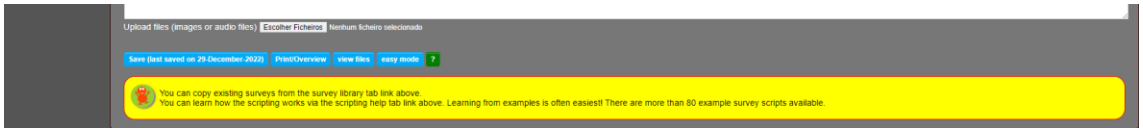


Figure 3. Menu of creation of an online survey on Psytoolkit.

After reviewing the system of *PsyToolKit*, it is possible to say that, although it is functional, it does appear that it has some design limitations that should be addressed. The platform does not follow an intuitive and structured way of displaying GUI elements such as buttons and text boxes. Also, the excess of colours makes the platform look overwhelmed. Regarding the security of the respective *website*, there are basic topics that should be addressed. First, when creating a new account, the password cannot be defined by the user himself, but

instead by the web application, which is a problem; the defined password is made up of five characters, all of which are numbers. These passwords have an extremely low complexity, which makes the platform's authentication mechanism vulnerable to several password attacks [47]**.** Beyond that, as all the passwords have the same size and it is considerably short, it is reasonable to deduce that the platform does not have a sufficient and robust password policy. Consequently, it is suspected that the mentioned platform is not compliant with the *General Data Protection Rules (GDPR)* [30] that frequently refers to terms like "*appropriate safeguards"*, "*appropriate security"* and *"appropriate measures"* [48]**.** The *GDPR* will be covered more in-depth in the following section as well as some relevant security topics regarding technology in healthcare.

2.3. *Cybersecurity standards in healthcare technology*

Cybersecurity is of critical importance in health applications, as the consequences of a cybersecurity breach can be severe and potentially life-threatening [49]. As mentioned before, healthcare applications often handle sensitive personal and medical information, and a breach of this information can have serious consequences for both the individuals affected and the organizations involved [19], [24], [25]. In addition to the risk of personal information being compromised, a cybersecurity breach in a health application can also lead to disruptions in the delivery of care. For example, if a healthcare organization's electronic medical records system is compromised, it can lead to delays in patient care and potentially even harm patients [49].

To address these potential risks, healthcare organizations should adopt and consistently update robust cybersecurity standards, measures, and protocols. There are several cybersecurity standards and protocols that are commonly used in healthcare technology to protect patient data and ensure the security and privacy of electronic health records.

The *Health Insurance Portability and Accountability Act* (HIPAA) [29] is a North American federal law that was enacted in 1996 to protect the privacy of individuals' personal and medical information. HIPAA applies to health-related entities, and it sets standards for the protection of personal health information (PHI) that is created, received, used, or maintained by these entities. To be compliant with HIPAA, covered entities must implement appropriate physical, technical, and administrative safeguards to protect the confidentiality, integrity, and availability (*CIA Triad*) [50] of PHI such as two-factor authentication in authentication mechanisms. HIPAA also requires covered entities to provide individuals with certain rights with respect to their PHI, including right to access, correct and request a copy of their PHI. Although it is not used in European organizations because HIPAA is considered an American law, it is one of the oldest active standards for personal data protection and served as inspiration for more recent standards [29], [51], [52]**.**

ISO/EIC 27001 [31] is a standard used worldwide to manage information security. This standard is not restricted to healthcare technology, but to organizations of all sectors that deal with any kind of assets such as financial information, intellectual property and employee data [31]. It provides a framework for organizations to establish, implement, maintain and continually improve cybersecurity practices. The standard is organized into fourteen sections that cover different aspects of an information security management system (ISMS). Some of the most relevant sections are: *Scope* that defines the scope of the ISMS, including the types of information that are covered and the boundaries of the system; *Planning* that defines the requirements for planning and implementing the ISMS, including the development of policies and procedures (e.g. Password policy); *Operation* that covers day-to-day activities required to operate the system, including the management of information security risks and the implementation of controls, technology and processes (e.g. *WAF – Web Application Firewall*); *Performance Evaluation* that covers the monitoring and measurement of the ISMS, including the evaluation of its effectiveness and the identification of areas for improvement [53]. Moreover, the measurement of the ISMS can be supported by the execution of internal and external scans, vulnerability assessments and penetration tests that will help identify security weaknesses, vulnerabilities, and flaws in the respective system, as well as possible mitigations.

According to the *International Organization for Standardization* (ISO) [54], this standard benefits organizations in several ways: maintains information secure in all forms, increases resilience to cyber-attacks, provides a centrally managed framework that secures information in one place, responds to evolving security threats, reduces costs and spending on ineffective defence technology and protects the integrity, confidentiality and availability of data, also known as the *CIA triad* [31], [50], [53].

The *CIA triad* is a model used to guide the development and implementation of cybersecurity measures in an organization. It stands for **Confidentiality**, **Integrity** and **Availability**, that are considered the three main pillars of cybersecurity [53].

1. **Confidentiality** – This refers to the protection of information from unauthorized access or disclosure. It is concerned with ensuring that only authorized individuals have access to sensitive information. (e.g. implement a solid access control mechanism supported by a trustable authentication process) [53].
2. **Integrity** – This refers to the accuracy and completeness of information, as well as the protection of that information from unauthorized modification. It is concerned with ensuring that information is not corrupted or altered in any way [53] (e.g. information stored in a secured and reliable way which is hard to read and edit, can be encrypted).

3. **Availability** – This refers to the ability of authorized users to access information when they need it. It is concerned with ensuring that information systems are available and always functioning properly [53] (e.g. implement a *Web Application Firewall* where suspect traffic such as high amount of login attempts from a single IP address makes the *WAF* block the IP from accessing the server since it might be a *brute-force* attack).

The Figure 5, illustrated below, represents the *CIA triad*.

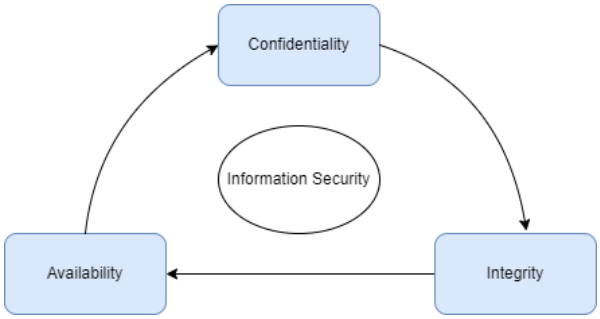


Figure 5. CIA Triad Diagram - Confidentiality, Integrity and Availability.

The *General Data Protection Regulation* (GDPR) [30] is a regulation in EU law on data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA). According to [55], GDPR "*is the toughest privacy and security law in the world"*. It was put into effect in 2018 and aims to give EU citizens and residents more control over personal data and to simplify the regulatory environment for international business by unifying the regulation within the EU.

Under GDPR, Personal Data is defined as any information that can be used to identify directly or indirectly an individual. This includes names, addresses, IP addresses, etc. GDPR sets several rights for individuals regarding their personal data giving them complete control of their data, even though the application's organization has it. The individuals have the right to be informed about the collection and use of their data, the right to access, to rectify, to erase and to know how their data's processing is done. To be compliant with this regulation and avoid very high fines, organizations need to have appropriate technical implementations and measures to protect personal data from unauthorized access, use, disclosure, and destruction. In case of a data breach that compromises *Personally Identifiable Information (*PII), the individuals involved should be notified about the situation [30], [55].

Organizations and their data controllers are required to "*handle data securely by implementing appropriate technical and organizational measures.*" [48], [55]. Technically, "*technical measures*" can be anything implemented to provide security such as two-factor authentication, end-to-end encryption, server-side validations, or a strong password policy.

After thoroughly reviewing the preceding standards, it becomes evident that they share common characteristics. Confidentiality, Integrity, and Availability are fundamental security principles that must be present in all applications. Systems must be designed considering the data they will be dealing with and the existing limitations, so it is easier to build them solidly, without any infrastructure gaps or flaws, as they should be robust and reliable. From a technical point of view, there are multiple techniques and implementations to ensure the security of software [49], that should be used in the present project. Additionally, applications should be the target of external and internal tests often [44]. Finally, all the reviewed standards focus on data protection since data is the most asset in healthcare applications.

2.4. *Most common risks in healthcare web applications*

As mentioned before, the security of current web applications is becoming increasingly crucial as their usage grows. Because millions of people rely on these services, their confidentiality, integrity, and availability (*CIA Triad*) [50], [53] are critical [56]. As so, it is fundamental to understand and review the most common vulnerabilities in web applications to implement a system that is specifically designed to be protected against all of these potential flaws and eliminate any possibility of discovering a vulnerability that is considered common in this type of technology. Because the current work focuses on the development of a web-based healthcare platform, this chapter will focus on web vulnerabilities.

Open Web Application Security Project (OWASP) [57] is a non-profit organization that provides resources and guidance for improving the security of web applications. *OWASP Foundation* works through community-led open-source software projects, has tens of thousands of members, and hosts global conferences worldwide. One of their most famous projects is the *Open Web Application Security Project Top 10* [58] which is a list of the most common web application security risks, as determined through analysis of data from various sources, including the *OWASP* community, industry experts, and the media. According to OWASP *Top 10* [58] most recent list, the top 10 web application security risks are:

2.4.1. Broken Access Control

Broken-Access Control (BAC) occurs when an application does not properly enforce access controls, leading to the disclosure of unauthorized information, modification or destruction of data, or executing business functions beyond the user's limits [59], that can result in a privilege escalation and even bigger damages [60]. BAC means that a user can act outside of his intended permissions. An attacker can get unauthorized access by leveraging this vulnerability by exploiting an inadequate user input validation and/or sanitization, usage of outdated functions, unmanaged exception handlers, uncontrolled redirection of pages [60], server misconfigurations (HTTP verb tampering), cookie tampering, token tampering, bad object referencing and others.

To prevent BAC, developers should prioritize the implementation of server-side validations since server-side code is the only trustable code, because clients can be easily manipulated most of the time [61]. The system should avoid external resources as much as possible, implement access control mechanisms, and repeat the verification process, when possible, within the application, filter and sanitize the user input always in a strict manner (e.g. using black/whitelists). Moreover, systems should disable web server directory listing and have a login failure log system that can notify the administrators after a certain amount of login attempts by a certain IP address. Finally, it should have short time-limited tokens that expire on the server as soon as the individual user logs out. This shortens the window attackers might have to read sensitive data [59].

2.4.2. Cryptographic Failures

Cryptographic Failures occur when sensitive and critical data is exposed. In web applications, data communication flows bidirectionally between server and client(s). For instance, during a communication that uses the *HTTP/HTTPS* protocol, data can be intercepted. This data can be a password, a credit card number, health record, or other type of PII. An attacker can leverage this vulnerability checking for any data communication flow happening in the system and trying to intercept it for further analysis. The problem is not the interception of the data, but the fact that if it can be decrypted and used, such as an encrypted cookie or a password hash. Missing out on safeguarding such data leads to theft, public listing, breaches [62] and big fines from standard-defining organizations (e.g. EU's General Data Protection Regulation) [55].

To prevent Cryptographic Failures, every single piece of data should be encrypted. The encryption algorithms should be strong, advanced, and up to date. Also, cryptographic functions such as *MD5* and *SHA1* should be avoided since "*(…) popular functions like MD5 and SHA-1 (…) are (much) weaker than originally anticipated (…)"* [63, p. 5]. Developers must not use legacy protocols *(…)* [64] for transporting data. Moreover, source code should never be revealed as it can compromise the encryption practiced throughout the whole application.

2.4.3. Injection

Injection occurs when an attacker can execute malicious code by injecting it into a vulnerable application. For this vulnerability to exist, an attacker must be able to manipulate a value of user-input parameters used as part of a command, a function or a query [65]. The respective parameter, if not validated and/or sanitized properly, will execute the same task including the user input exactly as it was inputted, malicious or not. Consequently, an application might be vulnerable to an injection if it does not filter, validate, or sanitize the user input properly [66].

There are various types of injections such as SQL, NoSQL, Operative System command, and *Cross-Site Scripting* (XSS) [65]. SQL Injections are flaws that enable a user to manipulate queries

of an SQL database. This can allow a user to access sensitive data, modify the database, or even gain control of the server as the database service user. The same logic applies to NoSQL databases. OS command injections are flaws that enable a user to inject malicious commands into a system shell. This can allow a user to access sensitive data (e.g. *SSH* keys), execute arbitrary code on the system potentially giving full control over the system. XSS is a flaw that enables a user to inject malicious scripts (e.g., *Javascript*) in the client's web browser. When the user visits an exploited web page, the code gets executed. This can allow a user to access sensitive data and hijack other user's sessions by stealing cookies [65].

To prevent Injections, developers must implement server-side input validations, as mentioned before [61], [66]. This, combined with blacklists and whitelists of characters frequently used on injections (e.g. quotes, slashes, etc.) [67] helps mitigate this vulnerability. Additionally, systems should never directly pass user input to any function, command, or query without precedent processing and/or escaping.

2.4.4. Insecure Design

Insecure Design can be defined as *"(...) missing or ineffective control design (...)*" [68]. It is not related to implementation, as it is possible to have a secure design with implementation issues. Although, an insecure design comes from neglecting design and architectural best practices. Designing a secure application requires security awareness from the planning stage, taking into account the specific characteristics and requirements of the system being developed [68], [69]. According to Rossi et. al [69], web application development is more than just visual design and user interface as *"(...) it involves planning, selection of an appropriate Web architecture, system design, (...), coding, (...), and its maintenance, testing, quality assurance, (...)"* [69].

Insecure Design is not a vulnerability itself, but an issue in the creation process that can lead to multiple vulnerabilities. Some of the potential vulnerabilities caused by a poor design are injections, path traversals, and broken access control (e.g. if rights, permissions and privileges are not effectively defined) [68]. These vulnerabilities mean the possibility of privilege escalation, system information enumeration, denial of service, and others.

To prevent Insecure Design, a secure development lifecycle must be in place, where there is frequent evaluations of the security controls, protections, and aspects [29]–[31]. Resource usage should also be limited to users and/or services. Quality assurance and penetration tests should be done often to maintain the system patched, secure, and up to date [68].

2.4.5. Security Misconfiguration

Security Misconfigurations occur when an application or its infrastructure is not properly configured, making it vulnerable to attack. According to [70], security misconfigurations reveal easy targets to attackers, since it is relatively easy to detect misconfigured web servers and applications [70]. These misconfigurations encompass everything that can expand an application or host's attack surface (e.g. unnecessary ports open) [71]. An application is deemed misconfigured when it lacks necessary security measures, possesses improperly configured permissions, includes unnecessary features, plugins, extensions, or services, or if it runs outdated components or software (may have a publicly listed *Common Vulnerability and Exposures (CVE)* [72] available to exploit it), if default credentials are still enabled and unchanged (e.g. *admin:admin*) and if security settings in the application's infrastructure, libraries, databases and others are not set to secure values [71].

To prevent Security Misconfigurations, default credentials must not be allowed, regularly patches and updates should be implemented, as well as suitable security controls and permissions set. Furthermore, developers should keep their systems minimal and simple as possible, not installing any unnecessary features or frameworks and components [71], as it can increase the attack surface.

2.4.6. Vulnerable and Outdated Components

Vulnerable and outdated components refer to open-source or proprietary code that is no longer maintained and/or may be vulnerable. This code can belong to any type of software, libraries, frameworks, web applications, individual scripts, and others. It can expose an application to attacks. Software with known vulnerabilities (e.g. published *CVEs* available) can allow an attacker to exploit these vulnerabilities and gain a foothold in the target system [73]. Consequently, an application may be vulnerable if any component of its infrastructure (e.g., OS, web server, database management system, APIs, libraries, and others) is discontinued.

To prevent vulnerable and outdated components, the versions of all components present in the platform (client and server-side) should be read and documented by the developers. Unnecessary features, dependencies, and files should be removed. Besides this, vulnerability assessments and penetration tests must be done often and the system should be checked frequently for patches and updates, as already mentioned by [74].

2.4.7. Identification and Authentication failures

Identification and Authentication failures occurs when an application's authentication and session management mechanisms are not properly implemented, allowing an attacker to gain unauthorized access to the system.

An application might exhibit authentication vulnerabilities if it permits automated attacks, such as dictionary attacks, brute-force attacks, and password sprays. Additionally, the use of default credentials or a weak password policy, coupled with an inadequate password recovery mechanism, could contribute to these weaknesses [75]. Regarding session management, an application may be vulnerable if exposes session identifiers (e.g. Session ID) in URLs, if it reuses session IDs after successful logins, and if the app does not properly validate session IDs or authentication tokens (e.g. during logout, the user's authentication token is removed and disassociated from any active session token database) [75].

To prevent this type of risk, multi-factor authentication can be implemented that reinforces the login mechanism and prevents automated attacks from taking place. Default credentials should not be used and/or accepted by the application. Also, password policies should be aligned to demand a specific length and complexity. It is important to make sure the application is robust enough to minimize damages caused by enumeration attacks (e.g., using the same error message) [75]. Finally, the system needs to limit failed login attempts as well as log all failures consequently alerting the administrators and rely on a server-side built-in session manager that can create and manage random session IDs [75].

2.4.8. Software and Data Integrity Failures

Software and Data Integrity failures occur when program code and/or its infrastructure do not ensure integrity [76]. Architectures frequently incorporate plugins, modules, and libraries sourced from public repositories and open-source code, which may originate from untrusted sources. Also, these additional developer tools may have public exploits available (CVEs) [72]. Owing to its intricacy, software and data integrity may falter, leading to the exposure of data without proper integrity verification. This outcome stems from a lack of sufficient validation, reliance on outdated third-party software, inadequate testing, and vulnerability scanning. Consequently, this issue is directly correlated with vulnerabilities such as Injections, Broken Access Control, and Insecure Design in the architecture.[76].

To prevent this, all third-party software sources should be verified. Libraries and dependencies should be using trusted repositories. Like other risk preventions mentioned before, the code should be reviewed more often and the application tested, to mitigate scenarios where sensitive data is leaked through the software workflow.

2.4.9. Security Logging and Monitoring Failures

Security Logging and Monitoring Failures occur when an application does not properly log and monitor activity, making it difficult to detect and respond to security incidents as attacks [77]. It is important to emit alerts if any malfunctions, errors happen, or suspicious activities are detected (e.g. if a user from a specific IP address attempts to log many times in a short amount of

time) [78]. Detecting such incidents can prevent significant damage to the protected system and even assist in troubleshooting potential issues, such as crashes. However, logs should not be vague as they should provide valuable information or insights about what is going on with the secured host. Moreover, logs must be protected because they can contain sensitive information. Additionally, message errors and warnings should be adequate, as mentioned before, not revealing any type of relevant information and logs should be backed up not only locally (e.g. cloud-based backups) [77].

To prevent this, developers must ensure all login failures are logged with proper user context to identify the source (user or IP address) and the component abused. Moreover, logs should be logically presented and understandable, and securely stored [77].

2.4.10. Server-Side Request Forgery (SSRF)

Server-Side Request Forgery (SSRF) occurs when an attacker targets a vulnerable application's backend server and tricks it into executing malicious requests specially crafted for performing unintended actions. By exploiting this vulnerability, an attacker gains the ability to access internal network resources or launch subsequent attacks, facilitating lateral movement or ensuring persistence. Technically, for this risk to exist on a web application, the app should be fetching a remote resource without validating the URL supplied by the user. According to [79], *"(...) when the services are accessed via URL, the attacker supplies or modifies a URL to access services on servers the he is not permitted to (...)"* [79]. Modern platforms offer a variety of features, and fetching a URL becomes a common scenario. The increasing usage of cloud services contributes to the existence of this risk [80].

To prevent this risk, developers should apply the rule *"deny by default"* in the application's policies so network traffic that is not previously defined as allowed is expressly blocked, allowing it to access only authorized services. Also, as mentioned before, the application back-end server should sanitize and validate all user-input data as well as avoid raw data responses to user's client side [80].

3. Methods

As previously outlined, we performed four consented interviews with psychologists; three of them were semi-structured interviews (see Appendices) and one was informal. The purpose of these interviews is to better understand 1) the process of administering a psychological test to a patient, 2) to understand what tests are the most used during that process, 3) to understand their data management and collection practices, 4) to identify what technological practices they employ in their daily work and 5) potential improvements in the process.

3.1. Participants

The psychologists interviewed are from different psychology areas and use different tests regarding mental health assessment. To better understand their background and potential correlations between their experience and problems with psychological assessments, we pointed out some characteristics of them. These characteristics are: Years of experience with testing patients in their daily work routine, main background of application of tests (target), and main field of experience within Psychology.

Table 1 contains all the previously mentioned information regarding the psychologists interviewed. The research group selected four psychologists with different years of experience so that the experience of different health professionals does not influence and/or condition the collected data and future application. The respective professionals are identified from P01 to P04.

Psychologists Identification (PID)	**Years of experience with testing**	**Background of application of tests**	**Main field of experience**
P01	4	Applying tests to children	Criminal psychology (justice)
P02	13	Applying tests to any individual	Clinical psychology and research
P03	3	Applying tests to any individual	Neuropsychology
P04	10	Applying tests to elderly individuals	Clinical psychology

Table 1. Information about the psychologists interviewed.

3.2. Experimental Setup

The experimental setup of the interviews consisted of three semi-structured interviews that followed a script (see Appendices) and one informal interview. All psychologists interviewed consented to their participation in this study, which allowed us to record the interview audio (see Appendices). Photo-taking was also allowed if necessary. However, the psychologists' identity is protected and confidential.

3.3. Results

Regarding the interviews, they are transcribed in full in this document (see Appendices). However, in this section, the most pertinent topics referred are presented. These are related to common practices in psychologists' daily work, critical aspects of their methodology, challenges, and potential improvements.

3.3.1. Interview with Psychologist P01 (*IP01*)

- Applied most of the tests with pencil and paper.
- Psychologists exchange tests with each other.
- Tests on the same patient performed at different times are compared to analyse the evolution of the diagnosis.
- Tests are confidential. They should be destroyed after some time to comply with data protections.
- Believes that there is room to evolve in terms of computerized tests.
- A few years ago, used acetate sheets in assessments.
- Mentioned the fact that there are recruitment processes in companies that take months to complete due to the delay in rating mental health tests.
- Used psychological tests such as *BSI* and *BDI*.

3.3.2. Interview with Psychologist P02 (*IP02*)

- Applied all the tests with pencil and paper.
- There are not many validated psychological artifacts regarding computerized tools.
- Psychologists share test results within the same clinical area.
- Stores tests data for around 10 years (for analysis).
- Stores assessments' data using the computer, identifies the patients with IDs, but stores the IDs in paper.
- Paper and pencil method is highly inefficient.
- Uses psychological tests such as *MoCA*, *WMS* and *WAIS*.
- Assessment process can and should be simplified.

3.3.3. Interview with Psychologist P03 (*IP03*)

- Psychologists exchange tests with each other.
- Some tests are adapted for illiterate patients.
- There are tests that might be hard to computerize.
- Stores assessment's data using paper and software provided by the institution. Also uses *Google Drive*.
- There are keywords in open-answer questions that helps interpreting and correcting them.
- Mentioned several tests such as EADS.
- Shares test results within the same clinical area.

3.3.4. Interview with Psychologists P04 (*IP04*)

- Tests performed are adapted to the specific organization (e.g. country, location, date).
- Patient answers orally and psychologist writes it down on the respective test.
- Applied all the tests with pencil and paper.
- Tests on the same patient performed at different times are compared to analyse evolution of the diagnosis.
- Some of the tests performed have a practical part (e.g. folding a piece of paper), which requires interpretation from the professional.
- Mentioned the fact that there are also rating problems due to interpretation issues with drawings.
- Uses tests such as *MMSE, MoCA* and *GDS*.
- Difficulties calculating *MoCA* test results.
- Tests are shared with a senior psychologist that rates and analyses the results of the specific tests.

Additionally, the informal interview with psychologists of IP04 was conducted in the context of a field study carried out at Casa de Saúde São João de Deus, where the process of administering a psychological test to patients was observed. The healthcare professionals of IP04 applied the *Mini-Mental State Examination* (*MMSE*) to two patients. Figure 6, present below, illustrates the moment when a patient is being tested.

Figure 6. Process of administering a psychological test to a patient (MMSE).

3.4. Thematic Analysis

A thematic analysis of the interview's feedback was completed in this section. The qualitative research approach assisted in the discovery of various patterns among psychologists in terms of assessment procedure issues, technologies used, common practices in their methodology, and their cybersecurity awareness and requirements.

3.4.1. Mental health assessment tests – issues

Regarding mental health assessment tools, the psychologist P01 indicated that there are tests that require a long time to assess such as the Brief Symptom Inventory [28], [35]. P01 mentioned that *"(...) the analysis of the BSI is complex because it has many thresholds, and we need to check it manually in the reference tables (...)*" and "*(...) we would like if applications could automatically rate the tests for us (...)*". He also pointed out that, as he cannot rate the test in front of the patient, he schedules a second session to continue the interaction with the patient - *"(...) the efficiency should be - we apply the test, and we get the result immediately to inform the patient and that's a case (...) technology professionals have an opportunity to help us (...)*".

Psychologist P02 revealed issues realizing the *Wechsler Memory Scale* [81] due to 1) the time taken by patients to respond, 2) the complexity of score calculations (sums), and 3) the fact that after realizing the test, a specific age-based referenced book needs to be checked to properly analyse the results, since the score is compared with values of reference (thresholds). Psychologist P03 also stated that paper-based psychological tests are "*too much work (...)*" because of their inefficiency (see Appendices – Interviews).

Lastly, psychologists of IP04 revealed issues calculating *MoCA* [8] test results and mentioned the fact that the rating of the *MMSE* is not as straightforward as others, and it requires qualitative

interpretation from him due to its practical part. Additionally, psychologists P01 and P02 mentioned the fact that some tests are registered trademarks as *"(...) most of the tests used by us are paid (...)*" and *"(...) we need to ask for some authorizations to use specific tests (...)*" (see Appendices – Interview with P01 and P02).

3.4.2. Common aspects in psychologists' methodology

All psychologists have their work methodology, a specific way to perform work tasks that are set according to their preferences, knowledge, and experience. Every single psychologist interviewed exchanges paper-based tests with colleagues. They also discuss some results that might be useful for further cases. Moreover, they compare the same test performed on the same patient at various points of time to evaluate how the diagnosis of the respective patient has changed over time. It should be noted that although all psychologists work in different sub-areas of psychology and have different industry experiences, they have similar practices.

3.4.3. Technologies used by psychologists in daily work

Regarding technologies used in their daily work, psychologists mentioned *Microsoft Word*, *Excel*, *Google Drive*, and email to edit, calculate results, and store tests. The psychologists interviewed do not use any specific software that fulfils all their requirements. As stated by psychologist P01, "*(...) we would like if applications could automatically rate the tests for us (...)*" and *"(...) technology professionals have an opportunity to help us (...)"* (see Appendices – Interviews).

3.4.4. Data security awareness and requirements

Regarding cybersecurity, psychologist P01 pointed out that healthcare applications must be compliant with the most common and recent data protection laws since patients' data and medical records are strictly confidential and must not be compromised. According to the psychologist P01, *"(...) tests must be confidential (...)*". He also added that *"(...) the only thing that I think is fundamental in this type of platform is that it always complies with the security of patient's data and the respective laws*". P01 also mentioned, as well as psychologist P02, that data must be stored and kept for a limited period and must be deleted afterward – *"(...) tests must be (...) destroyed after some time (...)"* and *"(...) patient data must not be kept for more than 5 years (...)*". Psychologist P03 pointed out the confidentiality aspect as *"(...) confidentiality must be in place (...)"* and "*(...) psychologists should not be capable of accessing old registers (...)*". Therefore, it is necessary to guarantee that an application that deals with sensitive health data respects the most distinguished security standards and guarantees the confidentiality of the information.

3.5. Discussion

A thematic analysis was performed on the preliminary interview data to gather information that could help our work. After reviewing the issues that psychologists pointed related to mental health tests, it is possible to retain that a supportive application for this specific task must have certain functionalities such as the ability to set thresholds for questions and tests, comparison with specific ranges of values, automatic and immediate scoring, ability to share tests with other health professionals and ability to edit and compare tests. Additionally, as some tests include a practical part [7] , the application should allow the upload of media (e.g. photograph of the drawing made by the patient). As mentioned before, psychologists *"(...) would like if applications could automatically rate the tests (...)*" (see Appendices – Interviews).

In addition, some psychologists pointed out that some tests are registered trademarks. However, the suggested application can aid in converting these tests to digital format. Due to its high degree of adaptability and customization, *Psyment* will help and enable mental health professionals to create and validate their own cognitive assessment tools.

Moreover, the psychologists interviewed have different years of experience and are from different backgrounds, such as clinical, criminal and neuropsychology. However, they all have similar methodologies since all of them exchange tests with colleagues and do comparisons between the applied tests. As so, to create a suitable application for psychologists to facilitate their work, it must have built in functionalities that allow mental health professionals to perform these daily tasks that psychologists from different areas practice.

These psychologists use general-purpose software, and none of them was specifically developed to support their work. As mentioned before, the software used by them does not match all their necessities since several problems and improvements were reported.

Regarding data security, psychologists praised this topic, demonstrating the need for cybersecurity in technologies related to health, as they deal with sensitive data. Surprisingly, all the interviewees outlined this topic, recognizing the importance to maintain sensitive data safe and secure. As so, there is a need for new and upcoming health-related technology to be security-ensured. According to the psychologists P01 and P03, "*(...) confidentiality must be in place (...)*" and *"(...) is fundamental in this type of platform is that it always complies with the security of patient data (...)*" (see Appendices – Interviews).

In general, as mentioned before, the interviews conducted with health professionals referred that there is a need to build an application capable of creating, managing, and assessing health-related tests since all the psychologists in the interviews clearly agreed and showed interest in this

idea. As pointed out by psychologist P01, *"(...) in my opinion steps need to be taken towards having the majority of the tests computerized (..)*" (see Appendices – Interviews).

Usability assessment

4. Usability Assessment

4.1. Prototyping

Following the feedback from the interviews, some crucial characteristics that an application of this type should address were highlighted. Following that, some prototypes were created. These are based on the suggestions of the health professionals interviewed. There are two types of prototypes: low-fidelity and high-fidelity prototypes.

Initially, low-fidelity prototypes were produced to get a sense of what was going to be developed and how it should look. Figure 7 presents one of the initial low-fidelity prototypes that was created.

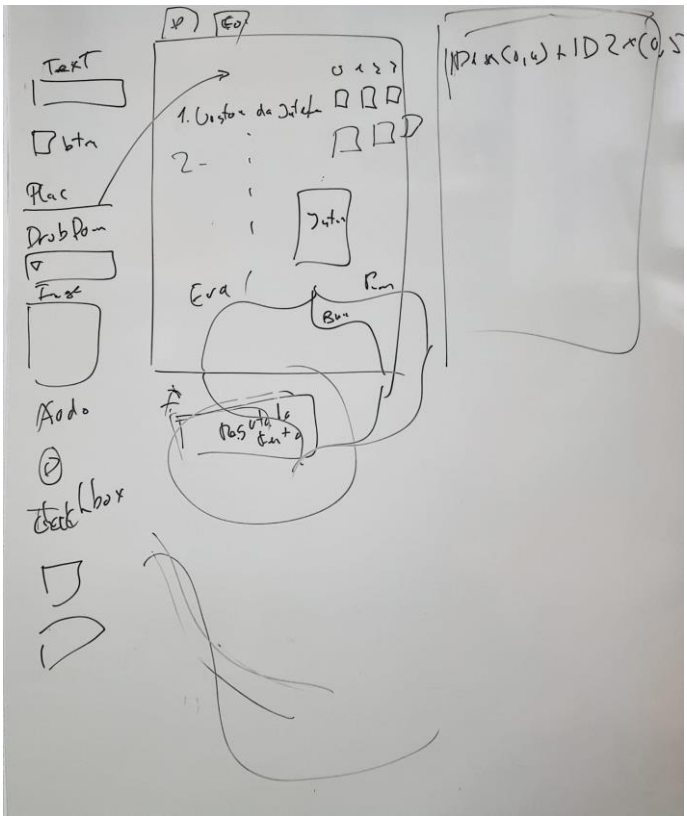


Figure 7. Low-fidelity prototype developed.

Then, high-fidelity prototypes were created. Because the low-fidelity prototypes were modified iteratively (according to the feedbacks received by the interviews), they differ from the high-fidelity ones. It is worth noting that the high-fidelity prototypes were created by the designer Eva Freitas, a member of the *Psyment* team, using the software *AdobeXD*. Figures 8 and 9 show two interfaces of the prototypes that were created.

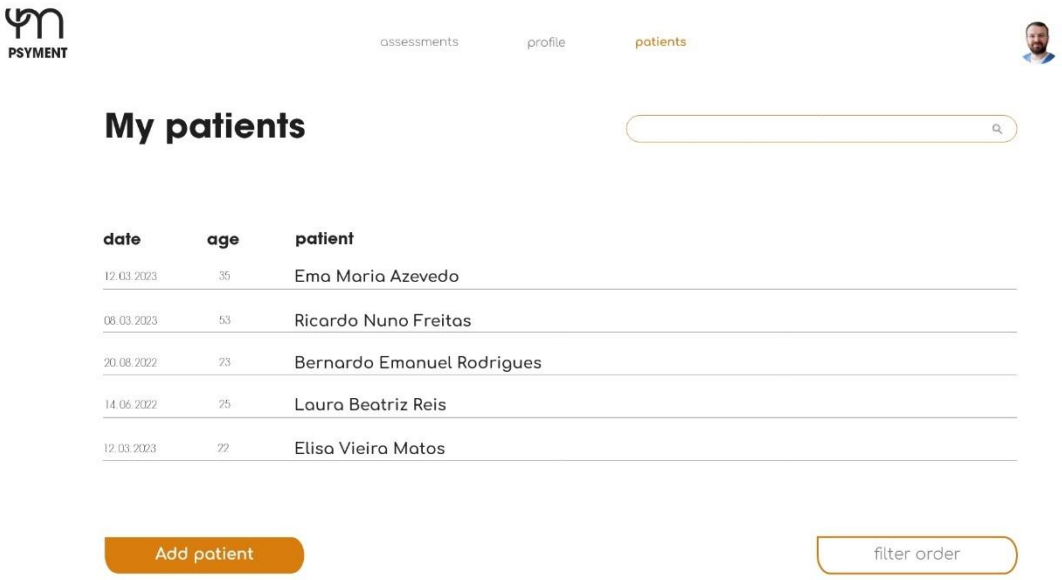


Figure 8. High-fidelity prototype - interface of the add patient functionality.

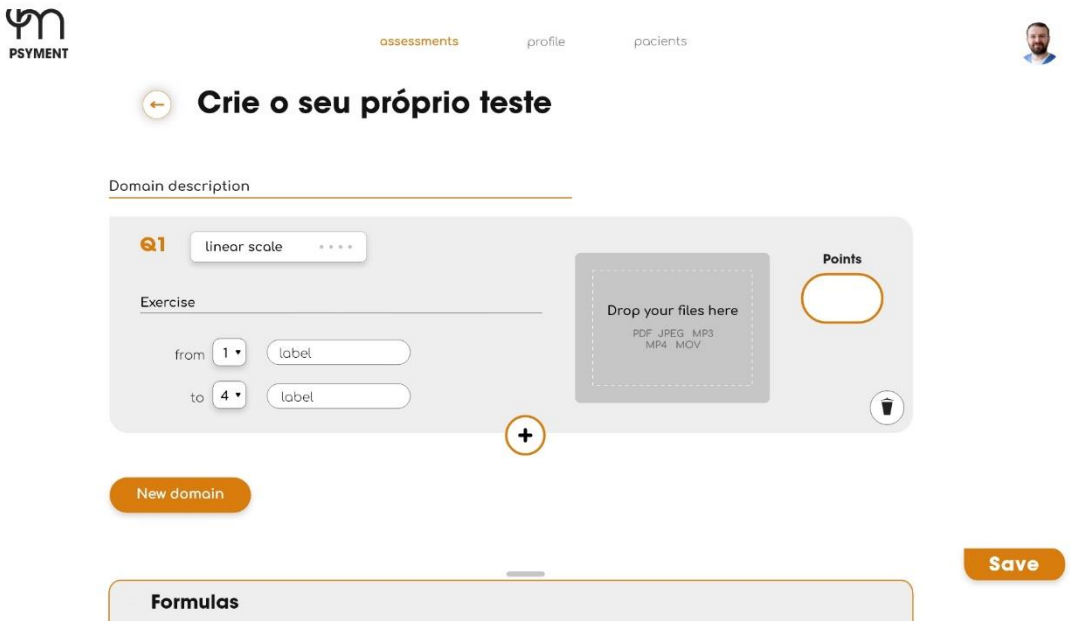


Figure 9. High-fidelity prototype - interface of the test creation functionality.

Note: The data used in the prototypes (see Figure 8) is arbitrary and it is not related to any real person in specific ("*dummy data"*).

4.2. Usability tests

4.2.1. Methods

In this part of the development process, the team conducted consented usability tests at Casa de Saúde São João de Deus with 7 psychologists and health professionals who apply conventional paper/pencil tests daily to evaluate the high-fidelity prototypes. The usability tests took place on April 6th and 10th, 2023. Several tasks (goals) were developed and assigned to health professionals. They were created to help health professionals interact with the *Psyment* application (see Appendices). The order of tasks for each user was determined at random, to avoid learning bias while using the platform. Furthermore, following the usability tests, health experts completed surveys to provide the group with feedback on the platform. This input comprised an assessment of the testing experience, as well as an assessment of the platform and functionality. The questionnaires used to evaluate the platform and the tests were *System Usability Scale (SUS)*, *NasaTLX* and *Intrinsic Motivation Inventory (IMI)* (see Appendices).

4.2.2. Results

Questions / **Participants**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**SUS Score**	**SUS Score (average)**
P01	5	1	5	1	5	1	5	1	5	1	100.0	88.6
P02	5	1	5	2	5	1	5	1	4	1	95.0	
P03	5	1	5	2	4	2	5	1	4	1	90.0	
P04	4	1	4	2	4	2	4	2	4	4	72.5	
P05	5	1	5	1	5	1	5	1	4	1	97.5	
P06	5	1	5	1	4	2	5	2	5	2	90.0	
P07	4	1	4	3	4	2	4	1	3	2	75.0	

Table 2. Quantitative results of SUS

Participants	Mental demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
1	2	2	4	9	4	1
2	2	1	1	2	2	1
3	2	1	2	2	2	1
4	2	1	1	6	4	1
5	2	1	2	3	1	1
6	5	4	6	1	5	1
7	7	2	5	3	5	1
Total (Average)	3.14	1.71	3.00	3.71	3.29	1.00

Table 3. Quantitative results of NasaTLX

		Participants						
		P01	**P02**	**P03**	**P04**	**P05**	**P06**	**P07**
Questions	**1**	7	7	7	6	7	7	6
	2	7	7	7	6	7	7	7
	3	7	6	6	7	7	7	7
	4	7	6	6	7	7	6	7
	5	7	6	7	6	7	6	5
	6	7	7	5	6	7	7	5
	7	7	6	7	6	7	6	6
	8	6	7	7	7	7	7	7
	9	6	7	7	7	7	6	7
	10	7	7	6	7	7	7	6
	11	7	7	6	7	7	7	6
	12	7	7	7	7	7	7	7
	13	5	6	5	6	7	7	5
	14	7	7	7	7	7	7	7
	15	7	6	5	6	7	7	6
	16	7	7	6	6	7	7	6

	17	7	6	7	6	7	7	6
	18	7	6	6	6	7	7	5
	19	7	7	7	6	7	7	5
	20	6	7	7	7	7	6	7
	21	7	6	1	7	7	7	5
	22	7	7	6	6	7	7	7
	23	7	6	6	6	7	4	5
	24	7	7	7	7	7	7	7
	25	7	6	6	6	7	6	5

Table 4. Quantitative results of IMI

Domain	Total per domain	Total (average)
Interest/enjoyment	6,48	6,39
Value/usefulness	6,35	
Perceived choice	6,45	

Table 5. Results of the IMI evaluation for the usability tests

4.2.3. Discussion

Regarding *SUS*, the average score was 88.6, which was satisfactory overall. However, there were two patients with scores clearly below the others, 72.5 and 75. This yet shows that there is room for improvement. It is also worth noting that the two healthcare professionals who granted the application with the lowest scores have the most professional experience within the group that was assessed, with 12 and 19 years, respectively. They may find it more difficult to use technology because they are the oldest members and have been dealing with outdated methods and instruments for years.

Regarding *NasaTLX*, the average was determined for each question's score that matched a particular workload domain. Performance domain had a maximum average of 3.71 out of 10, making it the measure in which they succeeded the most. However, it is determined that this is a low value and that certain improvements to the platform, as previously discussed, are required. Nonetheless, it is worth mentioning that health professionals did not feel frustrated while interacting with the platform since the result for this domain was the minimum possible, 1.0.

Regarding *IMI*, the average was determined for each question's score that matched a particular workload domain and then the calculations for each domain were performed. Some domains are calculated using the reversed value of the answer to the question, that is represented with (R). Domain *Interest/enjoyment* includes questions 3, 5, 7, 11, 12 (R), 15, 17 and 23. Domain *Value/usefulness* includes questions 1, 4, 6, 10, 13, 16, 19, 21 and 25. Domain *Perceived choice* includes questions 2, 8 (R), 9, 14 (R), 18 (R), 20 (R), 22 and 24 (R). As it is possible to see in Table 5, the domain with the lowest result was *Value/usefulness* with 6,35. Analysing the domain this result was registered, the result score is understandable, although it is very positive. The cognitive load associated with adapting to the new technology could have slightly diverted their focus from recognizing the tool's usefulness in improving the assessment process during the usability tests. On the other hand, the domain with the highest result was *Interest/enjoyment* with 6.48. It is considered an excellent result since the best score possible is 7.0. Examining the domain in which this outcome was recorded, the rationale becomes apparent. The health professionals, as indicated in the interviews (refer to Appendices), have explicitly conveyed that an application of this nature would significantly streamline their daily tasks, particularly in the context of assessment applications. Despite the technological difficulties, they revealed motivation to engage with the platform. Finally, the total domain average score for the usability tests of the 7 health professionals was 6,39 which is a very good score. Nonetheless, as seen in other usability tests, there is room for improvement in the application and the results can be improved.

Final considerations

Overall, according to the results, it is possible to consider that the usability tests were successful even though they had limitations such as the number of health professionals that were tested and the fact that all the health professions belong to the same organization and consequently may have similar routines/practices. All the users concluded all the tasks required. Regarding the difficulties felt by the health professionals, the font size was too small throughout the whole application. Moreover, they asked for a delete patient profile functionality since it was not available in the test. Moreover, the language was a barrier since the prototype of the application was developed in English. However, with the team assistance, they could understand the application either way. They also mentioned that in the register form, there should be a “job” field since the user can be a psychologist or a health professional (e.g. nurse).

Regarding the test creation functionality, they mentioned that some *UI* elements as the title and the points for each question needed a change as well. Also the section to add options was not very intuitive. The text box element from the answers was not intuitive as well.

Regarding the test sharing functionality, they mentioned that the application should have a search functionality to pick the user to share the test with based on their name, not only the mail address. The buttons to proceed in the test-sharing functionality were not very intuitive. Regarding the results of tests, the time is not relevant in the general view of the test. Also, the points should be grouped by domain since some tests evaluate different aspects and domains are useful to identify the different disturbs. Moreover, some more details of the *UI* on the different pages of the application were also mentioned by the participants of the usability tests (e.g., text alignments).

Lastly, according to the *IMI* and *NasaTLX*, healthcare professionals did not feel stressed by the tasks assigned to them and felt good while completing the tasks and interacting with *Psyment*, which is one more positive indicator of the platform's User Interface's quality.

Development

5. Development

In this chapter, are presented several topics regarding *Psyment* application development. Those topics are: 1) Requirements of the application, that will be used by psychologists and health professionals, 2) a review of technologies that were used for the current work, 3) a review of Architectural patterns that applied to the current work, 4) Diagrams of the developed software and 5) all the security aspects associated to the implementation of *Psymen*, attending to all psychologists' needs.

5.1. Requirements

In this section, the requirements regarding *Psyment* application are specified. These requirements were designed according to all the data collected in the research part *Psyment* application development, mentioned before, that included interviews with health professionals, and tests with a high-level prototype developed to address what is required by health professionals to carry out their work more effectively.

5.1.1. Functional requirements (F.R.)

F.R.1. The system should allow the user to create an account by entering a name, email, psychologist number and password.

F.R.2. The system should allow the user to login.

F.R.3. The system should allow password recovery.

F.R.4. The system should allow consulting the profile of each user.

F.R.5. The system should allow the user to edit account data.

F.R.6. The system should allow viewing the profile of users registered in the application.

F.R.7. The system should allow creating a test.

F.R.8. The system should allow the user to create their tests.

F.R.9. The system should allow the user to create their own questions.

F.R.10. The system should allow the user to ask quantitative and qualitative questions.

F.R.11. The system should allow the user to ask questions using multimedia.

F.R.12. The system should allow the user to create their test analysis metrics.

F.R.13. The system should allow the user to be able to create metrics for quantitative questions.

F.R.14. The system should allow editing of a test created by the user.

F.R.15. The system should allow editing of questions in the tests created.

F.R.16. The system should allow editing of the test name.

F.R.17. The system should allow the user to be able to change the analysis metrics.

F.R.18. The system should allow sharing of a test between users.

F.R.19. The system should allow editing of a test predefined by the system.

F.R.20. The system should allow consulting a test.

F.R.21. The system should allow viewing created tests.

F.R.22. The system should allow viewing tests shared by other users.

F.R.23. The system should allow viewing predefined tests.

F.R.24. The system should allow analysing test results.

F.R.25. The system should allow observation of test results individually for each patient.

F.R.26. The system should allow comparing the results of a test on the same patient.

F.R.27. The system should allow viewing of the results of different tests simultaneously.

F.R.28. The system should allow the sharing of test results between users.

F.R.29. The system should allow users to share a patient's results with another user.

5.1.2. Non-functional requirements

N.F.R.1. The system should run on any web platform.

N.F.R.2. The system must be interconnected with the database.

N.F.R.3. The system must be able to handle concurrent requests.

N.F.R.4. The system shall keep personal information confidential.

N.F.R.5. The system should alert users of any errors in filling out forms.

N.F.R.6. The system should indicate to users' which fields are filled in incorrectly.

N.F.R.7. The system must encrypt user data, such as passwords.

N.F.R.8. The system must authenticate the user's access credentials.

5.1.3. Technological requirements

T.R.1. The system must be compatible with the Windows operating system.

T.R.2. The system must be compatible with the Linux operating system.

T.R.3. The system must be compatible with the Mac operating system.

T.R.4. The system must be compatible with the iOS operating system.

T.R.5. The system must be compatible with the Android operating system.

T.R.6. The system must have internet access.

T.R.7. The system must have access to the device's file system.

T.R.8. The system should support different types of devices such as tablets and computers.

5.1.4. Security requirements

S.R.1. The system must provide an authentication mechanism to verify the identity of users.

S.R.2. The system must have a robust authorization management mechanism to give only the right access to the respective users.

S.R.3. The system must handle session tokens in a secure manner – securely generated, stored and invalidated after a certain period of time.

S.R.4. The system must not leak any sensitive data regarding users (HP and patients).

S.R.5. The system must validate and sanitize every single input passed to the application.

S.R.6. The system must handle errors and exceptions in a secure manner.

S.R.7. The system must have a two-factor authentication mechanism.

S.R.8. The system must have a log functionality that can be accessed by the administrators.

S.R.9. The system must store all the sensitive data in an encrypted format.

S.R.10. The system must utilize security headers.

S.R.11. The system must be tested frequently for security vulnerabilities.

S.R.12. The system must follow secure coding practices.

S.R.13. The system must require a minimum password complexity.

S.R.14. The system must lockout accounts after a determined amount of tries to login.

S.R.15. The system must hardly verify and sanitize file uploads (e.g. profile pictures).

S.R.16. The system must comply with the most common security standards.

S.R.17. The system must not depend on any third-party services for functionality.

5.2. Technologies

A secure and healthy software development starts with the choice of technologies and tools that will be used during the development process and they have a significant impact on the security of the resulting software, as mentioned before. The technologies used in the current work were chosen taking into account the specifications of the project, scalability and performance, maintenance and longevity, existing libraries, experience of the developers, and security.

5.2.1. MongoDB

MongoDB is an open-source NoSQL database system that stores and manages document-oriented data. It was designed to store JSON data natively and it is built on top of JSON and JavaScript. It enhances the efficiency and simplicity of storing, manipulating, and representing JSON data at various points throughout the application [82]. Mongo has its query language named Mongo Query Language which is used to query data from the different collections (tables) in the database. These queries work within RESTful API communication, where HTTP requests are used to post, read, and delete data [83].

Regarding security, MongoDB provides some features to secure the deployments such as the ability to implement a robust access control through role-based access control (RBAC) integrated within the Mongo database. It gives administrators the ability to create custom roles that tailor and specify different access needs and scope of access for different users. Besides this, mongo databases support various authentication mechanisms such as username/password, Kerberos authentication, and others. It ensures that only authorized users can access sensitive data however this configuration is not enabled by default. According to Singh et al. [84], "*MongoDB does not offer most security configurations by default, but it is less prone to injection attacks considering it does not deal directly with a query language in the form of string*" (like SQL for example) [84]. This is a notable security aspect because as stated before, injections are one of the most common web application vulnerabilities. However, in the last few years, several vulnerabilities were found related to MongoDB especially related to DoS attacks such as CVE-2021-32040 [86]. As so, it is important to use the latest version of the database software and keep it up to date.

As so, MongoDB was the database system chosen for several reasons. Mongo has a flexible security posture, where it is easy to implement security measures efficiently. Additionally, it is a flexible document-based model that makes storing complex data structures trivial and allows developers to modify or add fields without requiring a schema update. Its scalability is also a key factor since MongoDB was designed to be highly scalable and can handle large amounts of data. *Psyment* applications will deal with a large amount of data regarding personal information and medical records, making MongoDB a valid choice for storing purposes. Its performance is also notable because Mongo's architecture is highly optimized with features such as native indexing (which makes the query processing more efficient since it does not need to scan every document in a collection). Finally, as it is a non-relational database management system, it allows for ad hoc queries where developers do not need to predefine relationships between tables or specify them in the actual query, increasing flexibility and simplicity [87]–[89].

5.2.2. MongoDB Compass

MongoDB Compass is a free interactive tool for querying, optimizing, and analysing data from and to MongoDB. It was developed by *MongoDB* developers and provides a GUI to handle and interact with MongoDB. It has several features such as a versatile GUI that lets users have a detailed vision about the data intuitively. Moreover, it is possible to assess query performance in granular detail, inspect individual queries, break down multi-stage complex queries, evaluate their performance, and improve it. Finally, it is possible to access and control deployments through the MongoDB shell directly in Compass [90]. Due to all the features mentioned before, MongoDB Compass will be used in the software development cycle of the current project.

5.2.3. *Mongoose*

Mongoose is a library for MongoDB and Node.js. It provides a simple and easy-to-use API for working with MongoDB and enables developers to define schemas and models for their data, making it easier to interact with MongoDB in a structured and organized way. According to [87], it *"helps with many common MongoDB tasks, and removes some levels of complexity (...)"* [87].It also contains a *"whole suite of helper functions and methods*" for interacting with a mongo instance. According to the same source, *Mongoose* has several benefits – a) Schema-based modelling where it is possible to define schemas for data, making it easier to validate and manage data structures; b) Validation and middleware support because it provides built-in support for data validation, which can help to ensure the consistency of the data stored and allows an easier debugging of potential bugs; c) Query building because *Mongoose* provides a powerful and flexible API for constructing queries, which can help retrieving and manipulating data. Lastly, *Mongoose* is highly extensible [87].

5.2.4. *Node.js*

Node.js is an open-source, server-side asynchronous event-drive JavaScript runtime [86]. It is built on Chrome's V8 JavaScript engine and allows developers to run scripts outside of a web browser, enabling them to build scalable and high-performance applications on the server-side. This happens because Node.js was designed without threads which avoids dead-locking situations, since there are no locks. There are very few functions that directly perform I/O activities, so the web process never gets blocked. It is the ideal concept for a scalable application that needs to handle concurrent requests in real-time. Logically, it performs asynchronous processing based on an event loop (single thread) that performs all the call-backs required to successfully perform the task [86].

Node.js boasts notable features and advantages. For example, a) *Node.js* is fast and scalable since the *Chrome's V8 JavaScript* engine provides fast performance. Besides this, Node handles concurrent requests with ease, making it suitable for web applications to scale up; b) Node.js has a large ecosystem of open-source packages available through the *npm* (*Node Package Manager*). These packages allow developers to easily add functionalities to applications. However, using "wild packages" from untrusted sources is highly dangerous and should not be done at all costs, since this can compromise entirely an application due to vulnerabilities in package's source code; c) *Node.js* uses non-blocking asynchronous I/O operations, which allows to handle multiple requests concurrently, as mentioned above, without getting blocked or crashing; d) Node allows a microservice-based architecture, where web applications are divided into independent services that communicate with each other, such as APIs [91], [92].

5.2.5. *Express*

Express "acts as a light layer atop the Node.js web server, making it more pleasant to develop Node.js web applications" [87]. It is a minimal web application framework that provides a range of features and tools to help developers build robust and scalable web applications quickly and easily. It includes a powerful routing system – that allows developers to define various routes for different *HTTP* methods and URLs, which makes it easy to create clean and organized APIs, handle different types of requests, and map them to appropriate functions or controllers. These conditions combined with a robust implementation help create protections against attacks such as *HTTP* Verb Tampering that tests the application's response by sending several *HTTP* method requests. A system, if vulnerable to this attack, might leak sensitive data and/or reveal access-control flaws. Additionally, *Express* offers middleware support, and a range of integrated *HTTP* utilities through a robust API [88]. With the middleware support, instead of having one primitive request handler function, becomes reliable by calling several request handler functions that deal with a small chunk of the requests to the different pages and functionalities, leading to a more efficient and better performance.

5.2.6. *Embedded JavaScript (EJS)*

Embedded JavaScript (EJS) is a templating language that enables the embedding of JavaScript code within *HTML* templates. It lets developers generate *HTML* markup dynamically with plain *JavaScript* logic [89]. Between EJS special tags/delimiters (e.g., <% %>), it is possible to include JavaScript expressions, variables and logical algorithms.

According to [89], some of the core features of this technology are the usage of JavaScript vanilla without the need for understanding specific framework's syntax. As so, the development time will be faster. Moreover, EJS caches some commonly used *JavaScript* functions, which results in a faster execution of applications, and grants easy debugging, since EJS errors are plain *JavaScript* exceptions, as this template completely relies on vanilla *JavaScript*, as mentioned before. Finally, EJS is maintained by a large community of users and is under constant development [89]. Additionally, EJS can work along server-side frameworks like *NodeJS*, allowing dynamic content to be generated and sent to the client. Then, the embedded *JavaScript* code within the EJS template is evaluated by the server-side *JavaScript* and produces the final HTML output that is sent to the client dynamically and constructed by the browser accordingly [93].

From a security perspective, this feature allows the server to have more control over the application, defining what is sent to the client. This inevitably makes the system more secure. It should be noted that the usage of templates can lead to specific vulnerabilities such as Server-Side Template Injection, if flawlessly implemented. Additionally, since EJS is maintained by a

large community of users, as mentioned before, a potential vulnerability in the source code would be mitigated quickly.

5.3. Review of architectural design patterns

In this section, will be reviewed several architectural design patterns. These design patterns will be compared and discussed to define which is the most suitable option for the current project.

5.3.1. Model-View-ViewModel (MVVM)

Model-View-ViewModel (MVVM) is a software architectural pattern used to build user interfaces. According to [94], *"(...) the heart of the MVVM design pattern is the separation of a view's logic from its look"*. It separates an application into three interconnected main components: *Model*, *View* and *ViewModel*.

The *Model* component represents the data and business logic of the application that the user must work with. From a general point of view, it encapsulates the data and provides methods for accessing and manipulating it. As stated in [94], a model can be defined as an object that usually contains data collected by the server. The *Model* is responsible for managing the application's data and data-related operations. Although the *Model* is not directly connected to the user interface, its primary role is to ensure data integrity and consistency that will be displayed in the mentioned user interface.

The *View* is responsible for displaying the user interface elements to the end-user. It receives input from the user and communicates with the *ViewModel* to retrieve and update data. It does not contain any business logic. According to [94], *"(...) view's code-behind class should be considered essentially irrelevant, and little to no code should be written in it (...)"*. This component is considered a passive component since it only contains the code for the interface (*UI*) and as mentioned before, does not contain any business logic.

The *ViewModel* is responsible for maintaining the state of a view. It exposes data and operations to the view and handles the logic from the view and routes to the respective behaviours. As stated in [94], *"(...) its name may indicate that the purpose of a ViewModel is to provide a view of a model, abstracting the model for consumption by the view, it's really the other way around. The real purpose of a ViewModel is to maintain the state of the view and therefore should be considered a model of the view (...)".* It provides the necessary data and commands from the *Model* to the *View*, and handles the communication and interaction between both of these components. It facilitates the separation of concerns, ensuring the *View* focuses solely on user interface rendering while the business logic and data reside in the *Model*.

5.3.2. *Model-View-Controller* (MVC)

Model-View-Controller (MVC) is a software architectural pattern commonly used in the development of several types of applications such as web applications and rich clients. It separates an application into three interconnected components that are built to handle different aspects of the application and have different responsibilities: the Model, the View and the Controller [69], [95].

The *Model* component represents the data and business logic of the application that the user must work with. It contains the data structures, database models, database interactions and any rule of the application. This component manages the state of the application and provides methods for accessing and manipulating data. According to [95], in object-oriented terms, it consists of the *"(...) set of classes (...)"* that *"(...) support the underlying problem (...)"* [95]. Additionally, the Model component does not need to know anything about the user interface and the presentation logic, and it is completely independent of them. As stated in [95], *"How much should the model (classes) know about the connection to the outside world? Nothing, absolutely nothing*" [95].

The *View* component is used for all the *User Interface* (UI) logic of the web application. In other words, it is responsible for presenting the data to the user. It is in this component where the different pages of the application are defined in terms of interface. Most of the time, this component will get data from the Model component and render it in a form that is visually attractive and appealing. In this specific context, the "output" of the View component will be sent dynamically to the client to then be parsed and constructed by the browser. Unlike the Model component, the View needs to know something about the Model – its existence [95]. As this component will get data from the Model through the Controller, it needs to know how to differentiate the different data fetched so it can be placed in the correct UI elements. As mentioned in [95], "*They have to know something of its nature. A bookingDate entry field (...) might display a variable of some model class somewhere"* [95].

The *Controller* component acts as an intermediary between the Model and View components to process all the logic and incoming requests. It receives user input from the View and updates the Model accordingly to render the final output. It is also this component that processes all user interactions, triggers the appropriate operations in the Model, and determines the adequate View to display. It acts as a "CPU" for the web application to work and handle all the program flow correctly – coordinates the flow of data and events between the Model and the View.

Final considerations

The architectures reviewed were *Model-View-ViewModel* (MVVM) and *Model-View-Controller* (MVC), as mentioned before.

In general, the *MVC* and *MVVM* architectures are particularly similar. Both provide separation of concerns, making the application easier to develop, test, and maintain. Also, they allow for the independent modification of each component individually, enhancing testability, code reusability, and scalability. As the UI and the server-side code are separated into different components, it becomes easier to integrate both in the further phase of the development process. They also promote better collaboration between the designer and the developers in the respective projects [69], [94], [95].

However, they also have some differences in how they organize and manage the application's components. The biggest difference consists of how they handle the interaction between the *View* and the *Model*. In MVC, the View has a more direct dependency on the Controller, as the *Controller* is involved in updating the *View* with data from the Model. In MVVM, the *ViewModel* provides data to the View using data binding, where the *View* binds to the *ViewModel* to get the necessary data and automatically updates when the *ViewModel's* data changes. This difference is crucial because when the *ViewModel* exposes data to the *View*, and the *View* automatically reflects changes in the *ViewModel's* data due to data binding, this results in a looser coupling between both of these components, enhancing factors such as testability [94], [96].

Regarding security, these architectures themselves do not provide security mechanisms logically, but the rules reinforced by them contribute to a more secure application by promoting good software engineering practices. As they enforce a clear separation of concepts, it makes the application easier to understand and consequently easier to develop as well as security protections and mechanisms to secure the software. Additionally, they also reinforce indirectly security since unnecessary complexity means risk. In terms of a concrete example of implementation, regarding the *Model-View-Controller* (MVC) architecture, the *Model* component can focus on data validation and access control, the *View* can focus on protecting against client-side attacks such as *Cross-Site Scripting (XSS)* and the Controller component can implement authentication and authorization logic.

In summary, both MVC and MVVM are architectural patterns used for organizing user interfaces in web applications using different frameworks. However, the developers decided to use the *Model-View-Controller* (MVC) architecture in *Psyment* application's development. The decision was based on several factors:

- **Familiarity**: *MVC* is a well-established architectural pattern that the current team's developers had already used in the past, and as so, they were already used to it and found it more straightforward and familiar.
- **Simplicity**: As the components (*Model* and *Controller*) are more loosely coupled, this separation of concerns makes it easier to implement them since they are less dependent of each other.
- **Debugging**: The debugging in MVC architecture is theoretically easier since as mentioned before, the components are strongly separated and can be tested almost independently;
- **Lower learning curve**: As the developers already used the MVC but not the MVVM, the learning curve is significantly lower since it does not involve data binding and a closer relationship between the *View* and *ViewModel*. This factor also makes the implementation process quicker, which is a positive factor since the development time was limited.

Figure 10, present below, represents the *MVC* architectural pattern applied to the current project. The respective figure illustrates the web application from a low level of abstraction and the different interactions between the several components are nominated from R1 to R8. As such, the following Routes (R) mean:

- **R1**: Model requests data to the database. For example, regarding the authentication process (e-mail and password), the Model would request a User with these attributes. If nothing is retrieved, there is no User with these credentials in the system.
- **R2**: Database retrieves data to the Model component, the result of a query realized to the application itself. The data is retrieved through HTTP requests between the database server and the mentioned application. These connections are done inside the internal network of the application, and the respective connection with the database is realized based on an authentication mechanism with a username and a secret. These values must be unguessable to keep the database safe.
- **R3**: Model returns the data to the Controller (in terms of implementation, for example a function will get the data required and send it to the View to be rendered dynamically).
- **R4**: Controller asks Model to provide data. For example, a user accesses a specific page, but there is data required to be displayed on the page. The Controller handles de request and needs to interact with the Model component to fetch the respective data.
- **R5**: Controller sends the UI resulted from the request initially made by the user and after the required operations server-side done, sends it to the client (User browser) to be parsed and constructed client-side.
- **R6**: The User tries to access or perform an action on a specific page and a request is made to the application. This request can be a GET or a POST, since all the other requests are

considered not necessary for the purpose of the developing application. Consequently, requests with different HTTP methods should be blocked and/or dropped.

R7: The View Component sends the UI page to the Controller, accordingly with the request made to the application. This request flow logic is realized by the Controller and it simply fetches the display from the View component.

R8: The Controller Component requests a specific page to the View based on the action that the User requested to the application. The View has the specific fields where data should be inserted within the page, that results on a dynamic render of the page.

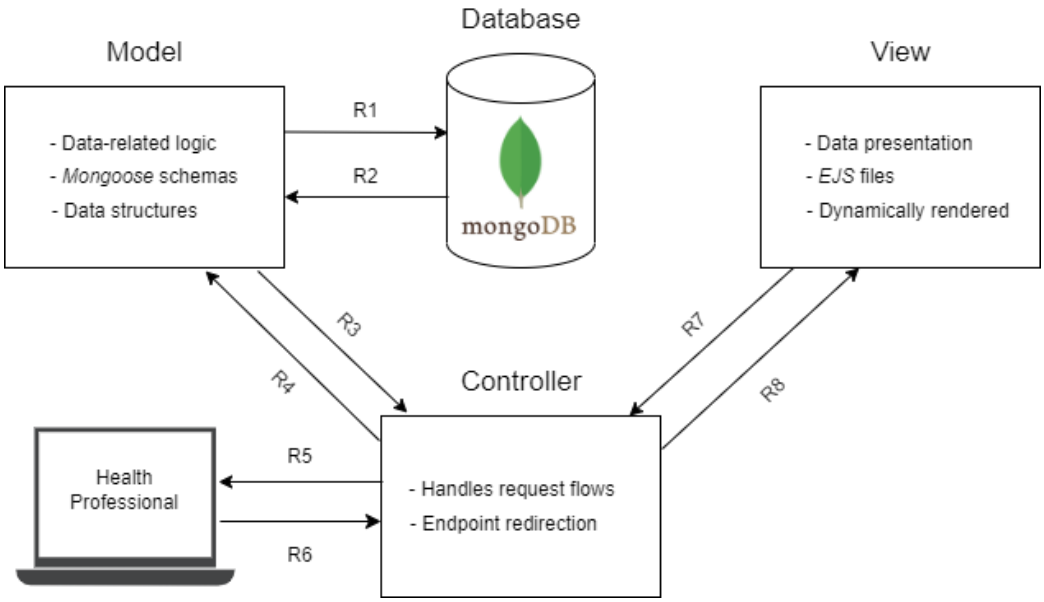
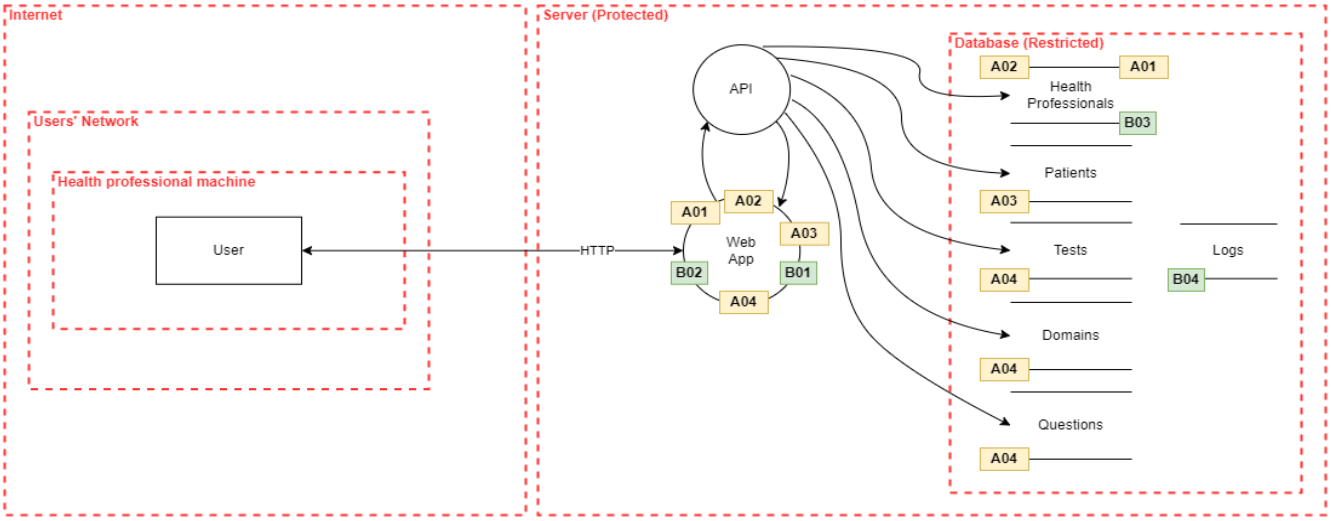


Figure 10. MVC architectural pattern of the current application.

5.4. Diagrams

5.4.1. Structural diagram

A structural diagram shows the static structure of the system's pieces, how the objects and components relate to another. It displays the elements, modules and components that are present in the system. In this instance, it also provides the primary security controls for the created system that include *Authentication, Access Control, Encryption* and *Logging*.



Assets	
ID	Description
A01	Health professional credentials
A02	Health professinal PII
A03	Patients PII
A04	Psychological and cognitive testes

Security Controls	
ID	Description
B01	Authentication
B02	Access Control
B03	Encryption
B04	Logging System

Figure 11. Structural diagram of the Psyment application.

Implementation

5.5. Implementation

The following section describes some specific well-known client-side vulnerabilities, and their implementation security measures regarding the developed application. The section is divided into two parts: **client-side** topics and **server-side** topics.

5.5.1. Client-side topics

Client-side refers to everything in a web application that happens on the end-user device, the client. This includes what the end-user sees, such as text and buttons, but also might include some actions that do not have any clear impact on the display such as the execution of some *JavaScript* scripts.

5.5.1.1. Cross-Site Scripting

Cross-Site Scripting (XSS), as mentioned before, is a vulnerability that allows an attacker to inject malicious *JavaScript* code into a vulnerable application in a way that the malicious code is returned to the clients (users) and executed in their browsers. When the code is executed in their browsers, the interaction between the user and the web app can be compromised. *XSS* attacks can allow an attacker to impersonate another user through cookie stealing, perform actions that the attacked user can do, change the aspect of a website (virtual defacement), inject malicious features on the website, capture the login credentials, and more [97]. There are three types of XSS – reflected, *DOM-based*, and stored [97].

Reflected XSS exists when an application receives data from an HTTP request and includes this data within the response in an insecure way. If the application does not perform any kind of processing of the data at all, this vulnerability can be quickly exploited. As the data contained in the response is passed to the app through an HTTP request (GET) via request parameters, the attacker can craft a malicious URL that passes the malicious parameter values. When the response is parsed and constructed in the client, the payload (*JavaScript* script for example) will be executed client-side in the context of the current user session of the application [97].

DOM-based XSS exists when a web application has some *JavaScript* code that processes data from an untrusted source in an insecure way, normally writing this data back to DOM. The untrusted source refers to an attacker-controlled source. If it passes the data to a function (also known as "sink") that allows dynamic code execution such as *eval()* or *innerHTML*, it might allow an attacker to run *JS* code. Another common source for this attack to happen is the *window.location* that accesses some specific URL fragmented parts.

Stored XSS also known as persistent XSS exists when an application receives data from an untrusted source and include this data later in the HTTP requests in an unsecured way. This attack is called persistent since the data sent through an HTTP request is stored in the back end, on a

database or any type of cache mechanism. As so, the malicious data is stored and every time a user makes a request to a single page, the response will contain the malicious code injected. Stored XSS is more dangerous than the other types of XSS because in this case, there is no need to induce the user to access a specific URL, since the server already responds with the page infected which makes it more critical.

According to [97], Cross-site scripting prevention can be achieved via two levels of defense generally: *Encoding* data on output and *Validation* data on arrival. When encoding data for output, it is advisable to implement the process directly before user-controlled input is written or reflected on a page. The contextual placement of the data plays a crucial role in assessing the associated risks and determining the appropriate encoding method to mitigate potential vulnerabilities. A value inside a *JavaScript* string requires a different type of encoding and/or escaping when compared with a value inside an HTML context (*value* for example) [97].

When validating input data, it is imperative to conduct thorough and strict verification. The specific validation criteria should align closely with the intended use-case for the data. For instance, if an input is anticipated to be numeric, the validation process should rigorously ensure that the entered value consists of integers and/or adheres strictly to an expected set of characters. [97].

However, as mentioned before, the application was developed using a template engine, *EJS*. This template engine was used to embed dynamic content in HTML tags, and they usually have their escaping system and special syntax. As so, it is important to block common syntax use-cases from dangerous inputs. This can be achieved via blacklisting and/or whitelisting characters. Whitelisting is a better practice since it is easier to define everything that is accepted instead of everything that can be harmful.

Regarding the development carried out in terms of *XSS* protection, it started in the conception phase. When developing the application, it was avoided as much as possible the usage of user-input directly in pages, reflected or within the *DOM*. This awareness is important as it helps decrease the number of potential vectors for *XSS* attacks, from a general point of view.

Initially, there were identified potential vectors for *XSS* attacks like search mechanisms such as the /*mypatients* functionality where a health professional can search for a patient by his name. As mentioned before, *EJS* provides automatic escaping of user input, which by itself already helps prevent *XSS* attacks. Figure 12, presented below, represents the */mypatients*, identified as a potential vector for *XSS* attacks. As so, it will show the encoding/escaping performed by *EJS*.

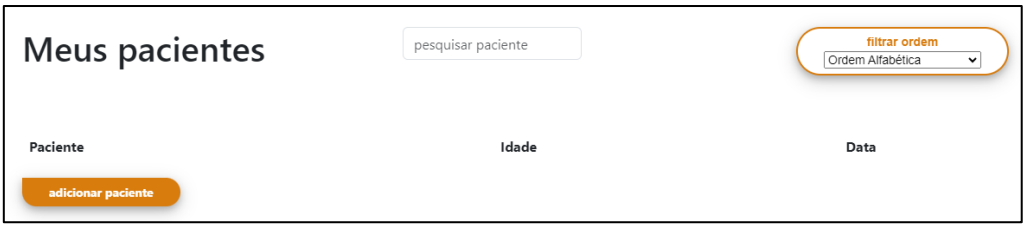


Figure 12. Mypatients page as potential vector for Cross-Site Scripting attacks

Input used for testing: *"><img src=x onerror=alert()>*

When searching for the string present above, the response of the app is represented in Figure 13, illustrated below. The input passed is reflected inside the placeholder used for the search.

URL: */mypatients?search="><img+src%3Dx+onerror%3Dalert%28%29>*

Figure 13. Mypatients page with in search input reflected in the placeholder

Moreover, when the source code of the page is inspected, it is observable some special characters encoded/escaped by *EJS* – Figure 14. It is worth mentioning that no encoding, escaping, or sanitization was performed in the server-side for instance, in this case.

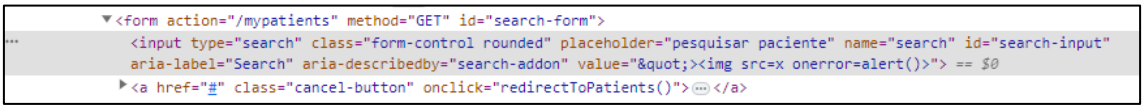


Figure 14. Part of the source code of the response with input characters encoded

As seen in Figure 14, double quotes were encoded to *"* and as so, it prevents the injected code from running in the client, when the rendering of the page is performed. However, this encoding, although useful, is not enough to fully prevent and protect the current application. Thus, as stated above, it is important to validate the input. To achieve this, the library *dompurify* [98] was used. It helps sanitize the input before rendering it in cases of reflection as the page specified before. The following snippet of code represents an implementation of validation of the data on arrival and encoding data outputted of the */mypatients*, as it is a good practice in terms of *XSS* protection [97]. This code is contained within the */mypatientsController.js*.

```
const createDOMPurify = require('dompurify');
const {JSDOM} = require('jsdom');
. . .
const window = new JSDOM('').window;
const DOMPurify = createDOMPurify(window);

const searchTerm = req.query.search;

//Validate input
const sanitizedInput = DOMPurify.sanitize(searchTerm);
. . .
```

From a general perspective, the code gets the value of the search with *req.query.search* and passes it to the *DOMPurify.sanitize()* method. This method is considered the core of this library and makes the input safe for rendering. Initially, *DOMPurify* creates a so-called *"clean-document"* to parse and sanitize the inputted *HTML* code. It creates a virtual *DOM* environment that mimics a browser's document structure. Once this is done, then it processes each element and attribute of the code passed as a parameter to the function and compares them to a whitelist of allowed elements and attributes. This prevents harmful code from being executed. It also checks if the resulting *HTML* code maintains a valid document structure, handling things like unclosed tags for instance. Finally, the sanitized input is validated and ready to be rendered [98].

For the same example given above, */mypatients*, with the *DOMPurify* implementation on top of the *EJS* encoding, the same functionality with the same input used for testing produces the following result, Figure 15.

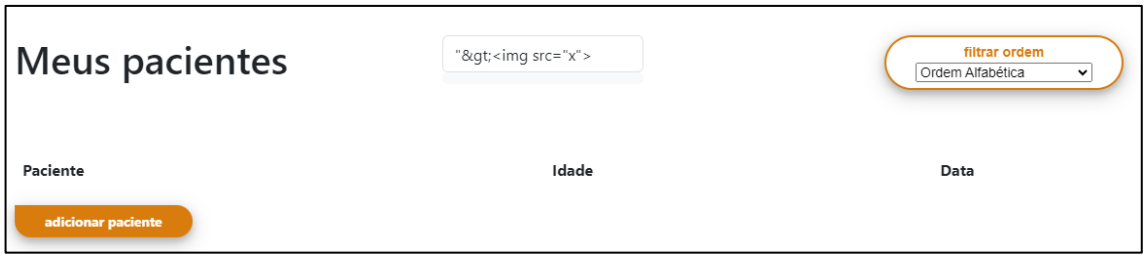


Figure 15. Mypatients page with the input sanitized by DOMPurify

When inspecting the code (Figure 16), the output is almost completely encoded, since most of the characters passed are considered dangerous and commonly used in *XSS* payloads.

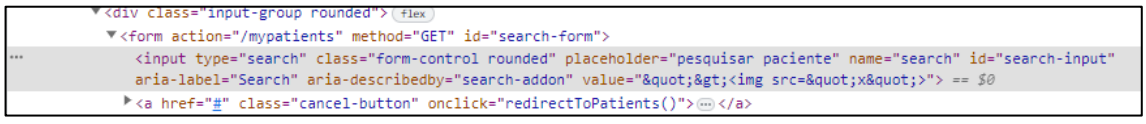


Figure 16. Part of the source code of the response with input characters sanitized

It is worth mentioning that this validation of input and encoding of output with *dompurify* was implemented throughout the whole application and for every single functionality that required user input.

Regarding the cookies, as *XSS* is a commonly used attack to impersonate another user through cookie stealing for instance, it is important to ensure cookies are securely used. The usage of attributes such as *httpOnly* prevents *JavaScript* code from accessing the cookie via the API *document.cookie*, mitigating the risk of theft through *XSS attacks*. Nonetheless, the cookie specification and configuration is explained, more in-depth in section *6.5.2.2. Access Control*.

5.5.1.2. Clickjacking

Clickjacking is a client-side attack in which a user is tricked into clicking on content on a hidden website by clicking on other content on a decoy. The attacker manipulates the presentation of a sensitive UI element by placing it outside its intended context (e.g., concealing it through transparency), leading the user to engage with it in a manner that is also out of context. [99], [100]. These actions out of context can be tricking the user into clicking on a button to win a prize and this results in the changing of his account's password. As so, clickjacking depends on the inclusion of a page that contains a button or a hidden link, normally within an *iframe* [99]. Overall, clickjacking is a browser-side behaviour and its success highly depends on browser functionality.

According to [99], server-side protections against clickjacking attacks consist of constraining the use of components as *iframes*. To achieve this, two mechanisms for server-side clickjacking protection are *X-Frame-Options* and *Content Security Policy*.

X-Frame-Options is a response header that provides the website owner with control over the use of *iframes* and objects so that the including a web page within an *iframe* can be blocked. This can be achieved with the directive *deny* [99].

Content Security Policy consists of a detection and prevention mechanism that provides defense against attacks such as clickjacking. It is usually implemented in a web application as a return header in the responses in the form of *Content-Security-Policy: policy*. This allows the clients to know information about allowed sources of web resources that the browser can rely on to intercept malicious activities. According to [99], the recommended protection regarding this prevention mechanism is to use the directive *frame-ancestors* 'none'.

Regarding the development carried out in terms of clickjacking, the middleware *helmet* [101] was used. It has various middleware functions that assist in setting HTTP headers that enhance the security of web applications. The following snippet of code shows the implementation of the already mentioned headers using *helmet*.

```
const helmet = require('helmet');
...
app.use((req, res, next) => {
  res.setHeader('X-Frame-Options', 'deny');
  res.setHeader('Content-Security-Policy',"frame-ancestors 'none'");
  next();
});
```

Then, when a request is performed to the platform and a response is provided, it is possible to see that these headers are present and, as so, the application is protected against clickjacking. The following figure, Figure 17, shows an arbitrary request and the following response that contains the referenced headers.

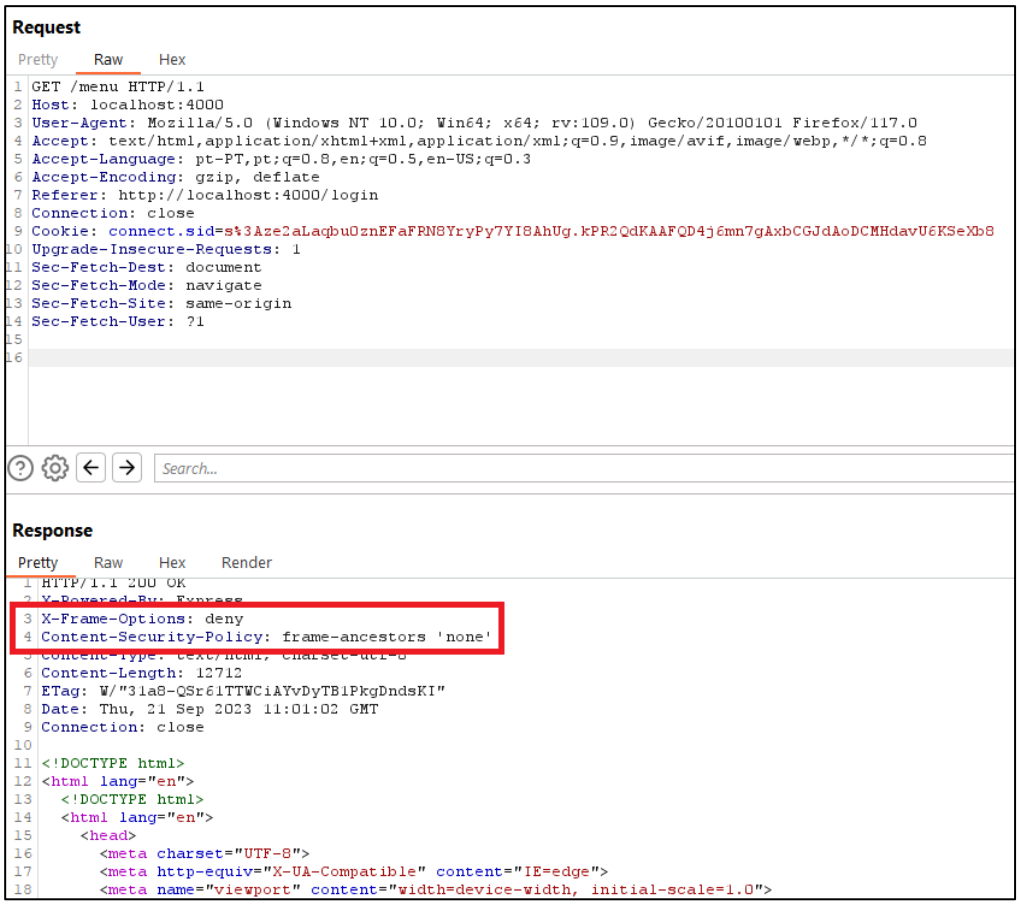


Figure 17. Request and Response with the headers regarding Clickjacking protection.

5.5.2. Server-side topics

Server-side topics refer to everything in a web application that happens on the server. This most of the time is not visible to the end-users but it is where all the business logic happens, including processing of data, interaction with the databases, and handling the flow of the program. It is also the most critical part of a web application since it controls the entire application. If an attacker can get to the server, it means the entire application is compromised as well as all the users. Logically, all the vulnerabilities that lead to server access are considered highly critical.

5.5.2.1. Authentication

Authentication is the process of validating a user's identity. It involves verifying that they are indeed who they claim to be. Most of the websites are designed to be accessible to anyone who has access to the internet. Thus, strong authentication systems are an essential component of good web security [75].

Regarding web applications, a typical example of an authentication procedure is the login mechanism, where a user needs to provide a password to prove he is who he says he is. This is frequently known as the knowledge factor, where the app verifies something that only a specific user **knows** to prove his identity [102].

According to [102], most authentication vulnerabilities happen because the authentication mechanisms are weak and vulnerable to brute-force attacks, where there is no limit to tries a user can do to prove his identity and logic flaws due to poor implementation, where the *"authentication wall"* can be bypassed without knowing any secret. This is also known as broken authentication and it is present in OWASP Top 10 [58], one of the top ten most common vulnerabilities according to [75].

To secure authentication mechanisms several practices should be applied. The application should take care of user credentials, demanding for complex passwords with certain rules such as minimum length, usage of characters, special characters, and numbers. As mentioned by Wheeler et al. [103], As mentioned by Wheeler et al. [103], *"(…) passwords should be a minimum of eight characters in length. Longer passwords are more secure and should contain upper and lower case characters, numbers, and special characters. This provides a character set of 95 possible characters. An eight-character password using a character set of 95 has a key space of 958, approximately 7×1015, or 7 quadrillion possible passwords. As the key space increases, the time required to perform an brute force attack on a password increases"* [103]. Moreover, username or e-mail enumeration should be prevented because it can get the attacker a step closer to compromise a user's account. Consequently, the application should use generic error messages and not something like '*Wrong password for current username.'*, because this type of message indirectly indicates that a user with the passed username exists. Additionally, the responses to any login request should have the same HTTP status code because it can make it harder to brute-force or detect some logic flaw [102]. These brute-force attacks can be stopped with the implementation of a robust brute-force protection such as a strict IP-based user rate limiting. This prevents users from trying to login several times in a short period, that might indicate a brute-force attack. Finally, applications can also implement a proper multi-factor authentication that can be much more secure than just a password-based authentication. Multi-factor authentication adds a new layer of security to the authentication (…) [104].

Regarding the development carried out, to verify the identity of the users, the primary implementation of authentication was done through a password-based method. From a general point of view, the user must provide a pair of credentials, e-mail and password that must match one of the records in the database, as far as the '*health_professionals'* table of the database is concerned.

In the first phase, the user's input undergoes a series of integrity checks, to know at a glance whether the data entered is correct. These verifications include checking the structure of the email address and whether it is a valid email address, whether the field is empty, and whether it contains spaces (and if so, they are removed). Regarding the password, if it contains spaces (and if so, they are removed) if the field is empty, and if it is a password considered strong (it has been defined that a strong password is a password with at least 8 characters, one capital letter, a lower-case letter, a number and a special character). These input validations were implemented with the support of *express-validator* [105], a well-known set of *Express* middleware that provides several functions, validators, and sanitizers to control the requests received by the main web server. The snippet of code present below represents an input validation implemented for the e-mail address and password (login step), as mentioned before.

```
const { body, validationResult} = require('express-validator');

exports.validateLoginFields = [

  body('email')
  .trim()
  .notEmpty()
  .withMessage('Email é obrigatório.')
  .isEmail()
  .withMessage('As credenciais introduzidas estão incorretas.'),

  body('password')
  .trim()
  .notEmpty()
  .withMessage('Password é obrigatório.')
  .isStrongPassword()
  .withMessage('As credenciais introduzidas estão incorretas.'),
]
```

Regarding the *email* parameter, initially it is called the *trim()* function that is used to remove leading and trailing whitespace from the input string before applying validation rules. It ensures that any extra whitespace surrounding the value is eliminated before the validation checks are performed, such as simple whitespaces, tabs, and line breaks. Then, the function *notEmpty()* is called to check if the input is empty or not. This function is called since the client-side validation

is not enough, as mentioned before. For example, with the attribute *required* in HTML forms, since an attacker can easily bypass this altering the client-side code or using a proxy to intercept requests and change the parameter values, removing any text that was sent through the form. If the parameter received by the server is empty, an error message is triggered and sent to the client, that will display it on the respective page. If not empty, the text passed is evaluated to verify if it is a valid e-mail address. The function *isEmail()* verifies the integrity of the address as it checks whether the input value matches the standard email format, following the pattern *local-part@domain*. It performs various checks to ensure that the email is valid, including verifying the presence of an @ symbol, the presence of a domain, and the correct arrangement of characters. If the email is not considered valid, an error message is triggered and sent to the client, that will display it on the respective page.

Regarding the *password* parameter, the *trim* function is also called as well as the *notEmpty* function, that were already used for the *email verifications*. Then, a verification of the password complexity is done with the support of the function *isStrongPassword*. It is possible to specify attributes to be required for the password. However, the default values are satisfactory for the password complexity and are used for the password policy of the web application – eight-character long password minimum, an upper case, a lower case, and a special character. If any of these conditions is false, the function will return false, and an error message will be triggered.

After this, a routine check is made to verify if there is a user logged in or not, through the session cookie. If yes, the user should not be able to login again and the process is immediately terminated. If not, the logic of the login mechanism is performed, and a query is made to the database to check the inputted values. If both values match an entry of the *'health_professionals'* table for the fields email and password, the user is successfully verified. This business logic was implemented with the support of *PassportJS* [106], a well-known *NodeJS* middleware that allows a simple and unobtrusive authentication. With *PassportJS*, the authentication logic relies on strategies that are methods of authentication used by the mentioned middleware. Although *PassportJS* website provides hundreds of strategies for several methods of authentication such as *OAuth*, *Google*, *Facebook*, username-password and others, the strategy used was implemented specifically for the current project.

To protect the authentication mechanism and block brute-force attacks, that as mentioned before, are one of the major causes of authentication vulnerabilities, it was implemented an account lockout system that locks accounts after 3 attempts to login failed for a certain email address for a defined interval of time of 5 minutes (300 seconds). During this time, even if the pair of credentials passed to the application is correct, the respective user simply cannot login to this account. This way, the chance of brute-force attacks decreases significantly as they become

an exhaustive and time-taking attack. The following snippet of code represents the account lockout system implemented for the current application.

```
const failedLoginAttempts = new Map();
const maxFailedAttempts = 3;
const lockoutDuration = 5 * 60 * 1000; // 5 minutes in milliseconds

… = async (email, password, done) => {
   const lockoutEnd = failedLoginAttempts.get(email);
   if (lockoutEnd && Date.now() < lockoutEnd) {
      return done(null, false, {
        message: `******** ${Math.ceil((lockoutEnd - Date.now()) / 1000)} `,
      });
    }
   const result = …

   if (result != null){
      …
   }
   else {
      const attempts = failedLoginAttempts.get(email) || 0;
      failedLoginAttempts.set(email, attempts+1);

      if (attempts + 1 >= maxFailedAttempts) {
         const lockoutEnd = Date.now() + lockoutDuration;
         failedLoginAttempts.set(email, lockoutEnd);
         return done(null, false,
            {message: 'Conta bloqueada devido a muitas tentativas falhadas.'}
         );}
      return done(null, false,
         {
            message: 'Credenciais inválidas.'
         })}}
```

This specific implementation relies on a ***Map*** data structure, where the login attempts are tracked. The email address is the key, and the lockout end timestamp is the value. After a failed login attempt, the code checks if the account is already locked by comparing the current time with the lockout end timestamp, after searching the data structure for the same key (email). If the account is locked, it returns an error response with the remaining lockout time. Otherwise, it proceeds with the authentication logic. If the authentication fails, the code increments the failed login attempts count for the user. If the maximum number of failed attempts is reached, it sets the lockout end timestamp and returns an error.

Because the account lockout is directly tied to the authentication logic, it is encapsulated within the aforementioned *Passport* authentication strategy. Additionally, the requirement that passwords must be at least eight characters long and contain at least one of each uppercase, lowercase, and special character strengthens security. Moreover, the errors messages regarding the authentication do not reveal which field is wrong to deny a vector for username/email enumeration. All the returning messages are highly generic (e.g. *Credenciais inválidas.*) as it is a good practice in this type of mechanisms, as mentioned before. Figure 18, illustrated below, summarizes the authentication logic implemented.

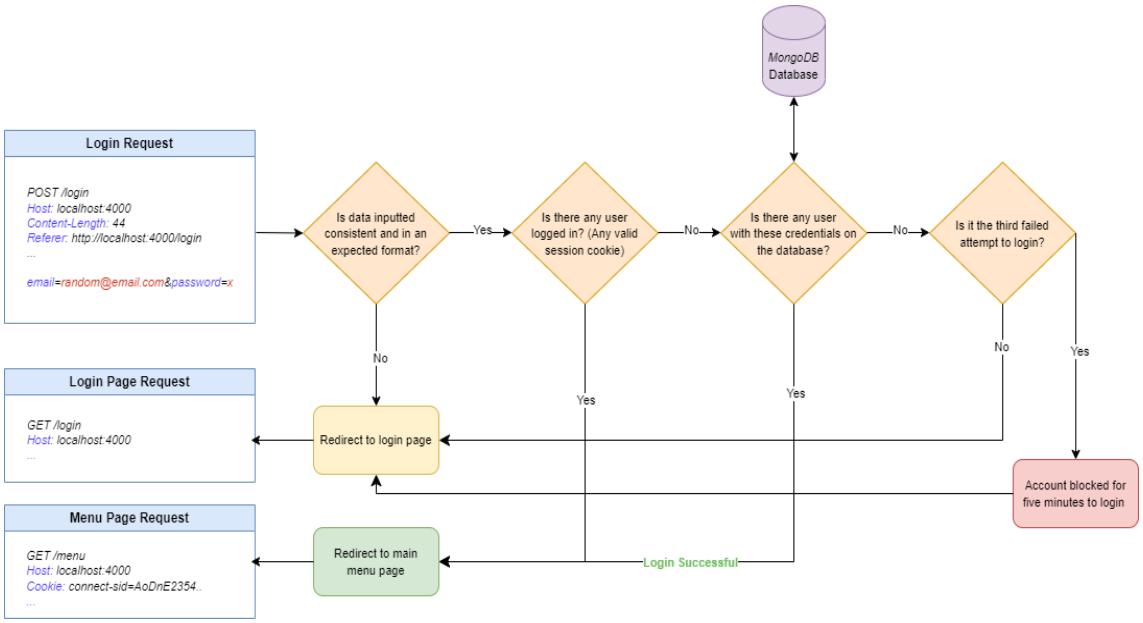


Figure 18. Diagram of the authentication logic implemented.

Moreover, a password recovery mechanism was also implemented. In the first phase, a user needs to provide the email address that is associated with his account. Then, this email is verified within the platform, and if there is any user registered with it, an email message is sent to the inserted address with a URL that contains a unique token, securely generated. This URL and consequently the token can only be used once, and it is directly associated with the user that is requesting the password reset. Thus, a user cannot random or forge an arbitrary token to change another user's password. The emails sent were supported by *nodemailer* [107] – a popular *Nodejs* module for email sending. The following snippet of code represents the logic implemented for the password reset functionality.

```
exports.sendResetEmail = async (req, res) => {

  const { email } = req.body;

  if (email) {
    // Validate and update 'email' if it's not empty
```

```
    await body('email').notEmpty().withMessage('Email cannot be empty').isEmail().withMessage('Email introduzido não é válido.').run(req);
  }

  const errors = validationResult(req);

  if (!errors.isEmpty()){

  } else {

    try {
      // Generate a secure token for password reset
      const token = crypto.randomBytes(32).toString('hex');
      // Find the user by email
      const user = await HealthProfessional.findOne({ email });

      if (!user) {
        //req.flash('error', 'Email not found.');
        return res.redirect('/forgotpassword');
      }

      // Set token and expiration in user document
      user.resetPasswordToken = token;
      user.resetPasswordExpires = Date.now() + 3600000; // 1 hour
      await user.save();

      // Send password reset email
      const transporter = nodemailer.createTransport({
        service: 'Gmail',
        auth: {
          // Precisa-se de criar um email para psyment
          user: '**********************',
          pass: '**********************',
        },
      });

      const mailOptions = {
        from: '************************',
        to: email,
        subject: 'Password Reset Request - Psyment',
        text: `******:\n\n
          ${req.protocol}://${req.get('host')}/resetpassword/${token}\n\n
          ********.`,
      };

      transporter.sendMail(mailOptions, (error, info) => {
        if (error) {
          console.log(error);
        } else {
```

```
            console.log('Email sent: ' + info.response);
          }
        });
        res.redirect('/forgotpassword');
      } catch (err) {
        res.status(500).send('Server Error');
      }
  }
};
```

Initially, the code extracts the email from the request body and validates it using the library *express-validator* [105]. Then, it generates a secure random token for a password reset using *crypto.randomBytes(32)* function and searches for the user in the database based on the provided email updating its field, and storing the token. Following this, the server configures an email *transporter* using *nodemailer* and sends the password reset email with a link containing the aforementioned token to the user. Finally, the server redirects the user to the */forgotpassword* page.

5.5.2.2. Access Control

Access Control consists of the constraints that the application imposes on who wants to perform a determined action or access resources requested by them. It is composed of the authentication part, mentioned before, and the session management. Session management refers to the process of effectively managing and maintaining user sessions on an arbitrary application. Starting from this, the application can identify which subsequent requests are being made by the same user or not [108]. This normally involves the generation of a unique session identifier and associating it with the user's session such as a session cookie. The cookies are stored on the client (browser) and when a request is made to the host, the cookie is passed in the request on a specific header – *Cookie*. Then, the server should process it determine if it is a valid session identifier or not and perform the necessary actions.

According to [108], most of the broken access control vulnerabilities happen because of unprotected functionality, where an attacker can gain access to functionality they are not supposed to, such as an administrative panel. Therefore, it is important to protect all the sensitive functionality that the application might have. Moreover, some applications determine the user's access rights or role based on the login and store this information in a user-controllable location such as a plain text cookie or a preset query string parameter. Then, the application can make the subsequent HTTP requests based on these values. However, an attacker can simply modify the value and gain access to certain functionality. Consequently, it is important to protect these stored values such as cookies in a way that cannot be understandable and editable [108].

As so, to protect against such vulnerabilities, applications should take a defense-in-depth approach that should rely on denying access by default to resources unless they are intended to be public, using a single mechanism throughout the whole application for enforcing access controls, since the overlap of several mechanisms can lead to functionality and information leaking. Additionally, all the input should be filtered and sanitized, as it is a common good practice to avoid most of the most common vulnerabilities. If a web application is accessing or referencing objects from the back-end directly, these references must be checked to ensure users cannot access or manipulate unauthorized resources by modifying these references [59], [108], [109].

Regarding the session management mechanism implemented in the current project, it starts on a robust authentication. After the user being authenticated, a session is created in the web server and a session cookie *connect.sid* is created. This session cookie is the result of a serialization process of the user that was fetched when the login attempt was successful. As so, the cookie holds some information regarding the user, specifically the *id*, but as long as the information is not legible or decrypted, the session mechanism is safe. Still, the session cookie follows some secure practices in order to safeguard them.

Each cookie is created with the support of a cookie secret – a value that is used to sign the session cookie and provide some cryptographic security to the cookie data. The secret used is 64 characters long and contains uppercase, lowercase, numbers, and special characters. The combinations possible for the secret are (26+26+10+30) ^ 64 which is a really large number, making the brute-force simply unfeasible.

Example: *s:FuPLwlIzRa4ljotJhLRkPiJLFYPJ0UHd.NzcXNuc0HkzPSpvBpC5+TpxKJZeGBEYKwsBTRquY/+0.*

Nome	Valor	Domain	Path	Expires /...	Tamanho	HttpOnly	Secure	SameSite	Partition...	Priority
connect.sid	s%3AFuPLwlIzRa4ljotJhLRkPiJLFYPJ0UHd.NzcXNuc0Hkz...	localhost	/	2023-08...	97	✓				Medium

Figure 19. Example of a session cookie generated by the server regarding an arbitrary valid session.

Each time a request is made to the server-side application, the integrity of the session cookie passed in the request is verified by checking the signature against the secret. If the signature is valid, the server knows that the cookie has not been tampered with while in transit and can trust the session data.

Cookies in the web app also have a special flag defined – *httpOnly* that prevents a cookie from being accessed through client-side scripting. As a result, even if a *XSS* vulnerability exists and a user accidentally accesses a link that exploits this flaw inside the web application, the browser will not reveal the cookie to a third party like an attacker's server. As so, if as mentioned

before, the cookie cannot be accessed through client-side scripting (in this case, *JavaScript* for instance).

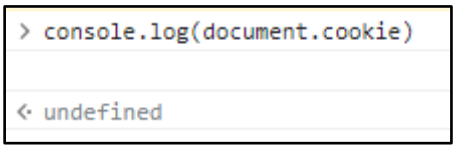


Figure 20. Access to the cookies through the browser built-in JavaScript.

Moreover, the cookies have a set expiration date specified to limit the attack damage if a user-session gets compromised. This is possible due to the flag *maxAge* with the value of 3600 * 1000 * 24 that results in a day in milliseconds. An attacker will only have 24 hours available to exploit the application to a deeper level if a session hijacking attack was performed successfully. When a user logs in successfully, the session cookie is generated and set through the HTTP header *Set-Cookie* which also contains an attribute *Expires* where the expiration date is placed. Figures 21 and 22, presented below, represent the POST request generated by the client to the server when a user is logging in with valid credentials (in this example, an arbitrary account) and its response by the server, with the header *Set-Cookie* already mentioned before, that assigns a session cookie to the current user, creating a session and setting a header for the follow-up requests made by that client. This way, it is known which user is performing the actions and consequently the application can respond accordingly.

Figure 21. POST request of the login.

Figure 22. Response to the login POST request.

Regarding the resource control management, when a specific resource or access to a specific page is requested, the server-side verifies the integrity of the cookie, as mentioned before, and if the user has access to the requested resource, using the session cookie data as base to make the logic decisions. From a higher layer of abstraction, it is simply necessary to verify if the current actions are being performed by a user with a valid session. The following snippet of code shows the verification performed when a user tries to access a specific resource that requires him to a have a valid session in the application.

```
const express = require('express');
const router = express.Router();
const passport = require('passport');

router.get('/menu', (req, res) => {
    if (!req.isAuthenticated()){
        const newUrl = '/login'
        res.redirect(401, newUrl);
    }
    else{
        res.render('mainmenu');
    }
})
```

This verification is contained within the *API* (middleware). When a GET request is performed to */menu*, the server verifies if the request received is authenticated with the method *req.isAuthenticated()*. It is a function provided by *Passport.js*, already mentioned before, that is typically used as middleware to check if a user is authenticated or not. It returns true if the user is authenticated and false otherwise. In this case, it verifies the session and the respective session cookie passed in the request, if the session contains all the necessary properties that indicate that the user is authenticated. As so, when accessing the endpoint */menu*, if the user is not authenticated, the user will be redirected to */login* with the HTTP status code 401 - *Unauthorized*. On the other hand, if the user has a valid session, he will be allowed to access */menu* and will be redirected to the required page.

Moreover, when data is fetched that only a specific health professional should see (e.g., patients of a single health professional), the queries are performed considering the current user session to not leak information that belongs to other users. The following snippet of code represents the case where a page is dynamically completed with data that is associated with a specific user, logged in with a valid session.

```
if (req.isAuthenticated()){
```

```
  const searchTerm = req.query.search;
  let query = {};
  . . .
  if (searchTerm){
    query = {hpId: req.user, name: { $regex: searchTerm, $options: 'i' }};
    console.log('Query: '+query);
    console.log('SearchTerm: '+searchTerm);
  }
. . .
await Patient.find(query)
    . . .
```

The data queried, in this case, consists of patients stored in the database with the *hpId* attribute equivalent to the current user session *Id*. It used the *req.user* key of the object *req* that uses the *Id* to identify each user associated with a session. Consequently, as mentioned before, the health professional can only see his patients, and not leak any information regarding other health professionals.

Finally, when the user logs out of the application or his cookie expires, the session cookie is no longer valid, and it gets deleted from the system (server-side). The cookie might still be stored in the client, but it will not pass the verifications when requesting a page or resource in the server-side. The following snippet of code represents the *API* work to logout a user from the application.

```
router.get('/logout', (req, res, next) => {
   req.logOut(function(err) {
      if (err) { return next(err); }
      res.clearCookie('connect.sid')
      res.redirect('/login');
      }
   );
});
```

Regarding the code, when an *HTTP GET* request is made to the endpoint */logout*, function *logOut()* is called. This function is exposed by *Passport*, and invoking it removes the property *req.user*, clearing the login session. As an additional measure, the function *clearCookie()* provided by *ExpressJS* is used to clear the session cookie – *connect.sid*. The following figures, Figure 23 and Figure 24 represent an arbitrary *GET* request to the application that a user with a valid session performed to logout from the platform and the server response, accordingly.

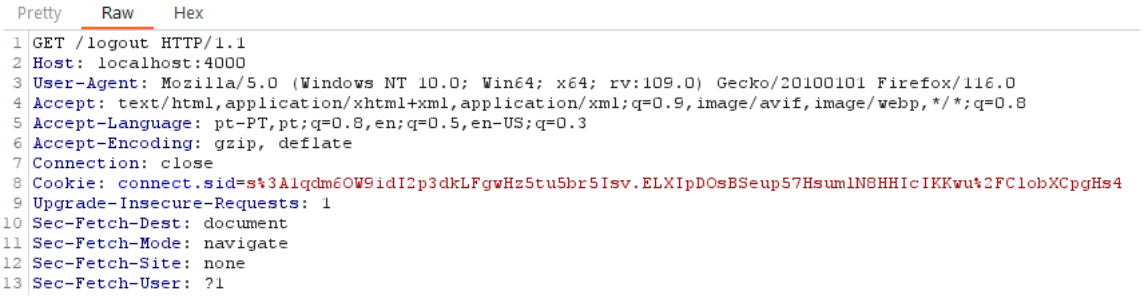


Figure 23. Arbitrary GET request to logout of the platform.

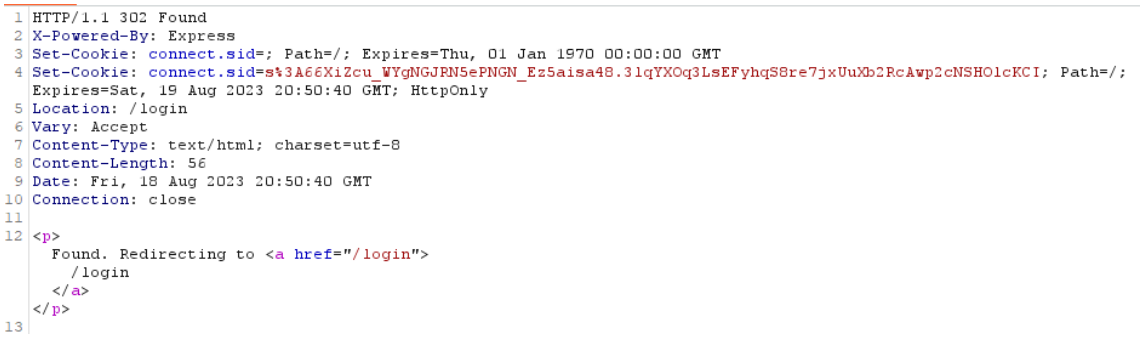


Figure 24. Server response to a valid logout request.

Additionally, as it is possible to verify in the response of the server, although the valid session cookie is deleted from the future requests, as mentioned before, the server automatically assigns a new session cookie to the client that is not associated with any user.

5.5.2.3. File Upload Vulnerabilities

File upload vulnerabilities allow an attacker to upload and execute files in web servers [110]. It does happen because applications do not sufficiently validate things like the filename, type, contents, or even size [111]. According to [111], *"(...) failing to properly enforce restrictions on these could mean that even a basic image upload function can be used to upload arbitrary and potentially dangerous files instead (...)"* [111]. This includes server-side script files that might enable remote code execution (RCE) and totally compromise the server [111]. However, uploading a file for itself is not enough to cause damage and as so, it typically involves a follow-up HTTP request for the respective file, to trigger its execution by the server. Nonetheless, if a file upload functionality validates robustly a file, it makes it harder to trigger a vulnerability even if the file can be requested later.

According to [111], file upload vulnerabilities arise because developers fail to implement robust validation. It is uncommon for an arbitrary website to not have any file upload protection, but this protection might flaw or be easily bypassed. Specifically, these vulnerabilities can happen

because of insufficient validation that might allow attackers to upload harmful files, incorrect file type checks that might allow attackers to trick the server into accepting malicious files that compromise the application itself, lack of file size limits that can lead to denial-of-service attacks due to overuse of server resources [110], [111].

The most common types of flawed validation are: flawed file type validation, Insufficient blacklisting of dangerous file types, flawed validation of the file's contents [111].

The flawed file type validation happens when a browser typically submits an HTML form. The data provided is sent in a *POST* request with a specific *Content-Type*, such as *multipart/form-data*. This *Content-Type* value informs the server about the MIME type of the data that was submitted using the previously mentioned input. If an arbitrary application is supposed to expect only images, it may only allow types such as *image/jpeg* and *image/png*. The following figure, Figure 25, represents an example of a *POST* request that contains an image *example.jpg* that was uploaded through a form, as stated in the example above.

```
POST /images HTTP/1.1
Host: normal-website.com
Content-Length: 12345
Content-Type: multipart/form-data; boundary=---------------------------01234

---------------------------012345678901234567890123456
Content-Disposition: form-data; name="image"; filename="example.jpg"
Content-Type: image/jpeg

[...binary content of example.jpg...]
```

Figure 25. POST request that contains an image uploaded through a form.

However, if the value of this header is implicitly trusted by the server, and it does not perform any further validation to check if the contents of the file match the supposed MIME type, this defense can be easily bypassed by manipulating the value of the header *Content-Type*. Consequently, an attacker might be able to upload a web shell and run commands on the server.

Regarding insufficient blacklisting of dangerous file types, it is straightforward. A web application built on top of the *.php* programming language that has the functionality of image upload should block the upload of files with extensions such as *.php* or *.php5*. Also, relying solely on blacklisting is intrinsically flawed [111] as it is difficult to completely block all extensions that can be used to execute code. Thus, blacklists can be bypassed sometimes by using lesser-known file extensions that may still execute code such as *.shtml* [111]. To protect against these types of vulnerabilities, the application should verify the file extension against a whitelist of permitted/allowed extensions rather than a blacklist of prohibited ones, since it is much easier to guess the extensions the application needs to allow to guarantee functionality instead of the ones an attacker might try to upload [111].

Finally, as a second-layer protection, file's contents should be verified, instead of just trusting the header *Content-Type* and its validation. There are several ways to verify that a specific file is an image for instance, such as verifying specific properties like its dimensions [111]. Additionally, certain file types contain a specific sequence of bytes in their header or footer, also known as magic bytes. According to [112], magic bytes *"(...) are specific to binary files and rely on matching signatures in their tails and in other places (...)"*. There are several hundred file types for which magic bytes are defined. However, magic bytes do not always explicitly identify a file type. For instance, the magic bytes *4D 5A* are the magic bytes used for executables and are commonly used by *.exe, .dll, .sys, (..)*. As so, it should be implemented as a second layer of protection, as mentioned before, along with a content validation based on the *Content-Type*.

About the development carried out in the current project, two clear functionalities required file uploads, the edition of the health professional' profile where the health professional can upload a profile picture, and the test creation where the health professional can upload a multimedia file to the respective test, such as an image or a sound. Thus, it was needed to implement a robust file validation for the mentioned functionalities. To support this implementation, a *Node.js* middleware named *multer* [113] was used. Although it was not guaranteed that these files could be requested, it is a good practice to validate them anyway, since it can prevent potential harm to the application.

Regarding the functionality that allows a health professional to upload a profile picture, the application must only accept the upload of image files. To limit the spectrum of files that can be uploaded, and to create a more robust, concise, and secure mechanism, it was defined that only *.jpg*, *.jpeg* and *.png* files could be uploaded. Initially, it was required to specify a set of rules and configurations for *multer* to work as intended for the implementation wanted. The following snippet of code represents a part of *multer-config.js* file, where the configuration for handling file uploads was coded, in the present app.

```
const multer = require('multer');

const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    cb(null, 'uploads/');
  },
  filename: (req, file, cb) => {
    const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() * 1E9);
    const extension = path.extname(file.originalname);
    cb(null, file.fieldname + '-' + uniqueSuffix + extension);
  }
});
```

Initially, it is specified the type of storage the *multer* instance wants to use. In this case, as the files are stored in the server, the suitable solution is *diskStorage*. Afterwards, the naming convention for the files is specified, where the current time is used as well as a random value that makes harder for files to be found. Even if an attacker bypasses the security measures implemented and can upload a malicious image that contains shell code for example, to request the file in the server, it will take a lot of time to try to brute-force the name.

For the image files validation, the following snippet of code illustrates the validation performed to the files uploaded that are supposed to be images.

```
const imageFilter = (req, file, cb) => {

  // Verify if there is a file uploaded
  if (!file) {
    cb(null, false);
    return false;
  }

  // Verify file size through Content-Length header
  const maxSize = 10 * 1024 * 1024;

  // Check if the 'content-length' header is present
  if (!req.headers['content-length']) {
    cb(null, false);
    return false;
  }

  const contentLength = parseInt(req.headers['content-length'], 10);
  console.log('Content-Length:',contentLength);

  // Verify if the content length exceeds the maximum size
  if (contentLength > maxSize) {
    cb(null, false);
    return false;
  }

  // Verify MIME type
  console.log('Detected MIME Type:', file.mimetype);
  if (!['image/jpeg', 'image/png'].includes(file.mimetype)) {
    console.log('Invalid MIME TYPE.')
    cb(null, false);
    return false;
  }

  // Verify filename and extension
  const validExtensions = ['.jpg', '.jpeg', '.png']; //Whitelist
  const extname = path.extname(file.originalname).toLowerCase();
```

```
    if (!validExtensions.includes(extname)) {
        console.log('Invalid extension and filename.')
        cb(null, false);
        return false;
    }
    cb(null, true);
    return true;
}
```

Initially, the function checks whether a file has been uploaded effectively. This simple verification is performed because it is relatively easy for an attacker to bypass the client-side validation of an upload with the property *required* with the usage of software with a built-in proxy like *Burp Suite*. Following, the size of the file is validated through the header *Content-Length* with its value being obtained through *req.headers['content-length']*. If its value is bigger than 10 MB, the upload process is canceled since 10 MB was the size limit defined for these types of files. It is important mentioning that this verification is not enough to fully validate the file size. Thus, the size is validated further in the second layer of validations, mentioned below. Then, the *MIME Type* of the file passed in the request is verified with *file.mimetype*, since the application does not trust the headers value sent in the request. As the file types allowed are *.jpg*, *.jpeg* and *.png*, as mentioned before, it means the app simply needs to verify two *MIME Types* – *image/jpeg* and *image/png*. If the *MIME Type* detected is different from both types, the upload job is immediately dropped, as observed in the code snippet. The next step is the verification of the extension. For this, a whitelist of allowed extensions – *jpg*, *jpeg* and *png* - was used, as it is a much better practice than blacklisting, as stated before. The name is normalized and then it is verified that if it contains one of the allowed extensions. If not, the upload process is also dropped.

Moreover, as mentioned before, the file's contents should be verified, and *magic bytes* are a valid way to achieve it. The *magic bytes* validation implemented in the project was structured as a second layer of security and that is why it is not implemented in the *multer-config.js* file but in an *uploadController.js* file that is called after the first layer of validations, explained above, is performed. The following snippet of code represents the logic behind the validation of the *magic bytes*, to verify the integrity of the image files uploaded.

```
const path = require('path');
const fs = require('fs');

// Formats allowed and magic bytes
const magicBytes = {
  'jpg': 'ffd8ff',
  'jpeg': 'ffd8ff',
  'png': '89504e'
```

```
};

const verifyMagicBytes = (fileExtension, filePath) => {
  // Read the content of the uploaded file
  const fileContent = fs.readFileSync(filePath);

  // Verify magic bytes
  // Get the first 3 bytes
  const fileMagicBytes = fileContent.toString('hex', 0, 3);

  // Remove the dot from the extension
  const expectedMagicBytes = magicBytes[fileExtension.substr(1)];

  console.log('FileMagicBytes:', fileMagicBytes);
  console.log('ExpectedMagicBytes:', expectedMagicBytes);

  return fileMagicBytes === expectedMagicBytes;
};

const uploadController = (req, res) => {

  // Access the uploaded file from req.file
  const file = req.file;

  // Verify that if there is a file (this is a second-layer protection)
  if (!file) {
    return res.redirect('/testupload?error=somethingwentwrong')
  }

  const fileExtension = path.extname(file.originalname);
  const filePath = file.path;

// Verify file size
  const maxSize = 10 * 1024 * 1024; // 10 MB
  fs.stat(filePath, (err, stats) => {
    if (err) {
      console.log('Something wrong with the file read.');
      return res.redirect('/testupload?error=somethingwentwrong');
    }

    const fileSize = stats.size;
    console.log('FileSize:', fileSize);
    if (fileSize > maxSize) {
      fs.unlinkSync(filePath);
      return res.redirect('/testupload?error=somethingwentwrong');
    }
  });

  // Verify magic bytes
```

```
const isValidFile = verifyMagicBytes(fileExtension, filePath);

if (!isValidFile) {
  // Delete the invalid file (this is second-layer of protection)
  fs.unlinkSync(filePath);
 return res.redirect('/testupload?error=somethingwentwrong')  }

return res.redirect('/testupload?error=none')
};
```

For the magic bytes' verification, the function *verifyMagicBytes()* was developed. Initially a specie of whitelist of allowed magic bytes was defined that correspond to the magic bytes for each image file format allowed – *ffd8ff* and *89504e*. Then, the content of the file is read with the support of the function *fs.readFileSync()*. At this point, the image file passed the first layer of validations, as mentioned before, and it is uploaded to the server, but in a temporary way. Only after this second round of validation the file will be stored effectively in the system. To avoid any potential issues, the system validates one more time the file size of the uploaded image. This validation consists of reading the file and obtaining its size. If the size is bigger than 10 megabytes, the file is immediately deleted from the system. It is important to mention that is only an additional measure since the size of the file was already verified once, in the first layer of validations.

Moreover, as the magic bytes are a kind of identity header for files, the first 3 bytes of the file's content are selected and assigned to a variable *fileMagicBytes*. Then, this variable is compared with the whitelist of magic bytes. If any of the entries match, the file is considered valid and keeps being stored in the system. Otherwise, the file is immediately deleted from the server with the function *fs.unlinkSync()*.

5.5.2.4. NoSQL Injection

NoSQL Injection is a type of security vulnerability that occurs in *NoSQL* databases, similar to how *SQL injection* affects relational databases [88]. In this type of attack, an attacker exploits a vulnerability in an application by manipulating the data or queries sent to a NoSQL database. The manipulation can be on letters, numbers, or sentences which will be all sent to query statements. If developers fail to examine the values or variables, a very dangerous situation will result. According to [88], the risk is that malicious code will be launched while the query function is active, having a very detrimental impact on database security. The malicious code will mess with condition statements, for example, by utilizing notations and segmentations in the NoSQL statements to make the result always true [88].

To prevent *NoSQL Injections*, it is important to perform User input validation in case it is required to use user input in the query statements and parameterization [114]. Throughout the

application, all the potential input vectors are validated and sanitized, according to their use-case, as mentioned before. Consequently, it significantly decreases the chance for any injection, including *NoSQL*.

Additionally, regarding the current application, parameterization is also used, where parameterized queries (prepared statements) are used. In all cases a query is required, the parameters are used within the query as an object property. This approach helps preventing direct injection of malicious input. Moreover, throughout the whole application, was used several times the functions *findOne()* and *find()* to query in different contexts. These methods are not inherently a parameterized query however they provide some level of protection because they expect an object that represents the query conditions.

5.5.2.5. Encryption of database records

Encryption is a critical security practice frequently implemented on the server-side to protect sensitive data. This data can be stored in an external server or database. Thus, according to [115], *"(...) database encryption refers to the use of encryption algorithms to transform plain text database into a (partially) encrypted database making it unreadable to anyone (...)"* [115]. These algorithms can be categorized into Symmetric and Asymmetric encryption. Symmetric encryption involves the usage of the same key for encryption and decryption. Asymmetric encryption involves the usage of a key for encryption and a different key (private) for decryption. There are several algorithms for both of these methods of encryption [116], [117].

Regarding symmetric encryption, multiple studies [116]–[119] mention *AES (Advanced Encryption Standard)* [118] as a robust algorithm since it provides *"(...) better data security, efficiency and high speed performance"* [120]. Several studies that contained performance analysis to different symmetric encryption algorithms [116], [117], [119] revealed that *AES* is known for having good performances when it comes to resource usage, time and strength of the algorithm. According to [116], [117], *AES "(...) has an advantage over others (...) in terms of time consumption and throughput"* [116] and *"(...) has a better performance than 3DES and DES (...)"* [117], other well-known encryption algorithms.

Advanced Encryption Standard (AES) [118] specifies a cryptographic algorithm that is used to protect electronic data, as mentioned before. It is capable of using cryptographic keys of 128, 192 and 256 bits to encrypt and decrypt data in blocks of 128 bits into its original form. As so, it is considered a block cipher [118]. *AES* performs operations on bytes rather than in bits. Since each block size is 128 bits, as mentioned before, the cipher processes 128 bits (or 16 bytes) of the input data at each point of time.

From a functional point of view, *AES* uses a secret encryption key to perform both encryption and decryption (since it is a symmetric encryption algorithm). Logically, the longer

the key, the stronger the encryption. However, more computational power is required. Thus, *AES* works in rounds, each of which consists of a sequence of operations that change both the data and the encryption key. For the different key sizes, a different number of rounds is required: 10 rounds for *AES-128*, 12 rounds for *AES-192* and 14 rounds for *AES-256*, depending on the length of the key. These iterations guarantee that the encryption process is sufficiently complex and secure.

Regarding the development carried out in the project, *AES-256* was the chosen cryptographic algorithm for encryption – the strongest *AES* version. To implement it, the library *crypto* [121] in *Node.js* was used. In a primordial phase of the project, only the passwords present in the database were encrypted. This ensures that even if an attacker gains access to the database, the stored passwords are unreadable without the encryption/decryption key. Additionally, this key was stored using secure key management practices – stored in a secure environment to prevent unauthorized access. To successfully implement the encryption, it was required to refine the authentication mechanism. As the database stores the encrypted passwords, when a user tries to login in the app, the platform needs to encrypt the password inputted by the user with the same configurations as before, and then compare both passwords to check if they match. Thus, it is possible to say that the encryption also helped to strengthen the authentication mechanism, since an additional verification is made to the password.

Nonetheless, as an extra layer of security and randomness to the encryption process, *Initialization Vectors (IVs)* [122] were used. They are components used in some encryption modes, including *Cipher Block Chaining (CBC)* [123] – used in the current project. *IV* is a random value that is generated for each encryption operation. It serves as an additional input to the encryption algorithm along with the encryption key [123]. This ensures that the same plaintext encrypted multiple times will result in different ciphertexts due to this randomized input, that ensures that the first encrypted block of data is random [124]. Also, it prevents attackers from discovering patterns or repetitions in the encrypted data, even if they know the corresponding plaintext and avoiding attacks like frequency analysis [125]. Yet, according to [122], [123], *IV* is not a secret and it is not often transmitted or stored alongside the ciphertext they correspond to. *IV* for each health professional is stored in its document in the database.

As the algorithm used was *AES-256*, the cryptographic key required needs to be 256 bits long, or 32 bytes. The key used in the current project was randomly generated using ***crypto***, with the function *randomBytes()*. The same happens to the *Initialization Vector* created in each operation (in this case, each registration of a health professional). The following figure, Figure 26, shows how the password and *IV* are stored in the health_professionals documents in the database.

Figure 26. Arbitrary document of a health professional in the database.

The following snippet of code shows how the encryption was implemented using ***crypto***, as mentioned before. In this case, the text intended to be encrypted corresponds to the passwords of the new users of the platform, when they register.

```
require('dotenv').config();
const crypto = require('crypto');
...
// Generate a random Initialization Vector (IV)
 const iv = crypto.randomBytes(16);

 // Get the cryptographic key
 const key = process.env.ENCRYPTION_KEY;

 // Create a cipher using AES-256-CBC algorithm, the encryption key and the IV
 const cipher = crypto.createCipheriv('aes-256-cbc', Buffer.from(key), iv);

 // Encrypt the password
 let encryptedPassword = cipher.update(password);
 encryptedPassword = Buffer.concat([encryptedPassword, cipher.final()]);
```

Initially, a random *IV* is created. Then, the cryptographic key – secret is obtained from an environment variable. To be possible to get the variable through *process.env*, *dotenv* library was used. Then, a cipher is created with the chosen algorithm – *AES-256-CBC*, the key and the *IV*. Finally, the method *cipher.update()* is used to partially encrypt the data, and the method *cipher.final()* concludes this process by providing the final encrypted portion. Then, this final piece of encrypted data is concatenated with the previously encrypted partial data stored in the variable *encryptedPassword*.

Regarding the login process, to verify the identity of a health professional, it is needed to verify both e-mail and password, as mentioned before. As the passwords are encrypted in the database, to compare the inputted one with the stored one, the user-supplied password needed to be encrypted using the same *IV* and key and then both can be compared. The following snippet

of code represents the implementation of this process, that followed the same logic of the aforementioned encryption - the code implemented for the registration part, represented above.

```
require('dotenv').config();
const crypto = require('crypto');
…
function encrypt(text, iv) {
    const key = process.env.ENCRYPTION_KEY;
    console.log('Key:', key.toString('hex'));

    // Create a cipher using AES-256-CBC algorithm, the encryption key and IV
    const cipher = crypto.createCipheriv('aes-256-cbc', Buffer.from(key), iv);

    // Encrypting the password
    let encryptedText = cipher.update(text);
    encryptedText = Buffer.concat([encryptedText, cipher.final()]);
    console.log('Inside function encryptedText:', encryptedText.toString());
    return encryptedText;
}

const result = await hp.findOne({email: email});
console.log('Enckey:', encryptionKey);
console.log('Result IV:', result.iv);
const iv = Buffer.from(result.iv, 'hex');
var encryptedPassword = encrypt(password, iv);
console.log('Encpassword:', encryptedPassword);

const result2 = await hp.findOne({email: email, password: encryptedPassword});
```

Regarding the code, a function *encrypt()* was defined that encrypts text with a determined *IV*, applying the same logic as the one mentioned before. This function is used to encrypt the password inputted by the user with the *IV* associated with the health professional that has the e-mail address provided. Then, with the encrypted password, a new query is performed to the database to verify if there is a user with both e-mail and the encrypted password stored in the system.

5.5.2.6. Log system

According to [126], *"(…) a log is a record of the events occurring within an organization's systems and networks (…)"* [126]. Logs are used for troubleshooting problems, but they have more functions within enterprises and applications such as the monitoring of performance and collecting of useful data for investigations of malicious activity. As mentioned before, applications should have logging systems that collect information about the user actions and errors that might occur in the arbitrary application. Logs should not have vague information and should be inaccessible for a normal user.

Regarding the development carried out in the current project, a log system was implemented that registers information about the user's interaction with the platform developed. To implement this, two popular *Node.js* libraries were used – *winston* [127] and *morgan* [128]. Regarding *Winston*, it was used to create a logger middleware that outputs the activity performed in the application to an external file (e.g., *logfile.log*). However, this logging mechanism was developed to trace the activity of the platform from an application layer. This means that only pages that are accessed by the user are logged and not the underlying requests performed by the browser (e.g., static files as images). In this log, when a user that is logged in the application performs an action, his *id* is registered as well as the request performed by him. Additionally, the timestamp of the action is also stored. The following snippet of code shows the implementation of the logger mechanism using *winston*.

```
const winston = require('winston');

const logger = winston.createLogger({
   level: 'info',
   format: winston.format.combine(
      winston.format.timestamp(),
      winston.format.json()
   ),
   transports: [
      new winston.transports.Console({level: 'silly'}),
      new winston.transports.File({ filename: 'logfile.log'}),
   ],
});

module.exports = logger;
```

Regarding the code, an object *logger* is created that will be used to register the logging information. Several properties were specified for this object such as the level that refers to the type of information being stored, the format of log information, where was specified that we want to store the timestamp and the rest of the information in *JSON* format. Lastly the *transports* property consists of the outputs of the logs, where they will be outputted to, in this case, to the console of our server and to the file *logfile.log*. Then, to log a specific action, the *logger* object needs to be called as follows.

```
const logger = require('./logger');
…
router.get('/myassessments', (req, res) => {
    if (!req.isAuthenticated()){
        logger.info('GET /myassessments (failed) - user not authenticated')
        const newUrl = '/login';
        res.redirect(401, newUrl);
    }
    else {
        logger.info('GET /myassessments (successful) - user '+req.user);
        res.render('myassessments');
    }
})
```

In the example given above, if a user accesses the */myassessments* endpoint, a successful message log is stored along with its *id*. If not, a failure message is stored in the logfile. This way, it is possible to track what each user did/is doing. The following figure, Figure 27, shows an example of logs registered by *Psyment* platform regarding the interaction between the application and an arbitrary user.

Figure 27. Example of logs registered during an arbitrary user interaction with Psyment.

In Figure 27, it is possible to observe the entire workflow of an arbitrary user. This user started by logging in the application successfully. He has the email *teste@gmail.com* and in the database, his *id* is *650acbb0fd9cc164fea96dbc*. Then, he accessed */menu*, */profile* and then */logout* respectively. The information provided in the logs is not vague and can clearly identify the user as well as his actions in the platform.

Regarding the development using *morgan*, it was used to create another logging file that works slightly different than the one mentioned above. In this case, the implementation is more focused on tracing the activity of the platform from a communication layer. Web applications use *HTTP* protocol to communicate. As so, *HTTP* requests performed are logged to the application as well as some useful information contained in these requests (headers). This information consists of the timestamp of the request, the *IP* address of the machine that performed the request, the resource/page requested, the status code of the response, the response time by the application

and the user-agent. The following snippet of code shows the implementation of the second logging system using *morgan*.

```
const morgan = require('morgan');
const fs = require('fs');
…
const requestFileLogPath = path.join(__dirname, 'requestlog.log');
const logStream = fs.createWriteStream(requestFileLogPath, { flags: 'a' });
const customMorganFormat = '[:date[clf]] :remote-addr :method :url HTTP/:http-version :status :res[content-length] - :response-time ms :user-agent';
app.use(morgan(customMorganFormat, {stream: logStream}) );
```

Regarding the code, initially it is specified the path where we wanted to store the log file, with the constant *requestFileLogPath*. Then, it was created a writable stream for the log file stored in the constant *logStream* with the support of the built-in *Node.js module* ***fs***. Then, a custom format for the logs was defined stored in *customMorganFormat*, where it is defined the information that we want to store, mentioned above. Finally, we set the *morgan* middleware to use the custom format and to store the logs in the mentioned stream.

The following figure, Figure 28, shows the output of this logging system regarding the activity of an arbitrary user throughout the application.

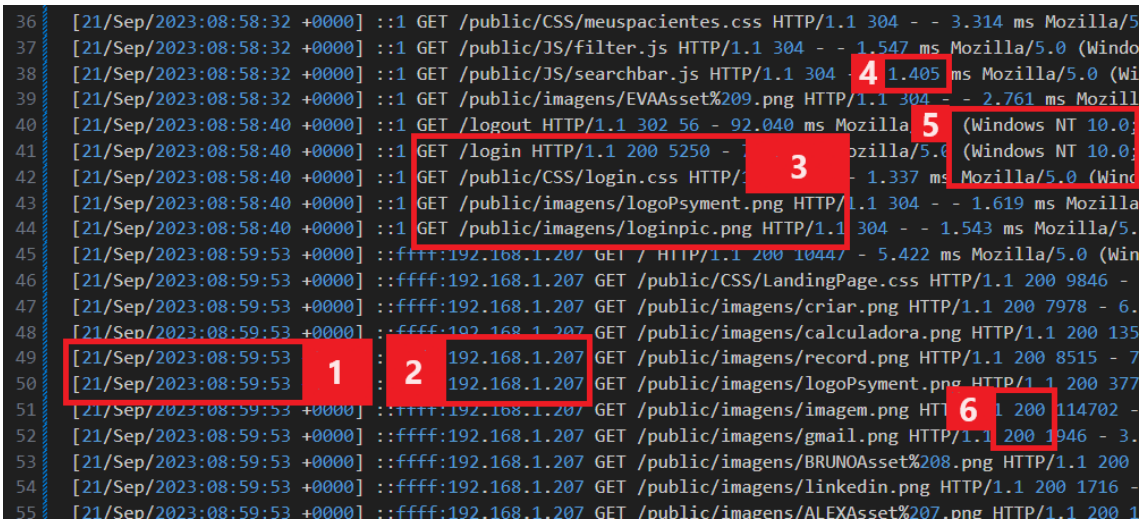


Figure 28. Logging output of the logging system implemented with morgan - (1) represents the timestamp of the request, (2) represents the origin IP address that performed the request, (3) represents the resource/page requested, (4) represents the response time by the application, (5) represents the User-agent and (6) represents the status code of the response.

Testing

6. Testing

6.1. Vulnerability Automated Scanning

In this section, an automated vulnerability scan was performed using *Burp Suite Professional* web scanner v2023.10.1.2. The work performed in [129] evaluated four well-known web scanners and their capabilities to find vulnerabilities in *Nodejs*-based applications. *Burp Suite Professional* had the best True Positive and Recall values [129]. It revealed to be a robust solution for the intended goal. Consequently, *Burp Suite Professional* was used to scan *Psyment* application as well. Moreover, the scanner used can verify over 160 types of vulnerabilities including all the most common vulnerabilities found in web apps, already mentioned before.

Regarding the scan performed using *Burp Suite Professional*, we ran a scan where the scanner did not have any credentials of the system (simulating a black-box assessment) The scan consequently included the auditing of the robustness of the authentication mechanisms but also gave a more in-depth view of the security posture of the whole web application.

For scanning, a custom configuration was defined to optimize the scanning process in terms of time. Thus, some vulnerabilities were removed from the scan scope since it did not make sense to audit them according to the technologies used (e.g., did not verify the "JWT signature not verified" issue as JWT tokens were not used in the implementation).

6.1.1. Scan system hardware specifications

The system used to perform the vulnerability scan has the following specifications:

- Processor: *12th Gen Intel® Core™ i7-12850HX*
- RAM: 32 GB
- Storage: 500 GB
- Graphics Card: Nvidia RTX A2000 8GB

6.1.2. Scan details

	Scan
Time taken	28 min and 25 sec
Requests performed	24875
Unique locations found	40
Issues found	19
Network errors	162

Table 6. Details of the vulnerability automated scanning performed.

6.1.3. Issues

As mentioned before, 19 issues were found in the scan (see Figure 29). These issues can be classified by Severity: Low, Medium, High and Information. Moreover, as it is an automated scan, sometimes the vulnerabilities cannot be completely exploited. For this reason, the scanner also provides a classification of each issue by Confidence that can be: Tentative, Certain and Firm.

Issue type	Host	Path	Insertion point	Severity	Confidence
Cross-domain Referer leakage	http://192.168.1.200:4...	/editprofile		Information	Certain
Cross-domain Referer leakage	http://192.168.1.200:4...	/approvedtests		Information	Certain
Cross-domain Referer leakage	http://192.168.1.200:4...	/newpatients		Information	Certain
Cross-domain Referer leakage	http://192.168.1.200:4...	/myassessments		Information	Certain
Cross-domain script include	http://192.168.1.200:4...	/		Information	Certain
Cross-domain script include	http://192.168.1.200:4...	/		Information	Certain
Cross-domain script include	http://192.168.1.200:4...	/		Information	Certain
Email addresses disclosed	http://192.168.1.200:4...	/editprofile		Information	Certain
Email addresses disclosed	http://192.168.1.200:4...	/profile		Information	Certain
Email addresses disclosed	http://192.168.1.200:4...	/		Information	Certain
File upload functionality	http://192.168.1.200:4...	/editprofile		Information	Certain
Input returned in response (reflected)	http://192.168.1.200:4...	/myassessments	search parameter	Information	Certain
JavaScript injection (DOM-based)	http://192.168.1.200:4...	/createtest		Medium	Tentative
Open redirection (DOM-based)	http://192.168.1.200:4...	/createtest		Low	Tentative
Open redirection (DOM-based)	http://192.168.1.200:4...	/createtest		Low	Tentative
Open redirection (DOM-based)	http://192.168.1.200:4...	/createtest		Low	Tentative
Open redirection (DOM-based)	http://192.168.1.200:4...	/createtest		Low	Tentative
Open redirection (DOM-based)	http://192.168.1.200:4...	/createtest		Low	Tentative
Vulnerable JavaScript dependency	http://192.168.1.200:4...	/		Low	Tentative

Figure 29. Issues found by the Burp Suite Pro scanner performed.

12 of the issues found of them were classified as Information, 6 of them as Low and only 1 as Medium. Fortunately, there were no High severity issues found within *Psyment's* platform in this scan. Moreover, all the worst findings (not informational) were classified as Tentatives (rating of confidence), which is a **positive** indicator because it means that none of them were successfully exploited and there is a chance that they are false positives.

The issues found in the scan were the following:

- *Cross-domain Referer leakage*

Endpoints: */myassessments*, */newpatients*, */approvedtests*, */editprofile*

Issue: When a browser requests a resource, it normally adds an HTTP header called "Referer", indicating the URL from the resource that originated the request (web page). This header is included in cross-domain requests and might lead to sensitive information if passed in the URL. In the case of *Psyment* application, cross-domain requests are performed to load external well-known scripts used for the implementation of the platform (UI) (e.g., *bootstrap – https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css*). Nonetheless,

the endpoints performing these requests do not contain any sensitive information in the URL, and as so, do not leak any kind of data. Follows part of the list of external domains fetched (Fig. 30).

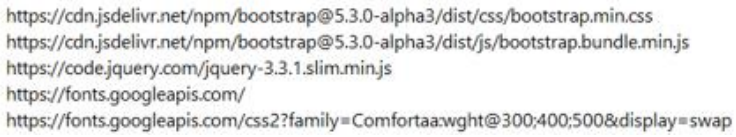


Figure 30. Part of the list of external domains fetched by Psyment app.

Impact: The issue does not expose a danger vulnerability since it cannot be exploited but yet a risk that the application faces.

- *Cross-domain script include*

Endpoints: */* (multiple locations)

Issue: When an application includes a script from an external domain, the script is executed by the browser of the user accessing the web app. These fetched scripts can do anything that the app's own scripts can do. Logically, when calling these scripts, the app is inherently trusting external domains to prevent an attacker from modifying the script to perform malicious activities (e.g., XSS). In the case of *Psyment* application, similarly to the *Cross-domain Referer leakage* finding, the external scripts fetched are well-known and maintained by a big community of developers (e.g., *https://code.jquery.com/jquery-3.3.1.slim.min.js*). However, it is still a risk to trust external sources.

Impact: The aforementioned vulnerability does not represent an active threat to the application. *Psyment* application can be threatened only if any of the external domain owners happens to be hacked. However, the hacking scope is reduced to client-side attacks (browser).

- *Email addresses disclosed*

Endpoints: */ , /profile, /editprofile*

Issue: The scanner detected the presence of email addresses within application responses. However, this does not necessarily mean a real security vulnerability. After analysing the endpoints and the requests/responses that triggered this issue, it was concluded that this is a false positive. The endpoints that contain e-mails addresses in the response are supposed to have them (e.g., user profile). Moreover, the endpoint " / " contains the email address of *Psyment* team (not sensitive). As so, this does not constitute a real issue of the application.

Impact: None.

- *File upload functionality*

Endpoints: */editprofile*

Issue: The scanner detected a page containing a form which is used to submit user-supplied files. The scanner itself recommends to manually review the file upload functionality (Figure 31). Nonetheless, this is not a real issue.

Impact: None.

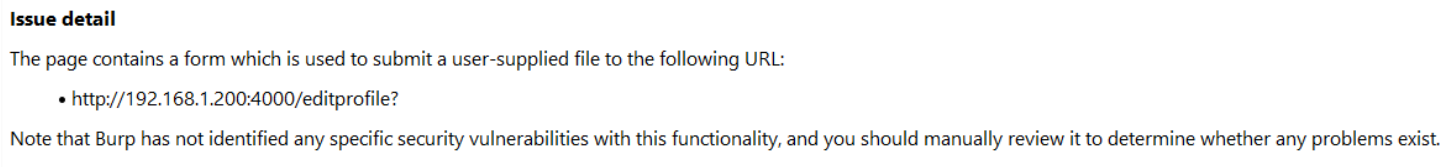


Figure 31. Scanner recommendation to manually review the file upload.

- *Input returned in response (reflected)*

Endpoints: */myassessments*

Issue: As mentioned before in the present document, reflection of input arises when data is copied from a request and placed in the response. In this specific case, the value of the parameter ***search*** is present into the application's response (Figure 32). However, *Psyment* application validates and sanitizes the input, encoding it on the output. Moreover, certain keywords typically included in payloads are blocked (Figure 33).

Figure 32. Parameter search reflected into the application's response.

Figure 33. Parameter search totally not reflected in the page (blocked due to keywords).

Impact: None since the output is rigorously encoded or blocked.

- *JavaScript injection (DOM-based) – DOM Cross-Site Scripting*

Endpoints: */createtest*

Issue: As mentioned before in the present document, *DOM-based* vulnerabilities arise when a client-side script reads data from the DOM. The scanner detected in the page */createtest* a potential vulnerability that could not be automatically exploited, where data is read from *input.value* and passed to an *eval* function (dangerous sink). This *input.value* is from the source element *test-formula* (test calculation formula). After digging into the application, and analysing the source code (Figure 34), we saw that this field is not correctly sanitized, probably because a specific formatted input is expected that contains "()" (parenthesis) characters with the test formula. The *test-formula* parameter was tested manually and successfully exploited. As PoC, we ran some JavaScript payloads through the vulnerability: the *alert()* function that is a common payload to show *XSS* (Figure 35) and a request to an external controllable server was made, to show how an attacker with this vulnerability could induce the victim to perform requests to an external domain (Figure 36).

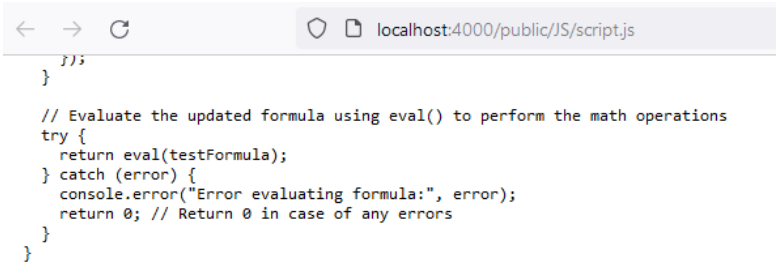


Figure 34. Part of the /createtest page that passes an user-input to an eval() function.

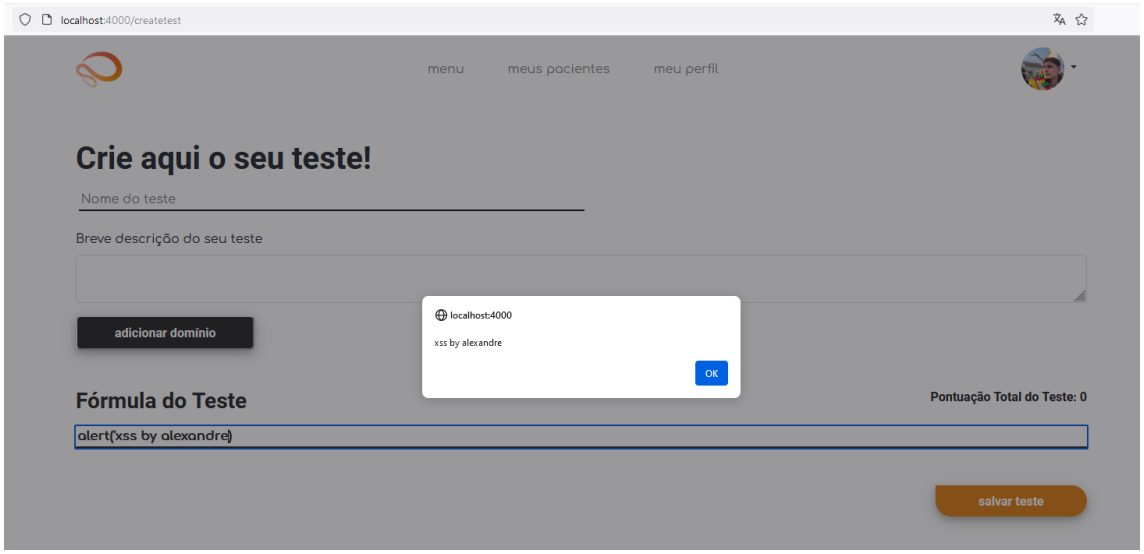


Figure 35. PoC of XSS with an alert() called in the /createtest page.

To make a request to the simulated external server controlled by the attacker, and to bypass possible filters or bad parsing of characters, the function *atob()* was used, which decodes *base64* strings. We created a base64 XSS payload that sends a request to the aforementioned server.

Javascript code used: *fetch('http://w85xwzhi8g86hex5de3vj4eh389zxple.oastify.com', {method: 'POST', mode: 'no-cors', body:document.domain})*

Code base64 encoded: *ZmV0Y2goJ2h0dHA6Ly93ODV4d3poaThnODZoZXg1ZGUzdmo0ZWgzODl6eHBsZS5vYXN0aWZ5LmNvbScsIHttZXRob2Q6ICdQT1NUJywgbW9kZTogJ25vLWNvcnMnLCBib2R5OmRvY3VtZW50LmRvbWFpbn0p*

Final payload: *eval(atob('ZmV0Y2goJ2h0dHA6Ly93ODV4d3poaThnODZoZXg1ZGUzdmo0ZWgzODl6eHBsZS5vYXN0aWZ5LmNvbScsIHttZXRob2Q6ICdQT1NUJywgbW9kZTogJ25vLWNvcnMnLCBib2R5OmRvY3VtZW50LmRvbWFpbn0p'))*

And when we pass this payload to the *eval()* DOM sink:

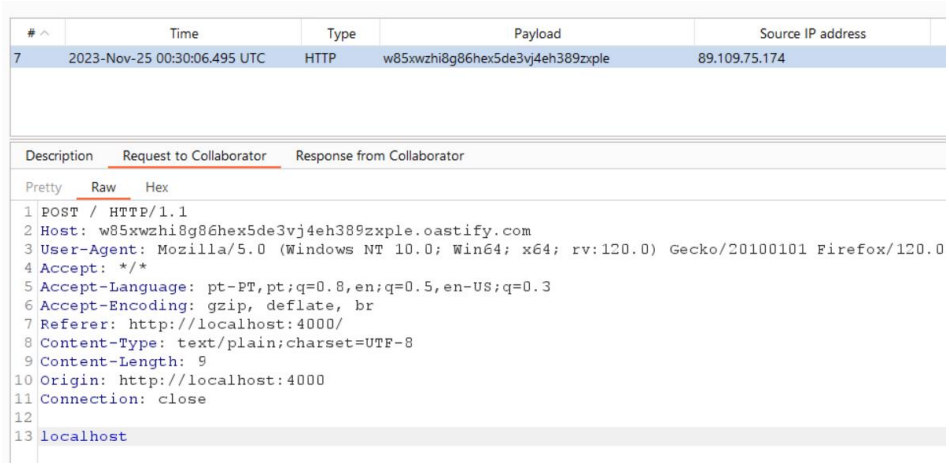


Figure 36. HTTP request received on our external controlled server by Psyment (see Referer)

And as we see, we got the request on our controlled-server with *document.location* on the body.

Impact: As mentioned before, this vulnerability allows the attacker to run *JavaScript* code in the victim's browser. Moreover, this type of vulnerability, depending on the target, can be chained with other vulnerabilities and put in danger a web application. In this case, as the vulnerability is not stored and the cookies are securely configured, the impact is significantly decreased.

- *Open redirection (DOM based)*

Endpoints: */createtest*

Issue: As mentioned before in the present document, DOM-based vulnerabilities arise when a client-side script reads data from the DOM. The scanner detected in the page /createtest a potential vulnerability that could not be exploited, where data such as *domainFormula, domainName, test-formula, description and testname* is passed to the *fetch()* API. However, after digging into these parameters, it was concluded that these parameters are all merged into an object, then they are converted to JSON format and finally passed into the *fetch* API. As so, it was not possible to exploit this Open redirection. Following, this does not constitute a real issue in the application.

Impact: None according to how the functionality is implemented.

- *Vulnerable JavaScript dependency*

Endpoints: /

Issue: The scanner detected that the application uses an outdated version of *jQuery* - 3.3.1. After manually reviewing this fact, it was confirmed (Figure 37). This version of the software has known security vulnerabilities (CVE-2019-11358, CVE-2020-11022, CVE-2020-11023 [130]). These vulnerabilities may be abused by an attacker to perform a wide range of attacks.

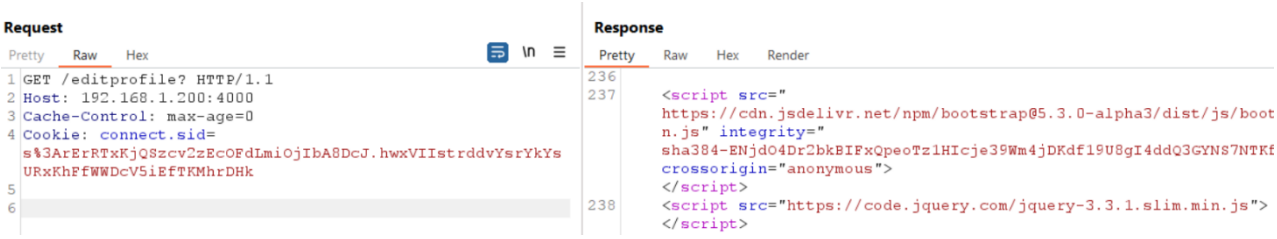


Figure 37. Outdated jQuery version used by Psyment web application.

6.1.4. Considerations

The web vulnerability scan conducted using *Burp Suite* has provided valuable insights into the security posture of our web application. The identified issues, including *Cross-domain Referer leakage*, *Cross-domain script inclusion, and DOM-based Cross-Site Scripting (XSS)*, underscore the critical need for a proactive approach to cybersecurity by developers.

The discovery of vulnerabilities such as *Cross-domain Referer leakage*, *Cross-domain script inclusion*, although not so dangerous in this context, highlights potential vectors for attackers to exploit the application's security. These findings emphasize the importance of implementing secure coding practices, including proper handling of cross-domain requests and validation of external script sources, to mitigate the risk of data exposure and injection attacks.

Furthermore, the detection of a *Vulnerable JavaScript dependency and* of a *DOM-based* XSS vulnerability signals a need for meticulous attention to client-side security. It is imperative to sanitize and validate user inputs in every single-entry point of the application and employ secure coding practices to prevent malicious scripts from compromising the integrity of our application, as mentioned before.

It is worth noting that the identification of false positives, such as email disclosure in the profile page, underscores the challenges in automated vulnerability scanning tools. It emphasizes the necessity for human validation and interpretation of scan results to differentiate between genuine vulnerabilities and benign behaviours specific to the application's intended functionality.

Another case where this is seen is in the *DOM-based XSS* where the scan could not exploit the vulnerability that existed, indeed.

In conclusion, this web vulnerability scan also served as a reminder that cybersecurity is an iterative, ongoing and dynamic process. By enforcing continuous improvement and staying aware of emerging threats, it is possible to get closer to what is a secure application and environment, essential in healthcare applications.

7. Discussion and limitations

The development of the *Psyment* web application for psychological assessments, with a focus on functionality and security, aimed to represent a valid solution for psychologists to test their patients, which combines the intersection between technology and health.

The comprehensive and healthy software development approach adopted throughout the project lifecycle contributed to the successful development of the proposed work.

The initial phase of requirements elicitation involved a study of user needs, system functionalities, and security considerations. The field study conducted in collaboration with a clinic, where health professionals actively engaged with and tested the application, added a crucial layer to the project. The positive feedback received from these professionals validated the functional aspects of the application and provided valuable insights into the user experience. This feedback, coupled with the requirements gathered during the field study, helped refining and expanding the platform's capabilities.

Following a final review of the requirements, 88% of the security requirements were successfully achieved. The security requirements that missed implementation were SR5 and SR7. SR5 was supposed to be met, but as the vulnerability *DOM-based XSS* was found (see section 7 - Testing), that is triggered by a flawed sanitization of a specific field, this requirement is considered not achieved. Regarding SR7, it was not implemented due to time constraints. Nonetheless, this high level of implementation underscores the project's commitment to meeting security standards and addressing the specific needs of healthcare applications. Additionally, most of the remaining types of requirements were also successfully incorporated, demonstrating a comprehensive and well-executed development process. Moreover, the in-depth study of existing standards, adherence to good practices, and consideration of fundamental security principles such as the *CIA* triad (Confidentiality, Integrity, and Availability) had a profound positive influence on the application's overall design, development and structure since the vast majority of the issues found were Informational. The extensive research into industry standards and best practices played a crucial role in shaping the application's security posture. The conscientious implementation of secure coding practices, data encryption, and access controls contributed to the creation of a robust and resilient web application.

The vulnerability assessment scan conducted aimed to validate the effectiveness of the security measures implemented during the development phase. While the scan revealed a few issues with low exploitability and potential consequences, the identification of a *DOM-based* Cross-Site Scripting (XSS) vulnerability was a critical finding with medium severity. Unlike the other issues, this vulnerability was manually exploited to demonstrate its existence, highlighting the significance of this particular security concern. Therefore, the *DOM-based XSS* finding served

as a reminder of the importance of frequent and thorough vulnerability assessments. This vulnerability reinforces the notion that even seemingly minor vulnerabilities can be potential entry points for attackers, especially in the context of healthcare applications that handle sensitive patient information. Moreover, given the size and complexity of the web application developed, including numerous endpoints and various attack vectors such as input fields, the discovery of vulnerabilities is considered a normal part of the development and testing process. It underscores the dynamic nature of web security with technologies always evolving and being patched, as well as the constant need for testing.

Although we completed the project successfully, we had some limitations. Firstly, the number of health professionals approached by the field studies and interviews was limited due to lack of time, availability and logistics. Secondly, regarding security, health professionals, as expected, did not provide extensive technical contributions during the usability testing phase. Given the highly specialized nature of security concerns, their lack of in-depth input was expected. However, the professionals did emphasize the importance of data security in the application under development, as seen in the interviews (see Appendices). Consequently, for instance, usability testing, from a security perspective, did not yield as much utility as it did for UI development. Regarding the implementation itself, as the platform developed has a large range of functionalities and for instance, the tests' part is highly flexible, it required the usage of several technologies and thus, the process of implementing them in a secure way took longer than expected. Finally, as already mentioned, time restrictions prevented the implementation of all specified security requirements.

Conclusion

In this work, a web platform was developed with the aim of supporting healthcare professionals in the process of applying psychological assessments to their patients.

The platform was developed taking into account the most common practices in software development, which include phases such as requirements elicitation based on interviews and field studies, prototyping and testing. It is worth highlighting the fact that the interviews and the field study were one of the most important points of the entire work, as they gave a clear and objective view of the difficulties and needs of psychologists. Consequently, most of the features mentioned by them were implemented, and the vast majority of requirements were met.

Regarding the prototypes initially developed, these were iteratively improved until the final version of the *Psyment* platform was reached (see Appendices) taking into account all the feedback collected throughout the development process.

Regarding security, it was a premise of this work. The application, as previously mentioned, from its conception to implementation has always included security as one of its pillars. The vulnerability scan carried out, although revealing problems in the application, was validation in terms of security of a successful implementation. This does not mean that the application has no more flaws than the detected ones, but rather that it is closer to being a safe application and that it exposes itself to as few risks as possible.

Regarding the difficulties experienced in this project, it is worth highlighting the fact that developing an application that involves security at all stages of development is a complex and time-consuming process. Furthermore, the fact that web application security consists of both the front-end (client side) and the back-end (server side) increases the level of complexity and the requirement for in-depth knowledge of the entire scope, technology and application functionality by the developer. Furthermore, the developer responsible for the application's security also had some contributions to the back-end development of the same application, as well as smaller contributions to the front-end, that are not deeply mentioned in the present document. This factor increased the project's development time. However, with good time and resource management, all of these problems were resolved, and the project was completed successfully, with all of the aforementioned goals met.

In conclusion, this work demonstrates the interconnection of functionality and security in web application development, particularly in the healthcare industry. The use of standards, best practices, and security principles created a solid foundation, while the vulnerability assessment identified areas for improvement. Moving forward, a commitment to regular and extensive testing is critical to enable the continuous protection of web applications against emerging threats and the maintenance of the robust security implementation required in the healthcare industry.

8. Future Work

In terms of future work, firstly fix the vulnerabilities discovered by the automated vulnerability scan. For instance, the usage of the out-of-date *jQuery* library and the *DOM-based* XSS (see Appendices). Following, implement the missing requirements. Regarding the security requirements, SR5 and SR7. Furthermore, re-scan the application as well as the new implemented requirements. Then, conduct a penetration test to the developed application and make sure the same application gets scanned and gets tested frequently (penetration test).

From the point of view of new functionalities, the incorporation of *AI* with the support of Machine Learning techniques and algorithms to automate the evaluation of answers to open questions based on patterns of phrases and expressions, in psychological assessments.

References

[1] «Mental health». [Em linha]. Disponível em: https://www.who.int/health-topics/mental-health

[2] «O que é a saúde mental? Não seja espectador passivo da vida!», Associação de Apoio aos Doentes Depressivos e Bipolares. [Em linha]. Disponível em: https://www.adeb.pt/pages/o-que-e-a-saude-mental

[3] M. Silva *et al.*, «Barriers to mental health services utilisation in Portugal – results from the National Mental Health Survey», *J Ment Health*, vol. 31, n.º 4, pp. 453–461, jul. 2022, doi: 10.1080/09638237.2020.1739249.

[4] A. Pinto-Meza *et al.*, «Social inequalities in mental health: results from the EU contribution to the World Mental Health Surveys Initiative», *Soc. Psychiatry Psychiatr. Epidemiol.*, vol. 48, n.º 2, pp. 173–181, fev. 2013, doi: 10.1007/s00127-012-0536-3.

[5] A. Loutsiou-Ladd, G. Panayiotou, e C. M. Kokkinos, «A Review of the Factorial Structure of the Brief Symptom Inventory (BSI): Greek Evidence», *Int. J. Test.*, vol. 8, n.º 1, pp. 90–110, fev. 2008, doi: 10.1080/15305050701808680.

[6] F. Fariña, L. Redondo, D. Seijo, M. Novo, e R. Arce, «A meta-analytic review of the MMPI validity scales and indexes to detect defensiveness in custody evaluations», *Int. J. Clin. Health Psychol. IJCHP*, vol. 17, n.º 2, pp. 128–138, 2017, doi: 10.1016/j.ijchp.2017.02.002.

[7] I. Santana *et al.*, «Mini-Mental State Examination: Screening and Diagnosis of Cognitive Decline, Using New Normative Data», *Acta Médica Port.*, vol. 29, pp. 240–248, abr. 2016, doi: 10.20344/amp.6889.

[8] «MoCA - Cognitive Assessment», MoCA – Cognitive Assessment. [Em linha]. Disponível em: https://www.mocatest.org/

[9] H. Heathfield, D. Pitty, e R. Hanka, «Evaluating information technology in health care: barriers and challenges», *BMJ*, vol. 316, n.º 7149, p. 1959, jun. 1998, doi: 10.1136/bmj.316.7149.1959.

[10] H. Thimbleby, «Technology and the Future of Healthcare», *J. Public Health Res.*, vol. 2, n.º 3, p. e28, dez. 2013, doi: 10.4081/jphr.2013.e28.

[11] É. F. D. Cunha, «Nexus BRaNT Backoffice para BRaNT», masterThesis, 2023. [Em linha]. Disponível em: https://digituma.uma.pt/handle/10400.13/5102

[12] M. Alloghani, D. Al-Jumeily, A. Hussain, A. J. Aljaaf, J. Mustafina, e E. Petrov, «Healthcare Services Innovations Based on the State of the Art Technology Trend Industry 4.0», em *2018 11th International Conference on Developments in eSystems Engineering (DeSE)*, set. 2018, pp. 64–70. doi: 10.1109/DeSE.2018.00016.

[13] W. Yao, C.-H. Chu, e Z. Li, «The Adoption and Implementation of RFID Technologies in Healthcare: A Literature Review», *J. Med. Syst.*, vol. 36, n.º 6, pp. 3507–3525, dez. 2012, doi: 10.1007/s10916-011-9789-8.

[14] P. Guo, K. Watts, e H. Wharrad, «An integrative review of the impact of mobile technologies used by healthcare professionals to support education and practice», *Nurs. Open*, vol. 3, n.º 2, pp. 66–78, 2016, doi: 10.1002/nop2.37.

[15] G. Aceto, V. Persico, e A. Pescapé, «The role of Information and Communication Technologies in healthcare: taxonomies, perspectives, and challenges», *J. Netw. Comput. Appl.*, vol. 107, pp. 125–154, abr. 2018, doi: 10.1016/j.jnca.2018.02.008.

[16] I. R. Bardhan e M. F. Thouin, «Health information technology and its impact on the quality and cost of healthcare delivery», *Decis. Support Syst.*, vol. 55, n.º 2, pp. 438–449, mai. 2013, doi: 10.1016/j.dss.2012.10.003.

[17] A. Greenberg, «The Untold Story of NotPetya, the Most Devastating Cyberattack in History», p. 14.

[18] R. K. Knake, «A Cyberattack on the U.S. Power Grid», p. 11.

[19] S. Ghafur, S. Kristensen, K. Honeyford, G. Martin, A. Darzi, e P. Aylin, «A retrospective impact analysis of the WannaCry cyberattack on the NHS», *Npj Digit. Med.*, vol. 2, n.º 1, Art. n.º 1, out. 2019, doi: 10.1038/s41746-019-0161-6.

[20] K. Pequenino, «Ciberataque obriga serviço de saúde irlandês a desligar sistemas informáticos», PÚBLICO. [Em linha]. Disponível em:

https://www.publico.pt/2021/05/14/tecnologia/noticia/ciberataque-obriga-servico-saude-irlandes-desligar-sistemas-informaticos-1962672
[21] «Ataques informáticos mais do que duplicaram em Portugal nos últimos dois anos - SIC Notícias». [Em linha]. Disponível em: https://sicnoticias.pt/pais/2022-05-04-ataques-informaticos-mais-do-que-duplicaram-em-portugal-nos-ultimos-dois-anos
[22] F. Lau, S. H. Rubin, M. H. Smith, e L. Trajkovic, «Distributed denial of service attacks», em *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. «cybernetics evolving to systems, humans, organizations, and their complex interactions» (cat. no.0*, out. 2000, pp. 2275–2280 vol.3. doi: 10.1109/ICSMC.2000.886455.
[23] «CNPD». [Em linha]. Disponível em: https://www.cnpd.pt/organizacoes/obrigacoes/violacao-de-dados-ou-data-breach/
[24] K. A. Whitler e P. W. Farris, «The Impact of Cyber Attacks On Brand Image: Why Proactive Marketing Expertise Is Needed for Managing Data Breaches», *J. Advert. Res.*, vol. 57, n.º 1, pp. 3–9, mar. 2017, doi: 10.2501/JAR-2017-005.
[25] S. Kamiya, J.-K. Kang, J. Kim, A. Milidonis, e R. M. Stulz, «What is the Impact of Successful Cyberattacks on Target Firms?» em Working Paper Series. National Bureau of Economic Research, março de 2018. doi: 10.3386/w24409.
[26] N. Tariq, «IMPACT OF CYBERATTACKS ON FINANCIAL INSTITUTIONS», p. 11.
[27] J. N. Butcher, «Minnesota Multiphasic Personality Inventory», em *The Corsini Encyclopedia of Psychology*, John Wiley & Sons, Ltd, 2010, pp. 1–3. doi: 10.1002/9780470479216.corpsy0573.
[28] C. Cox, «Brief Symptom Inventory», p. 13.
[29] «HIPAA Compliance – Your guide to HIPAA Compliance». [Em linha]. Disponível em: https://www.hipaa.com/
[30] «General Data Protection Regulation (GDPR) Compliance Guidelines», GDPR.eu. [Em linha]. Disponível em: https://gdpr.eu/
[31] «ISO - ISO/IEC 27001 and related standards — Information security management», ISO. [Em linha]. Disponível em: https://www.iso.org/isoiec-27001-information-security.html
[32] «DC4DM – DC4DM – Digital Creativity for Digital Maturity». [Em linha]. Disponível em: https://www.dc4dm.eu/
[33] «Journal of Entrepreneurial Researchers». [Em linha]. Disponível em: https://jer.ponteditora.org/index.php/jer/index
[34] R. M. Kaplan e D. P. Saccuzzo, *Psychological testing: principles, applications, and issues*, 7th ed. Belmont, CA: Wadsworth Cengage Learning, 2009.
[35] «Brief Symptom Inventory - PsycNET». [Em linha]. Disponível em: https://psycnet.apa.org/doiLanding?doi=10.1037%2Ft00789-000
[36] R. C. Peveler e C. G. Fairburn, «Measurement of neurotic symptoms by self-report questionnaire: validity of the SCL-90R», *Psychol. Med.*, vol. 20, n.º 4, pp. 873–879, nov. 1990, doi: 10.1017/S0033291700036576.
[37] Z. J. Lipowski, «Somatization: the experience and communication of psychological distress as somatic symptoms», *Psychother. Psychosom.*, vol. 47, n.º 3–4, pp. 160–167, 1987, doi: 10.1159/000288013.
[38] «Psychiatry.org - What is Somatic Symptom Disorder?». [Em linha]. Disponível em: https://www.psychiatry.org:443/patients-families/somatic-symptom-disorder/what-is-somatic-symptom-disorder
[39] E. Helmes e J. R. Reddon, «A Perspective on Developments in Assessing Psychopathology: A Critical Review of the MMPI and MMPI-2», p. 19.
[40] D. J. Stein *et al.*, «Obsessive–compulsive disorder», *Nat. Rev. Dis. Primer*, vol. 5, n.º 1, p. 52, ago. 2019, doi: 10.1038/s41572-019-0102-3.
[41] V. R. Preedy e R. R. Watson, Eds., «5-Point Likert Scale», em *Handbook of Disease Burdens and Quality of Life Measures*, New York, NY: Springer, 2010, pp. 4288–4288. doi: 10.1007/978-0-387-78665-0_6363.
[42] G. Groth-Marnat, *Handbook of Psychological Assessment*. John Wiley & Sons, 2009.

[43] «Minnesota Multiphasic Personality Inventory (MMPI) | SKYbrary Aviation Safety». [Em linha]. Disponível em: https://skybrary.aero/articles/minnesota-multiphasic-personality-inventory-mmpi
[44] «Online psychometric testing software for behavioral health professionals - PsyPack». [Em linha]. Disponível em: https://psypack.com
[45] G. Stoet, «PsyToolkit: A Novel Web-Based Method for Running Online Questionnaires and Reaction-Time Experiments», *Teach. Psychol.*, vol. 44, n.º 1, pp. 24–31, jan. 2017, doi: 10.1177/0098628316677643.
[46] J. Kim, U. Gabriel, e P. Gygax, «Testing the effectiveness of the Internet-based instrument PsyToolkit: A comparison between web-based (PsyToolkit) and lab-based (E-Prime 3.0) measurements of response choice and response time in a complex psycholinguistic task», *PLOS ONE*, vol. 14, n.º 9, p. e0221802, set. 2019, doi: 10.1371/journal.pone.0221802.
[47] W. Summers e E. Bosworth, «Password policy: The good, the bad, and the ugly», pp. 1–6, jan. 2004.
[48] G. News, «GDPR Password Requirements», HIPAA Journal. [Em linha]. Disponível em: https://www.hipaajournal.com/gdpr-password-requirements/
[49] M. Muthuppalaniappan LLB e K. Stevenson, «Healthcare cyber-attacks and the COVID-19 pandemic: an urgent threat to global health», *Int. J. Qual. Health Care*, vol. 33, n.º 1, p. mzaa117, jan. 2021, doi: 10.1093/intqhc/mzaa117.
[50] S. Samonas e D. Coss, «THE CIA STRIKES BACK: REDEFINING CONFIDENTIALITY, INTEGRITY AND AVAILABILITY IN SECURITY».
[51] I. G. Cohen e M. M. Mello, «HIPAA and Protecting Health Information in the 21st Century», *JAMA*, vol. 320, n.º 3, pp. 231–232, jul. 2018, doi: 10.1001/jama.2018.5630.
[52] «Health Insurance Portability and Accountability Act of 1996 (HIPAA) | CDC». [Em linha]. Disponível em: https://www.cdc.gov/phlp/publications/topic/hipaa.html
[53] J. Hintzbergen, K. Hintzbergen, A. Smulders, e H. Baars, *Fundamentos de Segurança da Informação: com base na ISO 27001 e na ISO 27002*. Brasport, 2018.
[54] «ISO - International Organization for Standardization», ISO. [Em linha]. Disponível em: https://www.iso.org/home.html
[55] «What is GDPR, the EU's new data protection law?», GDPR.eu. [Em linha]. Disponível em: https://gdpr.eu/what-is-gdpr/
[56] F. Holik e S. Neradova, «Vulnerabilities of modern web applications», em *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia: IEEE, mai. 2017, pp. 1256–1261. doi: 10.23919/MIPRO.2017.7973616.
[57] «OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation». [Em linha]. Disponível em: https://owasp.org/
[58] «OWASP Top Ten | OWASP Foundation». [Em linha]. Disponível em: https://owasp.org/www-project-top-ten/
[59] «A01 Broken Access Control - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A01_2021-Broken_Access_Control/
[60] M. M. Hassan, Md. A. Ali, T. Bhuiyan, M. H. Sharif, e S. Biswas, «Quantitative Assessment on Broken Access Control Vulnerability in Web Applications», out. 2018.
[61] A. Anis, M. Zulkernine, S. Iqbal, C. Liem, e C. Chambers, «Securing Web Applications with Secure Coding Practices and Integrity Verification», em *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, ago. 2018, pp. 618–625. doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00112.
[62] «A02 Cryptographic Failures - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
[63] F. Mendel, C. Rechberger, e M. Schläffer, «MD5 Is Weaker Than Weak: Attacks on Concatenated Combiners», em *Advances in Cryptology – ASIACRYPT 2009*, M. Matsui, Ed., em Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, pp. 144–161. doi: 10.1007/978-3-642-10366-7_9.

[64] R. Rahim *et al.*, «Prototype File Transfer Protocol Application for LAN and Wi-Fi Communication».
[65] G. Deepa e P. S. Thilagam, «Securing web applications from injection and logic vulnerabilities: Approaches and challenges», *Inf. Softw. Technol.*, vol. 74, pp. 160–180, jun. 2016, doi: 10.1016/j.infsof.2016.02.005.
[66] «A03 Injection - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A03_2021-Injection/
[67] O. Abikoye, A. Abubakar, H. Dokoro, A. OLUWATOBI, e A. Kayode, «A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm», *EURASIP J. Inf. Secur.*, vol. 2020, ago. 2020, doi: 10.1186/s13635-020-00113-y.
[68] «A04 Insecure Design - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A04_2021-Insecure_Design/
[69] G. Rossi, O. Pastor, D. Schwabe, e L. Olsina, *Web Engineering: Modelling and Implementing Web Applications*. Springer Science & Business Media, 2007.
[70] S. Loureiro, «Security misconfigurations and how to prevent them», *Netw. Secur.*, vol. 2021, n.º 5, pp. 13–16, mai. 2021, doi: 10.1016/S1353-4858(21)00053-2.
[71] «A05 Security Misconfiguration - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[72] «CVE - CVE». [Em linha]. Disponível em: https://cve.mitre.org/
[73] A. Zerouali, V. Cosentino, T. Mens, G. Robles, e J. Gonzalez-Barahona, *On the Impact of Outdated and Vulnerable Javascript Packages in Docker Images*. 2019.
[74] «A06 Vulnerable and Outdated Components - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/
[75] «A07 Identification and Authentication Failures - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
[76] «A08 Software and Data Integrity Failures - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/
[77] «A09 Security Logging and Monitoring Failures - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/
[78] C. Dalton, «How Simplicity Can Lead to Improved Security», 2021.
[79] «Detecting Server-Side Request Forgery (SSRF) Attack by using Deep Learning Techniques - ProQuest». [Em linha]. Disponível em: https://www.proquest.com/openview/afe7ca89f1656bc6f8e7195f5522e991/1?pq-origsite=gscholar&cbl=5444811
[80] «A10 Server Side Request Forgery (SSRF) - OWASP Top 10:2021». [Em linha]. Disponível em: https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/
[81] R. W. Elwood, «The Wechsler Memory Scale—Revised: Psychometric characteristics and clinical application», *Neuropsychol. Rev.*, vol. 2, n.º 2, pp. 179–201, jun. 1991, doi: 10.1007/BF01109053.
[82] R. A. Grier, A. Bangor, P. Kortum, e S. C. Peres, «The System Usability Scale: Beyond Standard Usability Testing», *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 57, n.º 1, pp. 187–191, set. 2013, doi: 10.1177/1541931213571042.
[83] «Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale - JUX», JUX - The Journal of User Experience. [Em linha]. Disponível em: https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/
[84] «NASA Task Load Index | Digital Healthcare Research». [Em linha]. Disponível em: https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/nasa-task-load-index
[85] «Intrinsic Motivation Inventory (IMI) – selfdeterminationtheory.org». [Em linha]. Disponível em: https://selfdeterminationtheory.org/intrinsic-motivation-inventory/

[86] «NVD - cve-2021-32040». [Em linha]. Disponível em: https://nvd.nist.gov/vuln/detail/cve-2021-32040
[87] S. Holmes, *Mongoose for Application Development*. Packt Publishing Ltd, 2013.
[88] B. Hou, K. Qian, L. Li, Y. Shi, L. Tao, e J. Liu, «MongoDB NoSQL Injection Analysis and Detection», em *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, jun. 2016, pp. 75–78. doi: 10.1109/CSCloud.2016.57.
[89] S. Singh, «Security Analysis of MongoDB», Open Science Framework, preprint, out. 2020. doi: 10.31219/osf.io/c3w7y.
[90] «MongoDB Compass | MongoDB». [Em linha]. Disponível em: https://www.mongodb.com/products/tools/compass
[91] S. Tilkov e S. Vinoski, «Node.js: Using JavaScript to Build High-Performance Network Programs», *IEEE Internet Comput.*, vol. 14, n.º 6, pp. 80–83, nov. 2010, doi: 10.1109/MIC.2010.145.
[92] «About», Node.js. [Em linha]. Disponível em: https://nodejs.org/en/about
[93] «EJS -- Embedded JavaScript templates». [Em linha]. Disponível em: https://ejs.co/#features
[94] C. Anderson, «The Model-View-ViewModel (MVVM) Design Pattern», em *Pro Business Applications with Silverlight 5*, C. Anderson, Ed., Berkeley, CA: Apress, 2012, pp. 461–499. doi: 10.1007/978-1-4302-3501-9_13.
[95] J. Deacon, «Model-View-Controller (MVC) Architecture», 2009.
[96] «Introdução ao MVVM (Model-View-ViewModel)», Fabio de Oliveira. [Em linha]. Disponível em: https://fabiosoliveira.wordpress.com/2011/04/18/introducao-ao-mvvm-model-view-viewmodel/
[97] «What is cross-site scripting (XSS) and how to prevent it? | Web Security Academy». [Em linha]. Disponível em: https://portswigger.net/web-security/cross-site-scripting
[98] «dompurify», npm. [Em linha]. Disponível em: https://www.npmjs.com/package/dompurify
[99] «What is Clickjacking? Tutorial & Examples | Web Security Academy». [Em linha]. Disponível em: https://portswigger.net/web-security/clickjacking
[100] «Clickjacking: Attacks and Defenses | USENIX». [Em linha]. Disponível em: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/huang
[101] «helmet», npm. [Em linha]. Disponível em: https://www.npmjs.com/package/helmet
[102] «Authentication vulnerabilities | Web Security Academy». [Em linha]. Disponível em: https://portswigger.net/web-security/authentication
[103] A. Wheeler e M. Winburn, «Chapter 6 - Best Practices», em *Cloud Storage Security*, A. Wheeler e M. Winburn, Eds., em Computer Science Reviews and Trends. , Boston: Elsevier, 2015, pp. 113–123. doi: 10.1016/B978-0-12-802930-5.00006-X.
[104] «How to secure your authentication mechanisms | Web Security Academy». [Em linha]. Disponível em: https://portswigger.net/web-security/authentication/securing
[105] «express-validator | express-validator». [Em linha]. Disponível em: https://express-validator.github.io/docs
[106] «Passport.js», Passport.js. [Em linha]. Disponível em: https://www.passportjs.org/
[107] «Nodemailer :: Nodemailer». [Em linha]. Disponível em: https://nodemailer.com/
[108] «Access control vulnerabilities and privilege escalation | Web Security Academy». [Em linha]. Disponível em: https://portswigger.net/web-security/access-control
[109] Md. M. Hassan *et al.*, «Broken Authentication and Session Management Vulnerability: A Case Study of Web Application», *Int. J. Simul. Syst. Sci. Technol.*, mai. 2018, doi: 10.5013/IJSSST.a.19.02.06.
[110] J. Huang, Y. Li, J. Zhang, e R. Dai, «UChecker: Automatically Detecting PHP-Based Unrestricted File Upload Vulnerabilities», em *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, jun. 2019, pp. 581–592. doi: 10.1109/DSN.2019.00064.
[111] «File uploads | Web Security Academy». [Em linha]. Disponível em: https://portswigger.net/web-security/file-upload
[112] D. J. Hickok, D. R. Lesniak, e M. C. Rowe, «File Type Detection Technology».
[113] «multer», npm. [Em linha]. Disponível em: https://www.npmjs.com/package/multer

[114] M. Shachi, N. Shourav, A. S. Sajid Ahmed, A. Brishty, e N. Sakib, «A Survey on Detection and Prevention of SQL and NoSQL Injection Attack on Server-side Applications», *Int. J. Comput. Appl.*, vol. 183, pp. 1–7, jun. 2021, doi: 10.5120/ijca2021921396.
[115] «Database encryption - Archive ouverte HAL». [Em linha]. Disponível em: https://hal.science/hal-00623915/
[116] D. S. A. Elminaam, H. M. A. Kader, e M. M. Hadhoud, «Evaluating The Performance of Symmetric Encryption Algorithms», 2010.
[117] A. R. A. Tamimi, «Performance Analysis of Data Encryption Algorithms».
[118] M. J. Dworkin, «Advanced Encryption Standard (AES)», National Institute of Standards and Technology, Gaithersburg, MD, NIST FIPS 197-upd1, 2023. doi: 10.6028/NIST.FIPS.197-upd1.
[119] D. Rihan, A. Salih, S. Eldin, e F. Osman, *A Performance Comparison of Encryption Algorithms AES and DES*. 2015.
[120] R. Lanjewar e G. Pande, «Implementation of AES-256 Bit: A Review», vol. 2015, n.º 3, 2015.
[121] «Crypto | Node.js v21.2.0 Documentation». [Em linha]. Disponível em: https://nodejs.org/api/crypto.html
[122] C. C. Editor, «Initialization Vector (IV) - Glossary | CSRC». [Em linha]. Disponível em: https://csrc.nist.gov/glossary/term/initialization_vector
[123] M. Dworkin, «Recommendation for Block Cipher Modes of Operation: Methods and Techniques», National Institute of Standards and Technology, NIST Special Publication (SP) 800-38A, dez. 2001. doi: 10.6028/NIST.SP.800-38A.
[124] «Initialization Vector - an overview | ScienceDirect Topics». [Em linha]. Disponível em: https://www.sciencedirect.com/topics/computer-science/initialization-vector
[125] N. Sharma, H. Meghwal, M. Mehta, e T. Kumar, «A Review on Playfair Substitution Cipher and Frequency Analysis Attack on Playfair», em *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, mai. 2018, pp. 1–9. doi: 10.1109/ICOEI.2018.8553837.
[126] K. Kent e M. P. Souppaya, «Guide to computer security log management», National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-92, 2006. doi: 10.6028/NIST.SP.800-92.
[127] «winston», npm. [Em linha]. Disponível em: https://www.npmjs.com/package/winston
[128] «morgan», npm. [Em linha]. Disponível em: https://www.npmjs.com/package/morgan
[129] A. Al Anhar e Y. Suryanto, «Evaluation of Web Application Vulnerability Scanner for Modern Web Application», em *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*, jun. 2021, pp. 200–204. doi: 10.1109/ICAICST53116.2021.9497831.
[130] «Snyk Vulnerability Database | Snyk», Learn more about npm with Snyk Open Source Vulnerability Database. [Em linha]. Disponível em: https://security.snyk.io/

(Blank page)

Appendices

Informed consent for the interviews

Informação ao Participante de Investigação e Consentimento Informado

Título do Estudo: Desenvolvimento de uma plataforma web de criação e gestão de formulários
Investigador(es) Principa(l/is): Nome: Alexandre Romão, Bruno Rodrigues, Eva Freitas
Instituto: Universidade da Madeira
E-mail: romaoalexandre10@gmail.com, bruno2000rodri@gmail.com, evaazevedofreitas@gmail.com
Tel: 964099411, 968191201, 967022825

Objetivo do Estudo
O objetivo deste estudo é inquirir sobre o processo de avaliação de pacientes por parte dos psicólogos, metodologia, eficiência e possibilidade de criação de uma plataforma de suporte a este procedimento.

Procedimento
O estudo longitudinal destina-se a psicólogos. Vão ser realizadas entrevistas com o objetivo de recolher o máximo de informação conveniente necessária para os investigadores poderem prosseguir com o seu estudo e esclarecer dúvidas que os mesmos possam ter sobre práticas, processos e métodos de avaliação dos psicólogos aos seus pacientes. Após as entrevistas, os resultados são alvo de uma análise.

Critérios de Inclusão
Será considerado elegível para participar neste estudo se:

- Possuir ciclos de estudos e/ou experiência relevantes na área de Psicologia.

Confidencialidade
A confidencialidade dos dados será mantida das seguintes formas:
Todos os **dados pessoais** fornecidos na experiencia serão guardados e não serão partilhados com terceiros. Contudo, os dados recolhidos durante a experiência poderão ser usados/publicados para fins científicos ou educativos. Os nomes de **TODOS** os participantes serão **OCULTADOS**, incluindo nos artigos científicos publicados.

Autorização Opcional
Entendo que os investigadores podem querer usar fotografias, vídeo ou áudio por razões ilustrativas nas apresentações e publicações deste trabalho, para fins científicos ou educativos. Eu dou autorização para fazê-lo, **DESDE** que o nome e rosto **NÃO** apareçam.

Assine no lugar pretendido: _____SIM _____NÃO

Direitos
A sua participação é voluntária. Você é livre de interromper a sua participação em qualquer momento. A recusa em participar ou interrupção da participação não resultará em qualquer penalização, ou perda de eventuais benefícios ou direitos. O investigador principal poderá decidir, de forma fundamentada, interromper a sua participação neste estudo. Caso se verifique esta situação, esta não resultará em qualquer penalização, ou perda de eventuais benefícios ou direitos.

Esclarecimento de Dúvidas & Contatos
Se você tem dúvidas sobre este estudo, poderá fazer agora todas as perguntas. Se quiser fazer perguntas mais tarde, desejar obter mais informações, ou desejar interromper a sua participação no estudo, entre

em contato com o Investigador Principal em pessoa, por telefone ou e-mail. A informação de contato está disponível no início da primeira página deste documento.

Consentimento Informado Voluntário

Ao assinar este documento, você confirma que leu a informação acima descrita sobre este estudo, e que todas as suas perguntas foram respondidas. Assim mesmo, você poderá fazer perguntas adicionais a qualquer momento durante o estudo, e mesmo após este ter terminado. Ao assinar este documento, você concorda em participar neste estudo de investigação. Irá receber uma cópia deste documento de consentimento informado assinada e datada.

_____ _____

ASSINATURA DO PARTICIPANTE DATA

Investigador que Obtém o Consentimento

Como membro da equipa de investigação, confirmo que expliquei ao participante acima referido a natureza e finalidade deste estudo de investigação, e que esclareci quais os potenciais benefícios e eventuais riscos da participação no estudo. Todas as perguntas foram respondidas e estou disponível para esclarecer quaisquer dúvidas que possam surgir ao longo do estudo.

_____ _____

ASSINATURA DO INVESTIGADOR 1 DATA

_____ _____

ASSINATURA DO INVESTIGADOR 2 DATA

_____ _____

ASSINATURA DO INVESTIGADOR 3 DATA

Ao assinar este documento, você confirma que leu a informação acima descrita sobre este estudo, e que todas as suas perguntas foram respondidas. Assim mesmo, você poderá fazer perguntas adicionais a qualquer momento durante o estudo, e mesmo após este ter terminado. Irá receber uma cópia deste documento de consentimento informado assinada e datada.

Guide for the interviews

Entrevista

Bom dia, obrigada por disponibilizar um pouco do seu tempo para esclarecer-nos dúvidas sobre o processo de avaliação dos psicólogos aos pacientes. A nossa tese consiste no desenvolvimento de uma plataforma que permite a criação de ferramentas de avaliação pelos psicólogos e investigadores de forma interativa e eficiente. Dito isto, gostaríamos de fazer-lhe algumas questões de modo a esclarecer algumas dúvidas que temos sobre a forma como o processo de avaliação de pacientes é realizado, desde a fase inicial de conhecimento do doente até a fase de análise e monitorização dos mesmos. Aceita, com base no seu conhecimento, contribuir para este estudo e permite-nos gravar/filmar intervenções?

Ferramentas / Testes

Análise de testes tradicionais

1.Quantos anos de experiência tem na sua área?

2. Quantos testes, em média, costuma realizar por dia?

3. Que métodos de avaliação ou ferramentas de teste utiliza mais frequentemente?

3.1. Essas ferramentas estão publicamente disponíveis, são feitas por você ou ambas?

3.3. A validação de novos testes é um processo demorado?

3.4. Das que foi você que fez (métodos de avaliação), baseou-se num template ou foram feitas por si?

3.4.1. Costuma personalizar as ferramentas públicas que há?

3.4.2. As que foram feitas por si, demorou quanto tempo, em média, a criar os testes?

3.4.3. Utilizou papel ou um meio digital?

3.4.4. Utiliza outros recursos para avaliação do estado psicológico dos pacientes? Vídeos, imagens, áudios, Jogos, outros?

Análise de testes digitais

4. Sente-se confortável em utilizar meios digitais?

4.5. Costuma utilizar ferramentas digitais no dia a dia?

4.5.1. Utiliza essas ferramentas para realizar os testes? Se sim, quais?

5. Prefere uma interação direta ou indireta? Supor um tablet ou desktop/ laptop

Análise das respostas

6. Durante a avaliação dos pacientes, como regista as suas respostas?

7. Onde armazena todos os testes?

8. Tem hábito de ver registos antigos?

9. Faz comparação de testes do mesmo paciente? (Evolução do paciente)

10. Qual o tempo de análise, em média?

11. Como avalia os dados registados?

12. Utiliza cálculos?

13. Como avalia respostas abertas?

14. É hábito partilhar formulários de avaliação entre vocês, psicólogos?

15. Tem costume de partilhar resultados de um paciente, anonimamente, com outro psicólogo fora e dentro do local de trabalho?

16. Acha pertinente a criação de uma plataforma digital que permita a realização, gestão, armazenamento de testes e dados de saúde mental dos pacientes de forma interativa e eficiente?

Conclusão

17. O que acha essencial ter no seu método de avaliação psicológica de pacientes e que está em falta atualmente?

18. O que acha que pode ser melhorado no processo de avaliação que é feito por si?

19. Se houvesse um sistema que conseguisse facilitar e acelerar o seu processo de avaliação de pacientes, que funcionalidades gostaria que esta tivesse? realizar funções, tais como registar, armazenar, gerir, criar e partilhar avaliações

Interview with Psychologist 1 - P01

Team: Bom dia, fale-nos um pouco sobre o seu percurso e que tipo de psicologia pratica atualmente.

P01: Eu sou psicólogo com especialidade avançada em psicologia clínica e psicologia da justiça. Trabalhei durante 4 anos na comissão de proteção de crianças e jovens em Câmara de Lobos. Depois, decidi fazer PhD em Psicologia numa Universidade estrangeira na Suécia. Depois então, quando quis regressar à Madeira, (...) tive oportunidade de fazer um pós-doutoramento, que estou a fazer agora, na área de Reabilitação cognitiva na demência, ou seja, com idosos com demência para tentarem recuperar as suas memórias (...).

Team: Ou seja, costuma trabalhar com pacientes diretamente e realizar testes com pacientes (?)

P01: Realizei testes, mais com adolescentes quando estava na Comissão de proteção de crianças e jovens sim. Neste momento, não estou diretamente ligado à clínica.

Team: Sabe em média quantos testes realizava por dia ou por semana?

P01: Em relação a testes de avaliação?

Team: Sim, testes em que realiza perguntas para avaliar pacientes.

P01: É assim, eu por dia recebia cerca de 5 adolescentes e digamos que a cada 5, 2 precisávamos de usar testes, porque nem sempre os psicólogos precisam de usar testes. Por exemplo, se existe a desconfiança de uma depressão severa ou de um início de uma depressão, às vezes é preciso validar esse diagnóstico com um teste para sabermos se estamos a apanhar todos os sinais que o paciente nos está a dar e para saber se o teste vai de encontro aquilo que consideramos. Por exemplo, às vezes o tribunal pedia-nos testes de personalidade para saber por exemplo se um adolescente tinha determinada personalidade, que era necessário saber, e nós realizávamos esses testes chamados perícias.

Team: E esses testes, você ia buscá-los a alguma plataforma, online, ou fazia manualmente?

P01: É assim, os testes para ser muito sincero com vocês, cá na Madeira, por falta de financiamento, muitas vezes não temos capacidade de comprar "legalmente" os testes como deveria ser feito, porque alguns destes testes estão protegidos por copyright e há outros que já expiraram. Há uns que conseguimos ir à net fazer o download. Eu já tinha a minha pasta no computador com os testes que precisava, abria a pasta, e imprimia e realizava o teste. (...) Já existem alguns testes que se podem realizar sem imprimir a folha, no próprio computador. Mas sim, a maioria é assim que se faz. (...) Há troca de conhecimento entre psicólogos, por exemplo, imagine-se que há um teste muito específico, por exemplo, para avaliar o autismo, e que o colega recebeu o paciente autista e não tem esse teste nem encontra na net que está protegido. Ele pede

a um colega que trabalha com autismo e este colega envia por e-mail ou facilita de alguma maneira. Normalmente é assim que funciona. O ideal seria as instituições terem os testes oficiais comprados, mas não é sempre assim que acontece.

Team: E esses testes específicos, havia a necessidade de criar alguma pergunta personalizada ou o teste era sempre feito de forma global e estandardizada?

P01: Há testes que teem respostas abertas, por exemplo um teste que estou a aplicar agora (...), tem respostas abertas em perguntas como "Que dia é hoje?", "Em que ano é que estamos?", "Qual é a estação do ano?", e há lá outras perguntas abertas que se pode colocar. Mas sim, existem alguns testes mais fechados, onde teem que ser aqueles parâmetros e perguntas que já veem incluídas no teste, definidas pelo autor, e depois existem outros testes e perguntas mais gerais. Uma coisa que os psicólogos usam muito no início é chamado uma ficha de Anamnese, e essa ficha normalmente pergunta todo o historial de vida da pessoa, e às vezes somos nós próprios que criamos porque existe umas fichas mais completas, outras menos completas, uns mais para idosos, outras mais relacionadas com a violência doméstica. Dependendo do paciente, tem-se que acertar a ficha.

Team: Você falou em validação, ou seja, você quando realiza um teste necessita de validação de uma entidade? Ou simplesmente faz o teste e pode realizar o teste ao paciente?

P01: Nós fazemos os testes, mas quando os recebemos já veem validados. Esses testes para serem validados, são alvo de estudos como se faz em instituições como ARDI-TI. Pegamos em 200 ou 300 pessoas, e utilizamos um teste, e validamos se tem qualidade, chamada validade externa e interna, e verificamos se o teste tem este tipo de validade. São investigados que fazem isto. Se o teste for considerado válido, depois é disponibilizado no mercado geral. Alguns são gratuitos, outros são pagos infelizmente.

Team: E teem algum período para expirar a sua validade?

P01: Sim, por exemplo, há quem faça uma patente. Por exemplo, imaginem que existem três investigadores (...) que fazem um teste em escala com 15 perguntas. Depois de testarem com 100 pessoas, veem que tem bastante validade (...) e vai ser aprovado para ser utilizado com pacientes. O que acontece é que eles para ganhar algum dinheiro, criam uma patente (...).

Team: Mas é o psicólogo que paga?

P01: Depende, se eu trabalhar numa clínica no Funchal, em que eu trabalhe sozinho ou com 3 ou 4 colegas psicólogos, nós temos que pagar porque é privado e não temos ninguém a nos garantir

nada. Agora se trabalharmos no hospital, ou no centro de saúde (...), podemos pedir à entidade (...).

Team: Que tipos de testes é que existem? E um teste em papel, você faz o teste e quem responde diretamente no papel é o paciente ou é o psicólogo.

P01: É o paciente normalmente. É assim, se houver caso de pessoas que tenham dificuldade como pessoas idosas, nós podemos fazer as perguntas e ir anotando, assinalando. Mas normalmente pede-se ao paciente. Até existem alguns colegas que dão o teste para a pessoal levar a casa. Porquê? Porque estão a responder, inventam ou porque nas primeiras sessões, existe aquela timidez. Às vezes são perguntas profundas, muito íntimas, e as pessoas até preferem levar para casa. Já tive casos de adolescentes que respondiam ao teste, dobravam, metiam dentro de um envelope e depois entregavam. Eles sentiam que era confidencial, e isso é muito importante. Sentir que há essa confidencialidade entre o psicólogo e o próprio paciente, mais ninguém tem acesso. Isso é outra questão sobre os testes, devem ser confidenciais, e ser destruídos após algum tempo. Ou seja, depois de serem respondidos. Não se deve, segundo a proteção de dados, manter por mais de 5 anos respostas de pessoas. (...) O psicólogo pode sair de lá e aceder a dados de pacientes que não são dele, isso não deve acontecer (...) Mas estavas a falar de que tipo de testes?

Team: De testes em papel, impressos, e testes no computador ou num tablet?

P01: Eu sinceramente acho que falta dar passos no sentido de termos muitos dos testes informatizados. E não é só informatizados, é que, por exemplo, a coisa mais chata dos testes para o psicólogo não é dar o teste para fazer, é depois quando recebemos o teste, existem testes mais completos e outros mais simples. Temos um teste de personalidade que tem 90 perguntas. O que acontece? Antigamente, há uns anos atrás (...) havia umas folhas de acetato (...) e nós o que fazíamos para despachar trabalho, era meter o teste, de sim ou não, e ver os quadradinhos e nós sabemos qual a correta e a incorreta. Colocamos o acetato em cima da resposta, com as respostas corretas e conseguimos ver quais foram as que eles acertaram ou que não acertaram. Mas o nosso ideal, nós gostaríamos nos testes é que as aplicações informáticas nos cotassem automaticamente, tipo por exemplo, o paciente responde e dissessem logo "O paciente tem 16 pontos no BDI - escalda de depressão (...) Mas o ideal é o paciente acaba de responder, e nós não mostramos a ele obviamente, mas clicávamos num botãozinho qualquer e obter-se-ia o resultado. Como não se pode fazer a cotação à frente do paciente, manda-se o paciente embora e agendamos uma segunda sessão. Sendo muito honesto com vocês, quem trabalha no privado, tem uma vantagem. Recebe mais um dinheirinho na segunda sessão, mas em termos de eficiência, a eficiência devia ser: nós aplicamos o teste e temos o resultado imediato. E nisso, os profissionais de design, informática e isso são uma grande mais-valia para nós.

Team: Ao calcular se a pessoa por exemplo tem depressão moderada, é por cálculos mesmo ou é por resposta certa/resposta errada?

P01: Há vários tipos de teste. Por isso, por exemplo, o que eu sugeria é que quem entrasse num projeto desses, fizesse um levantamento dos testes que são mais usados. Na Madeira, existem três tipos de psicólogos, das clínicas, dos alunos e das empresas para os recursos humanos, que trabalham no recrutamento de pessoal. O pessoal do recrutamento de pessoal também tem muito esse problema. Imaginem que vocês entrevistam 60 pessoas em 1 semana e todos tiveram que fazer testes psicométricos, que eles chamam psicotécnicos. O que acontece? Aquilo é tudo em papel. Quando acaba a fase de recrutamento, todos os aqueles 60 testes vão ter que ser cotados um a um à mão, quando a pessoa a ser recrutada pode ficar 1 mês ou 2 a ser chamada, porque aquilo dá muito trabalho. Se fosse instantâneo, demoraria 1 dia ou 2, ou at é mais rápido. Se calhar, o programa já poderia dizer as pessoas mais aptas, ou a pessoa mais apta foi X. Isto no caso das empresas. No caso da clínica, o que eu recomendava é falar com alguns colegas de clínica, saber o top 10 dos testes mais usados com os pacientes no vosso dia-a-dia. Depois, vocês poderiam ver as perguntas dos testes, de resposta aberta, fechada, se são testes numa escala de 1 a 10 ou se são escalas abertas, se são testes de sim ou não. Ou até se são testes que obrigam a uma interpretação, por exemplo, existem testes em que o paciente pode responder o que ele quiser. Eu penso que esses testes são mais complicados para a informática, porque aí é mais complicado, mas todos os outros poderiam estar mais automatizados.

Team: Os testes de resposta aberta ficam sempre no caso da interpretação dos psicólogos.

P01: Embora, (...) já tive colegas que trabalhavam com uns sistemas inteligentes, que detetavam as palavras na resposta do teste, um bocado como o Google faz. Acho que é Machine Learning, acho que tem a ver com isso.

Team: Mas isso não poderia dar alguns problemas? No sentido em que a palavra não podia significar negativo mas na frase ter outro sentido.

P01: Sim, tens razão. Eu acho que como estamos a lidar com pacientes, e a única forma que temos de avaliar é a palavra, realmente é preciso um cuidado extra aí nesse caso. (...) De qualquer forma, esses testes informatizados precisavam de ser validados, validade interna, externa (...) Ou se teem a mesma qualidade dos testes em papel.

Team: E te costume de partilhar ou há algo que costuma partilhar o resultado do teste do paciente com outros colegas?

P01: Às vezes o que se faz é (...) o scan e envia-se ao colega e ele manda de volta. Já se faz tudo por computador, envia-se aos colegas.

Team: Os dados pessoais do paciente não são revelados?

P01: Não são revelados. Mas sim, existe partilha e existe também, não no meu caso (...), mas no SESARAM por exemplo, tem uma coisa que é chamada supervisão, ou seja, imagina que eu aplico um teste e tive um resultado que não sei interpretar. Eu posso levar todos os meses, ou de 15 em 15 dias. Existe o que eles chamam a reunião com os membros mais antigos em que já estão há 20/30 anos (os experts), em que vamos a essa reunião e discutimos casos (...) resultados inconclusivos. "Alguém já teve um paciente parecido com estes sintomas?". Isso é muito importante. Na psicologia educacional, o psicólogo da escola reúne-se com a direção da escola, não divulga a identidade do aluno porque é proibido, mas sim isso acontece.

Team: Então acha pertinente a criação de uma plataforma digital que permite a realização, gestão, armazenamento de testes e de dados de saúde mental?

P01: Com certeza que sim, sou a favor. A única coisa que acho fundamental neste tipo de plataformas é que cumpra sempre com a segurança dos dados dos pacientes, seguindo todas as orientações das leis de proteção de dados recente. O que acontece? Algumas apps que recolhem dados atualmente ainda não tão programadas para esquecer os dados passado uns anos, e acho que os programadores devem ter em atenção.

Team: E esse esquecimento/delete, preferia que fosse automático ou manual?

P01: Acho que poderia ser algo do género de um popup. "O paciente X tem dados cá há 5 anos e não visitam a clínica.". O computador normalmente que faz a gestão dos clientes consegue ver se o cliente já não vai lá há 4 ou 5 anos, vocês percebem disso melhor que eu. Era muito bom que fosse assim com um popup. Às vezes vamos a computadores com dados de 1900 e muitos e ainda estão lá esses dados, coisas que não devia acontecer atualmente.

Team: E em termos de análise, referia que gostaria de um sistema que cotasse automaticamente. E uma análise em termos de gráficos e tabelas, que acha disso? Facilita alguma coisa?

P01: Depende muito de teste para teste. Existe um teste, BSI, eu utilizava muito, que é Brief Symptom Inventory (...), que em poucas perguntas consegue-se ver muitas sintomatologias: depressão, ansiedade, tendência suicida, pessoa a desenvolver mania, se tem traços de sociopatia (...) tudo num teste só em que a pessoa responde a umas perguntas. Depois, na cotação, a cotação funciona por thresholds, limites, valores de referência. Por exemplo, imaginem para paranóia o valor é 17 - o máximo, e uma pessoa teve 15, para a depressão grave é 28, se tem abaixo de 28 tem depressão moderada. Como esse teste faz tudo, se houvesse um gráfico no final que mostra os resultados para todas as patologias com cores e aparece os que excediam.

Team: Você falou em intervalos, gostaria que no final do teste aparecesse apenas a resposta, ou o sistema pudesse ser configurado com os intervalos definidos pelos testes validados (...)

P01: Por exemplo, existe um parâmetro para jovens, adultos, séniores, e séniores acamados, ambulatório, etc. Os números mudam. Se no início da aplicação do teste, se perguntasse o tipo de paciente, e depois fosse buscar os limites associados a esse teste era o ideal, mas não. Tem-se que ir à tabela do sénior ver os limites, naquela papelada toda. Acho que era muito giro um trabalho sobre o BSI, podem investigar sobre isso. é muito utilizado na psicologia clínica, é muito completo e a análise é um pouco complexa porque tem vários níveis. A nível informático deve ser um pouco mais desafiante que estar a fazer um teste mais simples.

Team: Uma das ambições que temos para este trabalho é a plataforma suportar qualquer tipo de teste, apesar de ser desafiante. Nem tanto a realização do teste, mas a parte da avaliação, por existirem tantas formas de avaliar o paciente, deve ser o mais desafiante.

P01: Vou vos dizer também que por volta de 90% dos testes existentes em papel só avaliam uma valência, ou seja, são mais fáceis de informatizar. Agora os que avaliam mais valências talvez seja mais desafiante para depois introduzir nessa tal plataforma que vocês referem. São testes quase multinível.

Team: É frequente você realizar várias vezes o mesmo teste a um paciente de modo a ver se houve algum tipo de desenvolvimento, evolução. Por exemplo, esse BSI faz hoje um teste a um paciente e tem X resultados e em 5 meses volta a fazer esse teste e faz a comparação para ver se existe algum tipo de evolução?

P01: Com certeza, isso é uma prática comum. Existe um teste muito comum que é o BDI que avalia a depressão, e por exemplo nesse teste é muito comum os médicos pedirem ao psicólogo para aplicar o BDI para ter a certeza que o paciente não deve tomar medicação anti-depressiva. Chega um cliente com depressão a uma psicologia, está 6 meses com esse paciente. O psicólogo faz o teste e um relatório de final de acompanhamento e manda ao médico. O médico pede alguma coisa ao psicólogo para justificar a paragem da medicação, e esse tipo de teste justifica. O grande problema dos testes psicotécnicos e psicologia clínica é que tem que haver boa fé da parte do paciente. Por exemplo, eu como trabalhava na Justiça muita gente não queria dizer a verdade, por problemas com a justiça, no caso de jovens envolvidos com droga ou violadores. Eles podem mentir em muitas das perguntas. É diferente de um paciente que vem pedir ajuda, vai dizer a verdade se quer ser ajudado. Temos que ter em atenção se o paciente está mesmo a responder a verdade. Há pessoas que entram também em processo de negação, doentes depressivos em depressão severa, e vamos ver o teste e está tudo bem, parece a pessoa mais feliz do mundo. O teste foi manipulado, não se sente à vontade com o psicólogo, é mais uma curiosidade.

Interview with Psychologist 2 – P02

Team: Entretanto, tem quantos anos de experiência na área?

P02: Terminei o curso em 2008, portanto por volta de 13 anos, estou no Laboratório há 10, tive 3 anos de experiência mais no terreno, e estou na área de investigação há 10 anos.

Team: E durante a sua carreira, já realizou muitos testes de avaliação (?)

P02: Sim.

Team: Quantos testes por semana/mês?

P02: Numa semana, num contexto clínico, podia realizar cerca de 20 testes assumindo testes individualizados, visto que quando avaliamos alguém, recorremos a uma bateria de testes: avaliação da atenção, da memória, das funções executivas, da sintomatologia depressiva, ansiosa, funcionalidade, capacidade das pessoas para desempenharem as atividades de vida diária, neste caso questionários de autorresposta. Se me perguntar quantas avaliações no contexto clínico uma pessoa pode fazer numa semana, serão cerca de 5 a 6 pessoas e essas mesmas pessoas poderá se aplicar cerca de 20 a 20 e tal testes porque também é importante e depende daquilo que se vai encontrando, mas vamos pensar numa média de 20 a 30 testes por semana, portanto muito variável. Em termos de investigação, praticamente não faço avaliações porque estou a trabalhar na docência. Quando estava a recolher dados e trabalhava nas avaliações, se calhar tinha 4 a 5 pessoas por semana, e fazia cerca de 40 testes por semana, porque estava a aplicar baterias fixas para validar a eficácia de uma intervenção e então toda a gente fazia X testes antes e X depois, mais ou menos por aí, tendo em conta que há certos testes que levam 5 minutos a aplicar, pode parecer muito mas não estou a falar numa bateria.

Team: E que ferramentas usava para realizar os testes? Papel, folha, computador, tablet?

P02: Portanto em papel e lápis, todos de avaliação. Embora eu trabalhe nas tecnologias, nós temos tecnologias para reabilitação. Quando falamos em avaliação, há a necessidade de ter uma base de dados normativos. Em princípio, se uma pessoa tem um desempenho de 26 em 30, para eu saber para aquela idade e escolaridade, se essa pontuação corresponde a um défice ou não. Portanto, não há muitos instrumentos de avaliação validados na área dos instrumentos computorizados.

Team: Era você que realizava as perguntas nos testes ou os testes já existiam?

P02: Todos existentes.

Team: Mas adaptava alguma pergunta?

P02: Não, são testes todos que já estão aprovados, com 50/60 anos e teem muitos dados normativos, e estão validados para a população portuguesa.

Team: E esses testes, ia buscá-los online, fazia o download e teria que imprimir, correto?

P02: Alguns testes são de livre circulação e consegue-se encontrar online. No entanto, a maioria dos testes sendo ferramentas de trabalho específico da psicologia, psiquiatria e neurologia são de circulação restrita. Por exemplo, alguns testes são vendidos por empresas da área e é necessário enviar a cédula profissional para se poder comprar.

Team: E o processo é esse, encontrar o teste que pretende, imprimir e realizar o teste?

P02: Sim, adquiri muitos testes ao longo do meu estágio curricular, durante o curso, depois na minha experiência clínica, fui adquirindo a testes de supervisores e colegas, e atualmente tenho acesso a um conjunto de testes. Se for planear um estudo, penso no que quero avaliar para aquele estudo, e dentro daquilo que tenho, é imprimir e utilizar.

Team: Tendo em conta isso, depois como era feita a medição dos resultados?

P02: É uma correção de forma MANUAL. Existem manuais de correção.

Team: Em que sentido esses manuais de correção.

P02: Por exemplo o teste do desenho do relógio, em que a pessoa tem que desenhar um relógio com os números todos e a pessoa tem que desenhar um relógio com todos os números das horas e que marque 11h10. Há manuais com vários desenhos de relógios e para não haver muitas dúvidas nos pontos e subjetividade na cotação, os números teem que estar alinhados nos 4 quadrantes, quanto vale determinado tipo de respostas. Há questões muito simples, se eu estiver a fazer um teste de memória em que digo 5 palavras e daí a meia hora a pessoa tem que voltar a repetir as palavras, é fácil cotar esta resposta. Quando se trata de capacidade visio-construtiva, há sempre um manual que diz quanto vale cada tipo de resposta, cópia do cubo, para ajudar a cotar de forma objetiva.

Team: Você ao criar um teste, é muito difícil que seja aprovado porque ainda precisa de resultados predominantes para servirem de comparação, correto?

P02: Na área da avaliação, gostávamos de seguir esse caminho. Na área da avaliação computorizada, há uma grande vantagem, podemos criar um instrumento. No instituto *Guttman* em Espanha, eles teem o *Guttman Neuro Personal Trainer*. Eles criaram uma série de exercícios computorizados que podem ser utilizados. O *Guttman* é um instituto de reabilitação muito grande em Espanha. Estes exercícios são aplicados lá mas as pessoas também fazem em casa, para fins

de estimulação e reabilitação cognitiva. Eles teem dados de milhares de pessoas. Através de técnicas de *Machine Learning* e inteligência artificial, é realmente possível começarmos a pensar numa validação remota. O processo da validação depende muito disto, de haver resultados de muitas pessoas e que se possa fazer uma comparação por idade e escolaridade. 24 pontos em 30 no *Montreal Cognitive Assessment* é pouco para um jovem com escolaridade mas é uma pontuação boa para uma pessoa de 70 anos com 4º de escolaridade, não apresenta défice. Estão a perceber aqui a questão dos dados normativos? Então passa muito por aí, e realmente sobre as novas tecnologias, gostaríamos de começar a apostar um bocadinho nisso, portanto poder utilizar os dados para neste caso uma validação mais robusta, apenas no contexto clínico, visto que sobre o que as pessoas fazem em casa, poderá ter uma margem de erro porque podem ter ajuda, poderão não ser tão fiáveis. Mas sim, acho que as novas tecnologias veem facilitar este processo de validação das novas escalas. Se conseguir enviar estes testes remotamente para um grande número de pacientes, o teste é mais rapidamente validado. E mesmo que não seja para pacientes, para colegas noutros centros de reabilitação que vão aplicar nos seus pacientes.

Team: Falando nisso, alguma vez teve que partilhar os resultados de um teste de um paciente porque se encontrava com dúvidas ou porque tinha esse hábito de partilhar com um colega seu?

P02: Portanto, durante a minha prática profissional, eu trabalhava com outra colega em que às vezes havia essa discussão. Agora na investigação, é um contexto diferente, é um contexto que não se aplica para fins de diagnóstico, mas para fins de intensidade dos programas de reabilitação, portanto nem se costuma olhar para os dados normativos.

Team: Entretanto, onde costuma armazenar os testes? No seu computador? Numa pasta?

P02: Os testes costumo armazenar em papel e lápis. Tenho os que usei para investigação todos digitalizados no meu computador. Não é o ideal, mas não tem o nome das pessoas, tem um código. Na investigação, não identificamos as pessoas com dados pessoais.

Team: Então aquilo que você tem no computador, você não sabe de quem é?

P02: Tem o ID, e esse ID tenho uma lista em papel.

Team: Já nos foi dito que só se deve armazenar dados de pacientes durante algum tempo, isso é uma prática atual?

P02: Sim, mas depende de para que fim for. Portanto, eu acho que não há um tempo limite. Eu tenho dados guardados de 10 anos em papel e lápis, e tenho digitalizados em computador. Se agora quiser fazer um estudo retrospetivo, posso fazer. No entanto, acredito que nos hospitais não se deva armazenar a partir de uma certa altura, também pela capacidade de armazenamento. Deve ficar lá apenas uma conclusão da avaliação, do relatório nos sistemas informáticos, mas o arquivo

de papel e lápis não deve ser guardado depois de determinado tempo, não sei qual é o tempo de referência.

Team: E durante todo o tempo que realizou testes, quais eram os tipos de testes mais predominantes que realizou?

P02: O *Montreal Cognitive Assessment*, um teste de screening cogntivo que nos dá uma ideia geral sobre o perfil cognitivo daquela pessoa. Depois, há uma bateria que é *Weschler Adult Intelligence Scale*, que é a escala mais utilizada em toda o mundo. Já foram realizados levantamentos pela Associação Americana de Psicólogos sobre as práticas da validação e é realmente a escala mais utilizada, é aquela que dá o quociente intelectual, depois dá o quociente intelectual nas competências verbais, não verbais, vídeo espaciais, acaba por ser muito completa e é uma escala de referência com dados normativos em muitos países, e é a que utilizamos mais. Também o *Trail Making Test*, A e B, também é uma das mais utilizados em todo o mundo. Também há a escala de *Weschler* para crianças e só para a avaliação de memória, são escalas mais standard. Não quer dizer que sejam as melhores, mas são as mais estabelecidas na prática, já são muito usadas. Mas vão aparecendo escalas novas. Fui fazer um estudo há pouco tempo e usei escalas muito novas, usei uma escala que encontrei muito boa do Reino Unido, mas que não é nada conhecida para avaliação funcional e coloquei num estudo. Depois, não há comparação com outros estudos.

Team: Ou seja, o que estamos a desenvolver, a plataforma de criação e de gerir formulários para testes e etc. e para visualização de resultados, vamos também tentar incluir a partilha de testes entre psicólogos e pessoas na área da saúde. Acha pertinente a criação de uma plataforma dessas e aqui na Madeira?

P02: Acho pertinente, sim. Eu pessoalmente prefiro muito mais. Acho que papel e lápis está a ficar muito obsoleto, e dá muito trabalho. E acho que ajudaria, porque na prática, o que acontece, é que há muito poucas pessoas a avaliar uma avaliação neuropsicológica bem feita, no sistema público, devido a estes recursos muito demorosos. No privado, as pessoas acabam por ter ainda ter, a faculdade de psicologia da Universidade de Coimbra tem muito trabalho na área da avaliação psicológica, e sabem quantas sessões eles levam para fazer uma avaliação neuropsicológica, para aplicar uma bateria extensa de testes? 5 sessões, cerca de 1 hora, 1 hora e pouco, é muito tempo. Aqui, não conheço ninguém que faça esse tipo de avaliações, estamos a falar num centro standard. Aqui no privado, as pessoas fazem 2 ou 3 sessões. Mas lá está, há testes para avaliar a validade de desempenho, se o utilizador está a colocar esforço suficiente na tarefa. Isto é importante para questões de tribunal, se as pessoas pedem avaliações para fins de tribunal, para justificar incapacidade, para ter invalidez, reforma antecipada, as questões de validade são importantes. Se nós usássemos mais tecnologia, sei lá *eye-tracking*, etc. quem sabe

eletrofisiologia, resposta galvânica da pele, talvez não precisássemos de passar 30 minutos a fazer testes de validade de desempenho. Acho que ainda há muito a fazer nesta área.

Team: E o que é que acha essencial incluir nessa plataforma?

P02: Visualização de resultados, essencial o armazenamento de resultados de forma que depois se possa tratar facilmente, ou seja, permita depois analisar os dados por variáveis importantes como idade, escolaridade, sexo masculino, feminino. Normalmente os dados normativos estão divididos nestas variáveis. Isto em termos das tarefas.

Team: Sim, em termos de funcionalidades, o que gostava que a plataforma tivesse? Por exemplo a partilha de formulários entre profissionais.

P02: O modo de visualização de são para são. Também quais os testes o paciente fez, ter os testes divididos por domínio, testes de atenção, de funções executivas. Acho que isso é o mais importante.

Team: E existir uma comparação, o paciente faz um teste e passado um tempo faz novamente o teste?

P02: Acho isso importante, uma comparação entre o pré e o pós.

Team: Preferia também fazer os seus próprios testes ou utilizar já feitos e validados?

P02: Pois, depende. É importante a variabilidade de estímulos. Vamos pensar numa tarefa de nomeação de imagens, em que há 10 imagens para nomear, o paciente tem de identificar o que e é. Uma limitação da avaliação neuropsicológica atualmente é o efeito de aprendizagem, um paciente faz um teste hoje faço uma intervenção, aplico daqui a um mês e tenho um efeito de aprendizagem de um teste, são todos semelhantes. Apenas o *MoCA* de triagem, tem três versões diferentes. Nas imagens, mudam as imagens, na memorização de 5 palavras, mudam as palavras. Eu acho que a liberdade para fazer os testes, no fundo não é ter uma liberdade, mas ter uma base de dados para uma tarefa de nomeação, se eu preciso de 10 imagens, talvez ter uma base de dados de 40 ou 50 e haver uma aleatorização para que haja sempre variabilidade nos vários momentos, ou seja, não ser sempre as mesmas imagens no segundo e terceiro momento. Agora, eu ter a liberdade para na tarefa de memória usar três ou cinco palavras, a comparação... eu acho que nesta fase em que vocês estão, para validar, acho que deve haver alguma rigidez no sistema em si. Por acaso a liberdade para uma área em que estou a trabalhar agora, testes de avaliação para cirurgias com o paciente acordado, cirurgias com tumores em determinadas áreas e a pessoa fica acordada para remover o tumor de forma a deixar o mínimo de défices na pessoa. Então a pessoa fica acordada para garantir que os cirurgiões não mexam em áreas de funcionalidade. Utiliza-se muito tarefas de nomeação, nomeação de imagens na área da linguagem, e nós estamos a propor que

isso seja feito através de óculos VR. E temos colegas em Lisboa que só querem utilizar imagens a preto e branco ou escala de cinzentos. Que a cor, a tridimensionalidade, a tarefa vai deixar de ser tão pura de linguagem e vai interferir com outras áreas. Em França, um grupo que já usa VR, quer que seja o mais realista possível, a três dimensões e eles até gostavam de fazer a comparação, porque teem métodos de avaliação de neuroimagens muito bons. Então no sistema, estamos a pensar fazer as duas coisas, imagens a preto e branco, imagens mais realistas. A liberdade neste sentido de usar preto e branco, realista. Para fins de avaliação, a liberdade vai interferir com a questão da uniformidade e dos dados normativos. Se fosse para reabilitação, não tínhamos de estar muito preocupados com isso.

Interview with Psychologist 3 – P03

Team: Comece por nos dizer quantos anos de experiência tem na sua área?

P03: Tenho que pensar um pedacinho... se for a incluir o estágio profissional, 3 anos.

Team: E já realizou testes com pacientes ao longo desses anos.

P03: Sim, já antes disso, no estágio curricular também já aplicava.

Team: E quantos testes já fazia em média por dia/semana.

P03: Dependia muito do contexto por exemplo no contexto de estágio curricular, era diário, tínhamos sempre avaliações, tínhamos sempre depois a parte da intervenção mas quase sempre diariamente tinha-se um novo paciente para avaliar. No contexto de estágio profissional, talvez semanalmente, nem sempre havia novas pessoas e aí nesse contexto muitas vezes envergávamos pela anamnese, mais pela entrevista do que pela propriamente pela aplicação de testes. Mas havia também alguns de rastreio. Em termos do que eu trabalhei asseguir, que foi num projeto de apoio domiciliário a idosos, era numa fase inicial e final de intervenção, e ali uma intermédia para ver um ponto de referência em relação ao que estávamos a trabalhar. Portanto aí já era muito menos frequente.

Team: E como adquiria os testes que pretendia fazer?

P03: Portanto, no contexto de clínicas e casas de saúde, ou centros de reabilitação, que foi onde tive os estágios, eles próprios já tinham os testes e aí nós utilizávamos, tínhamos essa autorização. No projeto, alguns ou uma colega emprestava ou alguns tinha que aceder através da internet. Em termos de testes de rastreio que não implica material, é mais fácil obter, não é a melhor forma mas é a mais fácil de obter online. Aqueles que requerem material já tem que ser mesmo adquirido de algum lado. Nos outros lados, eles tinham portanto nós utilizávamos.

Team: E os testes, realizava alguma alteração no teste?

P03: Por exemplo, se nós tivéssemos algum paciente analfabeto, com baixa escolaridade, às vezes tínhamos que fazer alguma adaptação nalguma questão, alguma modificação ligeira na maneira de apresentar, digamos. De resto, não. Adaptar às vezes a linguagem, é mais por aí.

Team: E realizar algum teste de raiz?

P03: Se eu fiz algum teste mesmo? Não, nunca trabalhei nessa área da investigação, de construir mesmo testes.

Team: E acha que isso é algo útil?

P03: Sim, sem dúvida. Quem trabalha nessa área, é importante porque há coisas que muitas vezes nós, ou pela experiência, vamos questionando as pessoas em alguns sentidos. Mais na parte da clínica. Damos as entrevistas estruturadas ou semi-estruturados que nos dão alguma orientação em termos de diagnóstico. Mas há outras coisas mais complexas, já temos que fazer ali mesmo uma avaliação e precisamos de teste. Não é só questionar, temos mesmo que optar pelos testes, portanto quando são desenvolvidos ou se notamos algum gap, digamos, em termos de avaliação há áreas que são mais difíceis como pessoas com afazias. É uma área para mim em particular, desafiante em termos de avaliação. Nem sempre é fácil, ah é este instrumento e aquele. Depende muito do que nos é apresentado à frente.

Team: Para ir buscar esses testes à internet, utilizou alguma ferramenta pública?

P03: Depende, alguns os próprios autores disponibilizam, faço o download e utilizo. Como é para fins de trabalho, utilizo, não é para estar a divulgar em termos de investigação. Outros é um bocadinho tentar encontrar onde estão disponíveis.

Team: E os testes que são feitos, são sempre feitos pessoalmente através de papel correto? Ou há algum que é feito por meio digital?

P03: Depende da intervenção. Em termos da neuropsicologia, eu acho mais difícil fazer online, há muita coisa para avaliar. Já vai sendo mais desenvolvido porque o covid exigiu isso. Acho mais difícil na psicologia fazer online. Em termos de psicologia clínica, é mais fácil de fazer online. Se eu estiver a aplicar um questionário, é mais fácil de fazer.

Team: Mas pessoalmente, em vez de usar folha e papel, usar um computador ou tablet acha isso beneficial?

P03: Sim, sem dúvida. Até é importante irmos tendo algumas alternativas e as pessoas muitas vezes dependendo da faixa etária estão recetivas a fazer, sim.

Team: E durante esses testes, como é feita a obtenção de respostas, ou seja, como regista as respostas dos pacientes.

P03: Depende, se for por exemplo testes género questionário, aí escrevo as respostas das pessoas diretamente. Se for teste ais específico de memória, de atenção, mais dentro da neuropsicologia, depende do próprio teste. Há testes que são tão bem limitados a nível de tempo, cronómetro, apontar tempos, apontar por vezes respostas, cruzinhas consoante a resposta da pessoa. Depende muito do teste, uma opção correta, a opção que a pessoa está a dizer, depende muito do próprio teste.

Team: E onde é que armazena essas respostas de testes? Folha de papel? Computador? Formato digital? Digitaliza?

P03: Quando estava no centro de reabilitação e casa de saúde, nós tínhamos sempre o suporte em papel mas depois inseríamos sempre essa informação no software que eles utilizavam. Tínhamos que fazer a cotação, e fazer um resumo, e depois colocar essa informação online, aquilo que fosse relevante para que outros técnicos que tivessem intervenção com esse paciente pudessem ter acesso a essa informação. Em termos do projeto, tenho também todo esse material em suporte papel e nós utilizávamos, como não tínhamos um software específico, o google drive. Tínhamos pastas para cada utente e púnhamos a informação em formato relatório, fosse o que fosse, sobre cada utente. Acaba por ser nos dois lados, é mais seguro também.

Team: E se pretender ver registos antigos?

P03: Em casas de saúde e centros de reabilitação, temos a questão da confidencialidade. A partir do momento que deixo de trabalhar lá, deixo de ter acesso a essa informação. A menos que tenha algum material meu, mas mesmo os próprios testes e resultados ficam lá. Do projeto, como era um projeto de apoio domiciliário, nós não tínhamos uma estrutura, uma sede, esse material ficou comigo também para garantir a confidencialidade dos dados.

Team: E como você referiu, você faz os testes iniciais, depois a meio do processo e no final.

P03: Depende da duração da intervenção.

Team: Esses testes feitos nas várias etapas são os mesmos testes para servir de comparação?

P03: Exatamente.

Team: Costuma fazer isso de forma regular? Ou é só com certos tipos de doenças/pacientes?

P03: Normalmente, é mais regular. Em termos da psicologia clínica, dependendo do acompanhamento, nem sempre se justifica isso. Mas a nível da neuropsicologia sim. A menos que seja só uma avaliação. Mas se houver intervenção, há interesse de saber a evolução, se melhorou

ou piorou. Muitas vezes há pedidos de relatórios por parte de médicos, de seguradores, depende. Mas mais na neuropsicologia que na clínica, é sempre possível.

Team: Para dar alta ou para ver se melhorou, a principal forma de fazer isso é fazer esses dois testes iguais e compará-los ou faz dois testes diferentes e no final esse teste já consegue aferir a melhoria?

P03: Depende da intervenção e contexto. No contexto do centro de reabilitação, se nós já sabíamos à partida que aquela pessoa ia ter alta, já tínhamos que fazer essa avaliação, visto que a pessoa tem que sair de lá com o relatório com essa informação do que evoluiu e não evoluiu. Em casas de saúde, nem sempre isso é necessário. As pessoas muitas vezes tinham alta e não fazíamos isso, não fazia entre aspas do protocolo deles repetir tudo aquilo que tínhamos feito. Havia outro tipo de formas de avaliar, que eles chamam os PII's (Protocolos de Intervenção Individual) e aí nós fazemos uma avaliação dessa evolução. Neste caso por exemplo do projeto, nós em equipa íamos vendo se aquela pessoa realmente está a ter um benefício na nossa área. De qualquer forma, avaliamos para ver se houve evolução. Mas por exemplo, num caso onde tivéssemos algum utente com uma demência muito avançada que já não beneficiasse daquilo que estávamos a fazer, aí podemos nem conseguir avaliar dependendo do estado do paciente. Mas tentamos sempre avaliar de alguma forma porque é importante para nós e para a pessoa ver essa evolução, e para a família.

Team: Essa avaliação às vezes tem respostas abertas? Como é que essas respostas abertas são avaliadas? De uma forma geral. Há palavras específicas que são procuradas?

P03: Por exemplo, se nós estivermos a avaliar um rastreio em termos de sintomatologia ansiosa ou depressiva, podemos aplicar diretamente o teste e aí as respostas são mais fechadas, e mesmo que as pessoas fujam um bocadinho, podemos orientá-las. Mas se tivermos a avaliar um recurso da pergunta género do questionário mas mais em termos de conversa, existe palavras-chave, em termos de resposta aberta que as pessoas dão, nós já identificamos em termos de resposta onde é que isso se situa. Mas eu prefiro usar o questionário porque é mais fácil orientar a nível de gravidade.

Team: E é hábito normal partilhar resultados de testes com outros profissionais?

P03: Depende dos contextos e depende da autorização que as pessoas nos dão, os pacientes. Num contexto de centro de reabilitação ou casa de saúde, à partida as pessoas já dão esse consentimento e então nós partilhamos aquilo que é relevante com os outros técnicos. É importante por exemplo a nível de enfermagem saberem certas informações da psicologia para saber como abordar certas questões. No contexto do projeto de investigação, passámos um consentimento no início e só os elementos da equipa tinham acesso à informação. Para fora, só mesmo se algum dos pacientes

tivesse que ir a um neurologista por exemplo, e pediam-nos um relatório. Pedíamos à pessoa se dava essa autorização, e então fazíamos o relatório e entregávamos.

Team: Portanto, acha pertinente a criação de uma plataforma digital que permitisse a criação e a edição de testes, de ferramentas?

P03: Mas em que sentido? Para utilizar? Para ter fácil acesso?

Team: Para ter fácil acesso e utilizar os testes.

P03: Sim, sem dúvida. Acho que iria facilitar imenso a nossa vida. Temos sempre essa questão de se eu não tenho determinado teste, tenho que ver aqui outra maneira de chegar a esse resultado. Se tivéssemos essa plataforma, já seria mais fácil em termos de acessibilidade. Por vezes, podemos mandar email a pedir autorização a um autor para nos enviar um teste, e se for preciso, temos resposta 1 ano depois. A questão tempo útil também é importante, era bom. Ao adquirir esta plataforma, já sei que vou ter esta informação aqui. É muito mais prático. E sim, acho que em termos de organização, sei que tenho isto aqui nesta área em vez de ter na minha pasta, na minha cloud, apesar de fazermos a nossa organização, acho que sim, era muito mais prático.

Team: E acha pertinente essa plataforma ser não só de criação e de ir buscar testes para depois serem realizados, mas também para realizar mesmo na plataforma esses testes com pacientes?

P03: Mas de quê? De forma independente?

Team: Ou seja, você realiza o teste na plataforma pessoalmente, e o paciente realiza o teste no computador e a análise de resultados é feita de forma interativa e eficiente.

P03: Acho que sim, para já diminui a margem de erro, porque é claro que nós a fazermos cotações de uma avaliação longa, o cansaço pode nos atingir e pode ser suficiente para fazer uma soma errada e assim reduzia a margem de erro. E sim, é concentrar tudo num local, também facilita na questão do papel, reduz-se aqui um impacto, acho que sim, acho que seria útil e importante.

Team: Acha que o estado do uso de tecnologia na psicologia ainda é rudimentar? Ou a tecnologia já está bem presente na área da psicologia? Na área específica de testagem.

P03: Em Portugal ou em geral?

Team: Em Portugal, ilha da madeira.

P03: Ok, eu acho que ainda não está a ser utilizada o suficiente para o que já deveria. Pelo que vejo em outros países que já preenchem tudo online. Acho que ainda não estamos a utilizar o suficiente. Acho que já vamos tentando mais, mas acho que muitas das vezes talvez até mais a nível de intervenção que a nível de avaliação. São mais utilizados tablets a nível de estimulação do que a aplicar os testes.

Team: Você referiu que realiza somas, etc. Quais os tipos de cálculos que costuma realizar nesses testes?

P03: Depende de cada teste. Depende, por exemplo, se nós utilizarmos aqui algum subteste da escala de Memória de Weschler, nós temos que fazer ali somas de pontuações, temos que ter em conta o tempo que as pessoas levam a realizar a prova, pode ser um fator para nem aplicar a prova toda, mas normalmente implica somas, temos um resultado e depois temos que aceder a um manual específico em que existem lá as pontuações específicas de referência para aquela faixa etária, depende. E aí já vejo se aquela pontuação que eles chamam direta em termos padronizados, qual é o valor que dá. E para eu perceber o quão desviada da norma a pessoa está. Se é um desvio ligeiro, mais moderado ou mais grave. Se é um défice mais acentuado ou menos acentuado. As somas resultam numa pontuação direta que vamos enquadrar dentro do que já está estabelecido para aquela população. Existem algumas provas que já utilizei online, que também normalmente são de papel e lápis, que foram disponibilizadas gratuitamente por um centro de investigação, e aí já nos facilitavam muito a vida visto que já colocámos os resultados e já nos dava esse resultado padronizado. Facilitava bastante, não se tinha que verificar o manual. A plataforma seria boa nisso se já tivesse lá os resultados em termos de referência. Poupa tempo.

Team: Para além do referido, que é a criação e edição de testes, aplicar testes ver a análise dos resultados, temos a ideia de partilhar os resultados dos testes com os profissionais de saúde. O que acha mais essencial colocar na plataforma?

P03: Vão ter já o consentimento incluído na plataforma?

Team: Não tínhamos pensado muito nisso. O consentimento do paciente?

P03: Exato, poderia ser importante caso algum dia ele diga "ah mas eu não autorizei". Tem a informação disponível para toda a gente a quem autorizou. É importante. Acho que devo haver uma parte que seja mais privada, para os psicólogos, para colocarem as suas conclusões e outra que esteja acessível para os outros técnicos. Porque há informações relevantes para nós em termos de avaliação, mas que podem não ser relevantes ou podem em termos de consentimento, ter coisas que não são relevantes para passar às outras pessoas. Pode estar no limite do que se deve ou não passar para outras pessoas. Sempre que aplicarmos uma avaliação, referimos sempre essa questão da confidencialidade dos dados. Só existe situações em que se pode quebrar essa confidencialidade se houver ameaça à vida da pessoa ou de outra pessoa. Portanto há sempre questões importantes que sabemos, mas o resto não precisa de saber. Se houver essa maneira de salvaguardar informação só para o psicólogo, e outra para outros acederem.

Team: Outra questão, quais os testes mais predominantes que realizou?

P03: Depende do contexto. Já trabalhei em diferentes contextos. Em termos de neuropsicologia, são muitos, por exemplo em termos de memória, existem escalas mais abrangentes que avaliam um pouco de tudo, porque memória é um "Mundo" digamos, e está localizada em diferentes partes do cérebro. Temos que usar diferentes testes. Podemos usar escalas completas ou escolher subtestes, se já temos suspeita de algo em específico. Ou se queremos fazer uma avaliação completa abrangente. Se já temos suspeitas visto que houve uma alteração qualquer numa parte específica do cérebro, consoante o que já sabemos vamos sondar alguma coisa em específica. A escala de memória de Weschler é importante, existem várias dentro desta, existe escala de inteligência que não mede só inteligência mas avalia outras coisas como a atenção. Estou a dizer a geral, mas depois existem vários subtestes. As matrizes progressivas de Haven. A nível da atenção (...) existe o nível de Toulouse Pierron, o d2, o corte dos A's, o teste de Strup, o Trail Making Test. Um teste que eu acho bastante completo que é a figura complexa de Rey que nos dá informação sobre muita coisa, não sei se já existe uma versão online desse. Implica as pessoas desenharem, esse aspeto é interessante. A nível de rastreio, existe o MoCA, o Mini Mental mas eu prefiro usar o MoCA ou o Adam-Brooke Cognitive Examination. Porque o Mini Mental, apesar de ser muito utilizado a nível mundial, eu acho que não é dos mais fidedignos para nos dar ali informação sobre o declínio é subtil ou não. Aqui só nos dá se for mais acentuado. Outros testes... existe o teste de cancelamento de estrelas para Neglet, existe também uma escala CBS que é também utilizada pela terapia ocupacional se não estou em erro, para perceber se as pessoas estavam a explorar o espaço, digamos que estava negligenciado ou não. São muitos testes...

Team: Utiliza o BSI?

P03: Sim, na clínica pode se aplicar o BSI, o BDI-2 para rastreio a nível de sintomatologia depressiva. O EARES-21, Escala Ansiedade Depressão Stress. Existe o 21 e o 42, 21 é a forma reduzida e o 42 é a forma mais completa. Esse faz um rastreio de ansiedade, depressão e stress. De depressão uso mais o BDI, de sintomatologia ansiosa existem vários. Não me deparei com um que fosse mesmo bom, às vezes acabo por recorrer ao EADS e a mais algum, porque acho que falta ali qualquer coisa. Existe a escala de Zung, existe também a de Beck. Também existe a escala de pressão geriátrica específica para os idosos. Depois, existem várias provas se for algum mais específico, mais concreto. Gosto de aplicar essas para rastreio, porque muitas vezes é mais isso que me chega. Agora se for por exemplo alguém que já tenha sintomas positivos em relação a alucinações, delírios, alguma coisa que já envereda por outra parte específica da doença mental, aí ou utilizo entrevistas estruturadas, semi-estruturadas, ou tenho que recorrer a outros testes mais específicos. Agora não me estou a lembrar o nome, mas posso procurar e vos enviar. Podem aplicar em termos de autoestima ou qualidade de vida, esses também são importantes às vezes para nos dar uma referência. Existe a Houckel-brave, que é o da OMS. Existe o outro SWLS, que é mais reduzido em termos de qualidade de vida. O de auto estima não me estou a recordar do

nome. Estou aqui a pensar, são muitos testes e existem às vezes provas muito específicas de imagem corporal, existe 1001 coisas. Depois acho que isso é bom e pode ser um problema, há tanta informação e tantos questionários que por vezes já nem sabemos que eles existem ou não sei até que ponto, a menos que seja em áreas muito específicas, estão a ser utilizados. Existem escalas para tudo e mais alguma coisa.

Team: Pois, depende muito da área do paciente.

P03: Sim, e depois depende muito, acho que aí vai muito da experiência do próprio técnico, uma pessoa com mais experiência com determinadas questões vai compreendendo certas coisas, não precisa de recorrer a tantos testes. Com menos experiência acho que é importante recorrer aos testes. E depois depende se há esse protocolo de ter mesmo que utilizar esses testes ou não.

Team: Mesmo assim, estaria disposta mesmo assim a fazer os seus próprios testes ou preferia que já tivessem lá feitos na plataforma?

P03: Fazer não no sentido de criar testes novos ou é nesse sentido?

Team: É nesse sentido.

P03: Eu não queria fazer porque sei o quanto isso implica em termos de investigação, leva muito tempo a validar um teste para a população portuguesa, para aquela faixa etária em específico. Isso implica muita testagem, muito tempo e recursos que eu não tenho nem quero ter, honestamente. Mas seria bom ter acessível aquilo que já existe numa plataforma, criar os meus testes eu acho que por aí na parte da entrevista nós vamos tentando obter alguma informação adicional. Há testes que levam anos a serem validados. Então, já tive essa dose para a tese e nem foi para validar testes.

Informed consent for the usability tests

Informação ao Participante de Investigação e Consentimento Informado

---

Título do Estudo: Desenvolvimento de uma plataforma web de criação e gestão de formulários
Investigador(es) Principa(l/is): Nome: Alexandre Romão, Bruno Rodrigues, Eva Freitas
Instituto: Universidade da Madeira
E-mail: romaoalexandre10@gmail.com, bruno2000rodri@gmail.com, evaazevedofreitas@gmail.com
Tel: 964099411, 968191201, 967022825

Objetivo do Estudo
O objetivo deste estudo é inquirir sobre o protótipo de uma plataforma, que visa o processo de avaliação de pacientes por parte dos psicólogos, metodologia e eficiência, desenvolvido pelos investigadores.

Procedimento
O estudo de usabilidade destina-se a psicólogos. Vão ser realizados *user tests*, através do método *Think Aloud*, com o objetivo de recolher o máximo de informação conveniente necessária para os investigadores poderem prosseguir com o seu estudo e esclarecer dúvidas que os mesmos possam ter sobre práticas, processos e métodos de avaliação dos psicólogos aos seus pacientes. O método terá a duração de 20 a 30 minutos e após o seu término, os resultados são alvo de uma análise.

Critérios de Inclusão
Será considerado elegível para participar neste estudo se:

- Possuir ciclos de estudos e/ou experiência relevantes na área de Psicologia.

Confidencialidade
A confidencialidade dos dados será mantida das seguintes formas:
Todos os **dados pessoais** fornecidos na experiência serão guardados e não serão partilhados com terceiros. Contudo, os dados recolhidos durante a experiência poderão ser usados/publicados para fins científicos ou educativos. Os nomes de **TODOS** os participantes serão **OCULTADOS**, incluindo nos artigos científicos publicados.

Autorização Opcional
Entendo que os investigadores podem querer usar fotografias, vídeo ou áudio por razões ilustrativas nas apresentações e publicações deste trabalho, para fins científicos ou educativos. Eu dou autorização para fazê-lo, **DESDE** que o nome e rosto **NÃO** apareçam.

Assine no lugar pretendido: _____SIM _____NÃO

Direitos
A sua participação é voluntária. Você é livre de interromper a sua participação em qualquer momento. A recusa em participar ou interrupção da participação não resultará em qualquer penalização, ou perda de eventuais benefícios ou direitos. O investigador principal poderá decidir, de forma fundamentada, interromper a sua participação neste estudo. Caso se verifique esta situação, esta não resultará em qualquer penalização, ou perda de eventuais benefícios ou direitos.

Esclarecimento de Dúvidas & Contatos
Se você tem dúvidas sobre este estudo, poderá fazer agora todas as perguntas. Se quiser fazer perguntas mais tarde, desejar obter mais informações, ou desejar interromper a sua participação no estudo, entre

em contato com o Investigador Principal em pessoa, por telefone ou e-mail. A informação de contato está disponível no início da primeira página deste documento.

Consentimento Informado Voluntário
Ao assinar este documento, você confirma que leu a informação acima descrita sobre este estudo, e que todas as suas perguntas foram respondidas. Assim mesmo, você poderá fazer perguntas adicionais a qualquer momento durante o estudo, e mesmo após este ter terminado. Ao assinar este documento, você concorda em participar neste estudo de investigação. Irá receber uma cópia deste documento de consentimento informado assinada e datada.

_____ _____

ASSINATURA DO PARTICIPANTE DATA

Investigador que Obtém o Consentimento

Como membro da equipa de investigação, confirmo que expliquei ao participante acima referido a natureza e finalidade deste estudo de investigação, e que esclareci quais os potenciais benefícios e eventuais riscos da participação no estudo. Todas as perguntas foram respondidas e estou disponível para esclarecer quaisquer dúvidas que possam surgir ao longo do estudo.

_____ _____

ASSINATURA DO INVESTIGADOR 1 DATA

_____ _____

ASSINATURA DO INVESTIGADOR 2 DATA

_____ _____

ASSINATURA DO INVESTIGADOR 3 DATA

Ao assinar este documento, você confirma que leu a informação acima descrita sobre este estudo, e que todas as suas perguntas foram respondidas. Assim mesmo, você poderá fazer perguntas adicionais a qualquer momento durante o estudo, e mesmo após este ter terminado. Irá receber uma cópia deste documento de consentimento informado assinada e datada.

Guide with tasks for the usability tests

User testing

Bom dia/ Boa tarde (*nome psicólogo*), este estudo está sendo realizado por um grupo de 3 elementos: Alexandre Romão, Bruno Rodrigues e Eva Freitas; e antes de começarmos o exercício para o qual pedimos a sua presença gostaria de lhe agradecer por disponibilizar um pouco do seu tempo para realizar este estudo de domínio académico.

Para dar início ao nosso teste, preparei um pequeno resumo como contexto breve do que se trata o projeto. Para o desenvolvimento das nossas Teses de Mestrado, referentes aos cursos de Engenharia Informática e Design de media Interativos, orientadas pelo professor Engenheiro Luís Ferreira, iremos testar a facilidade e capacidade rápida de compreensão das funcionalidades e do design da plataforma digital Psyment. A plataforma Psyment tem como principal foco permitir com que o processo de avaliação dos profissionais de saúde, nomeadamente os psicólogos, seja mais fácil, rápido e eficaz. Dito isto, esta plataforma foi desenhada a pensar nas suas necessidades como utilizador e lhe irá dispor de funcionalidades como a realização, gestão, armazenamento de testes e de dados de saúde referentes aos pacientes. Isto tudo de forma interativa e eficiente.

O estudo de usabilidade da plataforma destina-se a psicólogos. A nossa plataforma vai ser testada pelo nosso público-alvo, com o objetivo de recolher o máximo de informação conveniente necessário para que nós, como investigadores, possamos lhe oferecer uma melhor experiência da nossa plataforma digital durante a sua utilização.

Dito isto, pedimos para que seja sincero/a ao avaliar as funcionalidades e organização do Psyment. Relembro que este teste não é para avaliar a sua performance, mas sim as funcionalidades da interface desenvolvida ao qual irá testar. Por favor, sinta-se à vontade para exprimir todos os seus pensamentos em voz alta, tanto sejam esses pensamentos negativos, positivos ou confusos. Isto irá nos permitir uma melhor compreensão do aproveitamento do utilizador ao manusear o nosso produto digital interativo. Gostaria também de salientar que ao longo da sua avaliação, se gostar de algo ou não justifique o porquê, indique algumas sugestões caso lhe seja relevante em ambas as interfaces e por fim não hesite em perguntar algo em caso de dúvida.

Tem alguma questão a fazer?

Se não tem nenhuma questão a fazer, então iremos dar início ao teste de usabilidade!

Tarefa 1

Primeiramente, qual é a sua primeira impressão desta interface?

Contexto

A/O (nome do utilizador) encontra-se na landing page do Psyment e pretende entender melhor sobre esta plataforma, desde os seus criadores até ao que esta tem para lhe oferecer. Isto para tirar as suas dúvidas referentes a esta interface e saber se a mesma satisfaz as suas necessidades antes de criar uma conta na mesma, tendo ainda como base as reviews de alguns utilizadores.

Você foi informado/a sobre a existência desta plataforma através da sugestão de outros colegas profissionais de saúde como também pela recomendação do seu estabelecimento de trabalho atual e decidiu experimentar.

Execução

1. Para dar início à sua experiência de navegação pelo Psyment você quer saber as promessas desta interface de modo a verificar se correspondem ao que procura. Indique onde estão representadas essas ofertas relativas à nossa proposta.

2. Agora deseja saber mais sobre a nossa equipa e o processo de desenvolvimento da plataforma. Onde deve carregar para verificar isso?

3. Posto isto, quer saber a opinião de outros profissionais de saúde que utilizam a plataforma?

4. Para o caso de querer obter mais informações, vai verificar os contactos que lhe disponibilizamos. Indique os contactos do Psyment.

Contexto

Agora pretende efetuar o registo/login na plataforma.

Execução

1. Imaginemos que quer realizar o registo na plataforma. Onde deve clicar?

1.1. Acha os dados de registo pertinentes? Deveriam ser adicionados ou removidos alguns dos dados?

2. Imaginemos que agora pretende efetuar o login na plataforma. Onde deve clicar?

2.1. Acha que deveria ser adicionado algum método de recuperação de palavra-passe ou outro dado essencial ao login?

Tarefa 2

Contexto

Encontra-se no menu **assessments.** Pretende criar o seu próprio teste (Cognitive Assessment).

Execução

1. Que informação está a visualizar nesta página?
    i. Na sua opinião, o objetivo desta página e sua funcionalidade são fáceis de compreender?
    b. Antes de começar a criar o seu teste deve nomeá-lo. Onde deve colocar o nome do seu teste?
    c. Agora deseja adicionar uma fórmula para obter o cálculo automático dos resultados desse teste para quando for aplicá-lo num paciente. Onde clica?
        i. Que informação está a visualizar nesta secção das fórmulas?
        ii. Que cálculos acha que iria utilizar mais quando quisesse calcular os resultados de um teste?

iii. Para aplicar as suas fórmulas deve colocar o id/nº de cada pergunta correspondente a essa fórmula. Sabe identificar onde se encontram esses id's/nº em cada pergunta?

d. Após fechar a janela das fórmulas, vai começar a criar os seus exercícios. Para isso, vai ter de selecionar o tipo de pergunta que quer aplicar. Onde clica?

   i. Acha que as ilustrações ao lado direito de cada tipo de pergunta ajudam-no a entender como as mesmas funcionam?

   ii. Decide que o seu primeiro exercício vai ser uma pergunta aberta. Selecione essa opção e adicione mais uma pergunta na Q1.

   iii. Após adicionar todas as suas perguntas neste exercício terá de colocar o valor relativo a esse mesmo exercício. Sabe indicar onde deve colocar esse valor? Como gostaria que estes valores fossem dados (percentagem, peso, qualitativo/quantitativo, etc).

e. Agora pretende adicionar mais um exercício, onde deve clicar?

   6.5.1. Na sua opinião, acha que os exercícios estão bem organizados? Concorda que estejam em secções separadas? Gostaria que houvesse a funcionalidade de fazer drag nos exercícios para organizá-los na lista de exercícios para ficar na ordem que deseja?

   6.5.1. Se tivesse de apagar o último exercício que adicionou, o que faria?

f. Para além da opção de pergunta aberta existem outras opções. Explore as outras opções de perguntas.

   i. Na opção linear scale, o que visualiza? Descreva o que vê.

   ii. Na opção drop down, o que visualiza? Descreva o que vê.

iii. Como a opção de desenho só está disponível na versão tablet, terá de usar a opção alternativa, multimédia. Aqui poderá adicionar imagens, vídeos e áudios. Para onde deverá arrastar a sua mídia no exercício? E para adicionar mais uma questão, caso queira mais do que uma pergunta e média no mesmo exercício?

g. Quando acabar de criar o teste deverá salvar o mesmo. Onde você deve carregar?

   i. Após clicar no botão "Boa! Entendido", concorda com o redirecionamento para a menu de **"My assessments"**? Ou preferia que esta transição fosse feita de outra forma?

Tarefa 3

Contexto

Está / Tem de ir ao menu **assessments**. Quer fazer algumas alterações nos exercícios do seu teste "My version of MMSE". Vai assim editar o teste "My version of MMSE".

Execução

1.Encontra-se na página de edição do teste que criou. O que visualiza?

1.1. Não achou a segunda pergunta pertinente, então decide retirá-la. Como faz para apagar?

1.2. Passa directamente para o último exercício fazendo scroll na página (neste caso será clicar no ecrã). Descreva o tipo de exercício que visualiza.

1.2. Pretende verificar se os cálculos das fórmulas deste teste estão corretos. Onde pode verificar isso?

1.2. Após todas as edições, vai guardar as alterações feitas.

6.1.1 O que visualiza? Avance para continuar a efetuar outras tarefas na plataforma.

Tarefa 4

Contexto

Menu assessments. Deve partilhar um teste criado na secção “My assessments”.

Execução

1. Agora que deseja partilhar um teste seu, onde deve clicar?
2. Agora, está a ver a sua lista de testes de avaliação cognitiva (testes criados por si).
    a. Acha que a lista de testes está bem definida?
    b. Os dados dos testes listados (título e data) são suficientes para identificá-los facilmente?
3. Selecione os testes que deseja partilhar, neste caso, o primeiro teste.
4. Partilhe o teste com o colega José#1234.
    a. Acha este tipo de ID adequado para identificar o seu colega?
    b. Caso não, como gostaria de poder selecionar o profissional? (ex: a partir de uma lista / pesquisa pelo nome).
5. Agora, defina o tempo de acesso do seu colega José ao seu teste. Selecione 2 meses. Sentiu alguma dificuldade?
6. Finalmente, confirme a partilha ao clicar em “Boa, entendido!”

Tarefa 5

Contexto

Menu **assessments**. Deseja aplicar o seu teste "My version of MMSE".

Execução

1. Para aplicá-lo o que deve fazer?
2. Descreva o que visualizou nesta página e indique o que deve fazer para poder prosseguir com a aplicação deste teste.

   1.1. Depois de colocar o nome do paciente ao qual irá realizar o seu teste, pode começar a sua avaliação.

3. Onde deve colocar as respostas do seu paciente às suas perguntas de cada exercício?
4. Onde coloca a classificação desse exercício?
   a. Entende com facilidade o que deve fazer nesta página? Modificaria alguma coisa?
5. Após completar todas as respostas e dar a classificação correspondente a este exercício, quer passar para o exercício seguinte. Como faz?
6. Descreva o que vê nesta nova página.
   a. Neste exercício o paciente terá de desenhar a imagem que se encontra abaixo no lado esquerdo. Para conseguir colocar esse desenho físico neste exercício digital, como faz?
7. Terminou a sua avaliação e pretende visualizar os resultados.

Contexto

Após a avaliação do/a paciente utilizando um dos seus testes, foi redirecionado/a diretamente para a ficha desse paciente para ver os resultados do mesmo.

Execução

1. Que informações consegue obter do paciente nesta página?

    1.1. Na sua opinião tem necessidade de acrescentar ou remover alguma coisa?

    1.2. Os dados relativos ao paciente estão completos ou adicionava mais algumas informações? Se sim, indique quais.

2. Agora quer visualizar o teste "My version of MMSE" efetuado ao paciente. Onde clica?

    3.1. Acha adequada a forma como os resultados do teste aparecem na página?

    3.2. Agora quer visualizar as respostas que o paciente deu nesse teste. Onde clica?

        3.2.1. Acha que a disposição e forma como as respostas aparecem é adequada?

        3.2.2. Se quiser editar as respostas do paciente onde carrega?

            3.2.2.1. Acha que se deve poder editar as respostas? Se sim, sob que condições?

3. Esta página está bem explícita para o seu propósito?
4. Após fazer todas as edições pretendidas, deve salvar as mudanças efetuadas. E assim abre uma página que lhe informa que essas alterações foram salvas com sucesso.
    a. Confirme essa afirmação e clique em "Boa! Entendido".

Tarefa 6

Contexto

Volte ao menu **assessments.** Agora pretende ver a lista de pacientes, visualizar os detalhes dos pacientes e os testes realizados. Além disso, ainda pretende visualizar as respostas dadas pelo paciente.

Execução

5. Para visualizar a lista de pacientes deve clicar onde?
    a. Acha que a disposição e a organização da lista dos pacientes é boa e adequada?
6. Tem um paciente novo para adicionar à sua lista de pacientes. Onde deve clicar para o poder adicionar?
    a. Concorda com os dados necessários para adicionar este novo paciente? Adicionava ou removia algum destes pontos?
    b. Pretende alterar o género do paciente de feminino para masculino. Como faz?
    c. Após preencher todos os campos necessários para criar a ficha do paciente, clique em "adicionar paciente".

    1.2. Pretende filtrar a ordem pela qual os pacientes aparecem no ecrã. Onde deve clicar?

        1.2.1. Os filtros são adequados?

        1.2.2. Tem algum filtro para adicionar?

7. Quer visualizar o perfil do paciente "Ema Maria Azevedo". Onde você carrega?

Contexto

Pretende comparar o resultado de dois testes de avaliação cognitiva aplicados por si.

Execução

1. Acha os dados sobre os testes listados (nome, data de aplicação e cotação) suficientes para caracterizar cada um dos testes?
2. De seguida, selecione os **2 primeiros testes da lista** e compare os seus resultados.
3. Que dados sobre os testes gostava de ver **referidos** na página de comparação?
    a. O que acha dos dados que vê atualmente? (Cotação e tempo)
4. Após isto, analise as respostas do paciente ao primeiro teste.
    a. Acha útil poder revisitar as respostas dadas às questões do teste?
    b. É válido ter a possibilidade de editar respostas aos testes psicológicos nestas circunstâncias?
5. A funcionalidade de comparação é satisfatória ou tem alguma sugestão? (Por exemplo o facto de poder ver os testes lado a lado para mais fácil análise).

Tarefa 7

Contexto

Regressa ao menu **assessments** e pretende editar um dos testes existentes na plataforma.

Execução

Para ir para a página de edição de testes existentes onde deve clicar?

1.1. Acha que a disposição e a forma como os testes aparecem é boa e adequada?

1.2. Olhou para a lista de testes e pretende editar o Mini-Mental State Examination. Onde deve clicar?

Tarefa 8

Contexto

Regressa ao menu principal e pretende aplicar um teste partilhado por outro profissional de saúde na plataforma.

Execução

Para ir para a página dos testes partilhados a si, onde deve clicar?

1.1. Que informações consegue ver nesta página? Acha que está bem organizada e explícita?

1.2. Pretende aplicar o "My version of BSI" partilhado por Carina#2424. Onde deve clicar?

Tarefa 9

Contexto

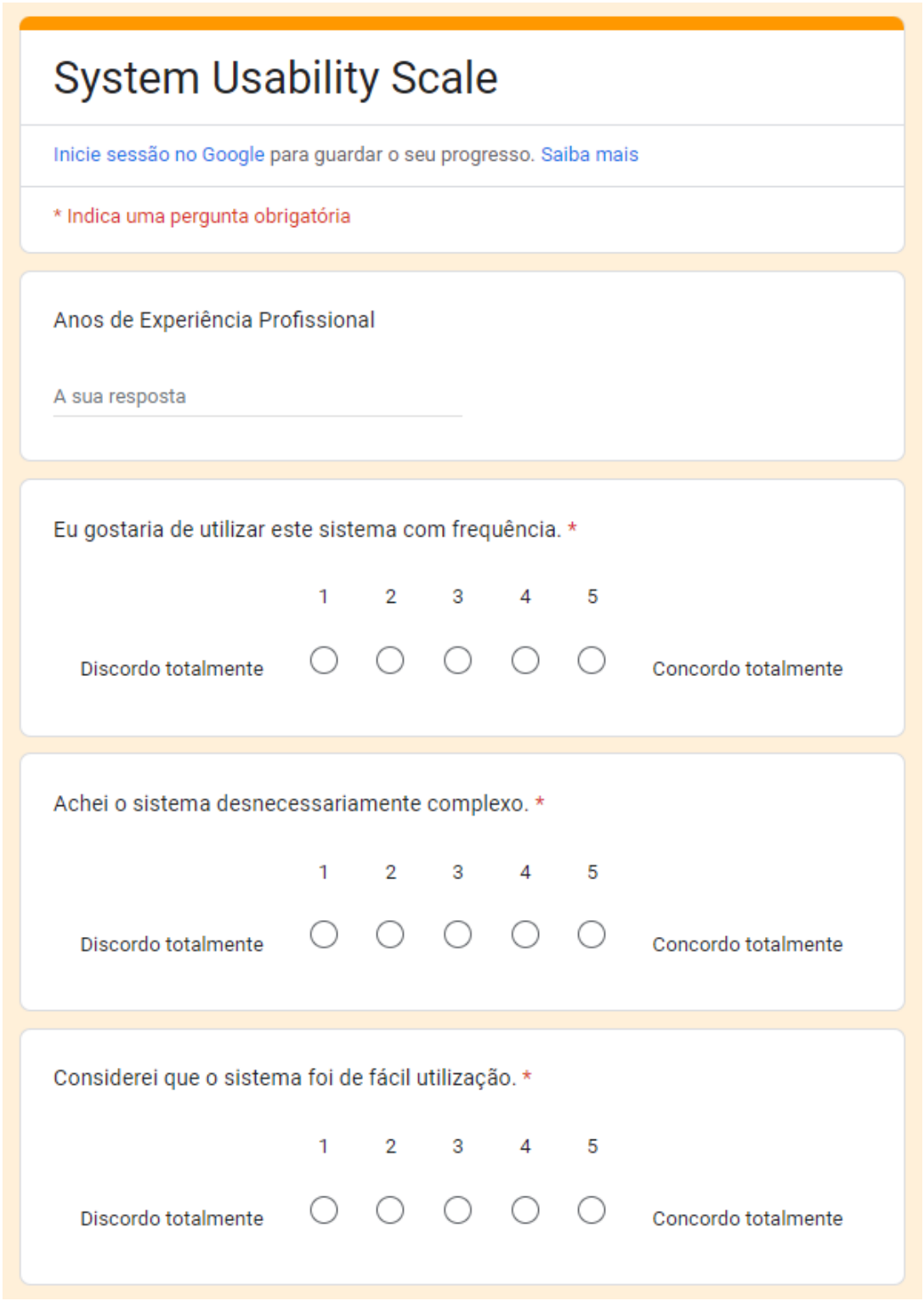
Encontra-se no menu **assessments**. Pretende alterar detalhes do perfil e depois cancelar a sua conta.

Execução

1. Navegue até à página do seu perfil.
2. Acha que os dados apresentados no perfil são úteis para definir um psicólogo? Falta algum dado? Cédula profissional é relevante?
3. Após isto, edite o seu perfil (clicar em **Edit profile**).
4. Depois guarde as alterações.
5. Agora, apague a sua conta através de um botão apresentado na aplicação.
    a. Este botão é claramente visível?
    b. ***Explicar que esta operação irá apagar os dados todos dos utilizadores e testes registados pelo profissional de saúde.***

6. Dado que esta é uma operação de grande importância, acha que deveria ser adicionada alguma medida adicional de confirmação? (ex: inserir palavra passe).
7. Confirme a operação feita e é redirecionado para a **landing page**.

System Usability Scale



System Usability Scale

Inicie sessão no Google para guardar o seu progresso. Saiba mais

* Indica uma pergunta obrigatória

Anos de Experiência Profissional

A sua resposta

Eu gostaria de utilizar este sistema com frequência. *

	1	2	3	4	5	
Discordo totalmente	○	○	○	○	○	Concordo totalmente

Achei o sistema desnecessariamente complexo. *

	1	2	3	4	5	
Discordo totalmente	○	○	○	○	○	Concordo totalmente

Considerei que o sistema foi de fácil utilização. *

	1	2	3	4	5	
Discordo totalmente	○	○	○	○	○	Concordo totalmente

Achei que precisaria da ajuda de uma pessoa técnica para conseguir utilizar o sistema. *

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

Achei que as várias funções deste sistema estavam bem integradas. *

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

Achei que haviam muita inconsistência na aplicação. *

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

Eu diria que a maioria das pessoas conseguiria aprender a utilizar este sistema muito rapidamente. *

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

Achei o sistema muito complicado de utilizar. *

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

Senti-me muito confiante na utilização deste sistema. *

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

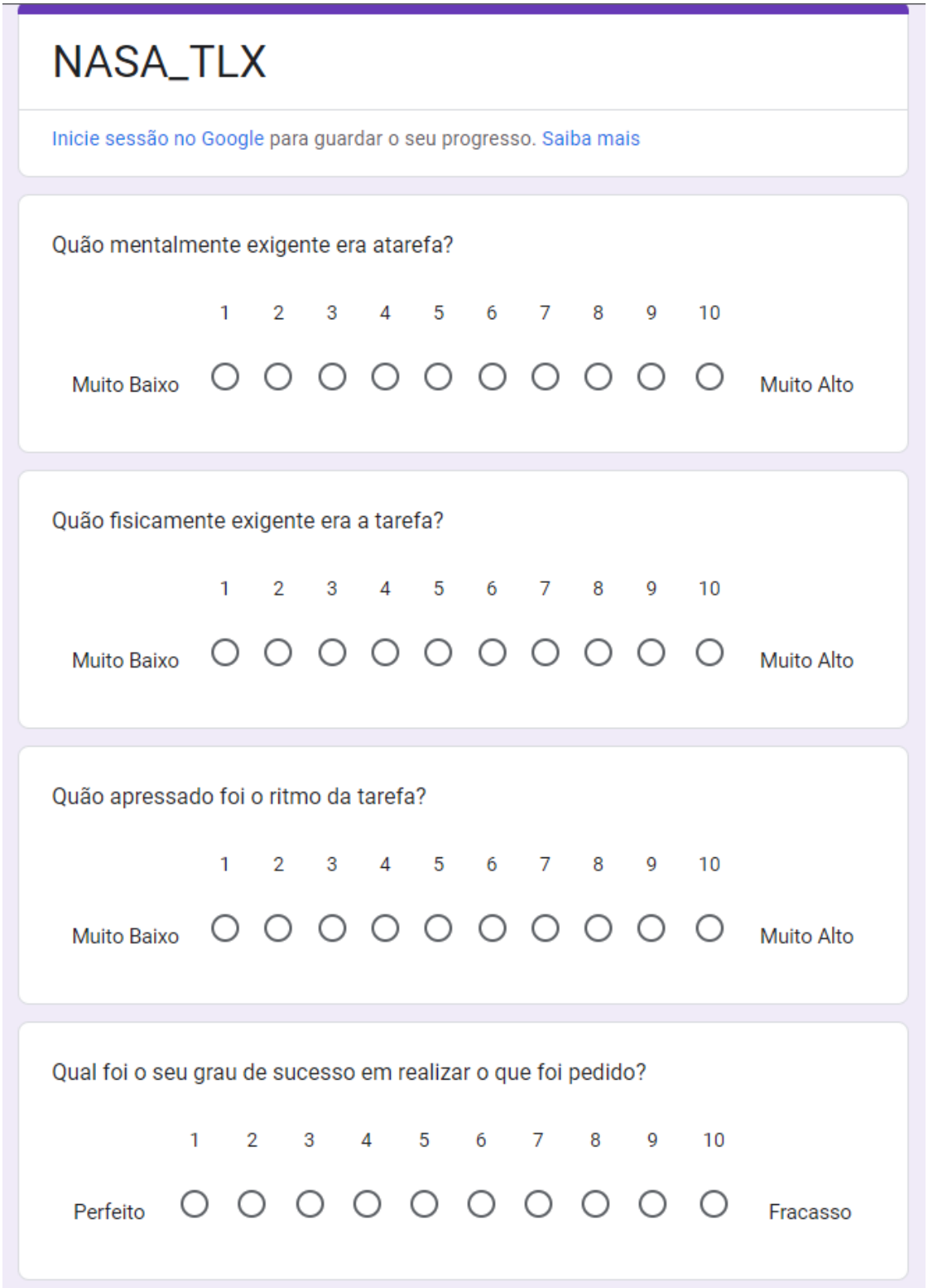
Precisava de aprender muitas coisas antes de continuar a utilizar este sistema. *

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

Enviar

Limpar formulário

NasaTLX



NASA_TLX

Inicie sessão no Google para guardar o seu progresso. Saiba mais

Quão mentalmente exigente era atarefa?

	1	2	3	4	5	6	7	8	9	10	
Muito Baixo	○	○	○	○	○	○	○	○	○	○	Muito Alto

Quão fisicamente exigente era a tarefa?

	1	2	3	4	5	6	7	8	9	10	
Muito Baixo	○	○	○	○	○	○	○	○	○	○	Muito Alto

Quão apressado foi o ritmo da tarefa?

	1	2	3	4	5	6	7	8	9	10	
Muito Baixo	○	○	○	○	○	○	○	○	○	○	Muito Alto

Qual foi o seu grau de sucesso em realizar o que foi pedido?

	1	2	3	4	5	6	7	8	9	10	
Perfeito	○	○	○	○	○	○	○	○	○	○	Fracasso

Quão duro teve que trabalhar para realizar seu nível de desempenho?

	1	2	3	4	5	6	7	8	9	10	
Muito Baixo	○	○	○	○	○	○	○	○	○	○	Muito Alto

Quão inseguro, desanimado, irritado, stressado e aborrecido estava?

	1	2	3	4	5	6	7	8	9	10	
Muito Baixo	○	○	○	○	○	○	○	○	○	○	Muito Alto

Enviar

Limpar formulário

Intrinsic Motivation Invetory (IMI)

Intrinsic Motivation Inventory (IMI)

(Traduzido) O Inventário de Motivação Intrínseca (IMI) é um dispositivo de medição multidimensional destinado a avaliar a experiência subjetiva dos participantes relacionada a uma atividade alvo em experimentos de laboratório.
Este estudo

ACTIVE PERCEPTION QUESTIONNAIRE

Os seguintes itens referem-se à sua experiência com as tarefas. Por favor, responda a todos os itens. Para cada item, indique quão verdadeira é a afirmação para si, usando a seguinte escala como guia:

Inicie sessão no Google para guardar o seu progresso. Saiba mais

1	2	3	4	5	6	7
○	○	○	○	○	○	○
Não é verdade de todo			Ligeiramente verdade			Completamente verdade

Anos de experiência

A sua resposta

Id

A sua resposta

1. Acredito que realizar esta atividade pode ter algum valor para mim.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

2. Acredito que tive alguma escolha em fazer esta atividade.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

3. Enquanto estava a fazer esta atividade, estava a pensar o quanto gostei dela.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

4. Acredito que fazer esta atividade é útil para melhorar a concentração.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

5. Esta atividade foi divertida de fazer.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

6. Acho que esta atividade é importante para a minha evolução.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

7. Gostei muito de fazer esta atividade.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

8. Não tive escolha sobre fazer esta atividade.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

9. Fiz esta atividade porque quis.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

10. Acho que esta é uma atividade importante.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

11. Senti que estava a gostar da atividade enquanto a estava a fazer.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

12. Achei que esta era uma atividade muito aborrecida.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

13. É possível que esta atividade possa melhorar os meus hábitos de estudo.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

14. Senti que não tinha outra escolha a não ser fazer esta atividade.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

15. Achei que esta era uma atividade muito interessante.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

16. Estou disposto(a) a fazer esta atividade novamente, porque acho que é útil.

1 2 3 4 5 6 7

17. Descreveria esta atividade como muito agradável.

1 2 3 4 5 6 7

18. Senti que tinha que fazer esta atividade.

1 2 3 4 5 6 7

19. Acredito que fazer esta atividade pode ser algo benéfico para mim.

1 2 3 4 5 6 7

20. Fiz esta atividade porque tive que o fazer.

1 2 3 4 5 6 7

21. Acredito que fazer esta atividade pode ajudar-me a ter melhores resultados na escola.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

22. Enquanto fazia esta atividade, senti que tinha escolha.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

23. Descreveria esta atividade como muito divertida.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

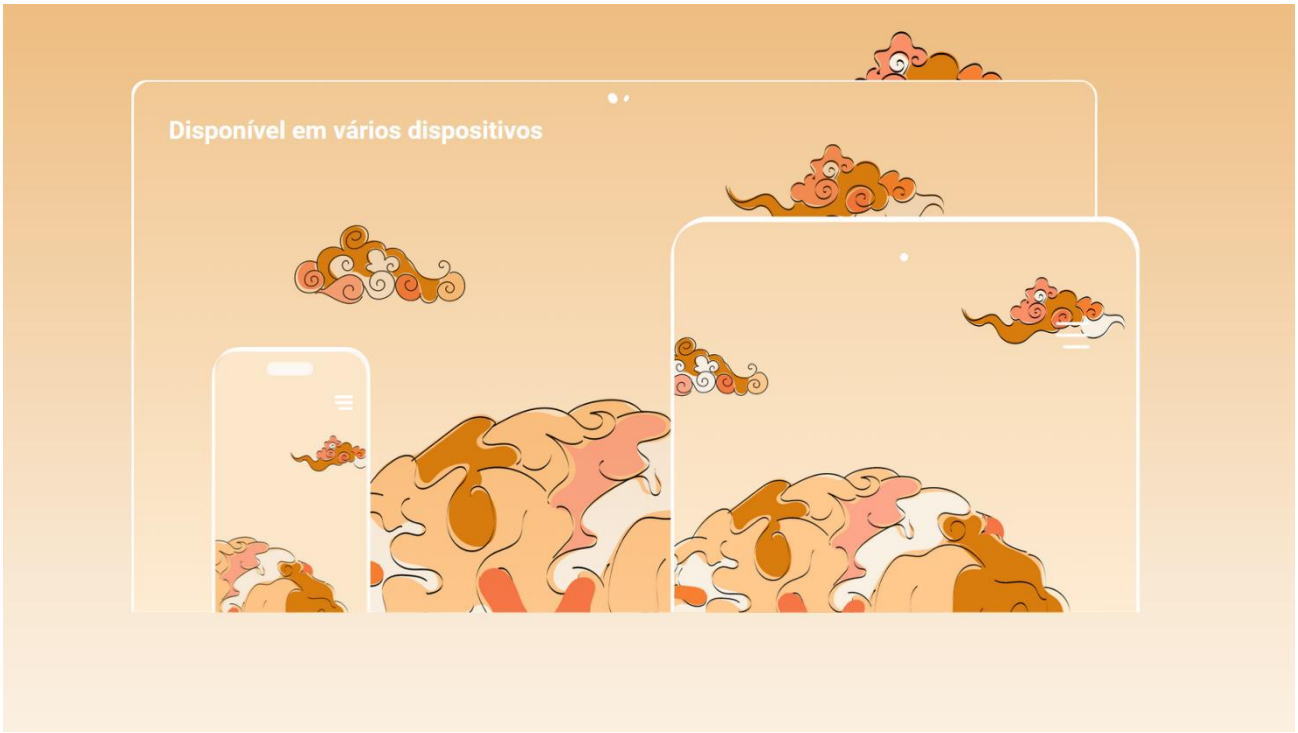
24. Senti que não foi minha escolha fazer esta atividade.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

25. Estaria disposto(a) a fazer esta atividade novamente, porque tem algum valor para mim.

1	2	3	4	5	6	7
○	○	○	○	○	○	○

Final application interfaces



entrar novo aqui?

Faça o seu registo

username

nome

profissão

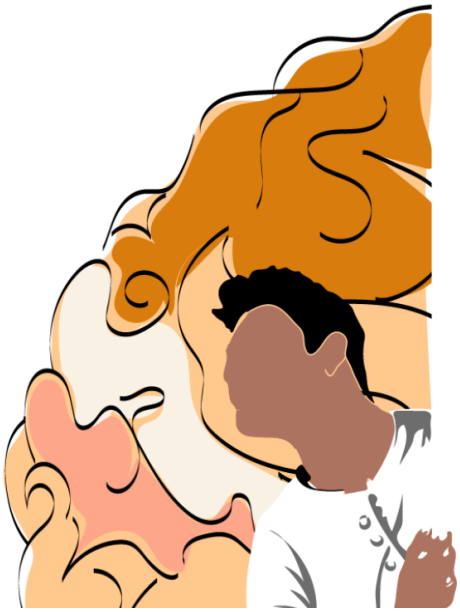
local de trabalho

email

palavra-passe

confirmar palavra-passe

criar a minha conta



entrar novo aqui?

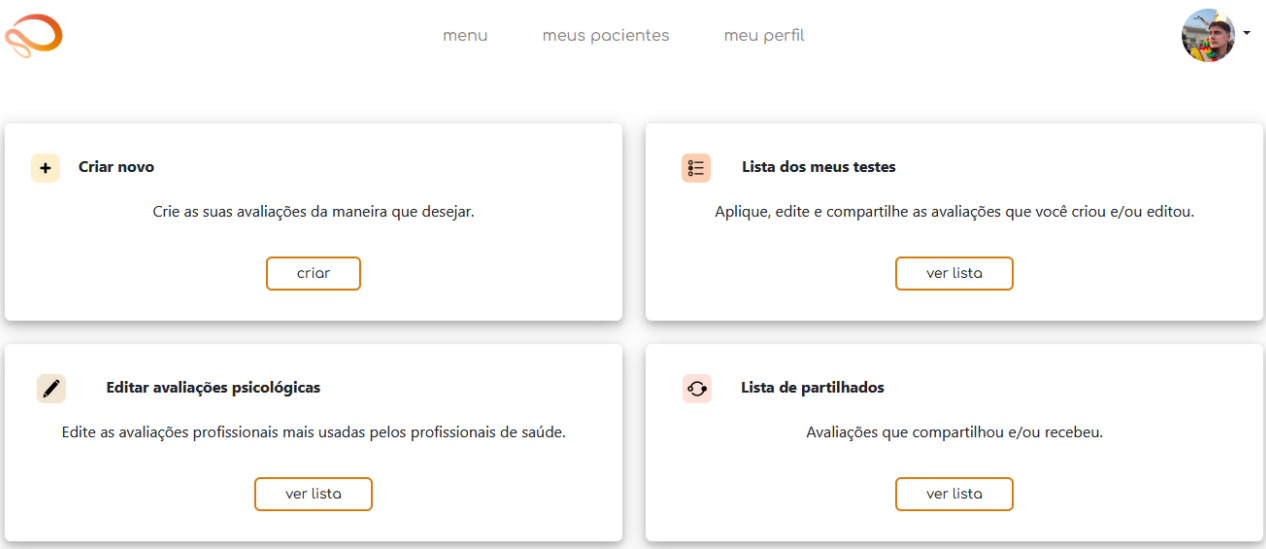
Bem-vindo de volta!

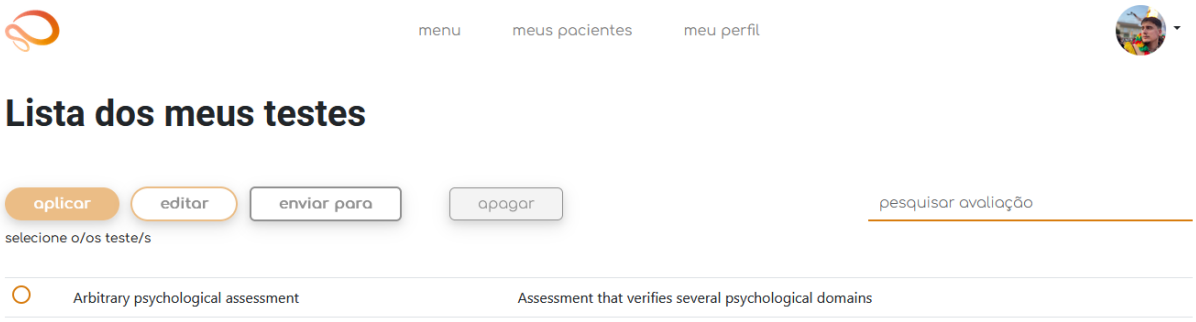
email ou nome de utilizador

palavra-passe

entrar na minha conta

esqueceu-se da palavra-passe?


menu
meus pacientes
meu perfil
Criar novo
Crie as suas avaliações da maneira que desejar.
criar
Lista dos meus testes
Aplique, edite e compartilhe as avaliações que você criou e/ou editou.
ver lista
Editar avaliações psicológicas
Edite as avaliações profissionais mais usadas pelos profissionais de saúde.
ver lista
Lista de partilhados
Avaliações que compartilhou e/ou recebeu.
ver lista


menu
meus pacientes
meu perfil
Lista dos meus testes
aplicar
editar
enviar para
apagar
pesquisar avaliação
selecione o/os teste/s
Arbitrary psychological assessment
Assessment that verifies several psychological domains

menu meus pacientes meu perfil

Crie aqui o seu teste!

Nome do teste

Breve descrição do seu teste

1. Nome do Domínio

1.

Resposta Aberta

Escreva a sua Questão

Explorar... Nenhum ficheiro selecionado.

pontuação

adicionar questão

Fórmula do Domínio (Ex: (Q1 + Q2) - Q3)

Pontuação Total do Domínio: 0

apagar domínio

adicionar domínio

Fórmula do Teste

Pontuação Total do Teste: 0

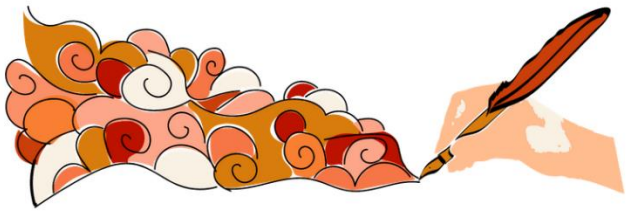
Ex: D1 + D2 - D3

salvar teste

menu meus pacientes meu perfil

Terminou a sua avaliação com sucesso

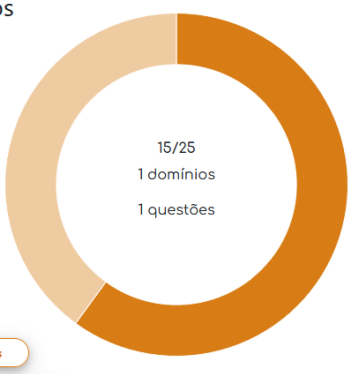
ver resultados



menu meus pacientes meu perfil

Arbitrary psychological assessment

Resultados



ver respostas

partilhar resultados

Mudar Gráfico

menu meus pacientes meu perfil

Perfil

Alexandre Romao

Psicologo

alexandreromao@gmail.com

Trabalha em Casa de Saúde Sao Joao de Deus

editar perfil