UNIVERSIDADE DE COIMBRA

Gui Botelho Silva Costa

# PLATAFORMA DE ESTÁGIOS @DEI V2

**Dissertation in the context of the master's in informatics engineering, specialization in Software Engineering, advised by Professor Nuno Laranjeiro and Professor César Teixeira and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.**

September of 2023

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Gui Botelho Silva Costa

# Plataforma de Estágios @DEI v2

Dissertation in the context of the Master in Informatics Engineering, specialisation in Software Engineering, advised by Professor Nuno Laranjeiro and Professor César Teixeira and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September 2023

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Gui Botelho Silva Costa

# Plataforma de Estágios @DEI v2

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de software, orientada pelo Professor Doutor Nuno Laranjeiro e pelo Professor Doutor César Teixeira e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro 2023

# Acknowledgements

I would like to thank thesis advisors, Professors Nuno Laranjeiro and Professor Cesar Teixeira for their support and guidance. As well as the department and the university for providing me with such a valuable opportunity with this internship. Finally, a big thanks to all my friends and family who, throughout this process showed immense support and care.

# Abstract

The main goal of this thesis is to develop a new internship platform to address the limitations and lack of automation of the previous platform. The development is based on the use of software development practices such as stakeholder specifications, functional requirement specifications and user stories. The stakeholders were classified as pertaining to a role in the platform or not. The functional requirements were prioritized using the MoSCoW method. The thesis also explores the architectural design of the platform, mainly using the c4 model and its three first levels (Context, Containers and Components). It talks about why certain technologies were chosen along with their benefits and their drawbacks. There is an emphasis on the adoption of Laravel for frontend and backend development and its relationship with the architecture chosen. During the development process a decision to restructure the database was made, and that involved the use of an Entity Relationship Diagram (ERD) which helped resolve some of the issues of the old platform. There were some challenges and setbacks during development mainly due to time constraints and technical issues, but despite these, some features were successfully implemented which is also discussed further in the document. Informal testing was also done by leveraging feedback from peers and faculty members, alongside basic security assessments made use of OWASP ZAP. These tools provided valuable insights for refinement and further improvement of the platform.

# Keywords

Internship management, Internship application, Web Application, Software development, Project planning.

# Resumo

O principal objetivo desta tese é desenvolver uma nova plataforma de estágios para abordar as limitações e a falta de automatização da plataforma anterior. O desenvolvimento é baseado no uso de práticas de desenvolvimento de software, como especificações de stakeholders, especificações de requisitos funcionais e user stories. Os stakeholders foram classificadas como tendo ou nao um papel na plataforma. Os requisitos funcionais foram priorizados usando o método MoSCoW. A tese também explora a arquitetura da plataforma, principalmente usando o modelo c4 e os seus três primeiros níveis (Context, Containers e Components). A tese fala sobre o porque de determinadas tecnologias terem sido escolhidas, juntamente com os seus benefícios e as suas desvantagens, havendo um ênfase na adoção do Laravel para desenvolvimento frontend e backend e sua relação com a arquitetura escolhida. Durante o processo de desenvolvimento, foi tomada a decisão de reestruturar a base de dados, o que envolveu o uso de um Diagrama de Relacionamento de Entidades (ERD), que ajudou a resolver alguns dos problemas da plataforma antiga. Houve alguns desafios e contratempos durante o desenvolvimento, principalmente devido a restrições de tempo e problemas técnicos, mas, apesar desses obstáculos, algumas funcionalidades foram implementadas com sucesso, o que também é discutido mais detalhadamente no documento. Foram também realizados testes informais, aproveitando o feedback de colegas e docentes, juntamente com avaliações básicas de segurança, fazendo uso do OWASP ZAP. Essas ferramentas forneceram informações valiosas para aprimorar e melhorar ainda mais a plataforma.

# Palavras-Chave

Gestão de estágios, Candidatura a estágios, Aplicação Web, Desenvolvimento de software, Planeamento de projetos.

# Contents

# Acronyms

**CSRF** Cross-site request forgery.

**DEI** Department of Informatics Engineering.

**ERD** Entity Relationship Diagram.

**FR** Functional Requirement.

**HTTPS** Hypertext Transfer Protocol Secure.

**MVC** Model-View-Controller.

**QoL** Quality of Life.

**XSS** Cross-site scripting.

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The present thesis was developed in the context of an Internship in a Software Engineering branch of the Master's degree in Computer Engineering at the Faculty of Sciences and Technology of the University of Coimbra, in the Department of Informatics Engineering (DEI).

The purpose of this thesis is the development of a new platform for internships, adding the missing features and overcoming the limitations of the existing Internship platform being developed by the University of Coimbra. The platform's objective is to manage the process of proposal, placement of students and realisation of thesis and internships for the Masters of the DEI. It should be noted that this document was started after the initiation of the project's development, thus it may not fully capture the entirety of the underlying process. Nevertheless, the aim is to provide a sufficient degree of documentation such that the process can be considered well-documented and can be followed up on as necessary. Additionally, this document will also detail the process undertaken by the author in regard to their own internship.

## 1.1 Objectives

In today's digital age, it's of paramount importance to keep the design of this platform modern and user-friendly. This is because the platform's design and usability will determine how easy it is for the different stakeholders to interact with it. A modern design can help ensure that the platform is easy to use and navigate for students, businesses, and teachers alike. This will make it easier for them to find the information they need and complete their tasks quickly and efficiently. The platform should be developed with best practices, to make it easy to use and navigate on both desktop and mobile devices. The platform should be accessible to all stakeholders and should be able to adapt to different screens, devices, and browsers. This will enable students and businesses to access the platform from any device, at any time, which can greatly improve the overall experience and engagement with the platform. Additionally, having a modern design can make the platform more attractive and visually appealing, which can encourage more people to use it and potentially increase the adoption of the platform.

The internship platform in question is a tool that is designed to facilitate the process of completing internships for students at the University of Coimbra. Its objectives include the following:

- Facilitate the process relating to internships for students, faculty members, and coordinators at the University of Coimbra as well as businesses looking to propose internships by improving the overall experience of the platform and by providing a centralised platform, developed with the best practices, for all stakeholders, such that key can fulfil their needs all in one platform;
- Develop the missing features in the Internship platform that were lacking or improperly developed in the current version of the platform (such as the ability to see meeting summaries);
- To automate some features of the internship platform in order to increase efficiency for all the involved stakeholders (Such as automated student selection).

## 1.2 Structure

In the following section, the structure of the document will be presented in detail. This document is organised into eight chapters, each of which covers a specific aspect of the internship platform and its development.

- **Chapter 2** provides a state-of-the-art analysis of the current platform, including its limitations, and tries to review its weaknesses or shortcomings. Additionally, the competing platforms are also mentioned;
- **Chapter 3** outlines the methodology employed during the development process, including the adaptation of the Scrum framework. This chapter explains the approach that was taken to develop the platform, including the specific processes and techniques that were used. **Section 3.2** of the chapter covers the planning aspect of the development process. It includes two sections, one detailing the plan of work for the first semester, and the other detailing the plan of work for the second semester. In this chapter, the authors present the project schedule and timelines, including Gantt charts that illustrate the tasks and milestones for each semester.;
- **Chapter 4** presents the results of the initial process and discusses the features and functionalities that the final product should have.
- **Chapter 5** provides details on the technology used. This chapter includes a description of the specific tools, technologies, and frameworks that were used to develop the platform. It also provides details on the architecture used.

- **Chapter 6** Talks about the development process of the platform along with its organisation. This chapter also talks about the tests performed.
- **Chapter 7** concludes the document with a summary of the key findings and contributions of the project. This chapter includes a summary of the main points covered in the preceding chapters and discusses the significance of the project's results.

# Chapter 2

# State of the art

In this state-of-the-art, an examination of the current internship platform currently used by the Department of Informatics Engineering (DEI) is conducted. The capabilities and features of the platform, including the ability for businesses, students, faculty members, and other stakeholders to manage various aspects of the internship process, are thoroughly examined. Additionally, the limitations and shortcomings of the current platform are also examined, providing an understanding of the current system and the need for the development of a new system. Each stakeholder, including businesses, students, faculty members, and other stakeholders (refer to Table 4.1, detailed in chapter 4 for a list), is thoroughly examined to gain an understanding of what the current platform does or doesn't do. Some of the issues mentioned in this section had been resolved in a newer version however some still persisted which we intend to resolve and improve upon in further iterations.

When it comes to a state-of-the-art analysis it is often customary to study and talk about competing products or technologies involved in the subject of the project, however, due to the nature of this project, obtaining access to such tools was a rather difficult thing to do. These tools would often be developed in-house and closed-source by other universities or entities, as such, it seemed rather difficult that access to the inner workings of these tools would be obtained. Thus, what could have been accessed would at most be the forward-facing features such as the student or company components but not the total of what is needed to properly study and describe them in this section, such as the administration and coordination parts of the tools. However, since the main point of this project is not to develop entirely new features but instead improve Quality of Life (QoL) (such as the lack of automation for some features) and end the limitations of the current platform, the problem of not having access to competing platforms is reduced.

The current internship platform is composed of three distinct sub-platforms, each of which is assigned to a different set of tasks, this might seem non-problematic but is however an issue as it requires higher maintenance effort and lowers QoL for the users if they must use different sides of the platform. It also makes it harder to maintain a consistent look and feel through the different sub-platforms, this is exacerbated by the fact that they were developed using different languages and tools. The platforms are split amongst the student sub-platform, the coordination sub-platform and the companies sub-platform.

## 2.1 Current platform & its limitations

There are some issues that do not apply to a single or specific stakeholder, thus, a few stakeholder-wide problems will be explored and revealed. The main one is the existence of multiple platforms, which require users to change between platforms if they want to switch profiles or execute different functions such as faculty members versus coordinator and administration. This certainly causes frustration as users are forced to login on to separate platforms, in today's day and age we must ensure a better user experience by reducing these factors as they can, over time, become cumbersome to the users, especially the ones that must use the platform on a regular basis.

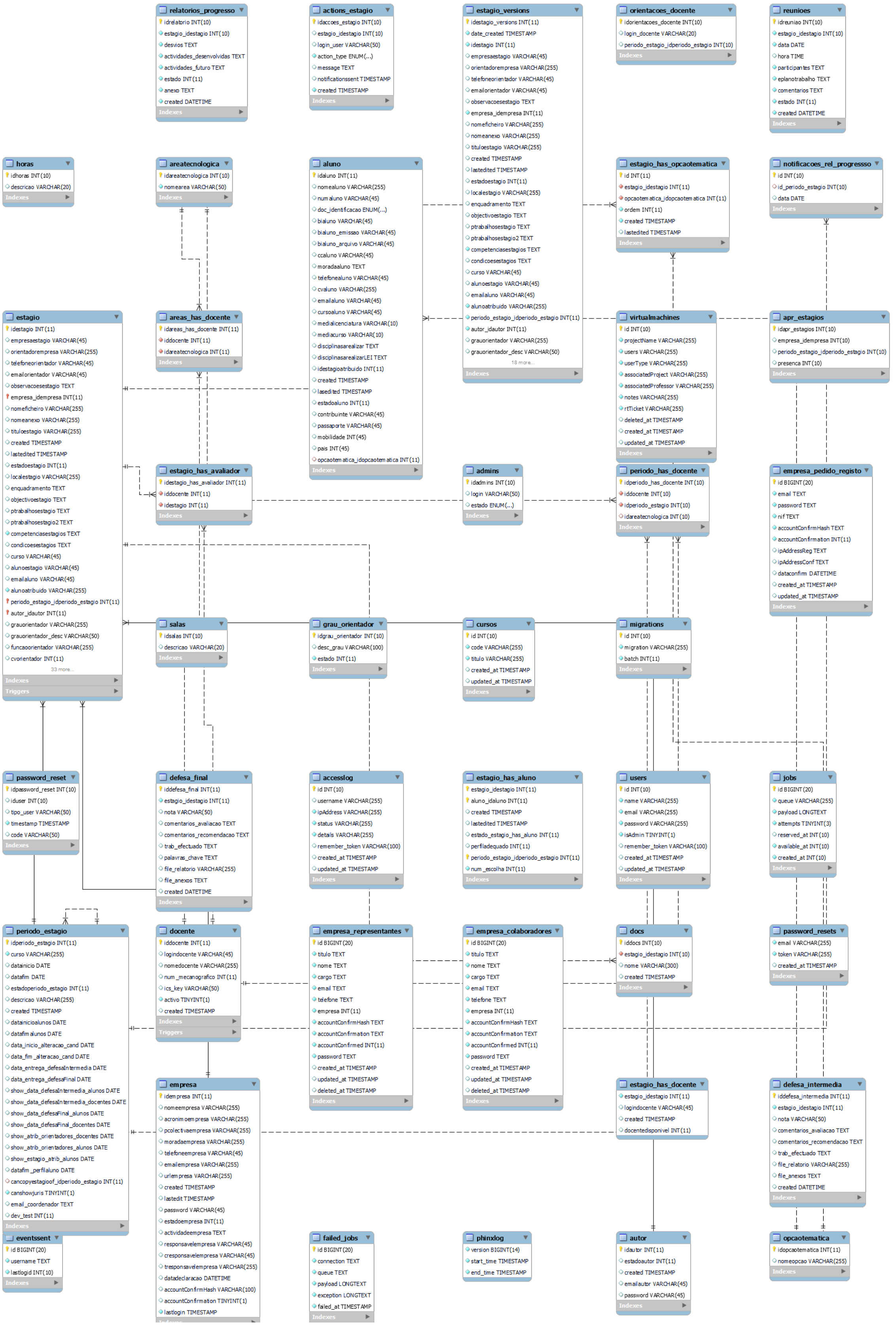
Two of the issues of the current platform is its outdated look and feel and its outdated backend too, such as the old database which is very fragmented. As it stands the database has 50 tables and about ten of them are redundant or no longer used as well as some of those have similar names with differently capitalised letters such as "eventsSent" and "eventssent" which do the exact same thing, with the same columns but one is not used, which makes maintenance on the platform extremely difficult even if there is documentation.

Another issue with the database is that some tables have fields that should either belong in an associative table or be entirely removed, for perspective, there is a table that contains 63 columns where over half of them should belong to another table as only the first should belong in the aforementioned table. Furthermore, there are some tables with columns that are repeated or haven't been used since previous revisions of the platform, often serving similar, if not the same, function as the new columns, this drastically increases the complexity of the old database and makes it harder to manage.

One major issue with the old database is that in some tables, tables that were used to store temporary credentials were stored in plaintext. This means that no encryption was used, this is a major breach of security for modern standards and must be addressed as it "may result in a system compromise" [OWASP]. Another point is that passwords are sent in plaintext but this is industry standard if we can ensure that the Hypertext Transfer Protocol Secure (HTTPS) protocol is used.

The Entity Relationship Diagram (ERD) shown in Figure 2.1 was obtained using the reverse engineering function of the MySQL workbench. This diagram shows the platform's old database and it is possible to see the issues mentioned above as well as other issues such as tables that should have relations to other tables but do not (lack of the appropriate foreign keys) such as table "*estagio_has_aluno*".

Figure 2.1: The old database's ERD



One issue pertaining to the advisor's role is that sometimes it would not be possible to consult the summaries and information submitted to the platform relative to the scheduled meetings, requiring them to ask IT personnel to retrieve this information.

### 2.1.1 Student

Although for the student the current platform does everything it needs to, there are some steps and some issues that can be easily identified by simply looking at the platform, such that there is a good amount of room for improvement. As such we will review what it does, what requires improvement and what it could do.

Firstly, the platform allows students to log in and review their data, this includes actual personal data such as their Full name; Student Number; Identification Number; Fiscal identification number; Curriculum Vitae; Email; Phone number; And lastly, address. Furthermore, the application asks students to fill in details of their academic course such as the course that the student graduated in, their average and the speciality of their current master's, the subjects that the student will be doing alongside their master's both in the 1st semester and second semester as well as any potential subjects that they might be doing from the preceding course.

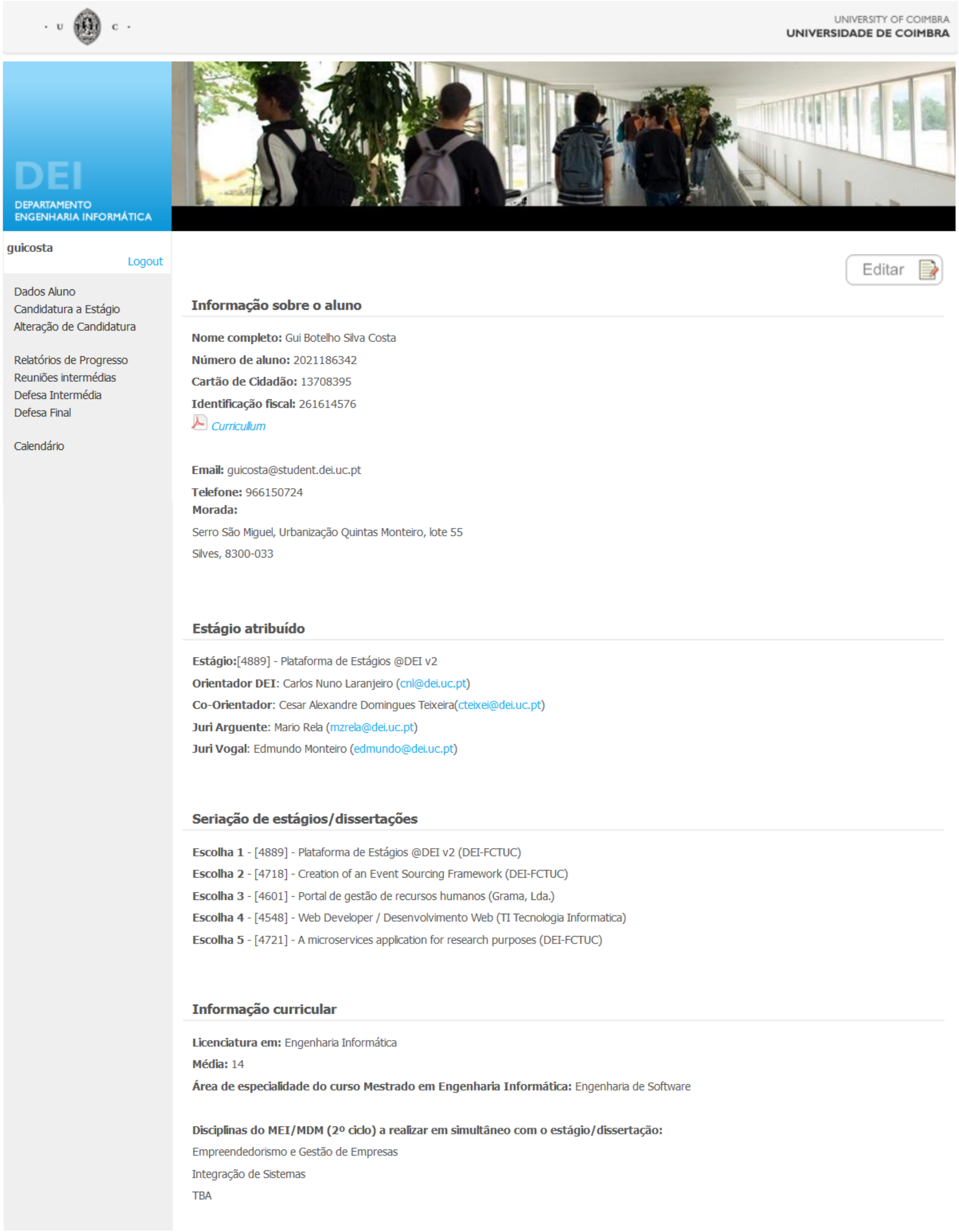
Secondly it allows students to review information regarding their internship if they have already been assigned one, such as the official title of the internship, current advisor and co-advisors and both participating Juris. Juris are stakeholders who are responsible for evaluating the student's performance, see Table 4.1. Moreover, the student is also able to see which internships they applied to. See Figure 2.2.

Thirdly, the platform allows students to review all the available internship proposals gathered, that have been accepted by the coordination or administration, and from that list, select 5 of the proposals. Those five proposals can also then be ordered by preference, the first being the most preferred and the last one the least preferred. Here we can start pointing out the first deficiencies, which go as follows:

- **The number of clicks** the student must do to reach the home page to the list of proposals is higher than it should be, in total if the student is on the home page a total of 4 clicks must be done before reaching the current list of proposals. If the student needs to switch between selection proposals and looking at the list this number increases even further. Additionally, the menus and links of the current platform are not quite obvious to navigate and therefore might lead to a higher-than-needed time to reach the destination page.

- **The differing platforms** force the students to perform a login on a site that is technically not the same as the one they use to check the proposals. Ideally, this would be joined in one platform or even on a single page such that the student could view and select the proposals as they go, in order to minimise the load on the student.

Figure 2.2: A demonstration of the current platform's personal data screen for students



**Informação sobre o aluno**

**Nome completo:** Gui Botelho Silva Costa
**Número de aluno:** 2021186342
**Cartão de Cidadão:** 13708395
**Identificação fiscal:** 261614576
*Curricullum*

**Email:** guicosta@student.dei.uc.pt
**Telefone:** 966150724
**Morada:**
Serro São Miguel, Urbanização Quintas Monteiro, lote 55
Silves, 8300-033

**Estágio atribuído**

**Estágio:**[4889] - Plataforma de Estágios @DEI v2
**Orientador DEI**: Carlos Nuno Laranjeiro (cnl@dei.uc.pt)
**Co-Orientador**: Cesar Alexandre Domingues Teixeira(cteixei@dei.uc.pt)
**Juri Arguente**: Mario Rela (mzrela@dei.uc.pt)
**Juri Vogal**: Edmundo Monteiro (edmundo@dei.uc.pt)

**Seriação de estágios/dissertações**

**Escolha 1** - [4889] - Plataforma de Estágios @DEI v2 (DEI-FCTUC)
**Escolha 2** - [4718] - Creation of an Event Sourcing Framework (DEI-FCTUC)
**Escolha 3** - [4601] - Portal de gestão de recursos humanos (Grama, Lda.)
**Escolha 4** - [4548] - Web Developer / Desenvolvimento Web (TI Tecnologia Informatica)
**Escolha 5** - [4721] - A microservices application for research purposes (DEI-FCTUC)

**Informação curricular**

**Licenciatura em:** Engenharia Informática
**Média:** 14
**Área de especialidade do curso Mestrado em Engenharia Informática:** Engenharia de Software

**Disciplinas do MEI/MDM (2º ciclo) a realizar em simultâneo com o estágio/dissertação:**
Empreendedorismo e Gestão de Empresas
Integração de Sistemas
TBA

- **Changing selected proposals** are in separate pages, considering the problem above, this issue need not exist.

Third and fourthly the platform allows students to submit documents related to the internship such that they can report the progress of their work and deliver what must be delivered. Additionally, it also allows the students to see a calendar of the deadlines and milestones, although, again, on a different platform, which is a thing that must be addressed.

One of the missing automation features relates to the student as well as the student selection process. Currently, this is done manually by other stakeholders, however, it need not be manual as a simple algorithm can be used to determine the student's placements. The proposed algorithm can be seen in chapter 4, subsection 4.3.7.

### 2.1.2 Administration & Coordinator

As a coordinator, one of the tasks that needs to be done is to assign students to the proposals that have been submitted and accepted. This process in the old platform was done via a table that showed all the students that were candidates for a given proposal as well as their other candidacies and their importance level (how they were ordered, 1st means the one the student wanted the most and 5th meant the one the student wanted the least). This table often proved to be "glitchy" such as having empty squares or showing information wrongly. This caused coordinators to have to figure out what was the supposed information and work around it. The new platform aims to improve how this data is displayed as well as fix these issues.

Another issue for the coordinator side is that when assigning juries, a process that happens after a student has been assigned, the coordinator has the select a list of internship periods to obtain information from. This information allows the coordinator to see what workload that specific jury has, which is useful because it improves the ability of the coordinator to distribute the juries through the proposals. However, the problem with this is that there should be no real need to select what internship periods they would want to see from because, realistically, only the active ones matter, even worse, every time a change is made they are forced to re-select everything from said list, leading to a disproportional amount of time and effort needed.

Furthermore, the layout of this information makes it so it's harder to see what workloads the potential juries have forcing coordinators to find workarounds such as copying and pasting this information on an Excel spreadsheet. The new platform aims to reduce the workload of the coordinator by providing a simpler way of obtaining the jury list and their workloads as well as improving how this information is displayed, reducing the time and effort needed.

On the administration side, when it comes to managing internship periods the interface is often confusing and prone to errors, specifically due to ambiguous fields and extra amounts of clicks to reach certain features. The new platform aims to reduce the amount of information by automating some of those fields and making it so the remaining fields are clearer and pre-filled with useful dates.

A combined issue of both the administration and coordination side, that has an impact on the users is that the way the platform is built forces the coordination and administration to often interact over external means, such as email. These cases occur, more specifically, when it comes to protocol signing and warning users. These warnings often refer to proposal reviews or actions that need to be done before certain dates, such as validating or interviewing candidates. The new platform aims to solve these issues by sending some of these emails automatically when a certain action is done or a certain date is approaching.

# Chapter 3

# Methodology & Planning

This chapter is split into two sections. The first, section 3.1 presents a detailed overview of the methodology used for the development of this project which is followed by a second section, section 3.2, which presents the planning for the project, which used the presented methodology.

## 3.1 Methodology

For this project, we employed a methodology of timeboxing that involved weekly meetings. The number of meetings was adjustable based on the needs of the project. When timeboxing, each task or activity is set a date or unit of time when it must be completed. Timeboxing is a technique that has a variety of uses, such as in project management and personal time management, for this, the schedule is divided into timeboxes according to certain constraints. Usually, these constraints are time, cost and scope. For this project, the most relevant constraints were scope and time [tim].

Timeboxing is successful in promoting iterative development and regular evaluation because, when a timebox reaches the end, the stakeholders, have an opportunity to meet and review the work achieved. This allows flexibility as it enables the adjustments or refinements of future timeboxes. This ensures that the platform aligns closely with the requirements (that may change over time) and the project's expectations [tim]. Most commonly, more important tasks with timeboxing tend to be the first ones to be approached, thus, this methodology aligns well with the chosen prioritization method *"MoSCoW"*, which is further detailed in chapter 4. This is very useful to promote transparency as it makes it very clear what is expected at each timebox, creating clear deadlines.

## 3.2 Planning

In this chapter, we will discuss the planning and scheduling process that was undertaken during the development of the internship platform. Planning is an essential part of any software development project, as it helps to ensure that resources are allocated effectively and that the project stays on track. The chapter is divided into two sections, the first focuses on the plan of work for the first semester, and the second one delves into the plan of work for the second semester. This separation serves to provide a clear distinction between the two semesters and the activities that will be undertaken in each one. Importantly, subsection 3.2.2 outlines what is ahead for the project and not what has transpired. This is essential as it allows us to anticipate and prepare for any difficulties that may arise during the development process, and to make any necessary adjustments to ensure the project's success

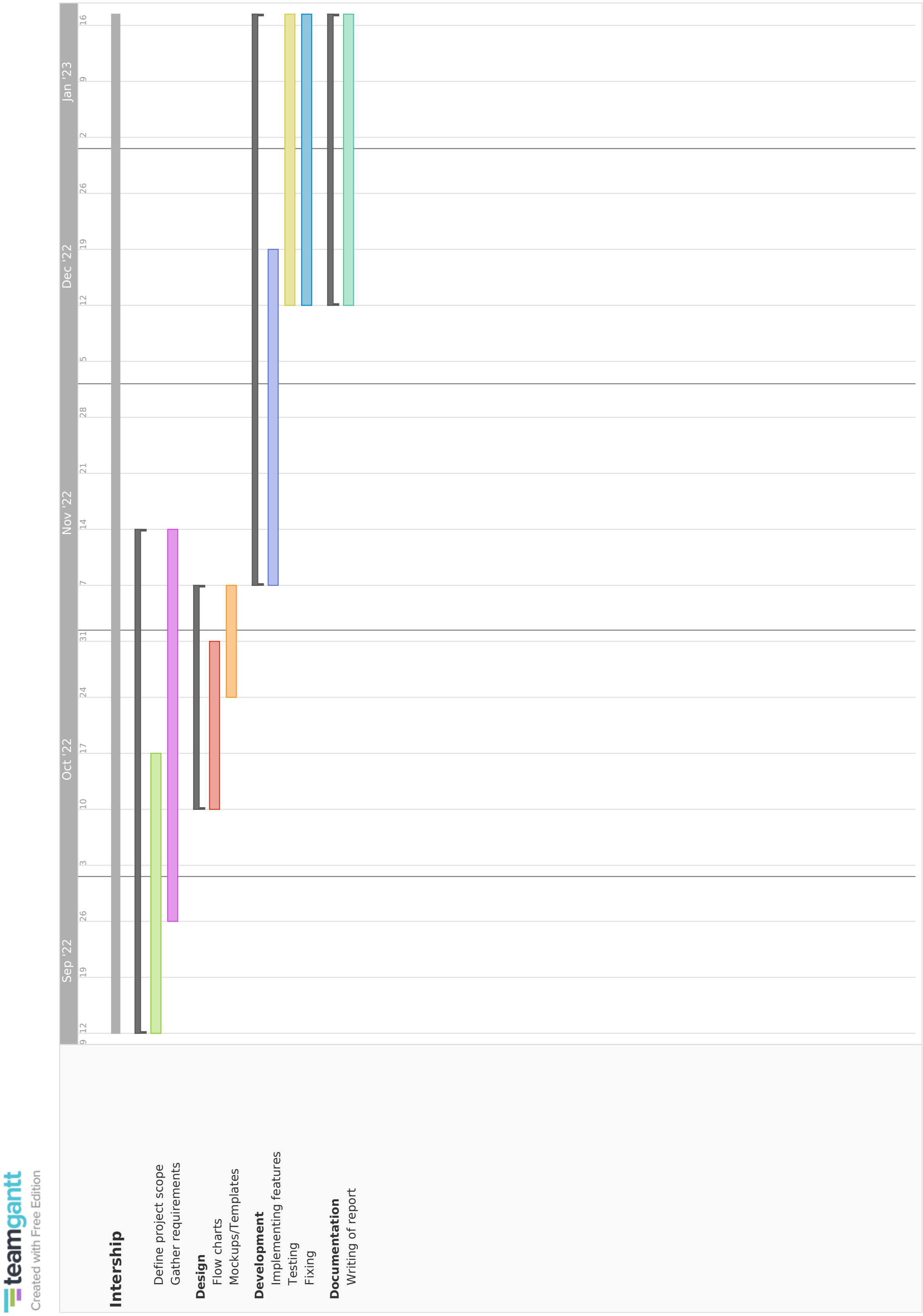
One of the key elements of the planning for this project was the use of Gantt charts which are a widely used tool that lets teams visually track the progress of a project and identify possible issues. These charts provide a clear and visual representation of the tasks that need to be completed. By using them it becomes easy to identify any likely obstacles and delays in the development process and take action to reduce or eliminate them. Therefore they were used to plan the project in two semesters, each with specific objectives, deliverables and milestones.

However, during the second semester it was not possible to follow the tasks set out by this Gantt chart. Most of the reasons are mentioned in section 6.1 but are as follows. The first issue was that the design was still not ready by the stipulated integration date, which pushed this date further away, and starting development with an ongoing design process introduces some issues as it might lead to an increased number of changes.

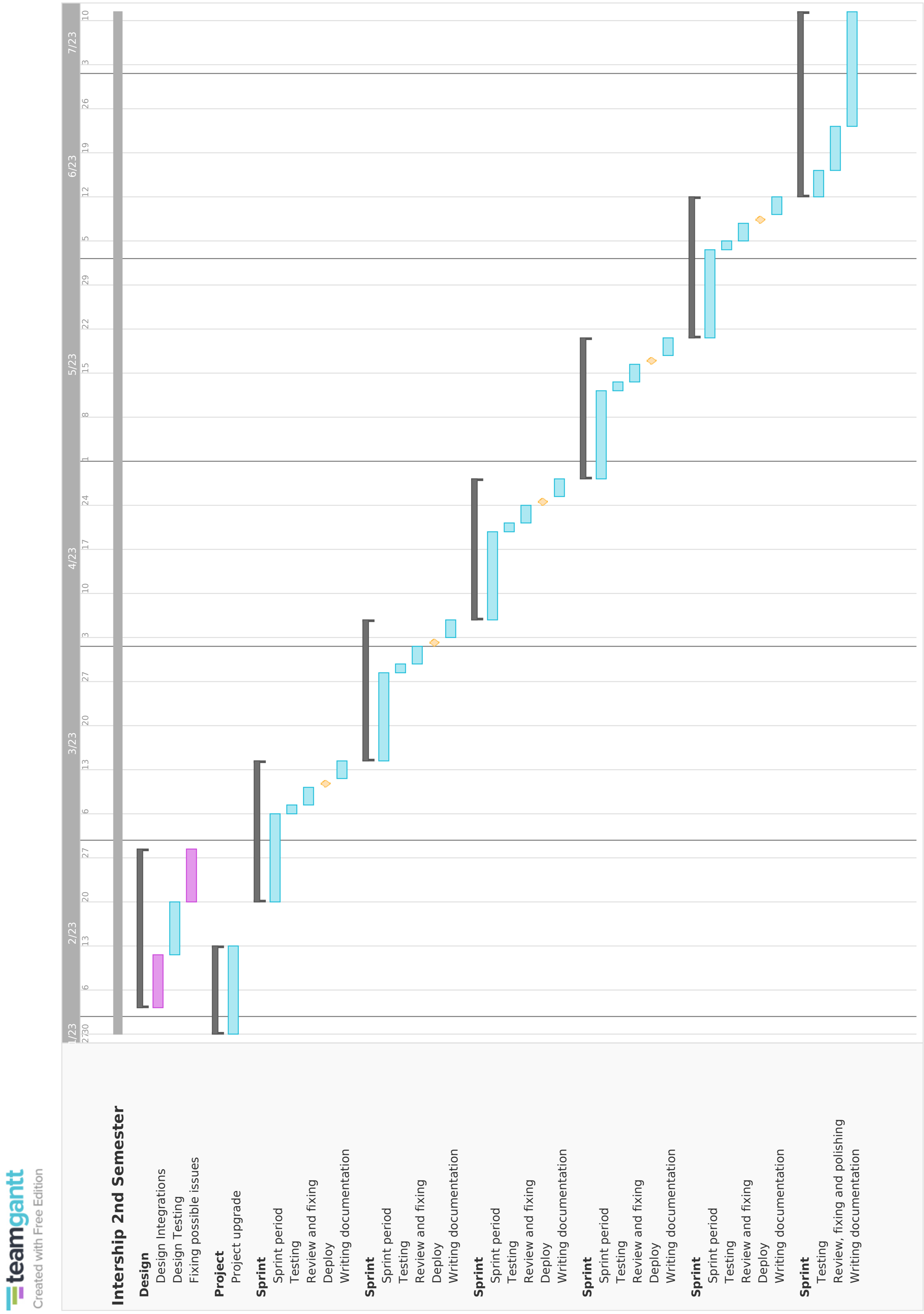
Secondly the issues shown during the first semester's deployment process led to a decision to redesign the database model and as a consequence a rewriting of most of the codebase. This further pushed back the dates, as the redesign itself required frequent meetings to nail down each detail of the model. Both the issues above pushed back the schedule by a couple of months. Furthermore, the framework and language update, along with the environment update also pushed back these dates by a a few weeks due to requiring meetings with IT personnel. The need for the design to be converted into HTML/CSS/JS code manually also led to a large delay in the schedule since it was not possible to automatically convert it.

However these delays, it was eventually possible to resume work on the features for the remainder of the available time while leaving the last month for the writing the the internship report. A detailed look at the implemented features and the setbacks can be seen in section 6.1. Nonetheless, these features were implemented with the use of timeboxing, which meant allocating a time or date for a specific feature to be ready.

### 3.2.1 First Semester



### 3.2.2 Second Semester



# Chapter 4

# Requirements Definition

In this section, we will approach all information regarding Functional Requirement (FR) obtained from several conversations with the advisors, helpdesk and others. Each table will represent a different stakeholder and their FRs. Some of these might have an additional sub-table to detail further their sub-requirements as well as their pre and post-conditions, input fields and similar information.

Each requirement will be assigned a priority level using the MOSCOW method, where 'Must Have' stands for the highest priority and is a requirement that must be implemented in order to meet the objectives of the project. 'Should Have' is the next level of priority, representing a requirement that should be implemented if possible, but not necessarily essential for the project's success. 'Could Have' is the next level of priority, representing a requirement that could be implemented but is not essential for the project's success. 'Won't Have' (this time) stands for the lowest priority and represents a requirement that will not be implemented in this iteration of the project but may be considered in a future implementation. These priority levels should be used to determine the overall importance of each requirement and its contribution to the platform as a whole. [Bradner, 1997] [MoS].

The MOSCOW method is a widely used prioritisation technique that helps to clearly define the level of importance of each requirement for a project. It allows teams to focus on the most essential requirements while also taking into account the feasibility and desirability of other requirements. This ensures that resources and efforts are directed towards the most critical aspects of the project, leading to a more efficient and effective development process. Additionally, it also helps in identifying requirements that may be delayed or assigned to future iterations of the project, ensuring that resources are directed towards the most important aspects of the project.

It is worth noting that using the MOSCOW method can also help in communication with stakeholders regarding the development of the platform as it helps explain the rationale for why certain features are included and why others are not.

## 4.1 Stakeholders

Stakeholders are an important part of the development process of most, if not all, projects and their success. It is critical to understand the stakeholders and what roles they play, such that it becomes easier to manage and communicate efficiently and effectively throughout the development process. Identifying stakeholders and their requirements is a necessity for the development of a platform that is meant to be designed to full fill their needs. It is of undoubtedly high importance to consider the different stakeholders such that we can ensure that the platform is fully functional and beneficial to all parties involved. By understanding the stakeholders we also open the door for easier maintenance and improvement of the platform, as their feedback and suggestions can and should be used to identify areas which require change and necessitate further improvement.

Another important aspect of project development is managing the expectations of stakeholders including ensuring that their requirements are fulfilled and that they are kept informed of progress and changes to the platform. This can also help with identifying problems that would otherwise not be detected, as it is a custom that concerns arise as development progresses that developers themselves do not see. Thus, we begin by identifying them in tables, assigning them an identification (ID) which is used to further reference that stakeholder, and a Name, specifying if they have an active role in the platform, that is if they have a 'Profile' assigned to them and a brief description of their needs and/or objectives.

Table 4.1 details the different actors that are involved in the internship platform. These stakeholders are identified by an ID, their name, and a profile description. The list of stakeholders includes companies, who submit proposals and manage the students who have applied for those proposals. They can also manage their data and add or remove representatives or Advisors. Students, who submit applications for different proposals can also see details about each application. The university is the entity that requests proposals and contains the students as well as the administration. Coordinators who are responsible for managing the application, phases and companies. and are responsible for managing proposals. Advisors and DEI Advisors who advise the student on their process, through the use of meetings. Juries, who evaluate the student's thesis or internship and Faculty Members who submit proposals and manage students who have sent applications for said proposals. This table provides a comprehensive overview of the different stakeholders that are involved in the internship platform and their specific roles within it.

Table 4.1: Stakeholders

| ID | Name | Profile | Description |
|---|---|---|---|
| STKH_1 | Company Manager | Yes | Companies submit proposals and manage students who have sent applications for said proposals. They can also manage their data and add or remove legal representatives, Advisors and managers. |
| STKH_2 | Student | Yes | Students submit applications for different proposals and can also see details about each application. |
| STKH_3 | University | No | The university is the entity that requests proposals and contains the students, as well as the administration. |
| STKH_4 | Coordinator | Yes | Responsible for managing proposals. |
| STKH_5 | Advisor (Company) | Yes | Advises the student on their process. |
| STKH_6 | DEI Advisor | Yes | Advises the student on their process. |
| STKH_7 | Jury | Yes | Evaluates the student's thesis or internship. |
| STKH_8 | Faculty Member | Yes | Faculty Members submit proposals and manage students who have sent applications for said proposals. |
| STKH_9 | Platform Admin | Yes | Platform admin is responsible for the management of the dates for the internship period as well as who is able to access the platform |
| STKH_10 | Legal Representative | Yes | Person responsible for the signing of the internship protocol, one must be assigned to each proposal |

## 4.2 Stakeholder Flowcharts

The flowchart in Figure A.1 represents the steps the student needs to follow to navigate the process. The flowchart in Figure A.2 represents what happens (from the proposal's point of view) during the placement phase of the process. The flowchart in Figure A.3 represents the process that the company must follow to submit a proposal for consideration by the administration.

## 4.3 Functional Requirements (FR)

This section of the document provides an overview of the functional requirements for the internship platform. The section is divided into several sub-sections, one for each of the needed stakeholders and provides a list of some of the functional requirements. For easier reference tables containing the complete list of requirements were also included with a description of each requirement. These tables can be found in Appendix B.

### 4.3.1 Company

The first requirement, **CMPY-1**, is the ability to register a company on the platform. This allows the company to create new internship proposals and manage the students who are participating in those proposals. This is considered a must-have requirement as it is essential for the company to be able to participate in the platform.

The second requirement, **CMPY-2**, is the ability to manage company information. This includes the ability to add or remove information related to the company, such as contact information and company details. This is also considered a must-have requirement as it is essential for the platform to have accurate information about the companies that are participating.

The third requirement, **CMPY-3**, is the ability of companies to create new internship proposals. This is a must-have requirement as it allows companies to offer internships to students. The remainder of the requirements can be seen in Table B.1.

### 4.3.2 Student

The table in Table B.2 presents the FR for the student stakeholder group within the internship platform. Each row in the table represents a different requirement. The table includes several requirements that are critical for students to be able to effectively use the platform. The following paragraphs will detail some of the requirements.

The requirement **STUD-1** is a must-have and allows students to fill out their personal information regarding their academic performance and curriculum vitae. This information is necessary for the company representatives to assess the student's suitableness for the proposal they applied to, ultimately helping decide whether a student is the right "fit".

The requirement **STUD-2** is also a must-have and allows students to view and review all available proposals and their details. This is essential for students to be able to make educated decisions about which proposals to apply for.

The requirement **STUD-3** allows the student to select up to 5 proposals in order of preference, meaning that the student can indicate which proposals are most desirable for them to participate in. This is an important feature as it helps match students with proposals that align with their interests and goals.

### 4.3.3 Coordinator

Requirement **CORD-1** is another must-have and provides the ability for the coordinator to assign juries to a given proposal, this happens after a student has been assigned. It enables the coordinators to review the schedules and subjects of each jury in a thoughtful manner, seeing as it is of some importance that the subjects of the thesis and the juries align, and assign them to a specific proposal.

Requirement **CORD-2** is also a must-have and provides the ability for the coordinator to manage all submitted proposals for their assigned degree, such as approving, rejecting or requesting an edit for a given proposal (reviewing).

Requirement **CORD-3** is also a must-have and provides the ability for the coordinator to review the details of a given proposal, including its candidates, and assign them a student while considering their and other candidate's preferences.Table B.1.

### 4.3.4 Platform Manager

The first requirement, **ADMIN-1**, is a must-have and provides the ability for the administration of the platform to set up new phases or alter previous ones. These phases should include dates that define the periods where each step of the process can occur, including start and end dates for the phase, the submittal of proposals, the submittal of candidacies and change of said candidacies, the interviews, the review period, defence (intermediate and final) dates, and others. This helps compose the proper schedule for a phase.

The third requirement, **ADMIN-3**, is also a must-have and provides the ability for the administration of the platform to manage the details of a given company, this is a rather important requirement as it enables the administration to perform otherwise forbidden tasks such as deleting proposals and recovering accesses. Recovering accesses is used by the administration when the company loses all access to the platform, see **CMPY-12** in Table B.1.

### 4.3.5 Advisors

The first requirement, **ADVS-1**, is a must-have requirement and it enables advisors, be them from a company or from Department of Informatics Engineering (DEI), to see what proposals they have been assigned to, including the ones they might have assigned as a co-advisor. It also enables them to see in which status the proposal currently sits.

The third requirement, **ADVS-3**, is a must-have and much and allows advisors to review the applications they have received from students and make decisions on whether to select that student or not.

The fourth requirement, **ADVS-5**, is also a must-have and it lets advisors submit reports from meetings, these are meetings summaries and are an important part of ensuring protocol is followed and that the process is on track,

### 4.3.6 Faculty Member

Requirement **FCLT-1** is the ability of faculty members to create new internship proposals. This is a must-have requirement as it allows faculty members such as professors to offer internships to students.

Requirement **FCLT-2** is complementary to the first one and allows faculty members to edit their proposal as long as their proposal has yet to be approved. It is a must-have requirement as it enables faculty members to correct any issue that may arise with their proposal

Requirement **FCLT-5** is a must-have requirement as it allows faculty members to select which candidates fit their needs by reviewing their information like average grade and preference order. They can accept or reject a given candidate, rejected candidates will be excluded from this proposal selection process unless accepted again by the faculty member.

### 4.3.7 System

In this section, the FRs work independently of a stakeholder and include requirements such as sending emails or other automated processes, these can be seen in table Table 4.2. The first requirement, **SYS-1**, is a must-have and it enables the system to send any email that may be required, such as account confirmation emails or alert emails that may alert a user of pending tasks.

The second requirement, **SYS-2**, is the ability of the system to automatically assign students to a proposal. The proposed algorithm would place accepted (students that have been accepted by the proposers) students would be placed according to their preference order first and if that conflicts then it would be placed by their grade average. If two students have the same preference for a given proposal and are both accepted their placement would be determined by their grades. Furthermore, if both of these parameters were to fail, the fallback would be a manual selection or, possibly, a random placement. This requirement is a should-have due to not being a core requirement for the functionality of the platform but would improve the Quality of Life (QoL) and reduce the load on other users.

The third and final requirement, **SYS-3**, similarly to the previous one, is a should-have as it is not the highest priority but would certainly increase QoL. It enables the system to select the best-fitting juries for a given proposal, the jury would be assigned to the proposal by matching the subjects of the proposal with the subjects and the jury and checking if the jury has a compatible schedule. The fallback would be manual placement.

Table 4.2: FRs for system

| ID | Name | Stkh | Description | Have |
|---|---|---|---|---|
| SYS-1 | send_email | SYSTEM | Allows the system to send any email that may be required, specifically when it comes to warning users about upcoming dates and pending reviews. | Must |
| SYS-2 | assign_students | | Allows the system to automatically assign students to a proposal. | Should |
| SYS-3 | assign_jury | | Allows the system to automatically assign juries. | Should |
| SYS-4 | display_info | | Shows information about the dates, degrees and proposals publicly without requiring authentication. | Must |
| SYS-5 | authentication | | Allows users to login or logout. | Must |
| SYS-6 | change_active _role | | Allows users to change active roles, switching roles allows a user to access different platform functions associated with that switched role. | Must |
| SYS-7 | change_language | | Allows users to change active language, switching language allows a user to access the platform in their preferred language | Must |

## 4.4 User Stories

In this section, user stories are presented. User stories are valuable because they present the user's needs from their perspective and allow for a better understanding of why that user needs that feature. Additionally, they allow for easier discussion with said users about the required features. Here are some of the user stories:

- *CMPY-1*: **As a** company representative, **I want** to be able to register my company on the platform **so that** I can create new proposals.
- *CMPY-2*: **As a** company representative, **I want** to be able to manage my company's information on the platform **so that** I can keep our information up-to-date and accurate.

- *CMPY-3*: **As a** company representative, **I want** to be able to create a new proposal for an internship on the platform **so that** I can offer opportunities to students.
- *CMPY-4*: **As a** company representative, **I want** to be able to edit an existent proposal for an internship on the platform **so that** I can correct or add data needed.
- *CMPY-5*: **As a** company representative, **I want** to be able to bring the proposal forward of an internship on the platform **so that** I can make use of a last period's proposal.
- *CMPY-6*: **As a** company representative, **I want** to be able to review the proposal of an internship on the platform **so that** I can make sure my proposal is accepted.
- *CMPY-7*: **As a** company representative, **I want** to be able to manage the legal representatives on the platform **so that** I can make sure the internship protocol is signed.
- *CMPY-8*: **As a** company representative, **I want** to be able to manage the collaborators on the platform **so that** I can make sure I assign the proper advisers to an internship.
- *CMPY-9*: **As a** company representative, **I want** to be able to see candidates on the platform **so that** I can see if my proposal has candidates.
- *CMPY-10*: **As a** company representative, **I want** to be able to manage candidates on the platform **so that** I can make sure only the appropriate candidates are considered.
- *CMPY-11*: **As a** company representative, **I want** to be able to digitally sign the internship protocol on the platform **so that** I can speed up the process.
- *CMPY-12*: **As a** company representative, **I want** to be able to recover access to the platform **so that** I can access my account in case I lose my access to it.
- *STUD-1*: **As a** student, **I want** to be able to fill out my personal data on the platform **so that** I can apply for proposals and complete the application process.
- *STUD-2*: **As a** student, **I want** to be able to see the details of each proposal that I have applied for **so that** I can stay informed about the status of my application and any next steps that I need to take.
- *STUD-3*: **As a** student, **I want** to be able to select my proposal preferences **so that** I can have a chance at being assigned an internship of my choosing.
- *STUD-4*: **As a** student, **I want** to be able to view the proposal that was assigned to me **so that** I can observe its details.
- *STUD-5*: **As a** student, **I want** to be able to view the dates of my deliveries and milestones **so that** I can stay informed and keep up with the process.

- *STUD-6*: **As a** student, **I want** to be able to submit my deliveries **so that** I can be evaluated.

- *STUD-7*: **As a** student, **I want** to be able to reorder my preferences of proposals **so that** I can change my choices in the allowed period of time.

- *CORD-1*: **As a** degree coordinator, **I want** to be able to assign juries to a given proposal **so that** I can make sure each proposal has the appropriate jury members.

- *CORD-2*: **As a** degree coordinator, **I want** to be able to manage all submitted proposals **so that** I can make sure that the submitted proposals all match the required format.

- *CORD-3*: **As a** degree coordinator, **I want** to be able to assign students to a given proposal **so that** I can ensure that students get assigned an internship of their choosing.

- *CORD-4*: **As a** degree coordinator, **I want** to be able to review and edit my personal data **so that** I can keep my personal information up-to-date and choose my preferred name.

- *ADMN-1* **As a** platform administrator, **I want** to be able to manage companies' data **so that** I can correct any issues if need arises.

- *ADMN-2* **As a** platform administrator, **I want** to be able to manage users' data **so that** I can make sure only the proper users have access to the system and have proper roles.

- *ADMN-3* **As a** platform administrator, **I want** to be able to manage internship periods **so that** I can allow the internship process to start.

- *ADVS-1* **As an** advisor, **I want** to be able to see assigned proposals **so that** I can keep track of the process and follow up if necessary.

- *ADVS-2* **As an** advisor, **I want** to be able to see assigned students **so that** I can get in touch with the student if necessary

- *ADVS-3* **As an** advisor, **I want** to be able to see candidates **so that** I can know if my proposal has candidates to interview.

- *ADVS-4* **As an** advisor, **I want** to be able to manage candidates **so that** I can make sure only the candidates with a certain profile get assigned the proposal.

- *ADVS-5* **As an** advisor, **I want** to be able to submit reports **so that** I can document the internship process properly.

- *ADVS-6* **As an** advisor, **I want** to be able to see submitted reports **so that** I can revisit previous information.

- *FCLT-1* **As a** faculty member, **I want** to be able create new proposals **so that** I can propose my ideas to students.

- *FCLT-2* **As a** faculty member, **I want** to be able to edit my proposal **so that** I can correct any issues.
- *FCLT-3* **As a** faculty member, **I want** to be able to review my proposal **so that** I can resolve any pending issues.
- *FCLT-4* **As a** faculty member, **I want** to be able to see candidates **so that** I can know if my proposal has candidates to interview.
- *FCLT-5* **As a** faculty member, **I want** to be able to manage candidates **so that** I can make sure only the candidates with a certain profile get assigned the proposal.
- *FCLT-6*: **As a** faculty member, **I want** to be able to review and edit my personal data **so that** I can keep my personal information up-to-date and choose my preferred name.

# Chapter 5

# Technology & Architecture

This chapter will discuss the technologies used to develop the internship platform. Some of these technologies were chosen based on prior knowledge of the tools and decisions taken before the start of this internship. The chapter will also expose the architecture implemented given the available tools. Section 5.1 discusses the implemented architecture enabled by the resources made available by Department of Informatics Engineering (DEI) followed by the architecture diagrams developed. Furthermore, section 5.3 and section 5.5 discuss the technologies used for both backend and frontend work respectively and their respective advantages and drawbacks.

## 5.1 Architecture

The decision to use a monolithic architecture, with Model-View-Controller (MVC) approach, for the platform presents both advantages and disadvantages. This architecture type is one where all the components of an application, such as the user interface, business logic, and database, are joined in a single codebase and deployed all together (as a monolith). This architecture type works well with the chosen frontend tool as it provides seamless integration with it, allowing for efficient communication and data exchange within the codebase.

Given the scope and size of the internship platform, a monolithic architecture can be a suitable choice. Since the simplicity and ease of development align well with the platform's objectives. However, there are also some drawbacks namely in scalability and lack of code boundaries. These drawbacks should be carefully considered and managed as the platform's scope and size evolve [Dushenin].

Some of the advantages of monolithic architecture are as follows:

1. **Simplicity**: Monolithic architectures are generally simple to develop and maintain. The single codebase and development environment streamline the development process and reduce the "cognitive overhead" [Harris] making it easier to understand the codebase and follow its flow [Dushenin].

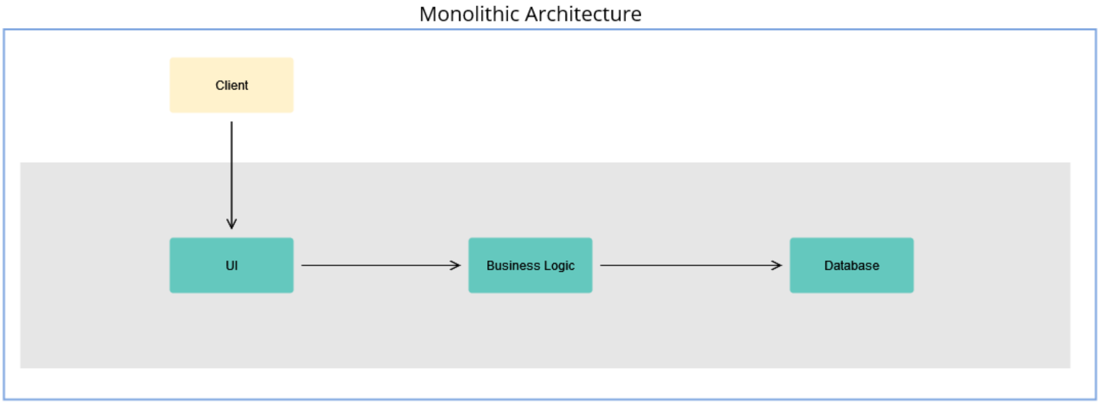


Figure 5.1: Simple example of monolithic architecture.

2. **Ease of Deployment**: This type of architecture can be easier and faster to deploy since its components are joined in a single unit, as such, there are usually no dependencies needed. Due to this, it is often simpler than an architecture like microservices due to possible dependencies and the existence of several "units" [Dushenin].
3. **Faster Development**: With all components in one place, i.e., a single codebase, it is usually easier to develop, especially as there is no need to manage communication between separate components. [Harris] [Dushenin].

Some of the disadvantages of monolithic architecture are as follows:

1. **Scalability difficulties**: Monolithic applications can become a bottleneck as they scale up. All components are tightly coupled as a unit, which makes it difficult if not impossible to scale individual parts, which may lead to inefficiencies when allocating resources [Harris].
2. **Deployment Risks**: A single monolithic codebase means that any change to any one part of the application requires redeploying the entire application which risks deploying unintended code and causing issues. This forces all features in development to be ready at a given time for the deployment to work properly. It can also cause an increase in development time due to the issues mentioned above [Dushenin].
3. **Team Collaboration**: Projects with larger teams using monolithic architecture can lead to collaboration challenges as different developers might need to work on the same codebase, leading to conflicts and difficulties, especially when managing version control. Larger projects can also have increased difficulty when onboarding due to higher codebase complexity. [Dushenin].

Considering these points and that the team was made of one developer and the previous codebase was already using monolithic architecture the most reasonable choice was the keep the same architecture for simplicity and speed.

## 5.2 C4 Model

The C4 Model is a set of hierarchical diagrams useful in software architecture, a clear and dynamic way of visualising an architecture, making it easier to understand these systems. The model offers four levels: **Context** (high-level system interactions), **Container** (which can be considered individual units that can be executed), **Component**, and **Code** (implementation details). Starting from the high level, it zooms in, increasing the level of detail, and creating a progressive view of the system [Brown]. In this case, only the first three levels of the diagrams were created.

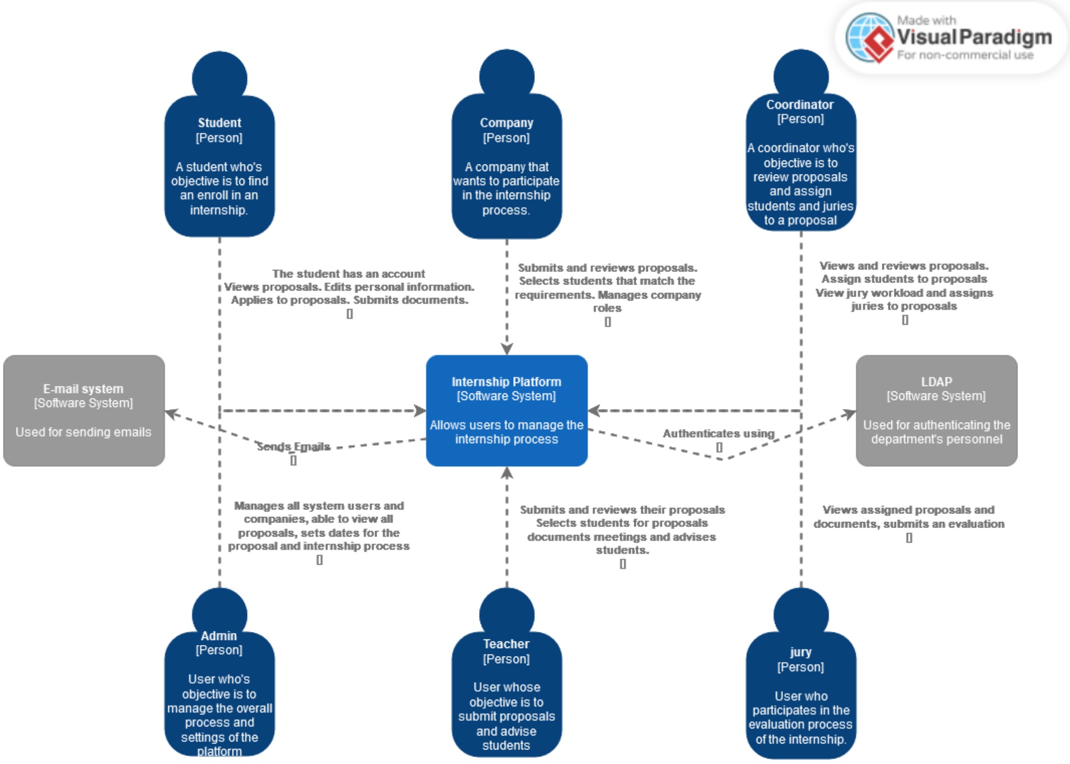

Made with Visual Paradigm For non-commercial use
Student
[Person]
A student who's objective is to find an enroll in an internship.
Company
[Person]
A company that wants to participate in the internship process.
Coordinator
[Person]
A coordinator who's objective is to review proposals and assign students and juries to a proposal
The student has an account Views proposals. Edits personal information. Applies to proposals. Submits documents.
[]
Submits and reviews proposals. Selects students that match the requirements. Manages company roles
[]
Views and reviews proposals. Assign students to proposals View jury workload and assigns juries to proposals
[]
E-mail system
[Software System]
Used for sending emails
Internship Platform
[Software System]
Allows users to manage the internship process
LDAP
[Software System]
Used for authenticating the department's personnel
Sends Emails
[]
Authenticates using
[]
Manages all system users and companies, able to view all proposals, sets dates for the proposal and internship process
[]
Submits and reviews their proposals Selects students for proposals documents meetings and advises students.
[]
Views assigned proposals and documents, submits an evaluation
[]
Admin
[Person]
User who's objective is to manage the overall process and settings of the platform
Teacher
[Person]
User whose objective is to submit proposals and advise students
jury
[Person]
User who participates in the evaluation process of the internship.

Figure 5.2: Context diagram of the platform.

The diagram shown in Figure 5.2 shows the high-level context diagram of the system, in this diagram, the persons/users interact directly with the "Internship Platform" system which contains all logic related to the platform, which in turn can communicate with other external systems given an appropriate condition. It communicates with the LDAP System in cases where it is needed to authenticate users, specifically for the department's personnel and students. Besides the authentication information, it also provides details such as the user's full name. On the other hand, it can also communicate with an email system to facilitate some system-to-user and user-to-user communications such as when sending confirmation emails and informational emails.

The second diagram, seen in Figure 5.3, shows the system with more detail

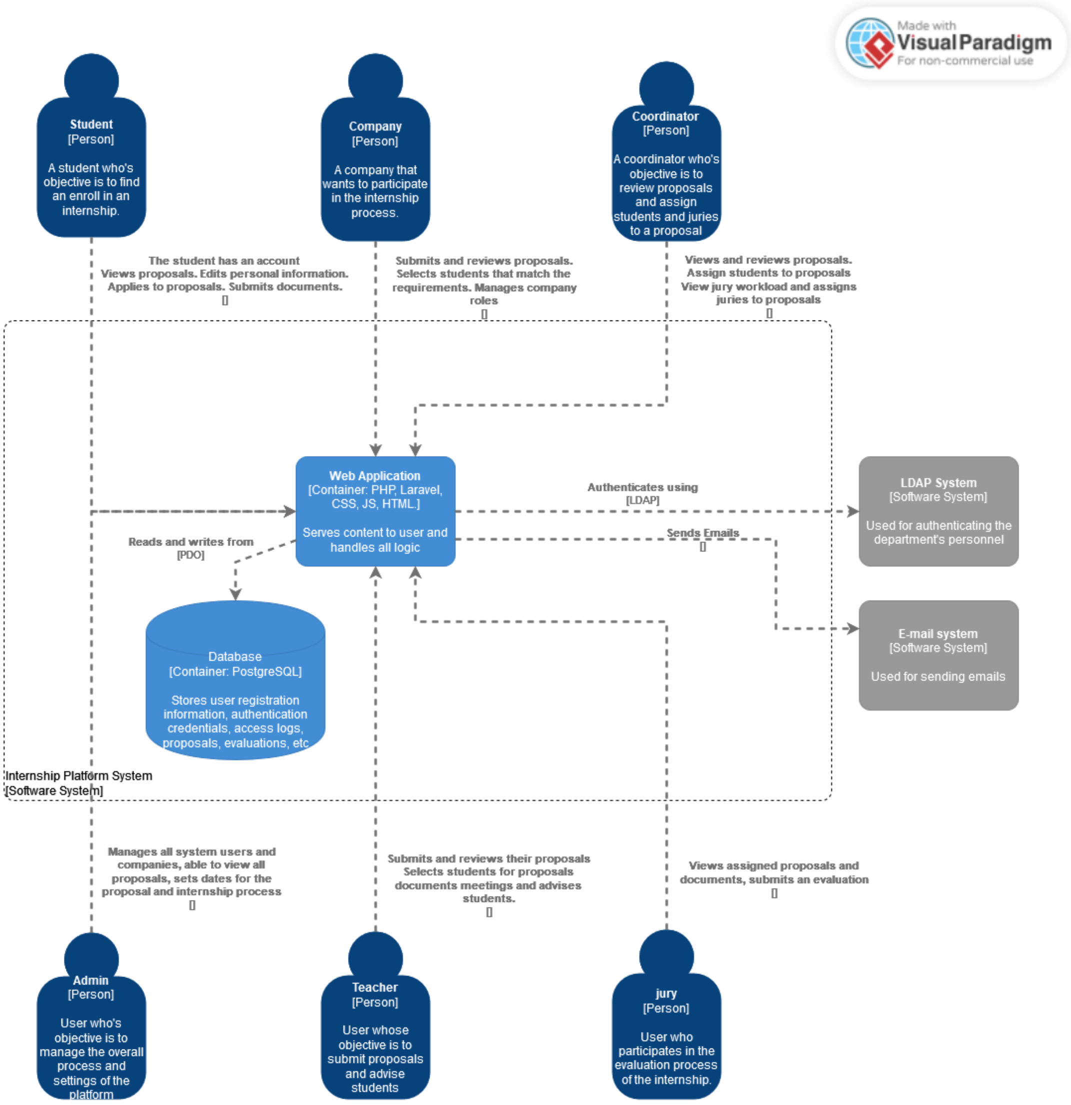

Made with Visual Paradigm For non-commercial use
Student [Person] A student who's objective is to find an enroll in an internship.
Company [Person] A company that wants to participate in the internship process.
Coordinator [Person] A coordinator who's objective is to review proposals and assign students and juries to a proposal
The student has an account Views proposals. Edits personal information. Applies to proposals. Submits documents. []
Submits and reviews proposals. Selects students that match the requirements. Manages company roles []
Views and reviews proposals. Assign students to proposals View jury workload and assigns juries to proposals []
Web Application [Container: PHP, Laravel, CSS, JS, HTML.] Serves content to user and handles all logic
Authenticates using [LDAP]
Sends Emails []
LDAP System [Software System] Used for authenticating the department's personnel
E-mail system [Software System] Used for sending emails
Reads and writes from [PDO]
Database [Container: PostgreSQL] Stores user registration information, authentication credentials, access logs, proposals, evaluations, etc
Internship Platform System [Software System]
Manages all system users and companies, able to view all proposals, sets dates for the proposal and internship process []
Submits and reviews their proposals Selects students for proposals documents meetings and advises students. []
Views assigned proposals and documents, submits an evaluation []
Admin [Person] User who's objective is to manage the overall process and settings of the platform
Teacher [Person] User whose objective is to submit proposals and advise students
jury [Person] User who participates in the evaluation process of the internship.

Figure 5.3: Container Diagram of the platform.

than Figure 5.2 and how the systems interact with each other, in this case, there's only one system and that's the web application. This container is responsible for most of the application's functionality and communicates only with 2 other external systems and a database system. The communication (both reads and writes) with the shown database system uses *PDO_PGSQL*, the *PDO* (PHP Data Object) driver for the PostgreSQL, which provides a consistent interface for PHP interaction with the RDBMS[pdo]. Additionally, the communication with the authentication system uses the LDAP (Lightweight Directory Access Protocol), which provides a centralised platform for several platforms to authenticate users from, including the internship platform. All of the platform's users interact with the "Web Application" container for their various tasks.

The third and last diagram, seen in Figure 5.4, shows the several components of the Web Application container and their major interactions. In this diagram, it is

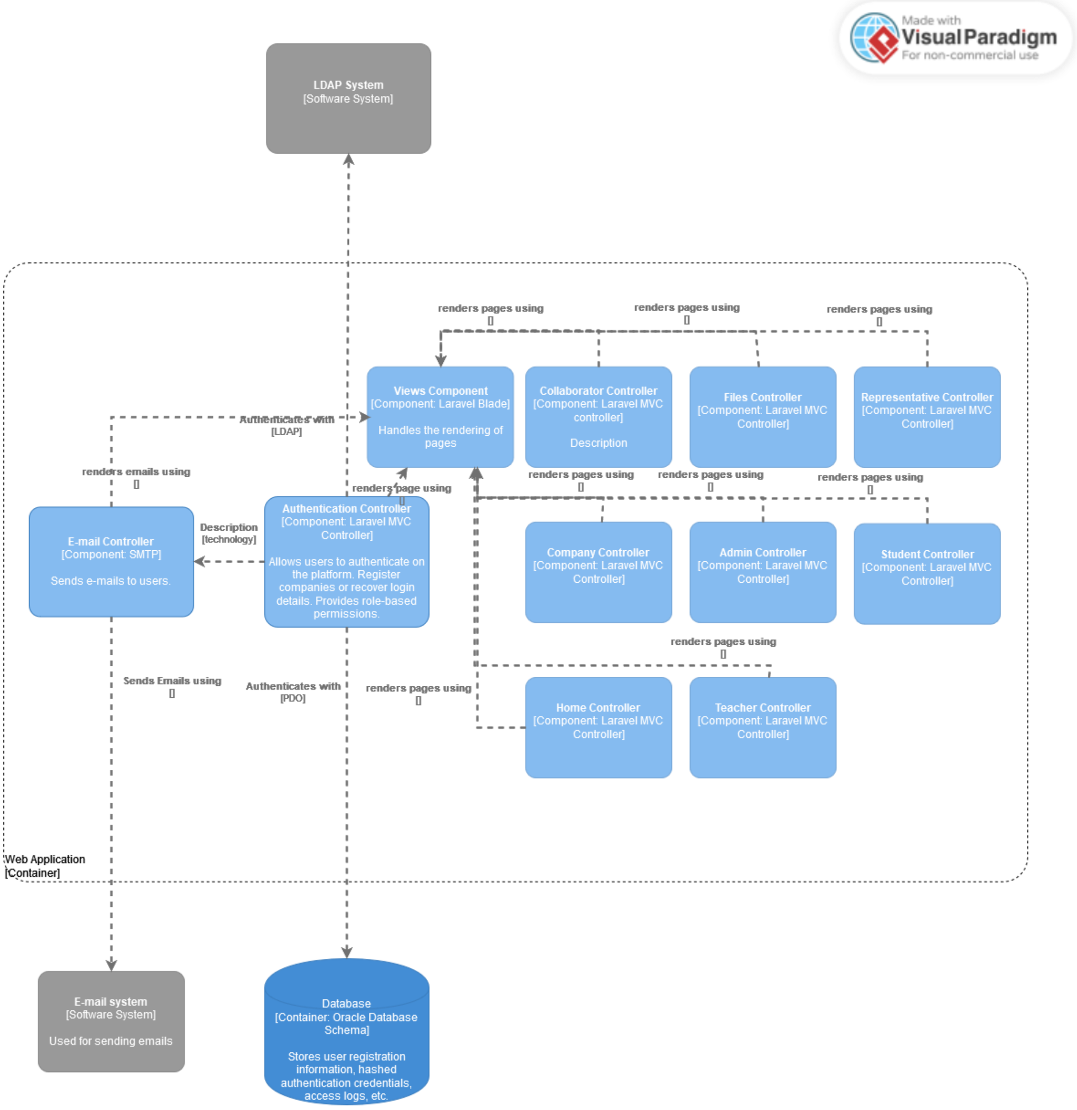

Made with Visual Paradigm For non-commercial use
LDAP System
[Software System]
renders pages using
[]
renders pages using
[]
renders pages using
[]
Views Component
[Component: Laravel Blade]
Handles the rendering of pages
Collaborator Controller
[Component: Laravel MVC controller]
Description
Files Controller
[Component: Laravel MVC Controller]
Representative Controller
[Component: Laravel MVC Controller]
Authenticates with
[LDAP]
renders emails using
[]
renders page using
[]
renders pages using
[]
renders pages using
[]
renders pages using
[]
E-mail Controller
[Component: SMTP]
Sends e-mails to users.
Description
[technology]
Authentication Controller
[Component: Laravel MVC Controller]
Allows users to authenticate on the platform. Register companies or recover login details. Provides role-based permissions.
Company Controller
[Component: Laravel MVC Controller]
Admin Controller
[Component: Laravel MVC Controller]
Student Controller
[Component: Laravel MVC Controller]
renders pages using
[]
Sends Emails using
[]
Authenticates with
[PDO]
renders pages using
[]
Home Controller
[Component: Laravel MVC Controller]
Teacher Controller
[Component: Laravel MVC Controller]
Web Application
[Container]
E-mail system
[Software System]
Used for sending emails
Database
[Container: Oracle Database Schema]
Stores user registration information, hashed authentication credentials, access logs, etc.

Figure 5.4: Component diagram of the web application container.

possible to see what specific component, or as they are called in the diagram, the controller interacts with which external system. In the case of the LDAP System, this interaction is done via the Authentication Controller that is responsible for handling the two methods of authentications required, one via database and one via LDAP, as well as any and all authentication-related features such as password recovery and account creation. Furthermore, there is one view component that is responsible for handling the rendering of all the views, all other controllers require the use of this component to display the pages associated with their functions.

Additionally, there exists one controller for each role in the platform and each one of these controllers is responsible for implementing the logic behind each and every one of the tasks that the respective profile or role entails. For simplicity of the diagram and to facilitate reading it, the relationships between the authentication controller and the remainder of the controllers were not included, however, each controller should make use of the authentication controller to ensure that each user has the required permissions to access the given feature.

There are, however, two outliers in this pattern, the Files controller and the Home controller. The File controller is responsible for handling the upload, retrieval and deletion of any files from the storage as well as handling who is allowed to access the given file. Lastly, there is the Home controller, this controller is responsible for handling the static pages of the platform, that is, the pages that do no require any role-specific functionality such as showing each available course a description of said course along with the dates for the internship period and the available proposals.

## 5.3 Backend Technologies

Given the scope of this project, a robust framework should be used, one with substantial use, support and maturity. As such, the framework chosen, albeit before the start of the internship, was Laravel, a PHP-based framework. It was selected by the project's previous developer based on familiarity with the framework, language and the available hardware. Nonetheless, Laravel offers a fair few advantages to any project.

Laravel was born in 2011 and became one of the most popular PHP frameworks. Its elegant syntax and powerful features have contributed to its widespread adoption. Laravel's elegant syntax makes it easy to write clean code, assisting with maintainability. Laravel is based on the MVC architecture which provides a clear separation of concerns and improved code organisation [lar, c].

The Laravel Eloquent simplifies database interactions by allowing developers to work with databases using a more intuitive, object-oriented and abstract way without having to handle database-specific problems or write database-specific code [lar, c] [elo].

When it comes to security Laravel offers a wide range of mechanisms to guard against several common web vulnerabilities like SQL injections, Cross-site scripting (XSS) and Cross-site request forgery (CSRF). It also provides an in-built authentication system that allows for easy implementation of users and access control to the platform, which can be easily modified to suit each developer's needs [lar, b]. Laravel also includes a very robust form validation system with over 60 different rules that automatically validate incoming requests, contributing to cleaner and more efficient code.

Laravel's extensive ecosystem is another advantage, due to having a significant amount of packages and extensions that can be easily integrated into projects with the PHP package manager named Composer, allowing developers to use existing solutions instead of having to write everything themselves [why].

The choice of Laravel as the framework for the internship platform aligns with its reputation for robustness, maintainability, and versatility. Its features and benefits not only facilitate development but contribute to the creation of a secure and efficient application. However, there are some drawbacks to the use of this framework.

One notable drawback of Laravel is its performance overhead, especially in comparison to other PHP frameworks, such as Spiral. Laravel's dynamic features and its ease of use can result in slower performance when handling high-traffic applications and complex computations, while Laravel's performance is satisfactory for many applications, it is behind some other frameworks that were optimised for speed [tec] [lar, a]. Furthermore, the heavy reliance on magic methods and dynamic method resolution, which is why it has dynamic and elegant syntax, can cause problems in debugging and following the flow of execution.

Laravel also has frequent updates which, whilst good for maintaining security and staying up-to-date with current practices, can lead to compatibility issues, creating a considerable amount of work for developers who have to ensure that their custom code and any third-party packages remain functional after the framework update.

Initially, the internship platform was built using MySQL as the chosen database management system. However, a decision was made to transition to PostgreSQL due to a higher level of familiarity with PostgreSQL's features and tools, which ultimately facilitated more efficient database design, management, and optimisation. This change was facilitated by and occurred during a database redesign that was done. Additionally, PostgreSQL might be considered a better choice due to having more advanced database features and being "ACID compliant in all configurations", while also being better for enterprise-level applications [mys].

The development process for the internship platform was facilitated through the utilization of the Department's GitLab. To maintain a structured and controlled workflow, two distinct branches were created in the GitLab repository. The "development" branch served as the place for ongoing coding and development. Once the code was deemed ready and tested on the "dev" site, it was then merged into the "deployment" branch and later deployed to the live site.

## 5.4 Frontend Technologies

The chosen frontend technology for the platform was Laravel Blade, a template engine integrated with the Laravel framework. It offers an expressive and intuitive syntax just like the rest of the framework. Blade templates provide features like template inheritance, sections, layouts, and components, increasing reusability and maintainability.

The decision to utilise Laravel Blade was primarily based on its simplicity and familiarity with the tool. Given that there was previous knowledge of Laravel and its associated technologies, including Blade, it made sense to leverage this knowledge. The main drawback of using this over some other frontend frameworks, like ones based in JavaScript, is that it's less dynamic and tends to lead to frontends that feel less polished [gro].

## 5.5 Entity Relationship Diagram

During the course of this internship, an Entity Relationship Diagram (ERD) was developed to model and visualize the structure of the internship platform's database, shown in Figure 5.5. An ERD is a fundamental tool in database design because it illustrates the relationships between different entities, providing a clearer representation of how data is organized, stored, and used.

ERDs are structured using relationships between entities (which can equate to tables). They include entities which are the main part on the diagram, they represent tables in the database. Each entity represents an object/data in the system, such as a user, or a proposal. Entities contain attributes that represent columns in the tables, these attributes are listed inside the entities and detail what the entity stores. Entities can be related to each other via relationships.

Initially the project started using the old database design, shown in **??**, this was done so the data on the previous platform would be fully compatible with the new platform. However, this decision was shown to not be effective as the old database design had severe problems, which could not have been overcome without a new design. This led to the use of workarounds for implementing features, which in turn led to features not working as initially intended. As such a decision to create a new database was made as out of this decision the ERD shown in Figure 5.5 was made. It was thought that after the implementation was done, an adaptation process could be created to retrieve the old information to the new database.

Some attributes in this diagram were omitted, such as *updated_at* and *created_at* so as to not overload the diagram and because they are pretty standard to include. These attributes help keep track of changes to the platform. Additionally, most of these entries can not be deleted, instead, they can be "deactivated" which causes them to lose access (in the case of users) or lose their effect (in the case of proposals, or degrees). This helps keep a history of the data and makes sure nothing important is lost. An example of this would be if a user was able to delete their profile, the proposals they submitted would be completely lost, which cannot happen.

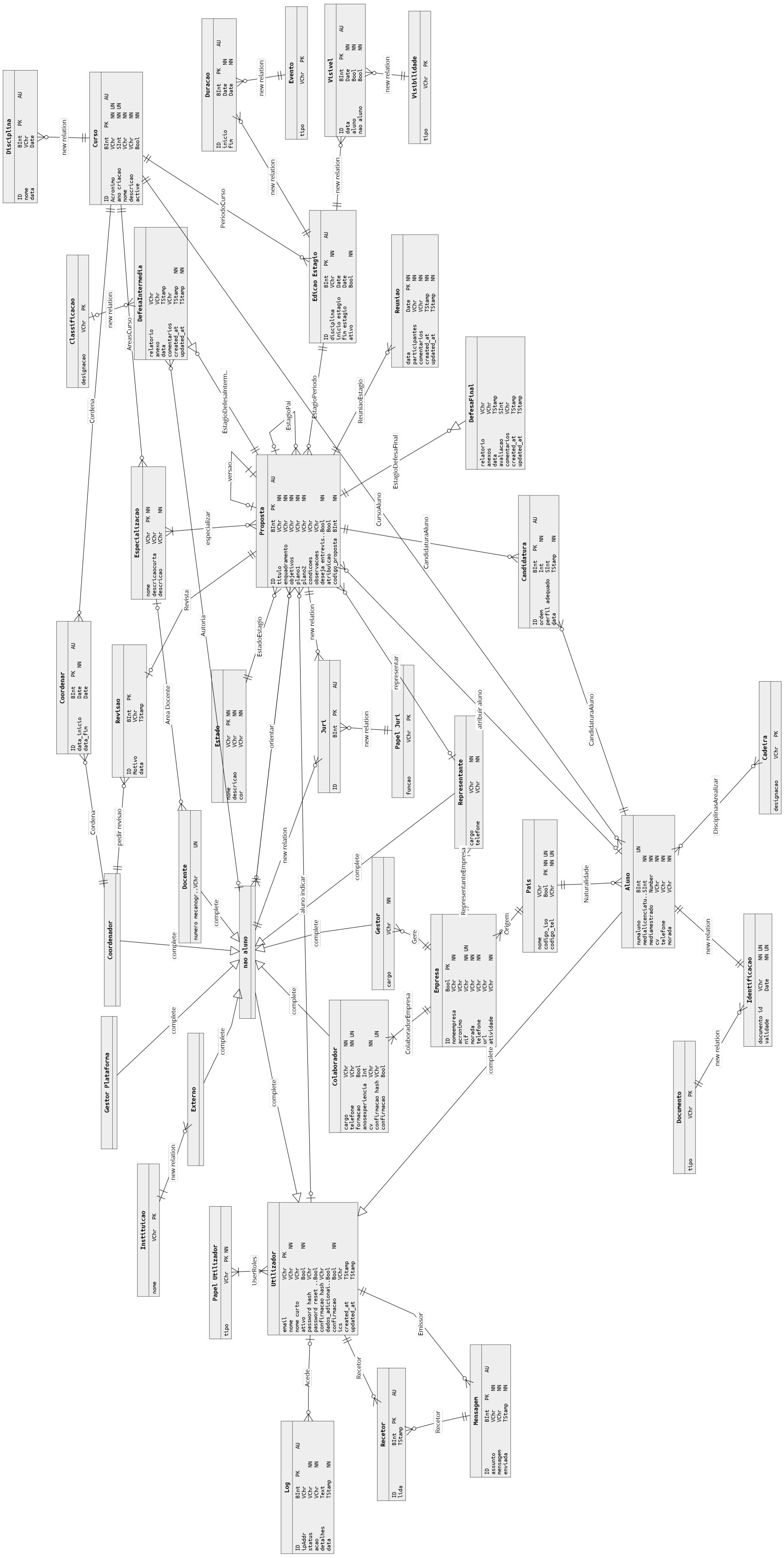


Figure 5.5: Conceptual model for the platform

# Chapter 6

# Development

During the start of the first semester, meetings between the members were conducted in order to evaluate the requirements of the project. Initially, an overview of the old platform was done along with all the problems said the platform had. These meetings were conducted with a few different stakeholders in order to obtain an accurate picture of the problems. Furthermore, some regular meetings with the previous developer were had in order to gauge the development situation and a more technical transfer of knowledge. The new platform had already started development but it was shown to contain some issues. During the first half of this semester work mainly was focused on learning and obtaining information, afterwards, the work shifted to the development of the platform

The first semester's work was primarily based on fixing what was already done on the new platform, creating workarounds for the database and implementing some simple features. This work included an update on the design using a Bootstrap 5-based template (Bootstrap 5 is a well-known CSS framework built for responsiveness [boo]). However, this was not the final design. The features finished during the first semester were the features for the company side and most of the features for the faculty members.

However, when deployment time came, around the end of the first semester (which equates to the multi-year phase of the internships), the problems with creating a new platform using the old database design were revealed and most of the features did not work as expected. More specifically, it failed to handle data that was previously inserted and data that was sourced from the different platforms that were still using the old code, such as the coordination and admin. This led to the decision to temporarily revert back to the old platform. These issues were not initially detected because the new platform had been developed on a separate environment without any old data. The lack of data was mainly due to the need to constantly change the design of the database model which led to frequent deletes of the content when conflicts appeared.

The issues encountered mentioned in the previous paragraph led to a decision the re-recreate the database model. This decision delayed the development a few weeks further as there had to be some extra meetings to discuss this new model, however, it was in the project's best interest as it would resolve a lot of the pending issues the old design had, as well as provide a better footing for the new code and features. This redesign of the database model caused a lot of the old code to become almost useless as this code depended on the specific models (of the Model-View-Controller (MVC) approach) associated with the tables. Nevertheless, this redesign provided an excellent opportunity to fix other issues with the project. These issues were solved during the second semester.

One of these issues was that the development environment was running in a very outdated virtual machine that was publicly accessible, meaning it allowed access to individuals who were outside of the Department of Informatics Engineering (DEI). This was fixed by upgrading said machine and moving it to a private group. A different issue was that the programming language's version was outdated which meant that the framework itself would also be outdated. To fix this the programming language's version and the framework's respective version were incrementally upgraded so as to not cause any issues, along the way, backups of the whole machine were created to prevent any problems. This upgrade of the framework was very beneficial because it provided better security features and other Quality of Life (QoL) improvements

Furthermore, a new design of the platform was being developed by a party outside of this internship, this design would be significantly more in line with the department's image. It would also prove to have a more modern look and improved user experience. The adoption of this new design necessitated regular and frequent meetings with the entire team to address various aspects and changes throughout the process. These meetings were critical in ensuring that everyone had a shared understanding of the evolving design, that any emerging issues were promptly addressed and any feedback was heard and improved upon. However, it's worth noting that, at times, the creation of a new design did introduce some delays in the implementation phase. Along with the redesign of the database model, it meant that most if not all of the codebase had to be rewritten, and due to the new design having more intricate details, this further delayed the process. These hurdles, while they temporarily impeded progress, are sure to have contributed to the creation of a more robust and well-considered platform.

Even after the design was mostly complete since the design was basically just images, with almost no way to reliably convert this design into a responsive and mobile-friendly layout in an automatic fashion. This made it so each screen had to be manually recreated from the ground up, and as mentioned in the next section, using a CSS framework and customising its traits. This required a lot of back-and-forth in order to line up the actual screens with the design.

## 6.1 Work developed

Despite the challenges and setbacks mentioned in the previous section, some amount of progress was made in terms of feature implementation during the second semester of the internship. These features were successfully developed with the new design concepts. These designs were implemented using the Bootstrap 5 framework. These features are primarily around the company side and faculty members as well as the student side.

Tables 6.1 through 6.4 show which features are either implemented or incomplete, along with a description explaining what they do or why they are incomplete. Each feature includes full backend validation of the submitted data as well as two forms of frontend validation, one via the default browser/HTML validators and one more comprehensive in JavaScript that includes custom validations for specific fields.

Table 6.1: System-wise features that have been started

| **Feature** | **Status** | **Description** |
|---|---|---|
| SYS-4 | Incomplete | Some pages are not yet finished. These pages include the page-specific courses with course-specific calendars. |
| SYS-5 | Complete | Authentication for all user types, external and internal. |
| SYS-6 | Complete | Any logged-in user is able to change their active role to access other features, only if they have more than one role. |
| SYS-7 | Incomplete | Backend is ready, lacking a frontend button to switch between languages. |

Table 6.2: Company-wise features that have been started

| **Feature** | **Status** | **Description** |
|---|---|---|
| CMPY-1 | Complete | Registering a company. Users select signup at login page, confirm their email and fill out any additional data required. |
| CMPY-2 | Complete | Allows company managers to edit any information about the company. |
| CMPY-7 and CMPY-8 | Complete | Allows company managers to view, edit, deactivate or add any collaborators or legal representatives. |
| SYS-12 | Incomplete | Backend for this feature is ready but requires frontend implementation. |

Table 6.3: Student-wise features that have been started

| **Feature** | **Status** | **Description** |
|---|---|---|
| STUD-1 | Complete | Student when first logs in is asked to fill out missing data, they are further able to edit this information. |

Table 6.4: Faculty member-wise features that have been started

| Features | Status | Description |
|---|---|---|
| FCLT-1 | Complete | Allows the faculty member the register a new internship propose for a specific internship phase. |
| FCLT-6 | Complete | when the faculty member first logs in, they are asked to fill out missing data, they are further able to edit this information |

Furthermore, in Appendix C it's possible to see the screens referring to some of these features. Firstly, screen Figure C.1 shows the platform's current landing page, this page is the first page available to any users of the platform and allows users to see any relevant information about the internship process as well as each degree. However, this is not entirely complete as some static information is missing as of now.

The second screen, seen in Figure C.2 shows the login screen for the platform, and allows users to sign in, with their created credentials or their LDAP credentials if appropriate. Furthermore, it allows users to create their own company to submit proposals, this can be seen in Figure C.3, which shows the simple form, which requires email and password, with confirmation. After this is done an email is sent, Figure C.4, which when clicked leads to a page, Figure C.5, which asks for the required company information. For a company to be able to submit any proposals, this form must be filled out and submitted. Afterwards, the user is required to login with their created information, if they do this they end up on page Figure C.6 which shows the appropriate options for the current active role (sign in the top right of the page).

The next screen shows the company data feature where company managers are allowed to edit important information about the company. Following that, in Figure C.8 we can see the page where it's possible to view, add, edit or deactivate/activate a specific collaborator, manager or legal representative.

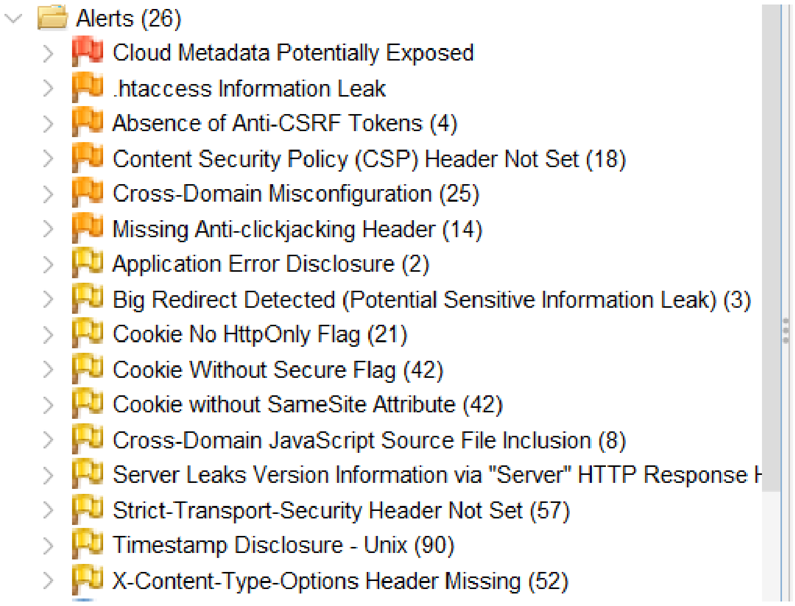
Finally, the last screen, Figure A.3 on the list shows the form for submitting proposals and all the relevant fields, Additionally it also shows examples of how form validation works in the platform, this functionality is currently only available for faculty members.

## 6.2 Testing

Testing for the project was conducted by a combination of faculty members and peers. The testing was primarily informal, by reviewing the functionality and user experience, as well as seeking feedback from other users. Formal testing was not a primary focus during the development process due to time constraints. Additionally, the informal testing also allowed for quick identification and resolution of any bugs or technical issues that arose during the development process. Valuable insights were gathered and necessary adjustments to the platform were made, which helped to ensure that the final product met the needs of the stakeholders.

This testing was done by repeatedly testing the developed features using the available interface and trying as many possible combinations of input as possible. This included but was not limited to, sending foreign characters, sending text longer than expected, copying and pasting text from other sources, and purposefully entering text in wrong formats (such as entering text in a number's field) while ignoring the errors from the frontend validation. Some forms of basic attacks were tried such as SQL injections.

A quick active scan using OWASP ZAP shows some concerning alerts, see Figure 6.1, however, upon further look most of these were shown to be false positives, more specifically the higher severity ones. The cloud Metadata one is a false flag because the page returned a JSON object as expected. The .htaccess leak is also a false positive as the page returns a 404 error. The error pertaining to the absence of anti-CSRF tokens is also a false flag as the tokens are there, just not in the format expected by the tool. Cross-domain Misconfigurations are mitigated because no sensitive file is available in an authenticated manner. Since this is the development environment and not everything is fully set up, some of these issues would not arise in the production environment. A couple of issues such as the cross-domain JavaScript Source file Inclusions should be fixed and can be done by downloading the respective file and including it locally.



Alerts (26)
- Cloud Metadata Potentially Exposed
- .htaccess Information Leak
- Absence of Anti-CSRF Tokens (4)
- Content Security Policy (CSP) Header Not Set (18)
- Cross-Domain Misconfiguration (25)
- Missing Anti-clickjacking Header (14)
- Application Error Disclosure (2)
- Big Redirect Detected (Potential Sensitive Information Leak) (3)
- Cookie No HttpOnly Flag (21)
- Cookie Without Secure Flag (42)
- Cookie without SameSite Attribute (42)
- Cross-Domain JavaScript Source File Inclusion (8)
- Server Leaks Version Information via "Server" HTTP Response H
- Strict-Transport-Security Header Not Set (57)
- Timestamp Disclosure - Unix (90)
- X-Content-Type-Options Header Missing (52)

Figure 6.1: Active scan on the platform

# Chapter 7

# Conclusion

This thesis aims to present the development and implementation of a new internship platform for internships. With the objective of addressing the multiple restrictions and lack of automation from the previous platform, through the use of various practices such are functional requirement specifications.

One of the key challenges faced during the development process in the first semester was time constraints stemming from several other subjects and exams as well as there being strict deadlines that needed to be met for the platform to implement certain features. In the second semester setbacks with the deployment, redesign and integration of the design as well as with the update to the environment and framework proved to be quite challenging to overcome and further contributed to the delay of the overall project.

Despite these challenges, some of the planned features for the platform are operations, namely for the company side and faculty member's side. The new design concept for the platform was successfully completed as well as the very needed upgrade, which provides a good base for future development of the platform, hopefully improving upon what was previously had. These challenges provided a valuable learning opportunity and experience when it comes to working in the real world with real constraints.

As this thesis wraps up, it's important to recognise that this project is not just a culmination but a starting point. There is more work to be done, the internship platform will continue to evolve and develop, addressing the upcoming challenges. Hopefully, this platform will enhance the internship experience for students, and facilitate operations for faculty members and companies.

# References

Chapter 10: Moscow prioritisation. URL https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html.

Introduction · bootstrap v5.0. URL https://getbootstrap.com/docs/5.0/getting-started/introduction/.

Eloquent: Getting started - the php framework for web artisans. URL https://laravel.com/docs/10.x/eloquent#introduction.

Frontend - laravel 10.x - the php framework for web artisans. URL https://laravel.com/docs/10.x/frontend#growing-expectations.

Laravel usage, laravel benefits, laravel drawback, the code box, a. URL https://shorturl.at/nPRZ7s.

Laravel: Celebrating 12 years of powering php development, b. URL https://www.linkedin.com/pulse/laravel-celebrating-12-years-powering-php-development.

Php laravel - pros and cons, c. URL https://www.linkedin.com/pulse/php-laravel-pros-cons-spdeveloper.

Postgresql vs mysql - difference between relational database management systems (rdbms) - aws. URL https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/?nc1=h_ls.

Php: Introduction - manual. URL https://www.php.net/manual/en/intro.pdo.php.

Round 21 results - techempower framework benchmarks. URL https://www.techempower.com/benchmarks/#section=data-r21&hw=ph&test=composite&s=1&c=a&l=zik073-6bj.

Timeboxing - wikipedia. URL https://en.wikipedia.org/wiki/Timeboxing.

Why laravel - the php framework for web artisans. URL https://laravel.com/docs/10.x#why-laravel.

S Bradner. Key words for use in rfcs to indicate requirement levels, 1997. URL https://www.rfc-editor.org/rfc/rfc2119.

Simon Brown. The c4 model for visualising software architecture. URL https://c4model.com/.

Oleksii Dushenin. Monolithic architecture. advantages and disadvantages | by oleksii dushenin | medium. URL https://datamify.medium.com/monolithic-architecture-advantages-and-disadvantages-e71a603eec89.

Chandler Harris. Microservices vs. monolithic architecture | atlassian. URL https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith.

OWASP. Password plaintext storage | owasp foundation. URL https://owasp.org/www-community/vulnerabilities/Password_Plaintext_Storage. Accessed on 15/01/2023.

# Appendices

# Appendix A

# Stakeholder flowcharts

.

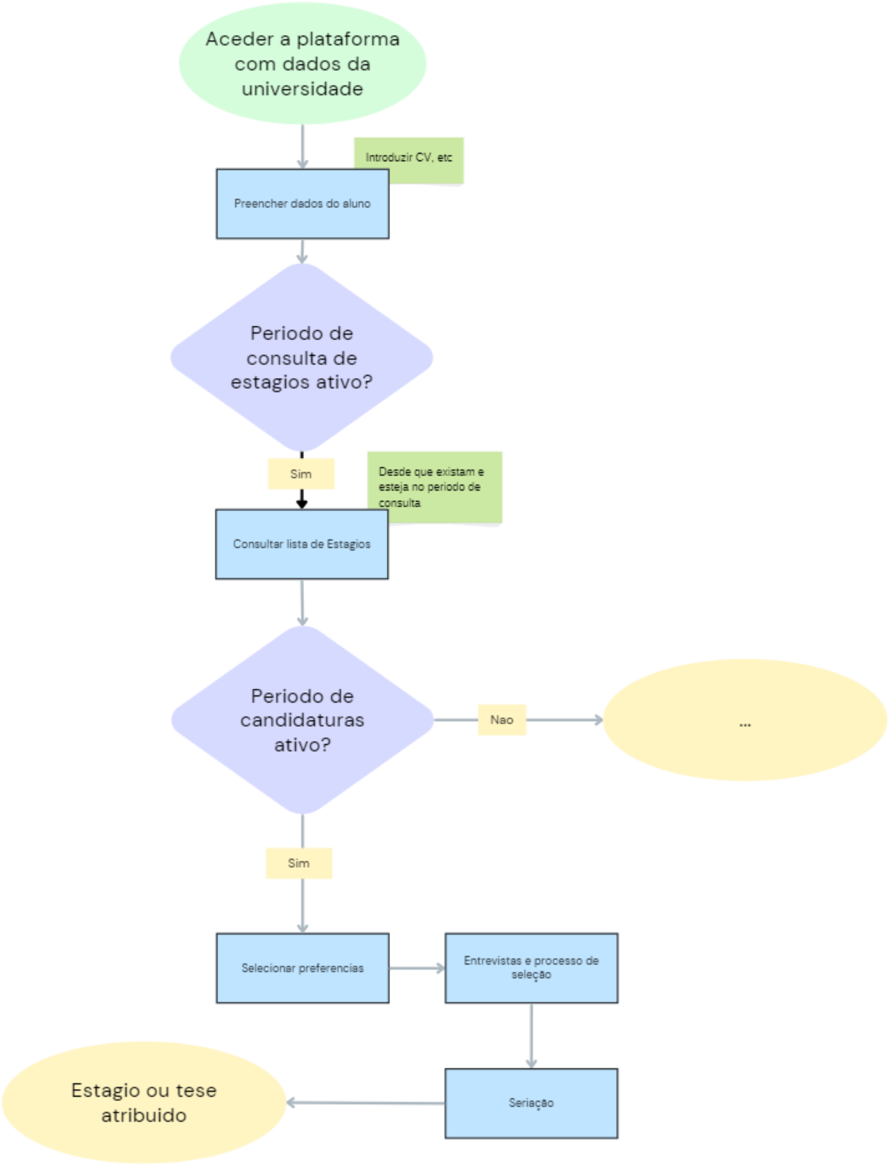

Aceder a plataforma com dados da universidade
Introduzir CV, etc
Preencher dados do aluno
Periodo de consulta de estagios ativo?
Sim
Desde que existam e esteja no periodo de consulta
Consultar lista de Estagios
Periodo de candidaturas ativo?
Nao
...
Sim
Selecionar preferencias
Entrevistas e processo de seleção
Seriação
Estagio ou tese atribuido

Figure A.1: Flowchart for the Student

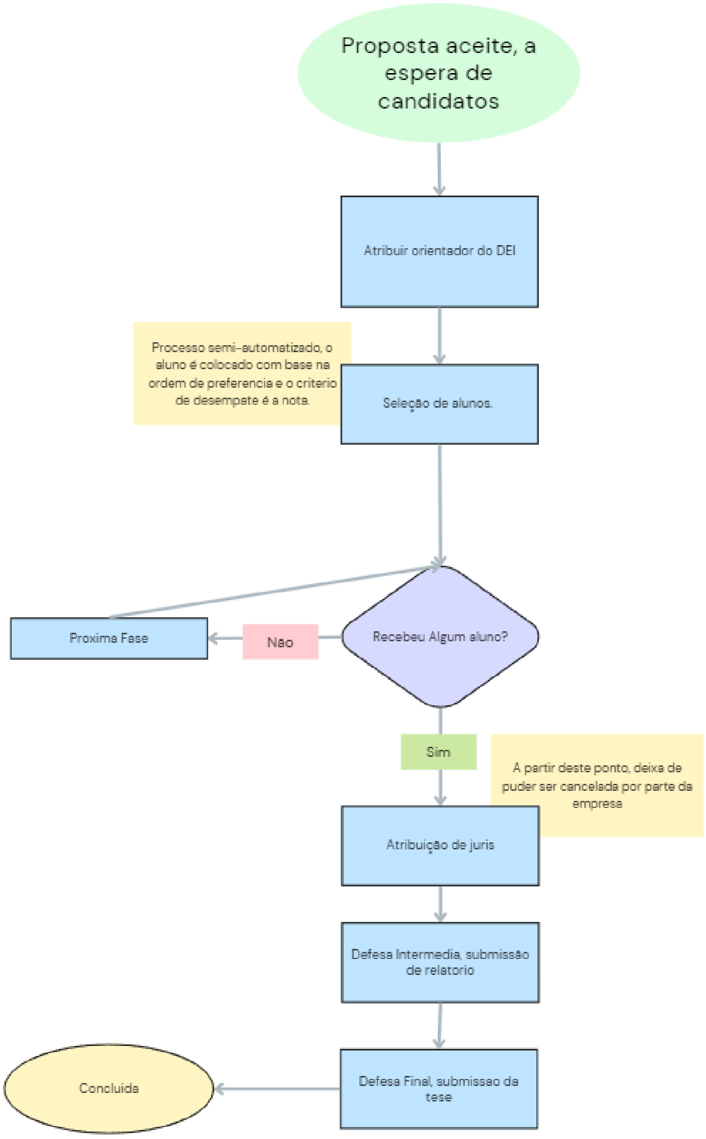

Proposta aceite, a espera de candidatos
Atribuir orientador do DEI
Processo semi-automatizado, o aluno é colocado com base na ordem de preferencia e o criterio de desempate é a nota.
Seleção de alunos.
Proxima Fase
Nao
Recebeu Algum aluno?
Sim
A partir deste ponto, deixa de puder ser cancelada por parte da empresa
Atribuição de juris
Defesa Intermedia, submissão de relatorio
Concluida
Defesa Final, submissao da tese

Figure A.2: Flowchart for the placement process

.

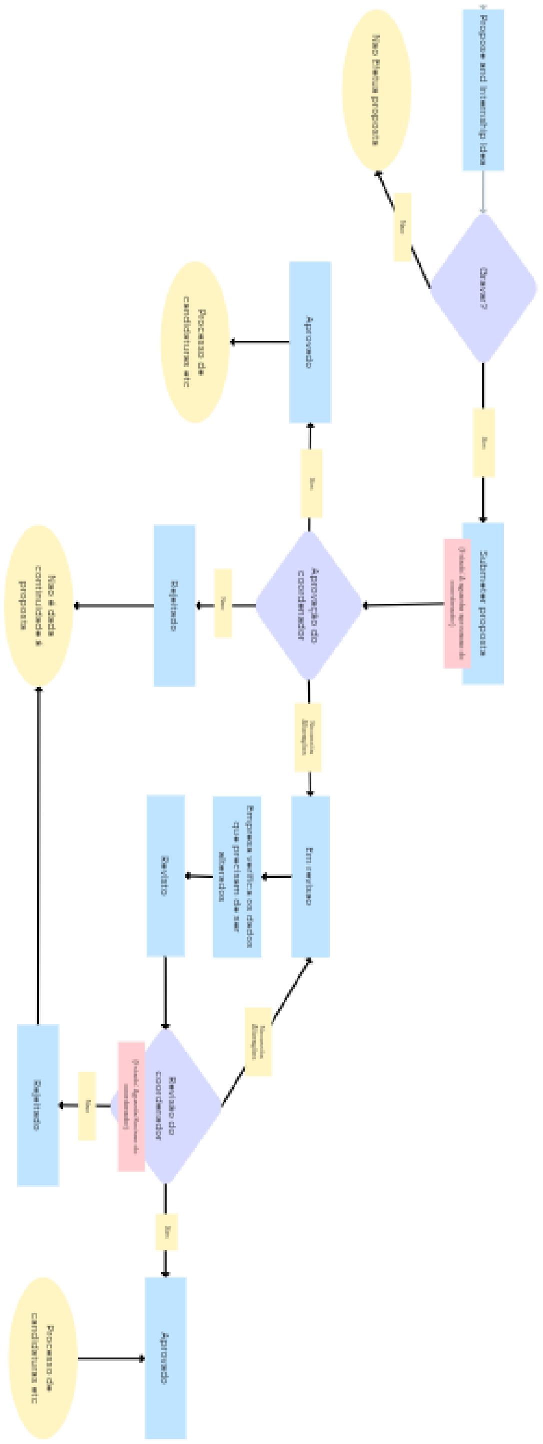

Propose and internship idea
Gravar?
Não Efetua proposta
Submeter proposta
Aprovação do coordenador
Aprovado
Processo de candidaturas etc
Rejeitado
Não é dada continuidade à proposta
Em revisão
Empresa verifica os dados que precisam de ser alterados
Revisto
Revisão do coordenador
Rejeitado
Aprovado
Processo de candidaturas etc

Figure A.3: Flowchart for the placement process

# Appendix B

# Functional requirements tables

Table B.1: Functional Requirements (FRs) for Company Manager

| ID | Name | Stkh | Description | Have |
|---|---|---|---|---|
| CMPY-1 | register_company | STKH_1 | Registering a company in the platform would allow the creation of new proposals, and managing students. | Must |
| CMPY-2 | manage_company_info | | Allows adding or removing information related to the company at hand. | Must |
| CMPY-3 | create_new_proposal | | Allows the company to create a new internship proposal. | Must |
| CMPY-4 | edit_proposal | | Allows editing details about the proposal. | Must |
| CMPY-5 | bring_forward_proposal | | Brings the proposal forward into a new period in case it was not completed in the last period. | Should |
| CMPY-6 | review_proposal | | Allows the company to review a proposal that has been sent back to be reviewed by the administration of the platform. | Must |
| CMPY-7 | manage_legal_rep | | Allows the addition or removal of Legal Representatives. | Must |

Continued on next page

Table B.1: FRs for Company Manager (Continued)

| | | | | |
|---|---|---|---|---|
| CMPY-8 | manage_colab | | Allows the addition or removal of collaborators. | Must |
| CMPY-9 | see_candidates | | Allows checking who has submitted a candidacy for a given proposal. | Must |
| CMPY-10 | manage_candidates | | Allows managing of candidates, meaning they can approve any candidate for a proposal. | Must |
| CMPY-11 | digital_signing_proto | | Allows companies to digitally sign the internship protocol. | Could |
| CMPY-12 | recover_access | | Allows companies to follow a procedure that lets them recover access to their account in case all accesses have been lost for any reason. | Should |

Table B.2: FRs for Student

| ID | Name | Stkh | Description | Have |
|---|---|---|---|---|
| STUD-1 | fill_personal_data | STKH_2 | The student is able to fill out his personal information regarding his academic performance and curriculum vitae. | Must |
| STUD-2 | view_proposals | | The student is able to view and review every proposal available and its details. | Must |
| STUD-3 | select_preferences | | The student is able to select 5 proposals in order of preference. | Must |
| STUD-4 | view_attributed | | The student is able to review the proposal to which they were assigned, as well as the details thereof. | Must |
| STUD-5 | view_dates | | The student is able to view a calendar of the dates his deliveries and milestones are due. | Should |
| STUD-6 | submit_delivery | | Allows the student to submit a delivery. | Must |
| STUD-7 | reorder_preferences | | Allows the student to reorder their preferences during the allowed period. | Must |

Table B.3: FRs for degree coordinators

| ID | Name | Stkh | Description | Have |
|---|---|---|---|---|
| CORD-1 | assign_jury | STKH_5 | Allows the coordinators to assign juries to a given proposal. | Must |
| CORD-2 | manage_proposal | STKH_5 | Allows the coordinator to manage all submitted proposals by approving, rejecting or requesting an edit for a given proposal | Must |
| CORD-3 | assign_student | STKH_5 | Allows the coordinator to assign students to a given proposal. | Must |
| CORD-4 | personal_data | STKH_5 | Allows the coordinator change their personal information | |

Table B.4: FRs for platform manager

| ID | Name | Stkh | Description | Have |
|---|---|---|---|---|
| ADMN-1 | manage_company | STKH_9 | Allows the coordinators to manage all aspects of a given company so that accesses (for example lost passwords or emails) can be restored or proposals deleted. | Must |
| ADMN-2 | manage_users | STKH_9 | Allows the platform manager to activate or deactivate any given user as well as add or remove roles. | Must |
| ADMN-3 | manage_periods | STKH_9 | Allows the coordinators to set up new internship phases. | Must |

Table B.5: FRs for DEI advisors and company advisors

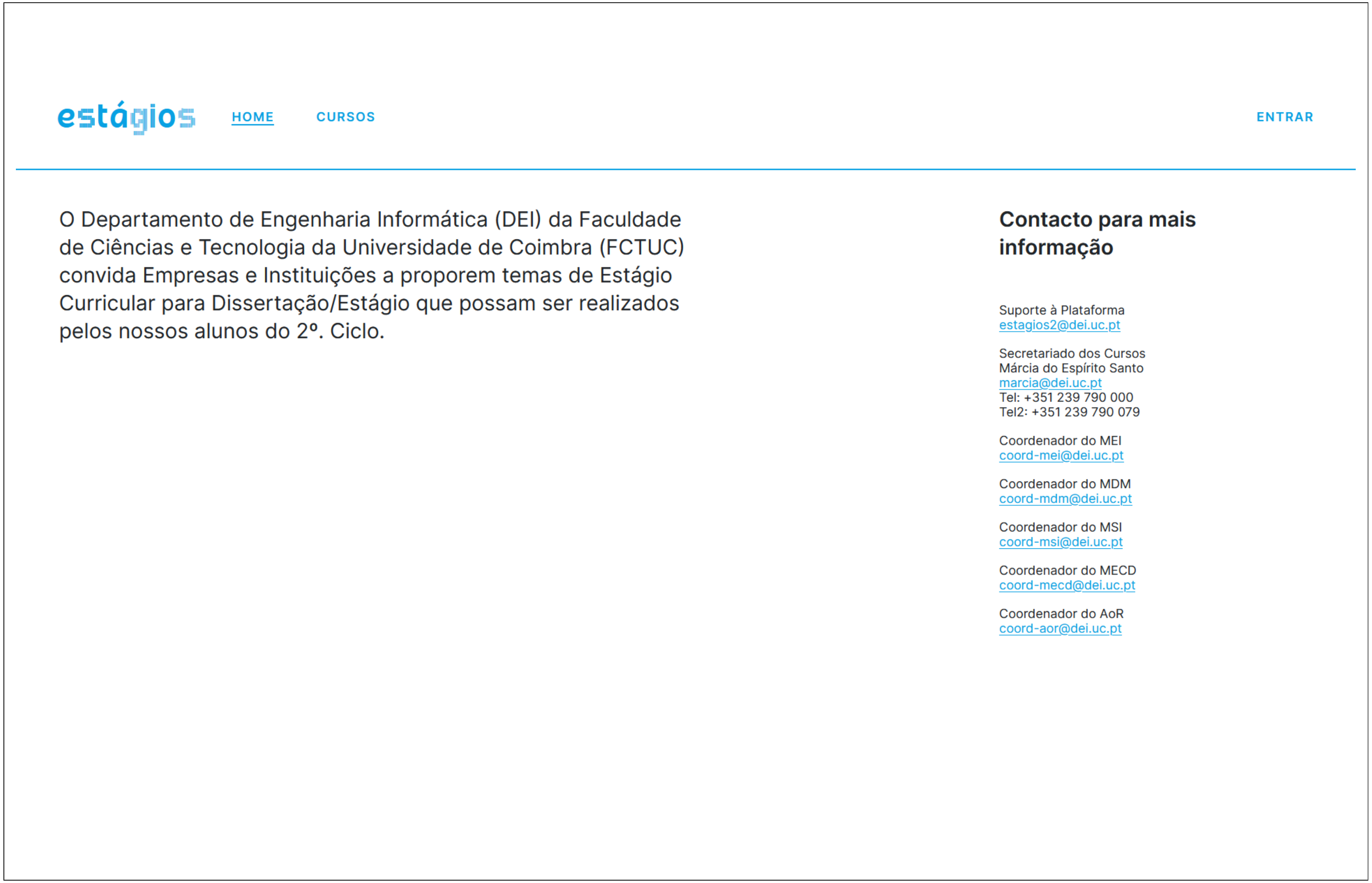
| ID | Name | Stkh | Description | Have |
|---|---|---|---|---|
| ADVS-1 | see_proposal | STKH_5, STKH_6 | Allows advisors the see which proposals they have been assigned to. | Must |
| ADVS-2 | see_student | | Allows advisors to see students assigned to their proposals. | Must |
| ADVS-3 | see_candidates | | Allows advisors to see candidates for their proposals. | Must |
| ADVS-4 | manage_candidates | STKH_5, STKH_6 | Allows advisors to manage candidates (accept or reject) for a given proposal. | Must |
| ADVS-5 | submit_reports | | Allows advisors to submit meeting summaries. | Must |
| ADVS-6 | see_reports | | Allows advisors to see meeting summaries for all their assigned proposals, even if they were submitted by a co-advisor. | Must |

Table B.6: FRs for faculty members

| ID | Name | Stkh | Description | Have |
|---|---|---|---|---|
| FCLT-1 | create_new_proposal | STKH_8 | Allows the faculty member to register a new internship proposal. | Must |
| FCLT-2 | edit_proposal | | Allows the faculty member to edit the details of an internship proposal. | Must |
| FCLT-3 | review_proposal | | Allows the faculty member to edit the proposal after it has been reviewed by the administration or coordination of the platform. | Must |
| FCLT-4 | see_candidates | | Allows the faculty members to see the candidates for a given proposal. | Must |
| FCLT-5 | manage_candidates | | Allows the faculty members to manage, meaning they can approve or reject the candidates for a given proposal. | Must |
| FCLT-6 | personal_data | | Allows the faculty member to change their personal information | Must |

# Appendix C

# Screenshot of current platform



O Departamento de Engenharia Informática (DEI) da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC) convida Empresas e Instituições a proporem temas de Estágio Curricular para Dissertação/Estágio que possam ser realizados pelos nossos alunos do 2º. Ciclo.

Contacto para mais informação

Suporte à Plataforma
estagios2@dei.uc.pt

Secretariado dos Cursos
Márcia do Espírito Santo
marcia@dei.uc.pt
Tel: +351 239 790 000
Tel2: +351 239 790 079

Coordenador do MEI
coord-mei@dei.uc.pt

Coordenador do MDM
coord-mdm@dei.uc.pt

Coordenador do MSI
coord-msi@dei.uc.pt

Coordenador do MECD
coord-mecd@dei.uc.pt

Coordenador do AoR
coord-aor@dei.uc.pt

Figure C.1: Platform's home page before any login.

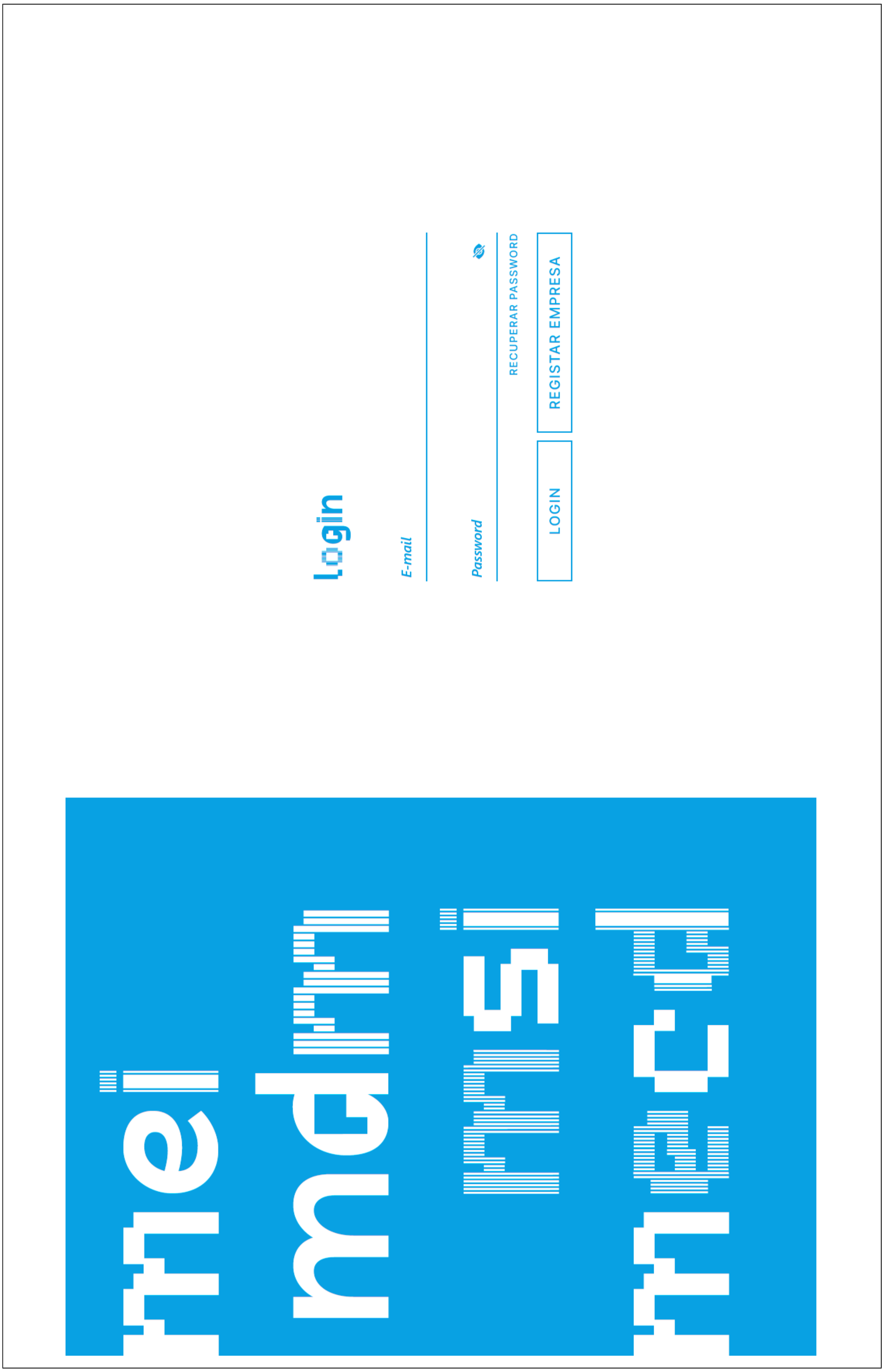


Figure C.2: Platform's login page

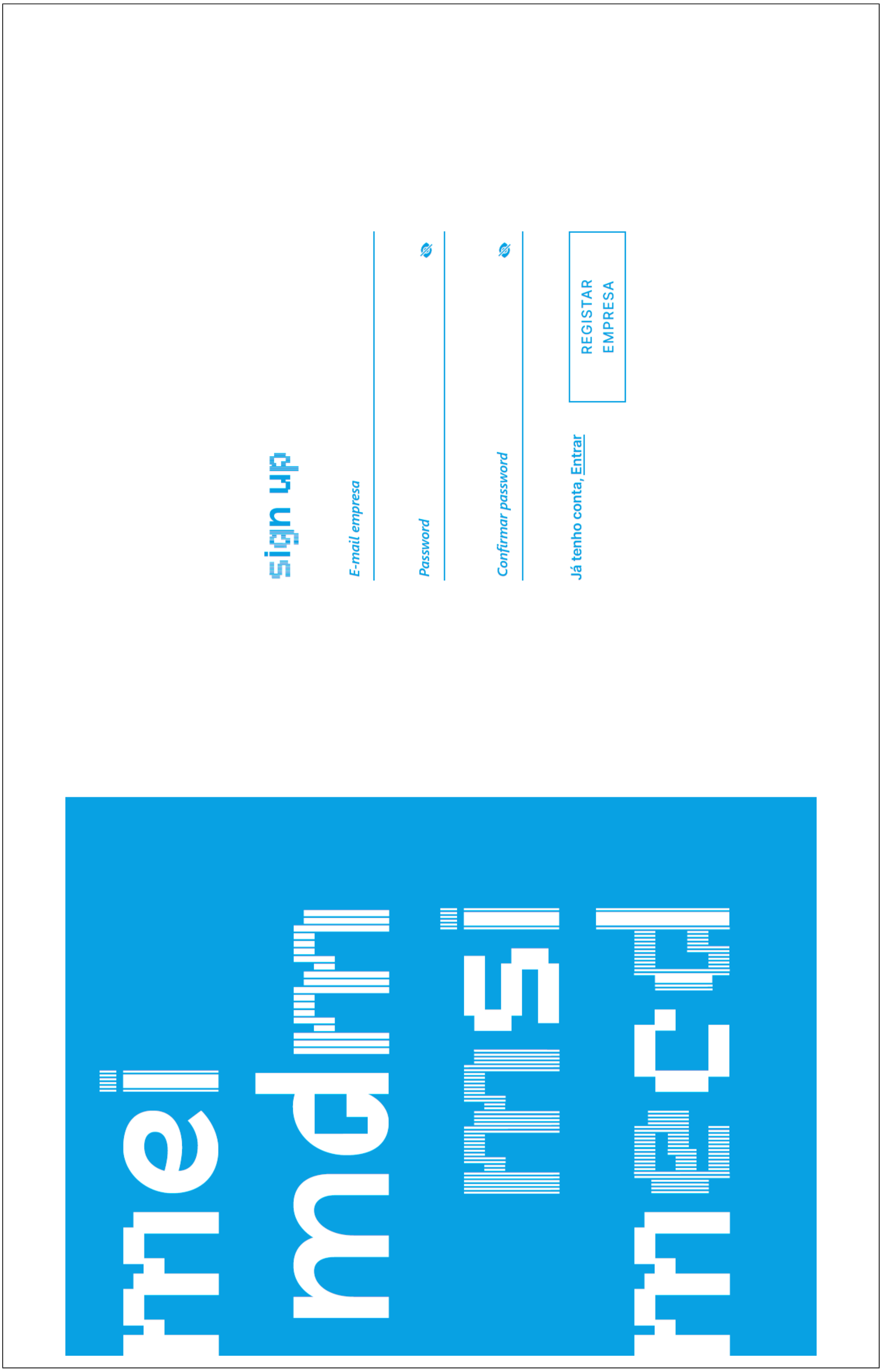
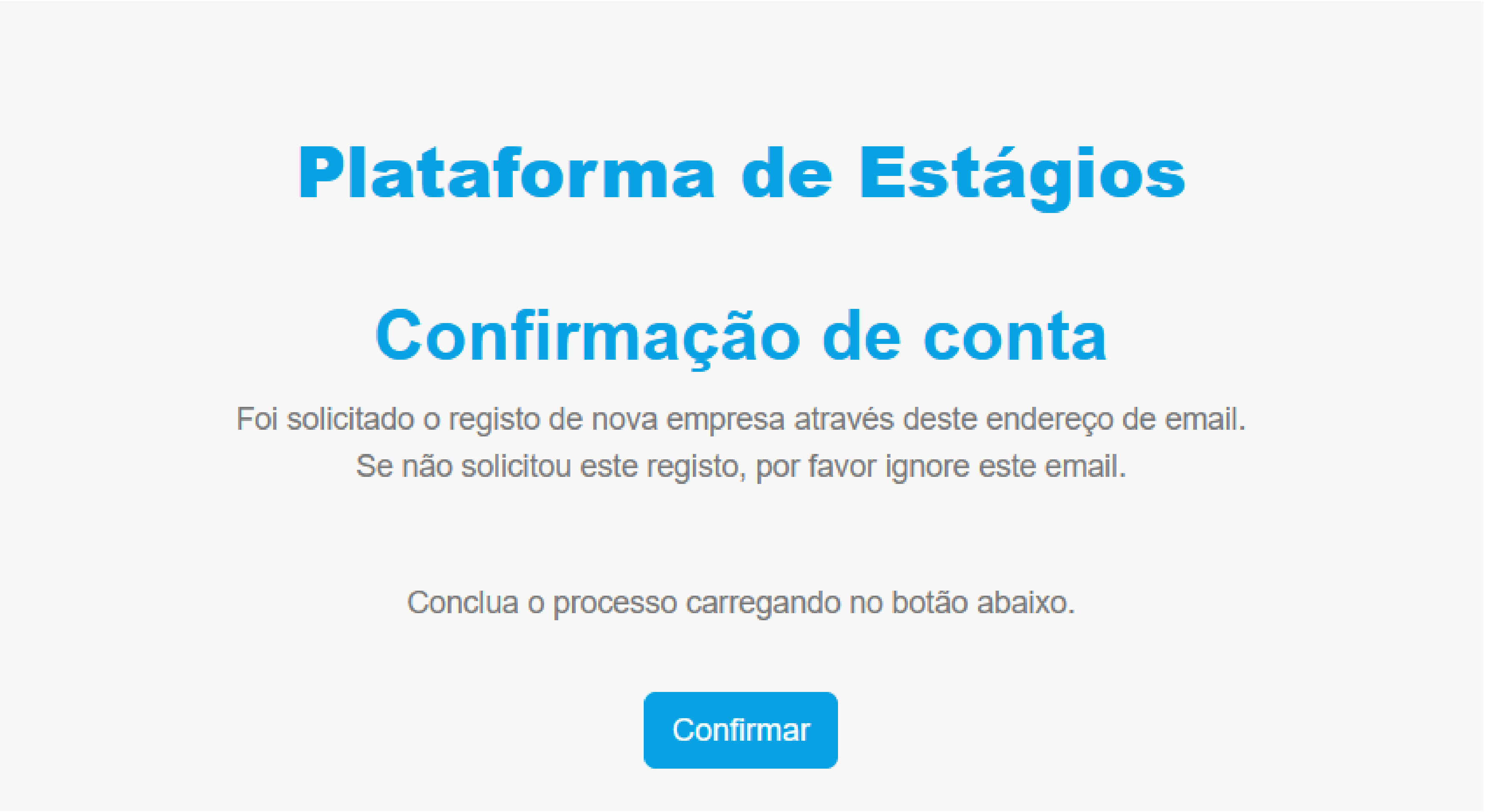


Figure C.3: Platform's company signup page



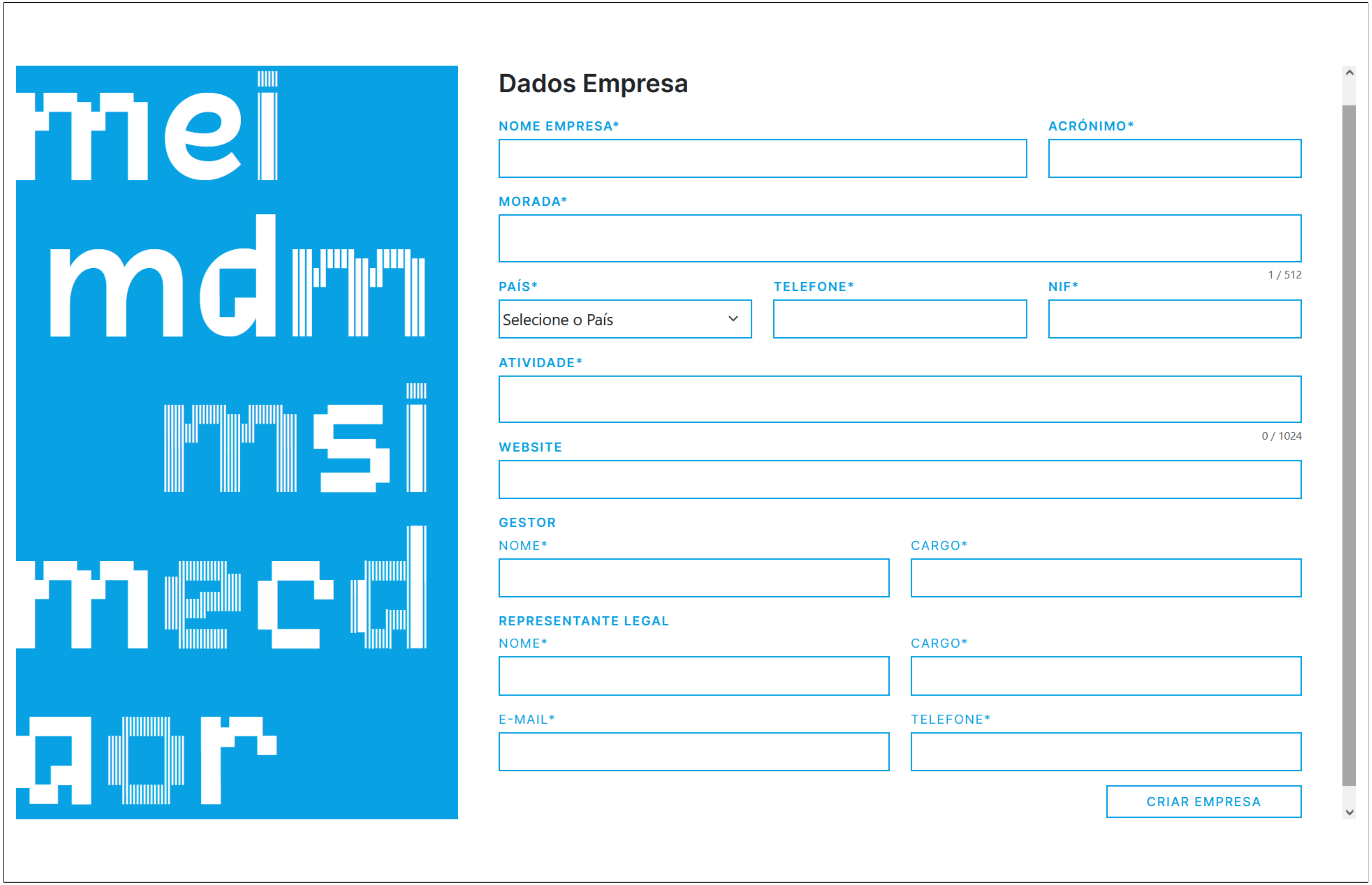
Plataforma de Estágios

Confirmação de conta

Foi solicitado o registo de nova empresa através deste endereço de email.
Se não solicitou este registo, por favor ignore este email.

Conclua o processo carregando no botão abaixo.

Figure C.4: Confirmation email sent to companies



## Dados Empresa

NOME EMPRESA*

ACRÓNIMO*

MORADA*

1 / 512

PAÍS*: Selecione o País ⌄

TELEFONE*

NIF*

ATIVIDADE*

0 / 1024

WEBSITE

GESTOR

NOME*

CARGO*

REPRESENTANTE LEGAL

NOME*

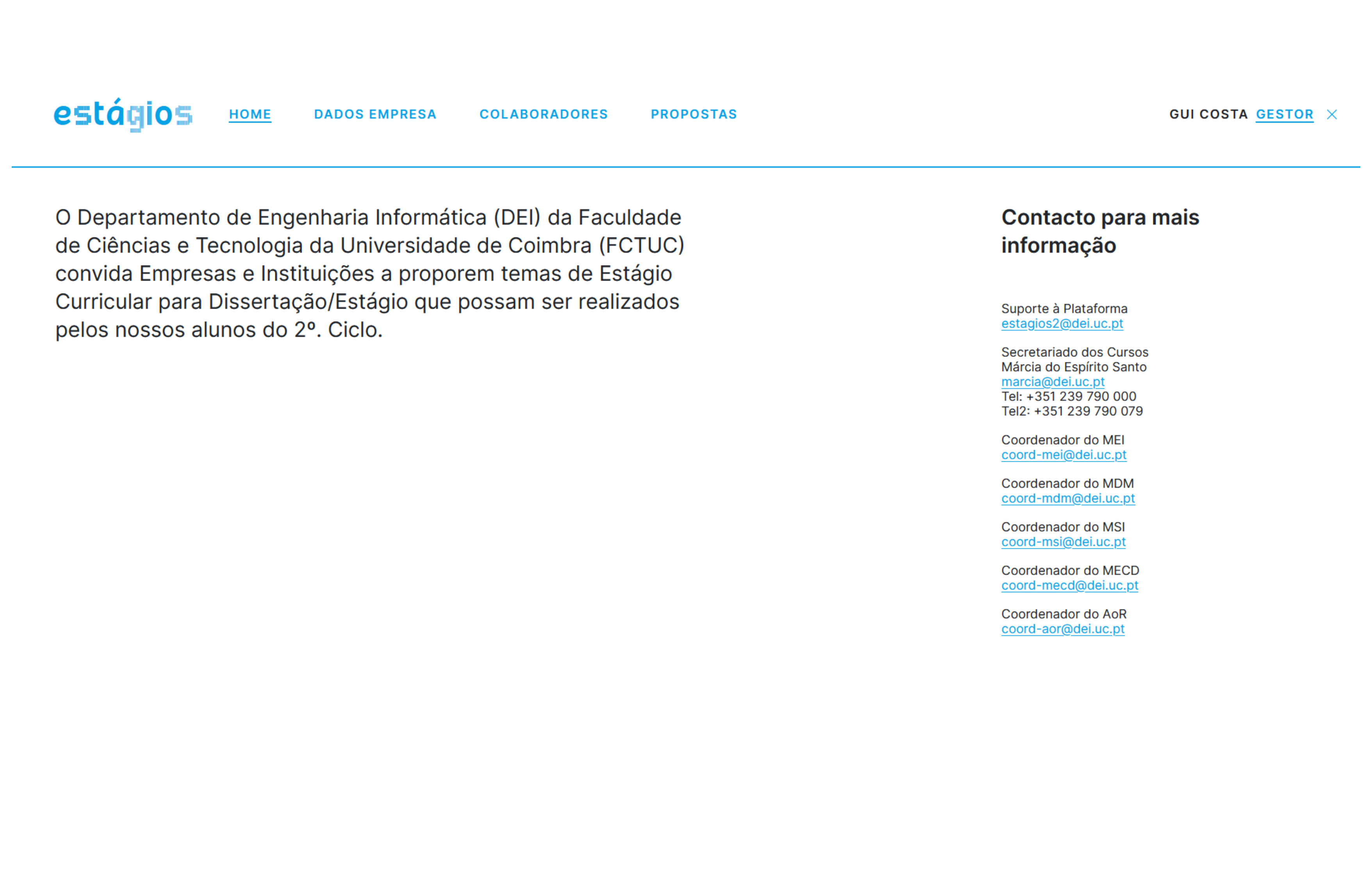
CARGO*

E-MAIL*

TELEFONE*

CRIAR EMPRESA

Figure C.5: Page shown after email confirmation



O Departamento de Engenharia Informática (DEI) da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC) convida Empresas e Instituições a proporem temas de Estágio Curricular para Dissertação/Estágio que possam ser realizados pelos nossos alunos do 2º. Ciclo.

Contacto para mais informação

Suporte à Plataforma
estagios2@dei.uc.pt

Secretariado dos Cursos
Márcia do Espírito Santo
marcia@dei.uc.pt
Tel: +351 239 790 000
Tel2: +351 239 790 079

Coordenador do MEI
coord-mei@dei.uc.pt

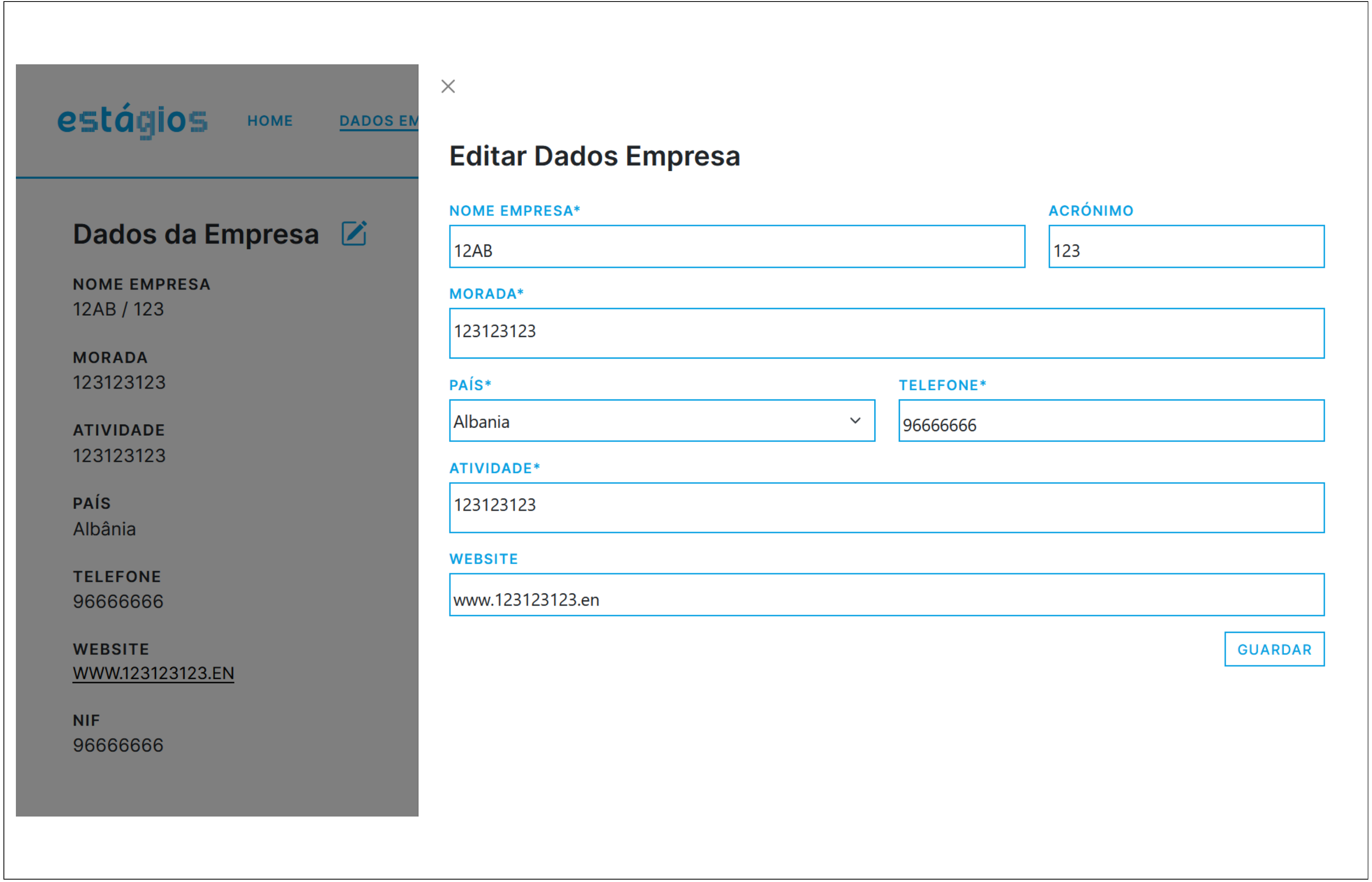
Coordenador do MDM
coord-mdm@dei.uc.pt

Coordenador do MSI
coord-msi@dei.uc.pt

Coordenador do MECD
coord-mecd@dei.uc.pt

Coordenador do AoR
coord-aor@dei.uc.pt

Figure C.6: Home page after company manager login



Dados da Empresa

NOME EMPRESA
12AB / 123

MORADA
123123123

ATIVIDADE
123123123

PAÍS
Albânia

TELEFONE
96666666

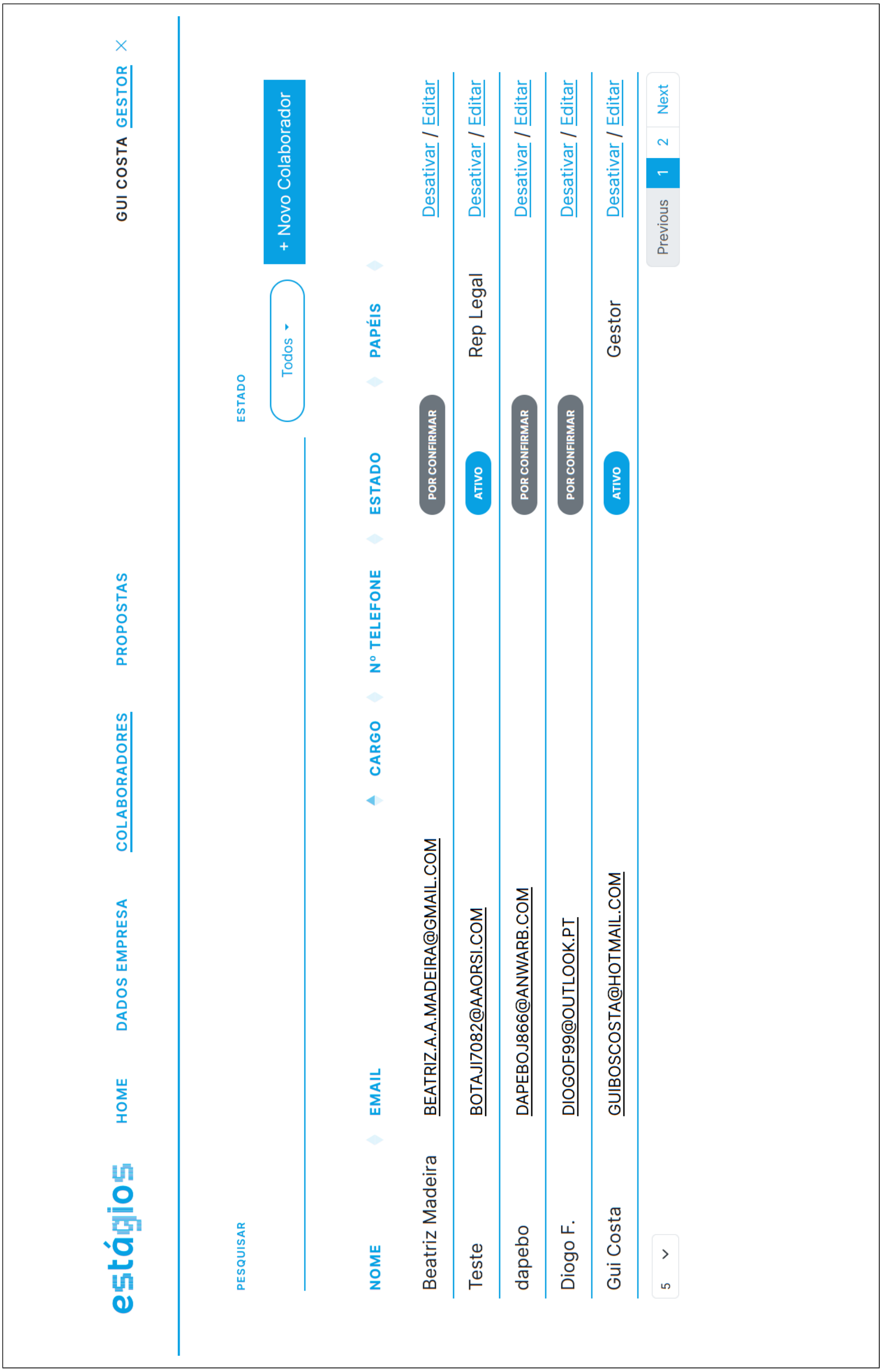
WEBSITE
WWW.123123123.EN

NIF
96666666

Editar Dados Empresa

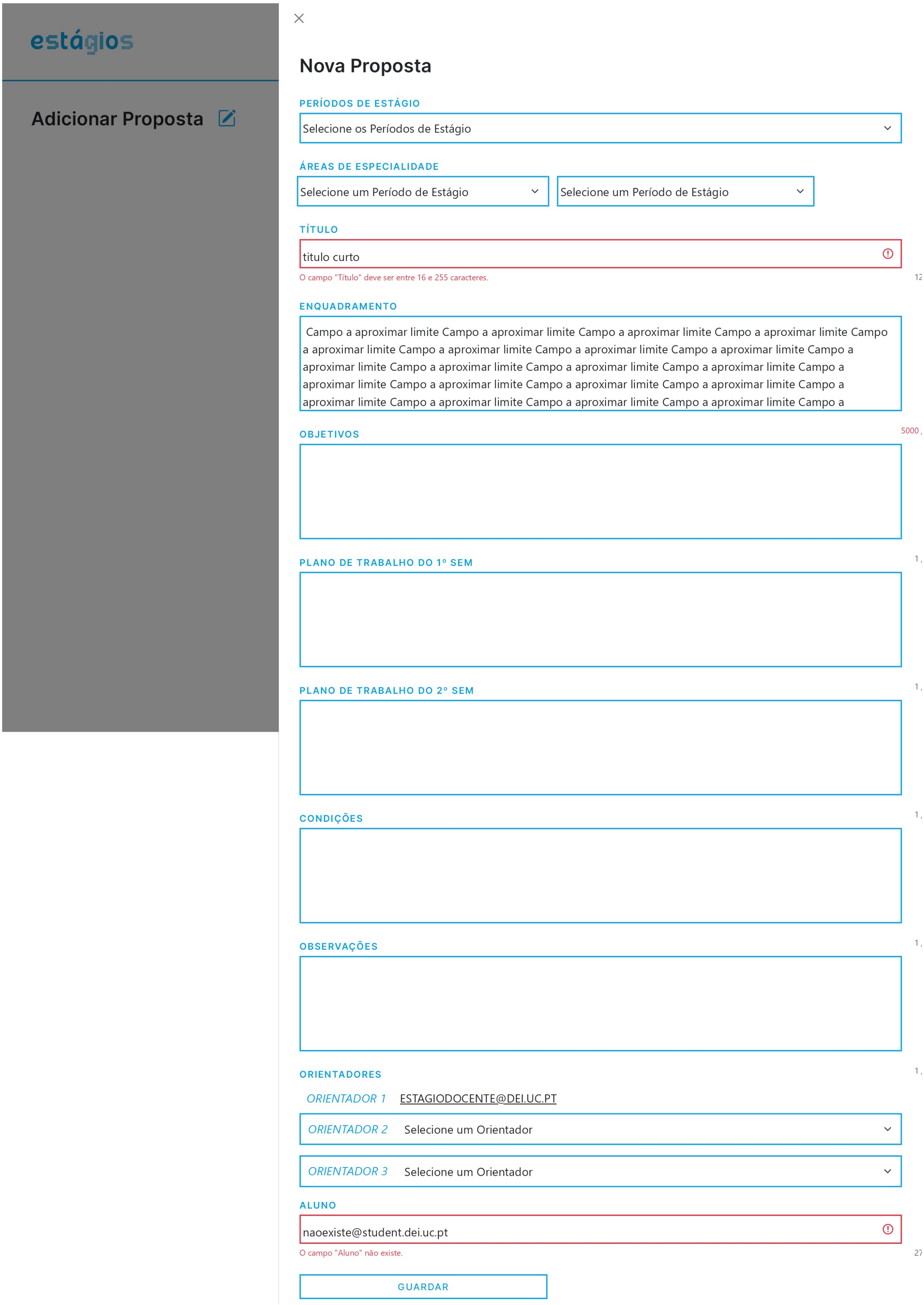
| | |
|---|---|
| NOME EMPRESA* | 12AB |
| ACRÓNIMO | 123 |
| MORADA* | 123123123 |
| PAÍS* | Albania |
| TELEFONE* | 96666666 |
| ATIVIDADE* | 123123123 |
| WEBSITE | www.123123123.en |

Figure C.7: Company’s data form



| NOME | EMAIL | CARGO | Nº TELEFONE | ESTADO | PAPÉIS | |
|---|---|---|---|---|---|---|
| Beatriz Madeira | BEATRIZ.A.A.MADEIRA@GMAIL.COM | | | POR CONFIRMAR | | Desativar / Editar |
| Teste | BOTAJI7082@AAORSI.COM | | | ATIVO | Rep Legal | Desativar / Editar |
| dapebo | DAPEBOJ866@ANWARB.COM | | | POR CONFIRMAR | | Desativar / Editar |
| Diogo F. | DIOGOF99@OUTLOOK.PT | | | POR CONFIRMAR | | Desativar / Editar |
| Gui Costa | GUIBOSCOSTA@HOTMAIL.COM | | | ATIVO | Gestor | Desativar / Editar |

Figure C.8: Colaborators page for compaby



estágios

Adicionar Proposta

×

Nova Proposta

PERÍODOS DE ESTÁGIO

Selecione os Períodos de Estágio

ÁREAS DE ESPECIALIDADE

Selecione um Período de Estágio | Selecione um Período de Estágio

TÍTULO

titulo curto

O campo "Título" deve ser entre 16 e 255 caracteres. 12

ENQUADRAMENTO

Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a aproximar limite Campo a

5000 ,

OBJETIVOS

1 ,

PLANO DE TRABALHO DO 1º SEM

1 ,

PLANO DE TRABALHO DO 2º SEM

1 ,

CONDIÇÕES

1 ,

OBSERVAÇÕES

1 ,

ORIENTADORES

ORIENTADOR 1 ESTAGIODOCENTE@DEI.UC.PT

ORIENTADOR 2 Selecione um Orientador

ORIENTADOR 3 Selecione um Orientador

ALUNO

naoexiste@student.dei.uc.pt

O campo "Aluno" não existe. 27

GUARDAR

Figure C.9: Sumbit proposal page for faculty members