



UNIVERSIDADE D
COIMBRA

Gonçalo Filipe Fernandes São Marcos

PORTAL DE GESTÃO DE OPERAÇÕES COM
EQUIPAMENTOS INFORMÁTICOS

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

Julho 2023



DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA
FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Gonçalo Filipe Fernandes São Marcos

PORTAL DE GESTÃO DE OPERAÇÕES COM EQUIPAMENTOS INFORMÁTICOS

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

Julho 2023



DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA
FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Gonçalo Filipe Fernandes São Marcos

MANAGEMENT PORTAL FOR COMPUTER EQUIPMENT OPERATIONS

**Dissertation in the context of the Master in Informatics Engineering,
specialization in Software Engineering, advised by Professor Nuno
Laranjeiro, PhD and Rui Abreu, MSc and presented to the Department of
Informatics Engineering of the Faculty of Sciences and Technology of the
University of Coimbra.**

July 2023

Agradecimentos

Agradeço ao Professor Orientador Nuno Laranjeiro e ao Rui Abreu por todo o trabalho realizado como orientadores ao longo de todo o estágio, disponibilizando sempre o seu tempo e conhecimento para auxiliar quando necessário.

À empresa Grama por ter apostado em mim para integrar a equipa de desenvolvimento de um projeto tão interessante e enriquecedor. Aos meus colegas Francisco e José que realizaram o seu estágio em simultâneo com o meu e que foram sempre uma fonte de ajuda, motivando o trabalho e aprendizagem mútua mas sempre com espírito de diversão e fraternidade. Agradeço também aos respetivos orientadores da Grama, Arnaldo Moura e Marcos Calvo por estarem sempre dispostos a ajudar, partilhando o seu vasto conhecimento técnico na área. Agradeço também à Susana Lourenço da Grama por toda a disponibilidade e por todo contributo direto e indireto que concedeu durante este estágio.

Quero agradecer também à minha família, especialmente aos meus pais por todo o apoio incondicional durante toda a minha vida e por acreditarem sempre em mim. Um agradecimento particular à minha tia por todo o apoio especialmente durante o meu percurso académico.

Por fim, agradecer também aos meus amigos por todos os momentos felizes e de amizade que me ajudaram a manter a motivação e alegria ao longo deste percurso.

Resumo

A gestão interna de uma empresa envolve diversos processos que por vezes se tornam complexos e cuja simplificação e otimização são um ponto chave para o bom funcionamento da mesma. Entre eles, podemos encontrar a gestão de colaboradores, equipamentos, férias e recrutamento que envolvem inúmeros desafios desde o simples armazenamento e organização das informações pessoais de cada interveniente até à alocação de equipamento e gestão dos orçamentos monetários e de férias dos colaboradores. Estes processos de gestão podem tornar-se complexos se não forem usadas ferramentas que auxiliem na sua realização. Atualmente as diversas funcionalidades de recursos humanos da Grama estão confiadas a uma plataforma externa contratualizada para o efeito e a gestão de equipamentos encontra-se espalhada por diversos documentos de uso manual. A informação encontra-se descentralizada tornando a sua consulta pouco eficiente e trabalhosa. A edição da informação torna-se suscetível a erros devido à sua descentralidade e aos métodos manuais utilizados.

Este estágio tem como objetivo construir uma primeira versão da Ferramenta Interna da Grama, que procura implementar uma base sólida para as atuais e futuras necessidades da empresa através do estudo, planeamento e implementação da mesma. O presente relatório e estágio focam-se na resolução de uma parte do problema: a gestão dos colaboradores da empresa e dos equipamentos. É importante a simplificação e automatização destes processos, eliminando a necessidade de documentos soltos, editados manualmente e com elevada suscetibilidade a erros.

Palavras-Chave

Gestão de Equipamentos Informáticos, Gestão de Recursos Humanos, Metodologia Ágil, Microsserviços, Requisitos de Software

Abstract

A company's internal management usually involves several different processes that can sometimes become complex and whose simplification and optimization are a key point for its proper functioning. Among them, we can find the management of employees, equipment, vacations and recruitment that involve numerous challenges from the simple storage and organization of the personal information of each participant to the allocation of equipment and management of the employees' monetary and vacation budgets. These management processes can become complex if the correct tools are not used. Currently, the various functionalities of Grama's human resources are entrusted to an external platform contracted for the purpose and the management of equipment is scattered over several manually edited documents. The information is therefore decentralized, making its consultation inefficient and strenuous. Editing the information becomes error-prone due to the decentralization and manual methods used.

This internship aims to build a first version of the Grama Internal Tool, which seeks to implement a solid foundation for the company's current and future needs through its study, planning and implementation. This report and internship focus on solving one part of the problem: the management of the company's employees and equipment. It is of utmost importance to simplify and automate the currently used processes, eliminating the need for loose documents, edited manually and with a high susceptibility to errors.

Keywords

Agile Development, Employee Management, IT Equipment Management, Microservices, Software Requirements

Conteúdo

1	Introdução	1
1.1	Gramma e Ferramenta Interna da Gramma	1
1.2	Objetivos	2
1.3	Estrutura	3
2	Estado da Arte	5
2.1	Plataformas Concorrentes	5
2.2	Discussão das Plataformas Concorrentes	8
3	Metodologia de Desenvolvimento	11
3.1	Metodologias Tradicionais vs Ágeis	11
3.2	Metodologia a Utilizar	12
3.3	Planeamento	15
4	Levantamento de Requisitos	21
4.1	Atores	21
4.2	<i>User Stories</i>	22
4.3	Requisitos Funcionais	23
4.4	Requisitos Não Funcionais	24
5	Tecnologias	29
5.1	<i>Backend</i>	29
5.2	<i>Frontend</i>	32
5.3	Fornecedor de Serviço <i>Cloud</i>	35
6	Arquitetura	37
6.1	Arquitetura Monolítica vs Microsserviços	37
6.1.1	Padrões de Microsserviços	38
6.2	Arquitetura a Utilizar	40
6.2.1	Resposta da Arquitetura aos Requisitos Identificados	41
6.3	Seleção das Tecnologias	43
6.4	Desenho da Arquitetura	43
6.5	Diagrama Entidade-Relacionamento	54
7	Desenvolvimento	59
7.1	Processo de Desenvolvimento	59
7.1.1	Organização das Tarefas	60
7.2	Funcionalidades Desenvolvidas	61
7.3	Evidências	68

7.4	Reflexão Sobre os Riscos Identificados	78
7.5	Trabalho Futuro	79
8	Testes	81
8.1	Testes Funcionais	81
8.2	Testes aos Requisitos Não Funcionais	83
9	Conclusão	93
	Referências	95
	Apêndice A Análise de Riscos	103
	Apêndice B <i>Epic Stories e User Stories</i>	107
	Apêndice C Tabelas de Requisitos Funcionais	129
	Apêndice D Vantagens e Desvantagens das Tecnologias Analisadas	135
	Apêndice E Evidências	137
	Apêndice F Testes Funcionais	151
	Apêndice G Testes aos Requisitos Não Funcionais	155

Acrónimos

API *Application Programming Interface.*

AWS *Amazon Web Services.*

CDN *Content Delivery Network.*

DEI *Departamento de Engenharia Informática.*

E-R *Entidade-Relacionamento.*

EC2 *Elastic Compute Cloud.*

ELB *Elastic Load Balancer.*

ES *Epic Story.*

HTTP *Hypertext Transfer Protocol.*

HTTPS *Hypertext Transfer Protocol Secure.*

IT *Information Technology.*

RDS *Relational Database Service.*

REST *Representational State Transfer.*

S3 *Simple Storage Service.*

SES *Simple Email Service.*

US *User Story.*

Lista de Figuras

3.1	Diagrama da <i>framework</i> ágil <i>Scrum</i> ^[54]	13
3.2	Planeamento 1º Semestre	17
3.3	Executado 1º Semestre	18
3.4	Planeamento 2º Semestre	20
6.1	Exemplo de uma <i>API Gateway</i>	39
6.2	Exemplo de dois serviços com uma base de dados privada para cada serviço ^[50]	40
6.3	Níveis do modelo C4 ^[17]	45
6.4	Diagrama de Contexto (<i>Context Diagram</i>) da Ferramenta Interna da Grama	46
6.5	<i>Container Diagram</i> da Ferramenta Interna da Grama	48
6.6	<i>Component Diagram</i> da <i>API Gateway</i>	49
6.7	<i>Component Diagram</i> do microsserviço de Gestão de Colaboradores	50
6.8	<i>Component Diagram</i> do microsserviço de Gestão de Equipamentos	52
6.9	<i>Component Diagram</i> da página <i>web</i>	53
6.10	Diagrama E-R - Modelo Conceptual	56
6.11	Diagrama E-R - Modelo Físico	58
7.1	Exemplo de um quadro e distribuição de tarefas no <i>Linear</i>	61
7.2	Ecrã de Início de Sessão	69
7.3	Ecrã de Primeiro Início de Sessão	69
7.4	Lista de Utilizadores Com Conta Ativa	70
7.5	Vista Detalhada do Perfil de um Utilizador - Separador da Informação Pessoal	71
7.6	Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual	71
7.7	Adicionar Novo Utilizador - Separador de Informação Pessoal	72
7.8	Lista de Departamentos	73
7.9	Lista de Equipamentos Ativos	74
7.10	Vista Detalhada de um Equipamento	75
7.11	Adição de um Novo Equipamento	75
7.12	Lista de Pedidos de Equipamentos Por Aprovar	76
7.13	Vista Detalhada de um Pedido de Equipamento Por Aprovar	76
7.14	Lista de Orçamentos dos Utilizadores	77
7.15	Vista Detalhada do Orçamento de um Utilizador	77
8.1	Exemplo do <i>Swagger</i> no microsserviço de Gestão de Equipamentos	82
8.2	Sinalética dos Alertas da Ferramenta OWASP ZAP ^{[44][43]}	87

8.3	Alertas da Ferramenta OWASP ZAP	88
E.1	Ecrã de recuperar palavra passe	137
E.2	<i>Modal</i> de terminar sessão	138
E.3	Lista de Utilizadores Com Conta Desativada	139
E.4	Vista Detalhada do Perfil de um Utilizador - Separador da Informação de Pagamento	139
E.5	Vista Detalhada do Perfil de um Utilizador - Separador da Informação da Conta	140
E.6	Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual - <i>Modal</i> de Adicionar Bónus	140
E.7	Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual - <i>Modal</i> de Adicionar Ficheiros	141
E.8	Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual - <i>Modal</i> de Adicionar Salário	141
E.9	Adicionar Novo Utilizador - Separador de Informação Contratual .	142
E.10	Adicionar Novo Utilizador - Separador de Pagamento	142
E.11	Adicionar Novo Utilizador - Separador de Informação da Conta . .	143
E.12	Vista Detalhada do Perfil de um Utilizador - Modo de Edição do Separador da Informação Pessoal	143
E.13	Adicionar Novo Departamento	144
E.14	Editar Departamento	144
E.15	Lista de Equipamentos Inativos	145
E.16	Lista de Equipamentos Obsoletos	145
E.17	Vista Detalhada de um Equipamento - Modo de Edição	146
E.18	Desativar um Equipamento	146
E.19	Reativar um Equipamento	147
E.20	Lista de Pedidos de Equipamentos Aprovados	147
E.21	Lista de Pedidos de Equipamentos Recusados	148
E.22	Vista Detalhada - Pedido de Equipamento Por Aprovar - Modo de Edição	148
E.23	Vista Detalhada - Aprovar Pedido de Equipamento	149
E.24	Vista Detalhada - Recusar Pedido de Equipamento	149
E.25	Edição do Orçamento de um Utilizador	150

Lista de Tabelas

2.1	Comparação das Funcionalidades das Plataformas Concorrentes . . .	10
4.1	Níveis de prioridade dos requisitos	23
4.2	Cenário de Segurança - Utilizador Não Autenticado	26
4.3	Cenário de Segurança - Utilizador Não Autorizado	27
4.4	Cenário de Desempenho	28
5.1	<i>"Most Popular Backend Framework January 2022" Usage of backend frameworks^[20]</i> - 10 Mais Populares	30
5.2	<i>"The State of JS 2021" Usage of frontend frameworks^[29]</i>	33
7.1	Requisitos Funcionais Implementados - Módulo de Autenticação . . .	62
7.2	Requisitos Funcionais Implementados - Módulo de Gestão de Colaboradores - Área de Gestão de Contas dos Utilizadores	62
7.3	Requisitos Funcionais Implementados - Módulo de Gestão de Colaboradores - Área de Gestão de Informações dos Utilizadores	63
7.4	Requisitos Funcionais Implementados - Módulo de Gestão de Colaboradores - Área de Gestão de Departamentos	63
7.5	Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de <i>Dashboard</i>	64
7.6	Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Equipamentos	65
7.7	Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Orçamentos dos Utilizadores	66
7.8	Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Pedidos de Equipamentos	66
7.9	Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Reparação de Equipamentos	67
8.1	Testes Realizados - Módulo de Autenticação	83
8.2	Testes de Segurança - Utilizador Não Autenticado	85
8.3	Testes de Segurança - Utilizador Não Autorizado	87
8.4	Testes ao Desempenho das Páginas <i>Web</i>	91
A.1	Matriz de Riscos	105
C.1	Tabela de Requisitos Funcionais - Módulo de Autenticação	129
C.2	Tabela de Requisitos Funcionais - Módulo de Gestão de Colaboradores - Área de Gestão de Informações dos Utilizadores	130

C.3	Tabela de Requisitos Funcionais - Módulo de Gestão de Colaboradores - Área de Gestão de Contas dos Utilizadores	130
C.4	Tabela de Requisitos Funcionais - Módulo de Gestão de Colaboradores - Área de Gestão de Departamentos	130
C.5	Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de <i>Dashboard</i>	131
C.6	Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Equipamentos	131
C.7	Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Orçamentos dos Utilizadores	132
C.8	Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Pedidos de Equipamentos	132
C.9	Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Reparação de Equipamentos	133
D.1	Vantagens e Desvantagens das Tecnologias - Backend	135
D.2	Vantagens e Desvantagens das Tecnologias - Frontend	136
D.3	Vantagens e Desvantagens das Tecnologias - <i>Cloud Service Provider</i> .	136
F.1	Testes Realizados - Módulo de Gestão de Colaboradores	152
F.2	Testes Realizados - Módulo de Gestão de Equipamentos	153
G.1	Teste de Indisponibilidade da Plataforma Durante Atualização de Microserviço	155
G.2	Testes de Segurança - Utilizador Não Autorizado (Cont.)	156

Capítulo 1

Introdução

O presente documento foi elaborado no âmbito da disciplina Dissertação/Estágio, especialização em Engenharia de *Software*, do Mestrado em Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

O estágio consiste no desenvolvimento de um portal de gestão de operações com equipamentos, a plataforma interna da empresa Grama, sob orientação do Professor Doutor Nuno Laranjeiro e do Mestre Rui Abreu, da empresa Grama.

1.1 Grama e Ferramenta Interna da Grama

A Grama foi fundada em 2017 e encontra-se sediada em Coimbra, trata-se de uma empresa que fornece diversos serviços de desenvolvimento de *software* em todas as suas fases, desde a gestão dos projetos ao *design* e desenvolvimento dos produtos de *software* em si. Atualmente a equipa conta com cerca de 30 colaboradores.

A gestão interna da empresa passa por diversas áreas de atuação, desde a gestão dos recursos humanos até a compra de equipamentos para uso dos colaboradores. Tratam-se de processos que a empresa necessita de ter bem definidos, para que não existam erros quando um colaborador deseja marcar férias ou requisitar um novo equipamento para a sua estação de trabalho.

Estas necessidades são atualmente tratadas com recurso a plataformas externas e também documentos de edição manual. Estas soluções apresentam algumas falhas como a necessidade de um grande esforço para a manutenção de documentos, a suscetibilidade a erros ou, no caso das plataformas externas, a falta de um serviço personalizado. Surge também uma descentralização de informações importantes para o correto funcionamento da empresa e das suas tarefas do dia a dia. Se um colaborador precisa de requisitar férias tem que aceder a uma plataforma externa, com credenciais exclusivas à mesma. Se deseja alterar algum dado pessoal necessita de contactar outro colaborador dos recursos humanos para que essa informação seja atualizada. Essa mesma informação pode estar presente em vários locais e pode ter que ser alterada manualmente em todos.

Existe, portanto, um problema claro de descentralização e de suscetibilidade a

erros nos diferentes processos da empresa. A Plataforma Interna da Grama surge como solução pensada para resolver estes problemas e também atendendo a necessidades futuras da empresa. Esta plataforma procura simplificar os processos referentes à gestão de recursos humanos e de equipamentos da empresa. Permitirá centralizar numa plataforma única que oferece a todos os colaboradores, sejam eles trabalhadores comuns ou gestores, a possibilidade de facilmente consultar informações e realizar ações nas diversas áreas que se encontrarão integradas na plataforma.

Desta forma, o desenvolvimento e implementação de uma plataforma deste tipo e que responde a estas necessidades não é trivial e, portanto, a empresa considerou ser necessário uma pesquisa, planeamento e desenvolvimento cuidados e estruturados de modo a que o produto final fosse de encontro ao desejado e pudesse substituir os processos implementados atualmente. Assim, a empresa preparou duas propostas de estágio para alunos do Mestrado em Engenharia de Software para implementar esta ferramenta, passando por todos os passos importantes no desenvolvimento de um sistema de *software*.

1.2 Objetivos

Este estágio faz parte da primeira fase de desenvolvimento da Ferramenta Interna da Grama, ou seja, esta será criada de raiz de modo a responder às atuais necessidades, resolvendo alguns dos problemas sentidos atualmente. No entanto, também tem como objetivo construir uma base sólida para as necessidades futuras, pensadas pela empresa. Os pontos sobre os quais serão desenvolvidos os primeiros módulos da plataforma durante este estágio dividem-se em **duas áreas** distintas:

- Gestão de Colaboradores;
- Gestão de Equipamentos.

O presente estágio é focado nestes dois módulos, sendo que outros dois (gestão de ausências e recrutamento) estão atribuídos a outro estágio curricular que decorre simultaneamente. Este segundo estágio está atribuído ao Francisco Bugalho, também aluno do Mestrado em Engenharia Informática. Durante o decorrer do estágio existirão decisões que serão tomadas em conjunto devido ao facto dos quatro módulos a desenvolver, dois por cada estágio, fazerem parte da mesma plataforma. A divisão de tarefas realizadas durante a implementação será apresentada em maior detalhe na Secção 7. Nos dois módulos que dizem respeito a este estágio, a empresa apresentou as seguintes funcionalidades que devem estar presentes nos mesmos:

- **Gestão de Colaboradores:**
 - Convites e estado de cada colaborador;
 - Informação pessoal do colaborador;

- Documentos associados;
 - Departamentos (gestores e colaboradores);
 - Histórico de bônus;
 - Histórico de salários.
- **Gestão de Equipamentos:**
 - Inventário;
 - Alocação;
 - Pedidos;
 - Reparações;
 - Orçamento para compra de equipamentos.

O desenvolvimento da Ferramenta Interna da Grama será pensado com vista a ser escalável para que possa crescer juntamente com as necessidades da empresa, quer esta tenha 30 colaboradores ou centenas daqui a alguns anos, facto que não é incomum em empresas de desenvolvimento de *software*. A mesma escalabilidade é um ponto importante caso esta plataforma venha a ser comercializada a outras empresas de maior dimensão que nela encontrem as soluções para os seus processos do dia a dia.

Futuramente, a plataforma idealizada pela empresa incluirá módulos adicionais que irão procurar solucionar outros problemas que a empresa enfrenta atualmente, como é o caso da gestão de formações e partilha de conhecimento. Apesar de serem questões que não vão ser implementadas de imediato, durante a realização deste estágio, é preciso que a primeira implementação desta plataforma seja já robusta e preparada para o futuro, tendo em conta, para além das necessidades atuais, as necessidades futuras da mesma.

1.3 Estrutura

O presente documento encontra-se organizado em **9 capítulos** que visam detalhar cada uma das fases necessárias ao desenvolvimento do projeto de estágio:

- No **Capítulo 2** é realizada uma análise da concorrência existente no mercado onde esta é comparada com as funcionalidades pensadas para a ferramenta que se pretende desenvolver;
- No **Capítulo 3** apresenta-se o modelo *Scrum* de desenvolvimento de software e a versão adaptada usada na realização deste estágio;
- O levantamento dos requisitos ocorre no **Capítulo 4**, onde são descritos os diferentes atores que interagem com a plataforma e onde são apresentadas as *User Stories* e *Epic Stories* elaboradas. Também se apresentam e priorizam os requisitos funcionais da plataforma. Por fim, são apresentados os requisitos não funcionais identificados;

- No **Capítulo 5** são analisadas diversas tecnologias para o desenvolvimento do *backend*, *frontend* e implementação dos serviços *cloud*. São analisadas as suas vantagens e desvantagens à luz dos requisitos e necessidades da plataforma;
- O **Capítulo 6** serve de apresentação da arquitetura escolhida para implementar neste projeto. É também neste capítulo que se definem as tecnologias a utilizar com base no que foi apresentado nos Capítulos 4 e 5. De seguida, é apresentado o desenho da arquitetura recorrendo ao modelo C4 e também o diagrama de Entidade-Relacionamento, com as diversas entidades necessárias para o funcionamento da plataforma;
- No **Capítulo 7** será detalhado o processo de desenvolvimento apresentando a distribuição de tarefas e as funcionalidades desenvolvidas;
- No **Capítulo 8** são descritos os testes à plataforma que foram realizados de modo a validar os requisitos e funcionalidades apresentados no Capítulo 4;
- Por fim, o **Capítulo 9** serve de conclusão para o presente documento.

Capítulo 2

Estado da Arte

Este capítulo encontra-se dividido em duas secções. Na Secção 2.1 é discutida a pesquisa inicial que foi realizada sobre as plataformas concorrentes. Posteriormente, na Secção 2.2, são apresentadas as cinco plataformas concorrentes escolhidas e exploradas do ponto de vista das suas funcionalidades, vantagens e desvantagens.

2.1 Plataformas Concorrentes

Esta secção tem como principal objetivo a exploração das ferramentas concorrentes encontradas e o respetivo registo e comparação de funcionalidades das mesmas.

No início do estágio foram realizadas reuniões com a *Product Owner* (apresentada em maior detalhe na Secção 3.2) onde esta expôs as ideias que a empresa tinha para os dois módulos da Plataforma Interna da Grama a implementar neste estágio. Estas foram inicialmente resumidas às funcionalidades que podemos encontrar na primeira coluna da Tabela 2.1, que se encontra no final deste capítulo.

Existem ferramentas no mercado que oferecem soluções para as funcionalidades que se pretendem implementar na plataforma a desenvolver. Deste modo, foi efetuada uma pesquisa com o objetivo de identificar estas ofertas, tal como as funcionalidades concretas que estas disponibilizam.

Inicialmente, o foco foi a análise da componente que se destacou como sendo a mais interessante e diferenciadora das duas que se pretendem desenvolver: a componente de gestão de equipamentos, onde se incluem orçamentos, pedidos de equipamentos de reparação e respetiva visualização dos dados recolhidos em *dashboards*.

A pesquisa por plataformas com este conjunto de funcionalidades revelou-se desafiante pois muitas das ferramentas encontradas contêm funcionalidades extra, que fogem ao âmbito deste projeto. É o caso de ferramentas como SuperOps.ai e NinjaOne que, apesar de conterem de forma inerente algumas funcionalidades que desejamos, focam-se no controlo e gestão mais abrangente da empresa

onde estas ferramentas são aplicadas. Estas plataformas englobam funcionalidades como a monitorização automática e detalhada de máquinas presentes na rede da empresa, controlo remoto de cada máquina, distribuição e instalação de *software* remotamente e a visibilidade total de todos os equipamentos. Apesar do foco principal desta pesquisa ser a componente de gestão de equipamentos, regra geral, todas as ferramentas que aplicam estas funcionalidades oferecem também as capacidades básicas de gestão de utilizadores.

As funcionalidades anteriores mostram-se úteis e interessantes, no entanto, não se encontram dentro do espectro das que são pretendidas para a Ferramenta Interna da Grama. A empresa não demonstrou interesse em existir a possibilidade de controlar remotamente os dispositivos nem existirem funcionalidades de *deployment* remoto de aplicações nas máquinas dos seus trabalhadores. Desta forma, não será realizado um maior aprofundamento deste tipo de ferramentas.

SnipeIT

SnipeIT é um *software open-source* feito a pensar na gestão de equipamentos informáticos de modo a auxiliar os departamentos de *Information Technology* (IT), identificando quem possui cada equipamento (e.g. um portátil), quando é que este foi comprado e que licenças de software e acessórios se encontram disponíveis.

Dentro das suas funcionalidades podemos encontrar a presença de um *dashboard* de administrador que permite observar a atividade recente, a vista geral dos equipamentos, os acessórios, consumíveis e componentes, assim como que equipamentos foram entregues ou levantados recentemente. Quanto à gestão de equipamentos em si, é possível saber a quem está atribuído cada equipamento e a sua localização física, alterar os atributos de cada equipamento, gerir e criar novos QR codes para identificação sendo que existe uma integração da aplicação com *scanners* de códigos de barras ou QR. É também mantida uma lista detalhada de ações passadas de cada equipamento (entregas, levantamentos ou manutenção). Existe também a possibilidade de integração com aplicações externas como o Slack para envio de notificações, assim como o envio de emails com informação relevante.

Para além das funcionalidades mencionadas, também fornece a possibilidade da gestão dos utilizadores da plataforma. Este módulo permite a criação, edição e remoção de utilizadores sendo que as vistas detalhadas do perfil de um utilizador oferecem a possibilidade de consultar um número elevado de informações como os seus atributos gerais, os equipamentos, as licenças ou os consumíveis atribuídos, os ficheiros carregados para esse utilizador e ainda uma janela com todo o histórico de ações do mesmo.

Sendo *open source*, existem versões grátis da ferramenta para serem implementadas por uma empresa que assim o queira fazer. No entanto, são disponibilizadas subscrições pagas com extras como suporte por email, capacidade de pedir funcionalidades aos programadores, *Transport Layer Security* (TLS) e *backups* automáticos.

Asset Panda

A plataforma Asset Panda oferece um elevado poder de customização, sendo possível adaptá-la a diferentes tipos de negócios e permite a gestão de qualquer tipo de inventário. Existe, no *website* desta aplicação, uma área dedicada a "*Fixed Asset Tracking*" que consiste na adaptação da ferramenta para casos de uso semelhantes ao que se está a estudar. Com esta solução, a ferramenta dispõe de funcionalidades como a gestão de equipamentos a partir de qualquer local, mantém um histórico acerca dos equipamentos, notificações personalizadas, controlo de acessos baseado em *roles*, atribuição de equipamentos a pessoas, geração de códigos de barras ou QR e histórico de ciclo de vida dos equipamentos. Uma funcionalidade que destaca esta ferramenta de outras é a oferta de uma aplicação móvel para uma maior facilidade e conveniência na gestão dos equipamentos.

No que toca à gestão de colaboradores da empresa, o Asset Panda oferece as funcionalidades básicas expectáveis neste tipo de ferramenta. Existem diferentes tipos de utilizadores com diferentes permissões e os equipamentos podem ser atribuídos aos utilizadores ou estes podem fazer o levantamento dos mesmos. O custo das funcionalidades oferecidas por esta plataforma apenas é obtido através do contacto com a empresa. O ponto que se mostra mais forte nesta ferramenta é a capacidade elevada de customização. Podem ser criados grupos para qualquer tipo de equipamento ou até mesmo licenças e consumíveis.

GoCodes

A GoCodes disponibiliza uma plataforma de gestão de equipamentos sendo que originalmente é destinada à gestão de um inventário de ferramentas e equipamentos de construção. No entanto, uma das áreas de implementação listada no *website* é a gestão de equipamentos de escritório, mostrando assim a versatilidade da aplicação que originalmente não era concebida para tal função. Também inclui a possibilidade do uso de códigos QR, *tracking* GPS de cada equipamento, *dashboards* de informação, funcionalidades de auditoria e customização fácil das informações dos equipamentos.

Como acontece com outras ferramentas anteriormente abordadas, o módulo de gestão de utilizadores da plataforma não é um foco central das funcionalidades da mesma. No entanto, a plataforma oferece a capacidade de gestão de utilizadores através da sua criação, consulta, edição e remoção.

Cheqroom

A ferramenta Cheqroom é um gestor de inventário que apresenta uma *interface* de utilizador muito moderna em comparação às outras ferramentas estudadas. Oferece a possibilidade de realizar pedidos de material, associar material a uma determinada pessoa, utilizar códigos QR para identificação dos equipamentos e tem uma aplicação móvel dedicada.

Uma das funcionalidades que se destacou na análise desta ferramenta foi a presença de um sistema de calendarização de recolha e entrega de equipamentos, isto é, existe um calendário onde é possível visualizar quem tem um determinado equipamento e durante que período de tempo esse equipamento vai estar reservado por essa pessoa. O caso de uso exemplificado no *website* é a requisição e utilização de material fotográfico e de gravação dentro de uma equipa sendo que, utilizando esta plataforma, é possível saber em que períodos cada material (câmaras, microfones ou lentes) estão reservados, quais os períodos de maior afluência no levantamento de equipamentos, quando é que um produto se encontra para arranjo, entre outros. Estas informações estatísticas adicionais podem ser consultadas através de *dashboards* apelativos e modernos.

PurelyHR

Esta ferramenta foi estudada pois uma vez que se foca na implementação de funcionalidades que são importantes à luz do módulo de gestão de colaboradores que se pretende desenvolver. Esta plataforma não oferece quaisquer funcionalidades de gestão de equipamentos nem de orçamento monetário de cada trabalhador. Outras ferramentas tais como o Absense.io ou o ClanHR poderiam também ter sido analisadas em maior detalhe, no entanto, apenas existiu um foco na análise do PurelyHR, a título ilustrativo, por se tratarem de plataformas idênticas no nível de funcionalidades que cobrem.

A PurelyHR implementa ferramentas de gestão de colaboradores, gestão do tempo de férias, análise do desempenho dos trabalhadores, detalhe do número de horas gastas em cada projeto, histórico de certificações, formações e licenças, bem como a gestão de informação contratual, histórico salarial e bónus atribuídos aos colaboradores. Para os módulos a implementar, interessa a análise dos dois últimos pontos, referentes à gestão de informação contratual e histórico salarial de bónus.

2.2 Discussão das Plataformas Concorrentes

Após a análise das plataformas, foi evidente a existência de algumas funcionalidades que são propostas pela Grama para a sua Ferramenta Interna e que não se encontram presentes nas mesmas. Nenhuma das plataformas estudadas implementa todas as funcionalidades pretendidas, sendo que é possível destacar a falta de implementação simultânea de gestão do orçamento de cada colaborador da empresa, de ferramentas que permitem a gestão contratual e histórico salarial, de funcionalidades de gestão de colaboradores e de gestão de equipamentos.

Para conseguir responder a todas as suas necessidades, não seria possível por parte da Grama a adoção de uma única plataforma das que foram estudadas. A integração das plataformas já existentes também não é viável devido aos elevados custos que iriam surgir, tanto de configuração como de manutenção das mesmas posteriormente. A criação de uma plataforma própria também terá os seus custos de desenvolvimento e manutenção porém, neste caso, existe controlo sobre

a sua evolução ao contrário de uma possível integração de diversas ferramentas cuja evolução dos custos seria imprevisível assim como a sustentabilidade dessa mesma solução.

É de notar, também, que a plataforma irá oferecer outras funcionalidades distintas noutros módulos e também existem já ideias para funcionalidades futuras. Deste modo, revela-se necessário o desenvolvimento da sua Ferramenta Interna que pretende responder a todas as funcionalidades pretendidas, englobando algumas das que já existem em outras plataformas. A questão da centralização de toda a informação e a melhoria dos processos atualmente existentes na empresa também são pontos fulcrais que justificam a criação de uma nova plataforma. Algumas das ferramentas já existentes no mercado também se focam em proporcionar este tipo de serviços a empresas de maior dimensão, na ordem das centenas ou até mesmo milhares de colaboradores, e, desta forma, a curva inicial de aprendizagem de utilização, configuração e personalização é algo que não se justifica para empresas mais pequenas como é o caso da Grama. O próprio custo monetário de algumas das ferramentas também não é justificável para a empresa.

Dentro das ferramentas estudadas foram encontradas algumas funcionalidades interessantes que foram discutidas junto da empresa, em reunião com a *Product Owner*, no entanto não serão implementadas na plataforma pois não vão de encontro às necessidades imediatas da empresa. É o caso da geração de códigos de barras/QR de modo a aceder a informações e histórico de um equipamento, a gestão de licenças informáticas e consumíveis e, por fim, a presença de um calendário onde seja possível visualizar os períodos onde cada equipamento está atribuído a um trabalhador.

A Tabela 2.1 descreve resumidamente as funcionalidades presentes em cada uma das plataformas estudadas e a comparação entre as mesmas de modo a melhor compreender quais implementam cada funcionalidade.

Tabela 2.1: Comparação das Funcionalidades das Plataformas Concorrentes

		Ferramenta Interna da Grama	SnipeIT	Asset Panda	GoCodes	Cheqroom	PurelyHR
Gestão de colaboradores e utilizadores	Autenticação e Controlo de Acessos (user, manager, admin)	Sim	Sim	Sim	Sim	Sim	Sim
	Gestão de utilizadores (CRUD)	Sim	Sim	Sim	Sim	Sim	Sim
	Adição de documentos a um utilizador	Sim	Sim	Sim	Sim	Sim	Sim
	Gestão da informação contratual	Sim	Não	Não	Não	Não	Sim
	Historico salarial e de bónus	Sim	Não	Não	Não	Não	Sim
	Vista detalhada do utilizador	Sim	Sim	Sim	Sim	Sim	Sim
	Gestão de departamentos	Sim	Sim	Sim	Sim	Sim	Sim
Gestão de Equipamentos Informáticos	Dashboard com resumo das informações gerais	Sim	Sim	Não	Sim	Sim	Não
	Listagem de equipamentos (vistas de filtro, estado atual, vista detalhada de cada equipamento)	Sim	Sim	Sim	Sim	Sim	Não
	Gestão de equipamentos (CRUD)	Sim	Sim	Sim	Sim	Sim	Não
	Associar o equipamento a uma pessoa	Sim	Sim	Sim	Sim	Sim	Não
	Informação e detalhes sobre o orçamento de cada utilizador	Sim	Não	Não	Não	Não	Não
	Gestão de pedidos de equipamentos e de reparação (aprovados, recusados e pendentes)	Sim	Sim	Sim	Sim	Sim	Não
	Carregamento de ficheiros associados a um equipamento	Sim	Sim	Sim	Não	Sim	Não
	Gestão de acessórios (ratos, teclados, etc.)	Sim	Sim	Sim	Sim	Sim	Não
	Gestão de licenças de produtos informáticos	Não	Sim	Sim	Sim	Não	Não
	Geração de Código de Barras ou QR com info do equipamento	Não	Sim	Sim	Sim	Sim	Não
	Depreciação dos equipamentos (EOL Rate, EOL Date)	Não	Sim	Sim	Sim	Sim	Não
	Gestão de consumíveis (papel, toner, etc.)	Não	Sim	Sim	Sim	Sim	Não
	Calendário para consulta de reservas de equipamentos	Não	Não	Não	Não	Sim	Não

Capítulo 3

Metodologia de Desenvolvimento

No Capítulo que se segue são expostas numa perspetiva sintética as metodologias tradicionais e ágeis assim como as vantagens de cada uma. É também apresentado o modelo *Scrum* na sua versão tradicional e a sua adaptação específica no contexto do projeto.

3.1 Metodologias Tradicionais vs Ágeis

Nas últimas décadas, verificaram-se grandes mudanças nas metodologias de desenvolvimento de *software*, sendo uma das mais notáveis a introdução do “desenvolvimento ágil”, descrito no “*The Agile Manifesto*”^[27]. Nesta secção será analisada não só esta metodologia, mas também as metodologias tradicionais, procurando identificar os pontos negativos e positivos de ambas, e procurando perceber qual se enquadra como a melhor alternativa para a realização deste projeto.

Projetos cuja necessidade principal seja a garantia de que o sistema é corretamente definido e implementado de forma a não existirem erros devem optar por um modelo onde a implementação só começa quando existe uma forte e segura definição de todos os requisitos e a garantia de que estes não se vão alterar ao longo do desenvolvimento. No entanto, no desenvolvimento de *software* não crítico, pode não fazer sentido aplicar este modelo pois acarreta custos temporais e monetários que não se justificam.

Como exemplo de metodologias tradicionais podemos encontrar, entre outras, o modelo *Waterfall* e o modelo em V. Dentro das metodologias ágeis existem diversas abordagens como é o caso do *Scrum*, *Kanban* ou *Extreme Programming* (XP).

De uma forma geral, as **metodologias tradicionais** são baseadas numa abordagem preditiva ao problema em causa, em que a equipa de desenvolvimento depende totalmente de um plano de trabalhos detalhado, com listas completas de características e funcionalidades a implementar, divididas em tarefas a realizar num determinado espaço de tempo, inalteradas durante o processo. Estas são dependentes da análise de requisitos e de um planeamento detalhado antes do início do processo de implementação de cada ciclo (caso a metodologia preveja a

existência de ciclos de desenvolvimento).

Os **métodos de desenvolvimento ágil**, por outro lado, adotam uma perspectiva mais dinâmica. Não existe um planeamento detalhado ao nível do que encontramos nas metodologias tradicionais, mas sim uma lista de tarefas a realizar, alinhadas com as funcionalidades desejadas pelo cliente. O produto é testado ao longo do desenvolvimento e de forma frequente para tentar minimizar o risco de falhas graves caso o processo de teste ocorresse apenas no final de cada ciclo^[58].

O uso de metodologias ágeis é recomendado para projetos de pequena e média escala e onde a prioridade mais elevada é a satisfação do cliente através de entregas de produto funcional frequentes. Também é uma abordagem que pode ser aplicada em projetos cuja lista completa de requisitos e funcionalidades não está totalmente definida desde o momento inicial^[27].

Alguns dos princípios deste tipo de metodologias passam por aceitar mudanças nos requisitos mesmo que seja numa fase tardia do desenvolvimento, entregar *software* funcional frequentemente, manter o contacto diário entre os programadores e os trabalhadores ligados ao negócio, confiar nas pessoas que têm a responsabilidade de desenvolver o projeto e dar-lhes o ambiente e suporte que estes necessitam, mantendo ao máximo reuniões cara a cara com os programadores^[27].

3.2 Metodologia a Utilizar

A metodologia utilizada durante a realização deste projeto será baseada em *Scrum*, tal como já é utilizada pela empresa para o desenvolvimento de projetos passados e atuais. Desta forma, será possível que a pequena equipa de desenvolvimento consiga desenvolver a plataforma de forma eficiente, contando com constante *feedback* da empresa devido às entregas regulares de produto funcional.

A metodologia *Scrum* insere-se nos métodos ágeis de desenvolvimento. É uma *framework* de desenvolvimento bastante leve, principalmente quando comparada às *frameworks* tradicionais. No geral procura ajudar as pessoas, equipas e organizações a gerar valor através de soluções adaptativas para problemas complexos^[53]. A Figura 3.1 descreve a *framework Scrum* através de um diagrama onde se encontram representadas as cerimónias que acontecem durante o processo e os artefactos que são gerados, assim como o fluxo desta metodologia.

No "*Scrum Guide*"^[53], documento elaborado por Ken Schwaber and Jeff Sutherland, a metodologia *Scrum* é resumida em quatro pontos essenciais:

- O **Product Owner** decide as tarefas a serem realizadas e coloca-as no *Backlog* do Produto;
- A **Equipa de Scrum** implementa o que considera mais importante e de valor durante um *Sprint*;
- A **Equipa de Scrum** e as partes interessadas analisam os resultados e ajustam o próximo *Sprint*;
- **Repetir.**

Também são gerados diversos artefactos durante o processo:

- **Backlog do Projeto** - Lista ordenada das melhorias que são necessárias aplicar ao produto;
- **Backlog do Sprint** - Contém o objetivo do *Sprint* (*Sprint Goal*), os itens que devem ser implementados nesse *Sprint* e um plano de como o incremento irá ser realizado;
- **Incremento** - Um passo concreto com vista ao objetivo final do projeto. Cada incremento é uma nova e melhorada versão do anterior que acrescenta valor ao produto.

As vantagens da metodologia *Scrum* passam principalmente por uma maior satisfação do cliente, promoção um ambiente saudável de trabalho em equipa, atribuição da responsabilidade necessária a cada interveniente e foco na melhoria do produto de forma contínua. Para além destas, também é importante referir que este processo é adaptativo, consegue ajustar-se às constantes mudanças durante o desenvolvimento, produz apenas a documentação necessária e reduz os riscos de desenvolvimento, pois as entregas são realizadas em pequenos incrementos.

No entanto, também existem desvantagens. No caso de existir uma falta de interação com o cliente, pode dificultar todo o processo visto que este é um ponto fulcral para que a metodologia *Scrum* funcione. Esta acontece principalmente quando o cliente não é claro acerca dos requisitos a implementar no produto, dificultando assim o trabalho a realizar. A possibilidade de constante mudança nos requisitos pode levar a um consumo maior de tempo e recursos caso o cliente não fique satisfeito com a iteração apresentada e altere os requisitos. A criação de pouca documentação pode também ser uma desvantagem caso entrem novos programadores a meio do projeto^[55].

Olhando para as vantagens e desvantagens da metodologia ágil *Scrum* podemos concluir que esta se aplica bem ao tipo de projeto que iremos realizar, pois as vantagens impõe-se claramente às desvantagens encontradas visto que a comunicação com a empresa é eficaz e contínua e, tratando-se de uma empresa de desenvolvimento de *software*, há um maior conhecimento técnico sobre as diversas funcionalidades a implementar. Este facto também leva a que mudanças nos requisitos que causem um maior consumo desnecessário de tempo não sejam comuns de acontecer.

A **metodologia baseada em *Scrum*** implementada durante a realização do estágio possui alguns pontos semelhantes e outros onde existem adaptações que procuram melhorar o processo de desenvolvimento do *software*. A equipa é constituída pelos seguintes elementos que desempenham as funções descritas:

- **Product Owner:** Susana Lourenço. Trabalhadora na empresa, *Head of Operations* e será utilizadora final da plataforma;
- **Scrum Master:** Rui Abreu. Orientador do estágio na empresa;
- **Equipa de desenvolvimento:**
 - Gonçalo São Marcos;

- Francisco Bugalho. Estagiário, a desenvolver os outros dois módulos brevemente apresentados na Secção 1.2;
- Marco Mendes. *Designer* da empresa alocado ao projeto da Ferramenta Interna da Grama.

As cerimónias realizadas durante a duração do estágio dependem do momento em que o desenvolvimento se encontra. Durante todo o **primeiro semestre** do ano letivo de 2022/2023, foram levadas a cabo reuniões semanais de acompanhamento denominadas "*Weeklies*" onde todos os elementos da equipa reuniam para discutir o progresso realizado durante a semana que passou e onde se definia o que iria ser realizado na semana seguinte.

No segundo semestre as reuniões foram diárias e denominadas "*Dailies*", têm o mesmo efeito das descritas anteriormente mas com a diferença da sua periodicidade e do facto de se tratar de uma fase de desenvolvimento o que implica que as mesmas incidam sobre temas mais técnicos do progresso na implementação das tarefas. Adicionalmente, existiram cerimónias de "*Sprint Planning*" onde foram realizadas análises das tarefas planeadas para o *Sprint* seguinte e onde era realizada uma estimativa de cada tarefa com base na sua complexidade. Para estimar cada tarefa foi usada a escala de Fibonnaci onde foram atribuídos apenas os valores 1, 2, 3, 5 e 8 desta escala. Tarefas estimadas acima de 8 deveriam ser repartidas em sub-tarefas de modo a planear melhor o *Sprint*. Cada *Sprint* teve a duração de **2 semanas** ao longo de todo o segundo semestre.

3.3 Planeamento

O planeamento deste estágio curricular está dividido em dois semestres devido à diferença do trabalho a realizar em cada um deles. O primeiro semestre consiste num trabalho de pesquisa, como é o caso da análise das plataformas concorrentes, e preparação, como é o caso da definição dos requisitos ou escolha da metodologia a utilizar. Os diagramas de *Gantt* foram elaborados recorrendo à ferramenta online *TeamGantt*^[61].

1º Semestre

No início do desenvolvimento foram definidas as diversas tarefas a realizar durante o decorrer do primeiro semestre do estágio curricular. Esta fase culmina com a entrega do relatório intermédio do estágio. As tarefas a realizar estão diretamente ligadas à estrutura deste relatório pois o trabalho realizado está presente no mesmo. Foram realizados dois diagramas de *Gantt* onde são especificadas as diferentes tarefas e o período de tempo que foi dedicado ao desenvolvimento das mesmas. O primeiro diagrama, Figura 3.2, mostra o que foi planeado no início do semestre para a realização durante a duração do mesmo. O segundo diagrama, Figura 3.3, apresenta o que foi executado na realidade. Existem algumas diferenças entre os dois diagramas, nomeadamente a duração da realização de algumas tarefas não foi a esperada, a existência de mais tarefas sobrepostas pelo facto do

processo de pesquisa, escrita e validação com os orientadores por vezes se sobrepõe com o início de outra tarefa. Também se pode observar a troca de ordem de tarefas que se deve a decisões de desenvolvimento tomadas juntamente com os orientadores.

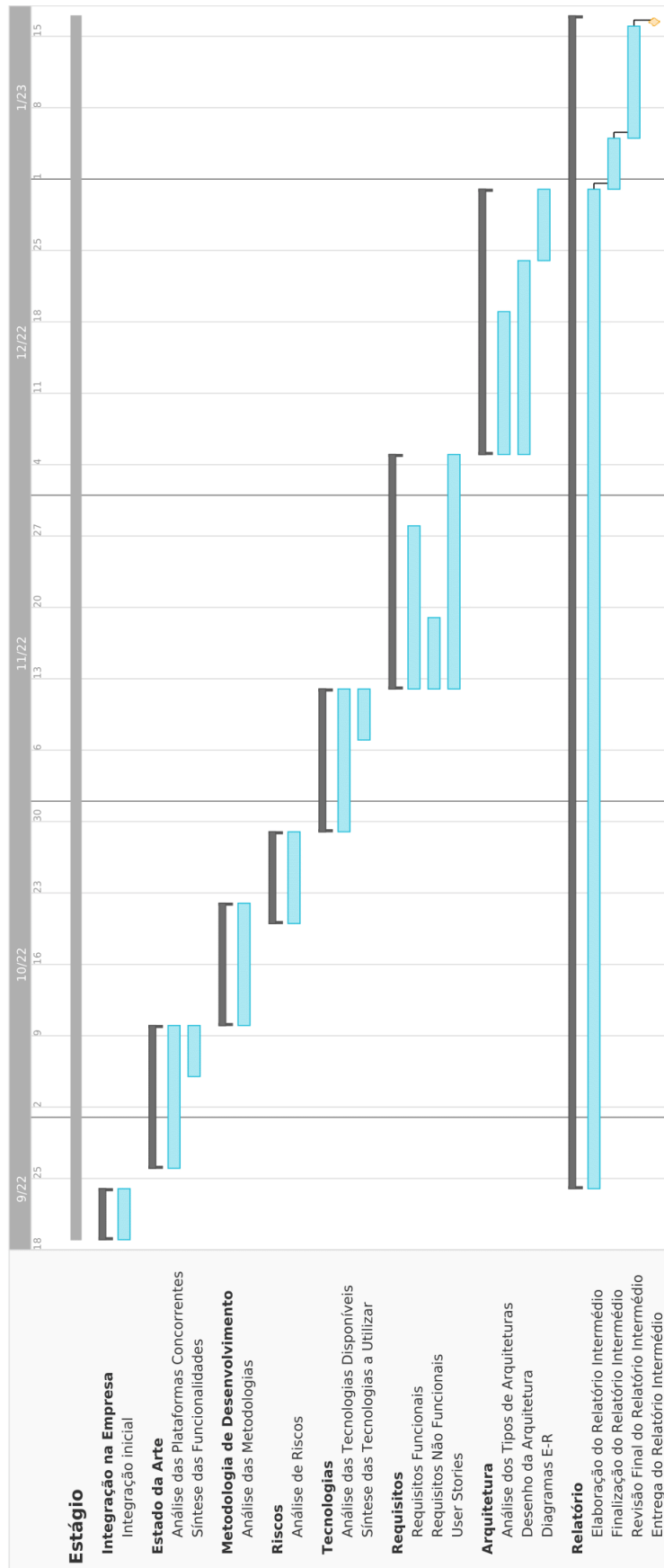


Figura 3.2: Planeamento 1º Semestre

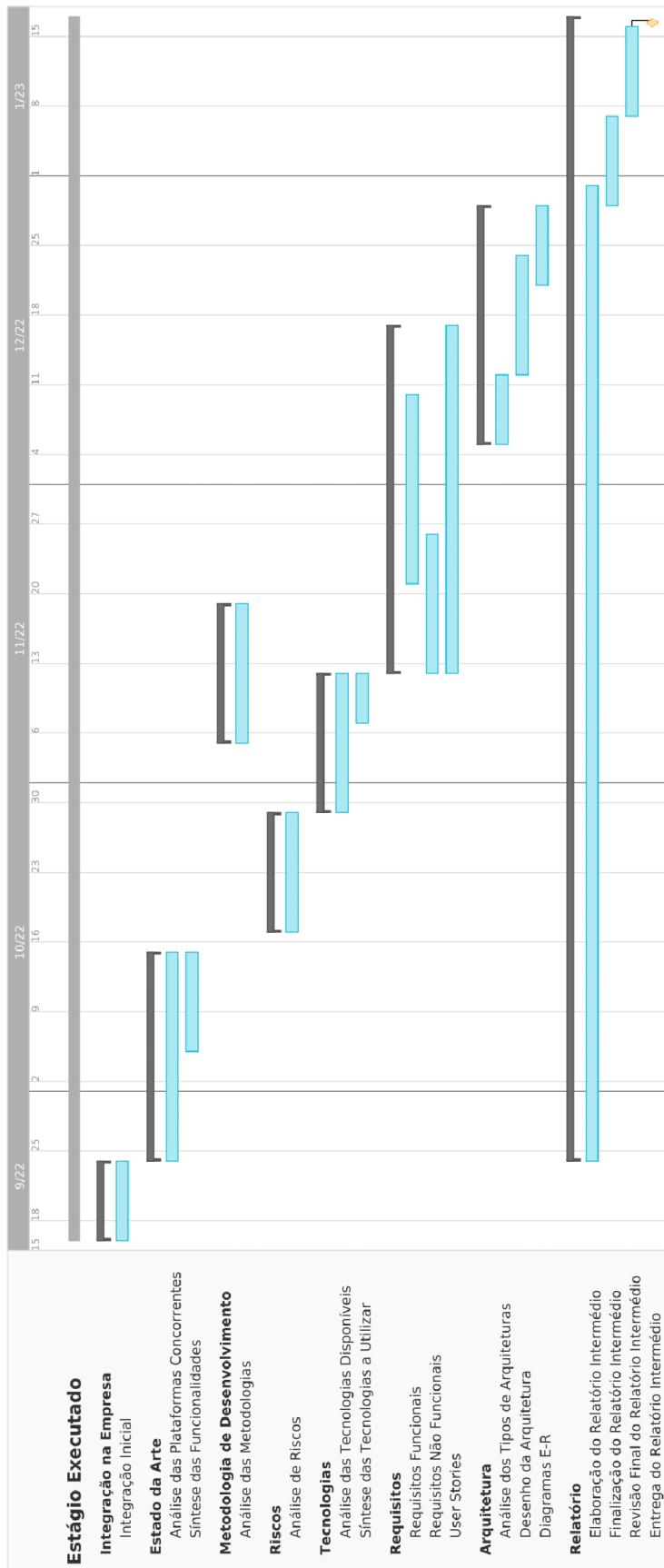


Figura 3.3: Executado 1º Semestre

2º Semestre

Foi também elaborado um diagrama de *Gantt* que descreve o planeamento do segundo semestre, Figura 3.4. Como já foi discutido no Capítulo 3, o desenvolvimento será feito seguindo uma metodologia baseada em *Scrum* com *sprints* de 2 semanas. No início de cada *sprint* existe um planeamento e no fim uma retrospectiva do mesmo. O processo de desenvolvimento consiste em 10 *sprints* sendo que o último é dedicado exclusivamente aos testes à plataforma. No final será feito também um diagrama de *Gantt* do que foi executado na verdade para poder existir uma comparação com este planeamento que foi agora feito. Ao longo de todo o segundo semestre será elaborado o relatório final de estágio com base no trabalho que será realizado, no final existe um momento de conclusão e revisão deste mesmo relatório.

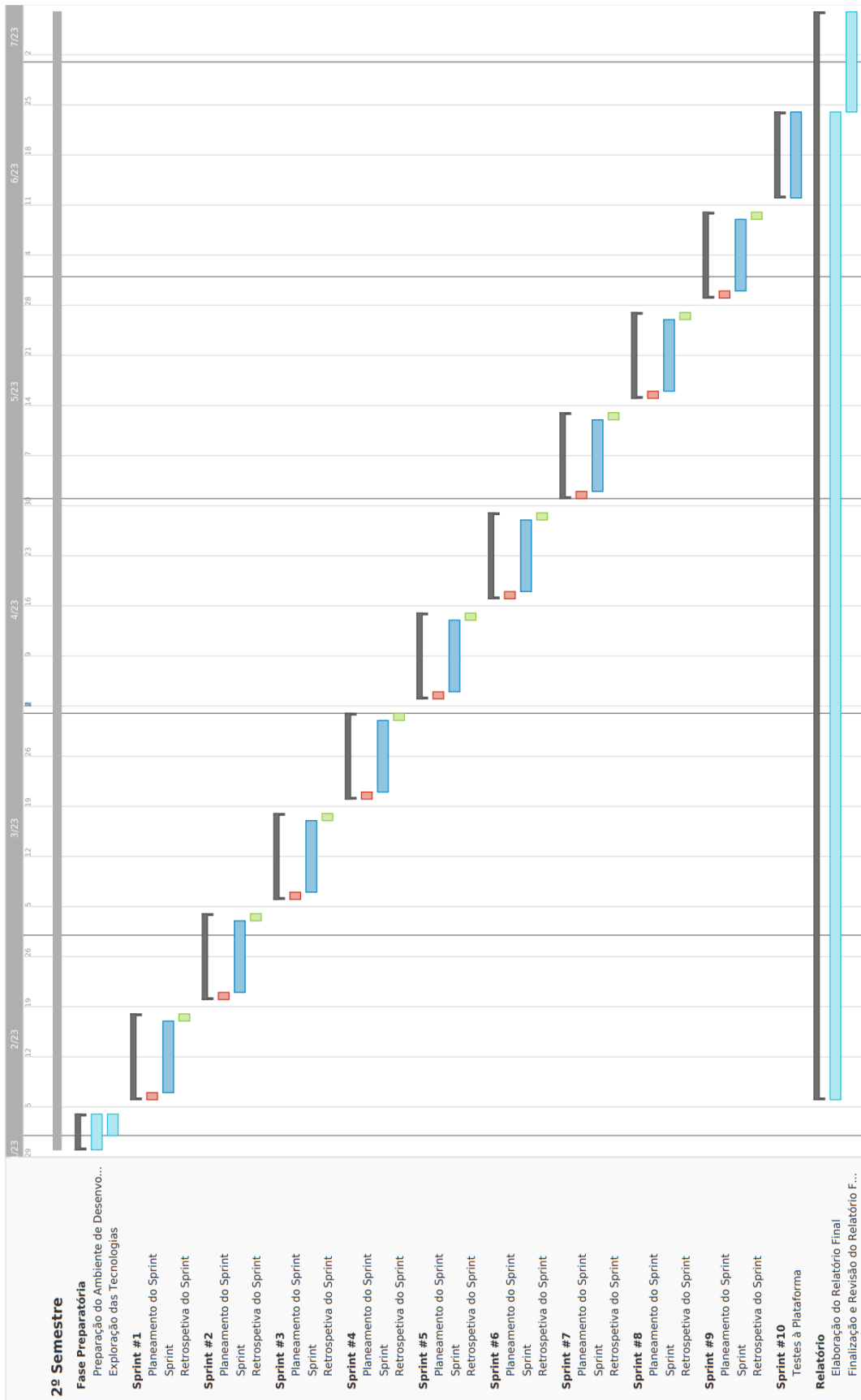


Figura 3.4: Planeamento 2º Semestre

Capítulo 4

Levantamento de Requisitos

Para a implementação dos dois módulos da Ferramenta Interna da Grama: Gestão de Colaboradores e Gestão de Equipamentos, é importante definir as funcionalidades e características que a empresa tem em mente para a plataforma. Para tal, é necessário fazer um levantamento dos requisitos funcionais e não funcionais de uma forma mais detalhada. Apenas desta forma é possível construir uma plataforma que vai de encontro às necessidades e desejos da empresa e do utilizador final.

Na Secção 4.1 são detalhados os diversos atores que interagem com a plataforma. Na Secção 4.2 apresentam-se as *Epic Stories* e *User Stories* desenvolvidas. Na Secção 4.3 são apresentados os requisitos funcionais da plataforma priorizados segundo o método *MoSCoW*^[19]. Por fim, na Secção 4.4 são apresentados os requisitos não funcionais identificados.

4.1 Atores

Na Ferramenta Interna da Grama deverão existir três tipos de atores diferentes, cada um com o seu papel e responsabilidades distintas. A descrição dos diferentes atores (*roles*) por parte da *Product Owner* foi resumida nos seguintes pontos:

- **Administrador** - Tem acesso a todas as funcionalidades dos módulos da Ferramenta Interna da Grama. Gere os utilizadores, as suas contas e informações tal como departamentos e os seus gestores. Gere os equipamentos tendo acesso a todas as funcionalidades de criação, adição, remoção;
- **Gestor** - Tem acesso à informação do(s) seu(s) departamento(s) e dos respetivos utilizadores do(s) mesmo(s). Não tem permissões para criação, edição e remoção de qualquer informação no sistema;
- **Colaborador** - Tem acesso à sua informação pessoal e à informação dos seus equipamentos e orçamento. Pode realizar pedidos de equipamento que têm que ser aceites por um Administrador.

É de notar que existe uma herança de funcionalidades e permissões entre os diferentes atores da plataforma. Os Colaboradores têm apenas acesso às suas próprias informações, departamentos, equipamentos e pedidos, podendo editar as suas informações pessoais. Os Gestores têm também estas permissões dos Colaboradores mas, adicionalmente, podem ver a informação de todos os utilizadores que fazem parte do(s) seu(s) departamento(s) assim como os seus equipamentos e pedidos realizados pelos mesmos, não tendo permissões para alterar nenhuma informação. Os Administradores, nível mais alto, têm as mesmas permissões e funcionalidades dos Colaboradores e Gestores e podem, para além dessas, ter acesso às informações de todos os utilizadores da plataforma com possibilidade de edição, ver todos os pedidos podendo aceitar e recusar os mesmos, adicionar e remover utilizadores, adicionar e remover departamentos e adicionar e remover equipamentos assim como editar os mesmos. Para que o administrador tenha acesso tanto à sua área pessoal como à área de administração, os esboços realizados pelo *designer* prevêm a existência de uma barra de navegação lateral onde é possível ao administrador trocar entre cada área

4.2 User Stories

Uma abordagem utilizada para o levantamento de requisitos em projetos que adotam a metodologia ágil é a formulação de *User Stories*. Neste tipo de metodologia é crucial colocar o utilizador final como prioridade. Uma *User Story* (US) é uma pequena descrição informal de funcionalidades escritas na perspetiva do utilizador final, tendo em conta como este utiliza o produto. Destas, será possível retirar o valor que é acrescentado à experiência do utilizador quando realiza uma certa tarefa.

Existem também as *Epic Stories* que são descrições de mais alto nível que podem ser separadas em diversas US que detalham melhor o objetivo global da *Epic Story* (ES).

A construção de uma US segue a seguinte estrutura^[46]:

“Como [tipo de utilizador], eu quero [intenção], para que [fim].”

Foram realizadas algumas reuniões com o orientador da empresa e com a *Product Owner* da Ferramenta Interna da Grama onde foram expostas e discutidas as funcionalidades desejadas pela empresa para a plataforma. Nesse momento existia já um trabalho prévio por parte do *designer* da empresa alocado ao projeto, que, a pedido da empresa e da *Product Owner*, já tinha realizado estudos de design e esboços de alto nível para a Ferramenta Interna da Grama. Desta forma, a maior parte das *Stories* foram extraídas dos esboços de alto nível e discutidas nas reuniões mencionadas, tendo influenciado o produto final. A lista total das US que foram desenvolvidas encontra-se no Apêndice B. As mesmas foram confirmadas pela *Product Owner* como estando de acordo com as funcionalidades desejadas pela empresa. Mostram-se de seguida exemplos de ES e US para que se possa perceber a estrutura das mesmas tal como a sua importância.

ES-1: Como utilizador da plataforma não autenticado, **eu quero** autenticar-me na plataforma, **para que** consiga aceder às diversas funcionalidades da mesma.

US-1: Iniciar sessão

(a) **Como** administrador/gestor/colaborador, **eu quero** iniciar sessão na plataforma, **para que** consiga aceder às suas funcionalidades. Quero introduzir os campos descritos abaixo para que a minha identidade seja validada no acesso à plataforma:

- Email*;
 - Palavra-passe*.
- (* campo obrigatório)

4.3 Requisitos Funcionais

Nesta secção serão apresentados os requisitos funcionais da Ferramenta Interna da Grama. Estes seguem o método de priorização *MoSCoW*^[19] (*Must, Should, Could, Won't Have*). Foi necessário escolher um método de priorização dos requisitos pelo facto de ser importante perceber quais os requisitos que devem ser implementados durante a realização do estágio. Dado que o estágio é de duração limitada, poderá não ser possível a implementação de todos os requisitos identificados e, portanto, como foi dito antes, é necessário priorizar quais devem ser implementados para que a plataforma vá de encontro ao que é esperado como fruto deste estágio. O método de priorização *MoSCoW*^[19] foi escolhido por ajudar a perceber e a gerir as prioridades dos requisitos segundo os quatro níveis que se encontram descritos na Tabela 4.1.

Tabela 4.1: Níveis de prioridade dos requisitos

Nível de Prioridade	Descrição
<i>Must (M)</i>	Requisito necessário para o correto funcionamento da plataforma segundo as funcionalidades que a empresa deseja
<i>Should (S)</i>	Requisito não obrigatório para o funcionamento da plataforma mas que acrescenta algum valor caso seja implementado
<i>Could (C)</i>	Requisito não obrigatório para o funcionamento da plataforma e que acrescenta valor reduzido, apenas deve ser implementado caso exista tempo adicional
<i>Won't (W)</i>	Requisito que não deverá ser implementado na fase atual. Poderá, no entanto, vir a ser implementado numa fase futura do projeto, fora do âmbito do presente estágio curricular

Os requisitos funcionais foram criados a partir das *Epic* e *User Stories* desenvolvidas e apresentadas na Secção 4.2. Procurou-se ter um nível elevado de detalhe

técnico para que existisse informação suficiente para realizar a implementação a partir dos mesmos. A *Product Owner* também teve um papel de confirmação e priorização dos requisitos funcionais de forma a que estes fossem de encontro ao desejado pela empresa.

A lista total de requisitos funcionais pode ser consultada no Apêndice C. Estes estão definidos em tabelas separadas por módulos. Cada tabela encontra-se separada em áreas e cada área tem os requisitos associados à mesma. Cada requisito tem um identificador único e cada um tem três colunas à sua frente que indicam que tipo de utilizador tem acesso a esse requisito em específico. Existe também uma coluna com a indicação da prioridade do requisito e outra com a dependência a outro(s) requisito(s).

O processo de levantamento de requisitos que foi levado a cabo teve como objetivo reunir todas as ideias e esboços da empresa de forma a criar *Stories* concretas, bem como requisitos funcionais que, em conjunto, geraram a informação necessária para a implementação. Durante o processo foram discutidas todas as funcionalidades estudadas, havendo espaço para a sua reformulação e clarificação por parte da *Product Owner*, permitindo o seu total entendimento e mitigando quaisquer dúvidas quanto à necessidade das mesmas. Este processo tornou-se, também, crucial para uma melhor compreensão da plataforma como um todo, tendo sido um exercício que permitiu olhar para todas as funcionalidades específicas da mesma e o seu enquadramento e interligação.

4.4 Requisitos Não Funcionais

Na literatura é possível encontrar diversas definições para "requisito não funcional" sendo que, no geral, dizem respeito aos atributos de qualidade do programa, isto é, às condições que o sistema deve cumprir. Desempenho, usabilidade, segurança, escalabilidade e confiabilidade^[18] são alguns exemplos de requisitos não funcionais.

De seguida, são apresentados os requisitos não funcionais que foram identificados para o projeto deste estágio assim como a sua prioridade. Cada requisito não funcional será apresentado, seguido da descrição e, caso aplicável, de um cenário de qualidade referente ao mesmo. Este cenário é caracterizado pela fonte do estímulo, o estímulo em si, o ambiente em que ocorre, o artefacto que é estimulado, a resposta do sistema e a medida concreta desta resposta^[16].

A cada requisito não funcional será atribuída uma prioridade que indica a relevância desse requisito no projeto em causa. A prioridade pode tomar três níveis: baixa, média ou alta.

Disponibilidade - Alta

A disponibilidade é um requisito de qualidade que define o tempo que um sistema se deve encontrar acessível por parte dos utilizadores que o querem utilizar. Por outras palavras, pode ser considerada como a chance do sistema responder de forma positiva quando existe um pedido ao mesmo^[3].

A métrica deste requisito pode ser descrita como a percentagem de um período de tempo em que o sistema deve estar disponível, por exemplo, 99.9% do tempo ao longo de um mês, isto é, o sistema apenas pode estar indisponível durante cerca de 43 minutos de um mês inteiro^[3].

Este requisito é considerado como sendo de alta prioridade pois a Ferramenta Interna da Grama deve estar disponível o máximo de tempo possível, com tempo de indisponibilidade mínimo, para ser acedida por parte dos colaboradores da empresa, assim como os gestores e administradores, de modo a fazer uso das suas funcionalidades que fazem parte do dia a dia da mesma.

Este projeto será elaborado com recurso à utilização dos serviços *cloud* da Amazon, os *Amazon Web Services* (como será discutido no Capítulo 5). Desta forma, a disponibilidade do sistema está ligada à disponibilidade que este serviço oferece. Segundo a documentação deste serviço, é garantida por parte da Amazon uma disponibilidade de pelo menos 99.99% ao nível das regiões e 99.95% para instâncias únicas das máquinas EC2^{[12][7]}. No entanto, existe também responsabilidade no desenvolvimento da plataforma, um sistema que apresente erros frequentes poderá ficar indisponível mesmo que os serviços *cloud* se encontrem funcionais. Sendo assim, é também importante garantir que ao longo do desenvolvimento do sistema sejam tidas em consideração estruturas no código e na arquitetura que permitam responder a este requisito.

Dado que a disponibilidade dos serviços da AWS não podem ser controlados, foca-se nos pontos em que é possível controlar partindo do princípio que os serviços se encontram funcionais. Quando existe uma atualização ao sistema, a plataforma não deve ficar indisponível por mais de 3 minutos em cada atualização. Este processo de atualização considera-se completo quando a plataforma retorna ao seu normal funcionamento após uma atualização.

Para testar este requisito não funcional deverá ser feito o *deploy* da plataforma com recurso aos serviços da AWS, conforme a arquitetura que irá ser apresentada na Secção 6.4. O tempo que o sistema fica indisponível quando existe uma atualização de um microsserviço pode ser contabilizado desde o momento em que este é desligado até ao momento em que está pronto a receber pedidos novamente. Desta forma, é possível confirmar se este tempo de indisponibilidade não excede os 3 minutos definidos como limite.

Segurança - Alta

O requisito de segurança diz respeito à capacidade do sistema de proteger os dados que nele existem ao impedir o uso do mesmo a utilizadores que não estão

autorizados para o fazer, enquanto permite a utilização por parte de utilizadores que, de facto, estão autorizados a usar o mesmo. Uma falha na segurança pode resultar em dados serem acedidos por utilizadores com intenções maliciosas, podendo até existir alteração dos mesmos, ou até ao impedimento da utilização do sistema a utilizadores legítimos^{[3][16]}.

Para este projeto, este requisito não funcional será dividido em duas secções que diferenciam a segurança do ponto de vista de utilizadores não autenticados e utilizadores não autorizados.

Tendo em conta a importância da privacidade dos dados pessoais de cada utilizador face às normas e leis aplicáveis, assim como o risco de exposição de informação privada da empresa, considera-se que a prioridade deste requisito é alta.

Este requisito pode ser testado realizando testes de integração onde cada funcionalidade é testada manualmente através do uso normal da plataforma *web*. Também é possível realizar testes específicos que procuram tentar aceder a informação não permitida alterando o URL manualmente, por exemplo. Outra possibilidade é a utilização de ferramentas existentes no mercado que realizem uma análise detalhada à plataforma quando esta se encontrar *deployed* e que revelam as possíveis vulnerabilidades. O OWASP ZAP^[44] é uma das ferramentas mais utilizadas para este efeito.

De seguida, serão detalhados dois cenários de segurança, um que diz respeito à interação com a plataforma por parte de utilizadores não autenticados, e outro por parte de utilizadores não autorizados.

Utilizador Não Autenticado

Um utilizador não autenticado é aquele que não tem permissão para aceder a qualquer parte do sistema pois não tem, por exemplo, uma conta pessoal que se possa autenticar perante o mesmo. Este utilizador não pode, portanto, aceder a qualquer informação ou funcionalidade. Apenas os utilizadores com credenciais de acesso válidas poderão autenticar-se na plataforma e aceder a informação da mesma, qualquer outro utilizador não deve conseguir obter qualquer informação. Um cenário deste atributo de qualidade encontra-se apresentado na Tabela 4.2.

Tabela 4.2: Cenário de Segurança - Utilizador Não Autenticado

Tópico do Cenário	Descrição
Fonte do Estímulo	Utilizador não autenticado do sistema
Estímulo	Tenta aceder a uma funcionalidade do sistema
Ambiente	Funcionamento normal do sistema
Artefacto	Sistema
Resposta do Sistema	Bloquear acesso ao sistema por parte do utilizador não autenticado; redirecionar o utilizador para uma página de erro
Medida da Resposta	As funcionalidades não são acedidas e os dados não são comprometidos

Utilizador Não Autorizado

Um utilizador não autorizado consegue autenticar-se na plataforma mas o sistema deve garantir que este apenas acede a informação que tem autorização para visualizar ou editar. Um colaborador, por exemplo, não poderá aceder a informações sobre outros utilizadores da plataforma da mesma forma que um administrador consegue.

Este comportamento pode ser implementado através da verificação da *role* do utilizador que realiza cada pedido e a comparação entre a informação que este realmente pode visualizar e ter acesso com a informação que este está a requisitar.

A Tabela 4.3 apresenta um cenário deste atributo de qualidade.

Tabela 4.3: Cenário de Segurança - Utilizador Não Autorizado

Tópico do Cenário	Descrição
Fonte do Estímulo	Utilizador autenticado
Estímulo	Tenta aceder a uma funcionalidade do sistema à qual não tem autorização
Ambiente	Funcionamento normal do sistema
Artefacto	Sistema
Resposta do Sistema	Bloquear acesso à funcionalidade/aos dados; redirecionar o utilizador para uma página de erro
Medida da Resposta	As funcionalidades não são acedidas e os dados não são comprometidos

Desempenho - Média

O desempenho é um atributo de qualidade que descreve a rapidez com que o sistema responde a pedidos do utilizador durante uma certa carga de uso^[3]. Desta forma, a prioridade é definida como média pois a Ferramenta Interna da Grama não tem como foco principal uma responsividade elevada mas sim a capacidade de permitir aos utilizadores que façam uso da funcionalidades da plataforma. É certo que esta não pode ser pouco responsiva, como qualquer plataforma *web*, necessitando de ter um bom nível de resposta às ações e pedidos do utilizador, de modo a não tornar frustrante a sua utilização.

O número atual de colaboradores da Grama (cerca de 30) leva a que este atributo de qualidade não faça sentido como uma preocupação atual, no entanto, devido aos planos futuros da empresa para a plataforma, este atributo deve ser tido já em consideração para prevenir falhas de desempenho em cenários futuros em que poderão existir centenas de utilizadores simultâneos na plataforma.

O desempenho pode ser medido objetivamente ao realizar acessos à plataforma quando esta se encontrar *deployed* e registando os tempos que as diversas páginas demoram a carregar de modo a obter valores médios para os tempos de carregamento. Assim será possível verificar se o tempo de carregamento excede os 2.5

segundos estabelecidos como limite no cenário da Tabela 4.4 que se apresenta de seguida.

Tabela 4.4: Cenário de Desempenho

Tópico do Cenário	Descrição
Fonte do Estímulo	Utilizador autenticado
Estímulo	Tenta aceder à plataforma
Ambiente	Funcionamento normal do sistema
Artefacto	Sistema
Resposta do Sistema	A página requisitada é apresentada
Medida da Resposta	A página é apresentada em tempo igual ou inferior a 2.5 segundos

Capítulo 5

Tecnologias

A implementação da Ferramenta Interna da Grama exige a decisão e seleção sobre as tecnologias de *backend* e *frontend* a utilizar tendo em conta os requisitos funcionais e não funcionais do sistema.

A empresa sugeriu, através do orientador do estágio na mesma, a exploração de algumas tecnologias atualmente utilizadas em diversos projetos dentro da empresa. Esses mesmos projetos são também de desenvolvimento de plataformas *web* com requisitos semelhantes e apresentam dimensões distintas, desde pequenos projetos a nível local até projetos de dimensão mundial. Foi dada a liberdade de escolha de qualquer tecnologia para a implementação do sistema, no entanto, foi sugerido que a escolha final passasse por alguma combinação de tecnologias familiares aos orientadores da empresa de modo a existir a possibilidade de um maior suporte técnico com as mesmas caso fosse necessário ao longo do desenvolvimento. Apesar das sugestões, foi sempre tido em conta como cada tecnologia poderia responder aos requisitos definidos para este projeto.

Na Secção 5.1 são apresentadas as tecnologias de *backend* escolhidas e as mesmas são analisadas brevemente, juntamente com as suas vantagens e desvantagens. Na Secção 5.2 são apresentadas as tecnologias de *frontend* onde também são analisadas as suas vantagens e desvantagens. A Secção 5.3 apresenta as três maiores ofertas de serviços de *cloud* no mercado que são posteriormente analisadas, recorrendo às suas vantagens e desvantagens.

5.1 *Backend*

Começamos por analisar algumas das tecnologias de *backend* mais populares atualmente, olhando para as suas vantagens e desvantagens de modo a procurar aquela que fará mais sentido adotar para o desenvolvimento da plataforma. As tecnologias foram escolhidas por popularidade, ou seja, pela sua adoção em massa que leva a uma maior maturidade e suporte da mesma, também pelo crescimento recente de utilização, sugestão de programadores mais experientes da empresa e curiosidade pessoal. Seria possível realizar o projeto recorrendo a qualquer uma das tecnologias estudadas, no entanto, as vantagens e desvantagens de cada uma,

a possibilidade de apoio dentro da empresa por parte de programadores mais experientes e também a resposta estas que conseguem dar aos requisitos do sistema, são fatores de decisão na sua escolha para o desenvolvimento.

Na Tabela 5.1, apresenta-se a popularidade de cada *framework* com base na evolução do número de estrelas dos repositórios do GitHub, podemos observar uma lista das *frameworks* mais populares de *backend* em janeiro de 2022.

Tabela 5.1: "Most Popular Backend Framework January 2022" Usage of backend frameworks^[20] - 10 Mais Populares

<i>Framework</i>	Popularidade
Laravel	67 902
Django	61 614
Flask	57 681
ExpressJS	55 520
Ruby on Rails	49 840
Spring	45 609
NestJS	43 408
Meteor	42 762
Strapi	42 206
Koa	32 281

Após a análise do *ranking* exposto na Tabela 5.1, foi decidido explorar algumas das tecnologias mais populares e também outras que atualmente se têm vindo a destacar mas que não foram consideradas pelos autores do *ranking*. Iremos explorar as seguintes tecnologias:

- Express.js;
- Django;
- Quarkus;
- Spring;
- Spark.

No Apêndice D, Figura D.1 podem ser consultadas as vantagens e desvantagens de cada uma das tecnologias de *backend* em formato de tabela de resumo, incluindo algumas informações adicionais como a data de lançamento das mesmas.

Express.js

O Express.js é uma *framework* para o Node.js com base na linguagem Javascript que oferece diversas vantagens para a criação de aplicações *web* e móveis, sendo considerada rápida e minimalista. Podemos destacar algumas vantagens como

a sua fácil aprendizagem, a facilidade de implementação de APIs, o desenvolvimento rápido do lado do servidor, por oferecer funcionalidades presentes no Node.js e também a vantagem do *backend* poder ser escrito na mesma linguagem que o *frontend*.^[62]

No entanto apresenta algumas desvantagens como o facto de ser *single-threaded* (por ser baseada em Node.js) e também não inclui predefinições de como implementar a segurança da aplicação, o que pode levar a uma menor segurança caso esta não seja bem tratada.^[62]

Django

Esta *framework* é implementada em *Python*, uma linguagem reconhecida pela sua simplicidade de escrita e maior facilidade de leitura. Oferece já muito código pré-desenvolvido para funcionalidades comuns o que torna o desenvolvimento mais rápido. É também uma *framework* escalável e com uma preocupação na segurança das aplicações^[23].

A sua estrutura é monolítica o que leva a que um problema afete todo o sistema de forma negativa. É também bastante pesado em termos de recursos e, portanto, não é considerado muito indicado para projetos pequenos onde existe uma maior importância na eficiência e em não gastar demasiados recursos^[23].

Quarkus

O Quarkus apresenta-se como sendo uma *framework full-stack* de Java para ser utilizada com contentores *Kubernetes*. Esta otimiza os tempos de compilação e consequentemente de inicialização das aplicações em comparação à outra oferta de tecnologia em Java, o Spring. A sua documentação é bastante detalhada, o que ajuda no desenvolvimento. O código editado é refletido de forma quase imediata para que o programador tenha logo acesso às alterações que realiza. É uma solução em Java que se mostra eficaz para arquiteturas *serverless*, *microserviços*, *function-as-a-service* (FaaS) e computação *cloud*^{[33][60]}.

A sua configuração pode ser complicada de realizar e a sua pequena comunidade torna-se numa barreira para a aprendizagem da *framework* devido à falta de suporte por parte de outros utilizadores, tornando-a mais difícil para iniciantes^[60].

Spring

A *framework* Spring é baseada em Java e é utilizada para desenvolver aplicações com uma escala empresarial ao invés das *frameworks* mais leves que são mais indicadas para projetos de média e pequena escala. É uma *framework* que, devido às suas vastas funcionalidades e integrações, é muitas vezes escolhida para projetos de *microserviços*. A documentação do Spring é detalhada e a comunidade é grande, existindo muitas soluções para problemas comuns. Em comparação com

o Quarkus apresenta operações de *Input/Output* (I/O) mais rápidas e também melhor segurança^[60].

Os seus tempos de inicialização são mais lentos que o Quarkus e também consome mais recursos de memória em comparação a esta. É dependente de módulos que podem conter muitas funcionalidades desnecessárias, reduzindo, assim, o desempenho geral da *framework*^[60].

Spark

O Spark Java apresenta-se como sendo muito eficaz para a implementação de arquiteturas de microsserviços oferecendo muita flexibilidade na implementação e acesso total à *Java Virtual Machine* (JVM).^[57]

Esta *framework* tem, no entanto, uma presença pequena na comunidade o que a torna mais complicada de aprender e dificulta a resolução de problemas. Existe pouco código pré-feito para funcionalidades comuns, pelo que o tempo de desenvolvimento aumenta em comparação a outras *frameworks* que oferecem bastantes bibliotecas para situações comuns.^[57]

5.2 Frontend

Começamos por analisar algumas das tecnologias de *frontend* mais populares atualmente, analisando as suas vantagens e desvantagens, procurando aquela que fará mais sentido escolher para o desenvolvimento da plataforma. Foi consultado o documento "*The State of JS*"^[29] na sua edição de 2021 de modo a observar a popularidade das *frameworks* e bibliotecas de *frontend* atualmente utilizadas em grande escala. A Tabela 5.2 apresenta os valores desse estudo, sendo que os valores representam a percentagem de utilizadores inquiridos que usariam novamente a *framework* ou biblioteca segundo a fórmula utilizada pelo autor.

Após a análise deste relatório sobre as tecnologias de *frontend* e a sua popularidade em termos de uso, foram escolhidas as seguintes para explorar como potenciais candidatas a utilizar durante a realização deste projeto:

- Vue.js;
- Angular;
- React;
- Svelte;
- Alpine.js.

No Apêndice D, Figura D.2 podem ser consultadas as vantagens e desvantagens de cada uma das tecnologias de *frontend* em formato de tabela de resumo, incluindo algumas informações extra como a data de lançamento das mesmas.

Tabela 5.2: "The State of JS 2021" Usage of frontend frameworks ^[29]

<i>Framework</i>	2019	2020	2021
React	80%	80%	80%
Angular	58%	56%	54%
Vue.js	47%	49%	51%
Svelte	8%	15%	20%
Preact	12%	14%	14%
Ember	12%	11%	9%
Lit		5%	7%
Alpine.js		3%	6%
Solid			3%
Stimulus		1%	2%

Vue.js

A *framework* Vue.js é uma das mais populares atualmente para o desenvolvimento de *frontend*. Recorre à DOM (*Document Object Model*) virtual o que permite que seja mais rápido que outras *frameworks*. Permite uma forte reusabilidade de componentes que facilita o desenvolvimento e leitura do código e facilita a sua aprendizagem. A documentação é concisa e de boa qualidade e, caso existam barreiras, a comunidade é grande e existe muito suporte online^[4].

É uma *framework* mais leve do que Angular ou React mas isto leva a que muitas funcionalidades não estejam presentes de raiz e tenham que ser adicionadas através do uso de bibliotecas complementares^[45].

Angular

Angular é uma *framework* de *frontend* apoiada pela Google o que pode ser visto em si como uma vantagem em comparação a outras tecnologias. A documentação é de qualidade, oferece níveis de performance elevados, suporte para desenvolvimento *mobile* e reusabilidade de componentes que torna o código mais fácil de ler e de testar. Por outro lado, é uma tecnologia cuja curva de aprendizagem é maior do que, por exemplo, React e Vue^{[45][56][2]}.

As funcionalidades são ricas mas implicam um conhecimento que, no começo, pode ser uma sobrecarga para o programador. O uso de RxJS, uma biblioteca para programação assíncrona, e a adoção de Typescript são também fatores que dificultam a aprendizagem da tecnologia.^[2]

React

O React é uma biblioteca muito popular nos dias que correm sendo que foi criada inicialmente pelo Facebook. A sua aprendizagem é considerada fácil e é uma tecnologia que se aplica bem a aplicações *web* dinâmicas. A reusabilidade de componentes permite uma fácil leitura e testabilidade do código. Recorre, tal como outras tecnologias, ao uso da DOM Virtual que melhora o desempenho das aplicações *web*^[36].

É uma biblioteca em constante melhoramento, principalmente através de atualizações de grande dimensão e a introdução de mudanças com grande impacto, o que por um lado é uma vantagem mas, por outro, pode ser uma barreira pois a documentação é alterada constantemente e pode ser menos rica para funcionalidades recentes, podendo também tornar-se um pouco confusa e de difícil consulta devido às mesmas razões. No entanto, considera-se, no geral, como sendo uma boa documentação apesar da sua extensão e os problemas referidos anteriormente. Apesar de ser considerada de fácil aprendizagem, a utilização de JSX pode ser uma barreira para iniciantes^[36].

Svelte

O Svelte tem vindo a crescer nos últimos tempos como uma escolha sólida para o desenvolvimento de *frontend*, como podemos comprovar pelo gráfico do "The State of JS" anteriormente mostrado. Esta tecnologia compila o código para Javascript nativo o que o torna muito eficiente e reduz o *overhead* da *framework*. O código apresenta-se como sendo de fácil leitura o que melhora a compreensão do mesmo^[39].

Sendo uma tecnologia recente e em crescimento, ainda existem poucas ferramentas de desenvolvimento disponíveis e a sua comunidade ainda se encontra em crescimento, podendo dificultar a resolução de problemas por falta de apoio^[39].

Alpine.js

O Alpine.js é uma *framework* interessante do ponto de vista da sua leveza e minimalismo pois isso torna-a simples, eficiente e fácil de aprender. No entanto, a sua presença ainda é muito pequena e isso pode ser uma barreira grande à adoção da mesma. É mais indicada para *websites* simples e mais estáticos, apresenta poucas funcionalidades e uma comunidade pequena de utilizadores. Estas características tornam o Alpine.js atrativo para projetos onde a simplicidade é um foco e não tanto para aplicações de um nível mais complexo^[28].

5.3 Fornecedor de Serviço *Cloud*

Os serviços *cloud* são praticamente imprescindíveis nos dias que correm quando se fala na implementação de uma plataforma *web* com alguma dimensão. Os principais fornecedores deste tipo de serviços atualmente são a Google, a Microsoft e a Amazon. Dados referentes ao terceiro quartil de 2022 revelam que o serviço *cloud* da Amazon detêm cerca de 34% do mercado dos fornecedores de serviços *cloud*, seguida da Microsoft com 21% e Google com 11%^[32]. Nas próximas secções serão analisadas as vantagens e desvantagens destas ofertas.

No Apêndice D, Figura D.3 podem ser consultadas as vantagens e desvantagens de cada dos serviços *cloud*, em formato de tabela de resumo.

Google Cloud Services

Dos três fornecedores de serviços de *cloud*, a oferta da Google é a que detêm menos percentagem do mercado atualmente. No entanto, oferece algumas vantagens que podem ser atrativas para certos casos de uso como é o caso da forte integração com os outros serviços oferecidos pela Google e também a maior facilidade de colaboração que é oferecida. O facto deste serviço ser desenhado a pensar em negócios que são baseados na *cloud* pode também ser um ponto forte para certas empresas. Os serviços *cloud* da Google oferecem também uma boa portabilidade e suporte *open-source*^[22].

Como pontos fracos podemos encontrar, por exemplo, o facto de ser baseado em tecnologia exclusiva da Google e existir pouco controlo sobre as máquinas virtuais. Existe também uma escolha limitada das linguagens de programação que podem ser utilizadas. Em termos de serviços, funcionalidades e *data centers* a oferta também é menor em comparação com outros serviços *cloud*^[22].

Microsoft Azure Cloud

Dentro das suas vantagens da *Microsoft Azure Cloud*, encontramos a sua disponibilidade alta, um foco grande na segurança, escalabilidade, suporte para *open-source* e *cloud* híbrida. Como desvantagens podemos encontrar uma maior necessidade de gestão e a necessidade de um conhecimento mais detalhado sobre o *Azure*. Existem também menos possibilidades de retrocompatibilidade, o preço *pay-as-you-go* fica menos em conta e o serviço de apoio não é tão eficiente^[22].

Amazon Web Services

Os serviços *cloud* oferecidos pela Amazon, conhecidos também como AWS, são os mais populares atualmente com uma grande presença no mercado. A sua maturidade e dominância no setor são pontos que tornam a sua adoção mais provável por parte de empresas à procura de um fornecedor de serviço *cloud*. Para além

destes factos, existem também vantagens identificadas como uma grande flexibilidade, muita oferta e diversidade de aplicações, *updates* e manutenções frequentes, existência de uma versão grátis, bom controlo sobre a segurança, escalabilidade, *deployment* rápido e suporte para empresas de grande dimensão^{[22][31][35]}.

Por outro lado, a grande diversidade e oferta de aplicações pode desencorajar empresas menos habituadas aos serviços *cloud*. Existem ainda limites impostos aos recursos dependendo das regiões a nível mundial e o *downtime* dos servidores é mais comum do que o desejável. Além disso, os custos são muito variáveis, também devido à grande diversidade de ofertas e funcionalidades^{[22][31]}.

Capítulo 6

Arquitetura

Neste capítulo será discutida a decisão de que tipo de arquitetura deve ser utilizada no desenvolvimento da Ferramenta Interna da Grama. Em primeiro lugar, na Secção 6.1 serão analisados os tipos de arquitetura monolítica e de micro-serviços, apresentando vantagens e desvantagens de cada uma. Na Secção 6.2 é apresentado o tipo de arquitetura que será utilizada durante o desenvolvimento deste projeto com base nos requisitos apresentados no Capítulo 4. Na Secção 6.3 são selecionadas as tecnologias a utilizar durante o desenvolvimento do mesmo tendo em conta a arquitetura escolhida. Na Secção 6.4 é detalhado o desenho da arquitetura da plataforma através de diagramas recorrendo ao modelo C4. Por fim, na Secção 6.5 será apresentado o diagrama de entidades e relacionamentos pensados para o projeto.

6.1 Arquitetura Monolítica vs Microserviços

Uma **arquitetura de *software* monolítica** é tipicamente composta por uma única unidade que inclui todas as funcionalidades e serviços do sistema que corre o processo, tradicionalmente conectada a uma base de dados única^[47]. Os serviços dentro desta unidade única podem comunicar com sistemas externos através de diferentes interfaces como serviços *web*, páginas HTML ou *Application Programming Interface (API) Representational State Transfer (REST)*^[1]. Este tipo de arquitetura traz vantagens devido à sua simplicidade de desenvolvimento, teste e *deployment*. É também possível escalar este tipo de programas horizontalmente através do *deployment* de várias instâncias do mesmo código e o recurso a *load balancers*^[64]. No entanto, o facto de todo o código estar concentrado numa única unidade, leva a que a introdução de novas funcionalidades ao longo do tempo obrigue os programadores a terem atenção se estas não interferem com os serviços já existentes^[64]. À medida que o programa cresce torna-se mais complexo de entender e, possivelmente, de testar. A implementação de mudanças rápidas e corretas torna-se mais difícil de garantir e cada nova iteração obriga a um novo *redeploy* da aplicação o que pode causar transtorno aos utilizadores^[38]. O facto de ser uma única unidade também leva a que exista um *single point of failure*, isto é, se algo falha na aplicação, todo o sistema fica indisponível^[64].

A ideia de uma **arquitetura de microsserviços** parte por dividir a aplicação em diversos serviços, mais pequenos, que comunicam entre si^[38]. Hoje em dia podemos encontrar este tipo de abordagem à arquitetura de sistemas, juntamente com a utilização de serviços *cloud*, em grandes empresas como é o exemplo da Netflix. O recurso a arquiteturas deste tipo e ao *deployment* na *cloud* permite uma grande flexibilidade no desenvolvimento e manutenção das aplicações. Cada serviço diferente pode ser desenvolvido e mantido por uma equipa independente de programadores^[64] o que permite uma melhor escalabilidade através do uso de várias pequenas equipas. Do ponto de vista do sistema em si também podemos encontrar vantagens como a eliminação do *single point of failure*. Neste caso, se um dos microsserviços falhar, o sistema já não fica totalmente indisponível^[64].

Na *cloud*, cada microsserviço pode escalar de forma independente para dar resposta às necessidades do sistema. A complexidade do sistema como um todo também fica mais simples de visualizar e entender devido à decomposição do sistema em aplicações mais rápidas de desenvolver, compreender e manter^[38]. A barreira de adoção de novas tecnologias deixa de existir de forma tão marcante pois cada aplicação/microsserviço pode ser desenvolvida com recurso a tecnologias diferentes para responder às necessidades do problema em causa^[64].

6.1.1 Padrões de Microsserviços

Dentro das arquiteturas de microsserviços é possível explorar diversos padrões de implementação dos serviços do ponto de vista dos dados, decomposição, comunicação (descoberta e estilo), *deployment*, *cross-cutting concerns*, testes, segurança, observação e interface de utilizador (UI)^[51].

Para o projeto em causa foi sugerido pela empresa explorar em maior detalhe alguns dos padrões de microsserviços. Serão explorados os padrões *API Gateway*^[49], *Access Token*^[48] e *Database Per Service*^[50]. Foram escolhidos os padrões atrás indicados devido ao facto de serem padrões muito implementados no desenvolvimento de plataformas *web* baseadas em microsserviços e que a empresa aplica em praticamente todos os seus projetos. De seguida, cada um destes padrões será apresentado brevemente.

API Gateway

No caso de existirem vários serviços aos quais o cliente (i.e. a aplicação que pede os dados) tem que aceder através de chamadas às API, poderá haver entre o cliente e os serviços uma *API Gateway* à qual os clientes (*web*, móvel ou aplicações de terceiros) fazem chamadas para obter os dados que necessitam. Desta forma, podem existir APIs específicas para tipos de cliente diferentes e também permite criar uma camada onde se pode realizar a autenticação do pedido antes desta enviar os pedidos a cada um dos serviços separados^[49].

A Figura 6.1 exemplifica um sistema que usa uma *API Gateway* para gerir os pedidos realizados por clientes. À esquerda temos dois tipos de clientes distintos: uma aplicação *web* tradicional e uma aplicação nativa. Estas realizam os seus pe-

dados ao sistema recorrendo a pedidos específicos de cada cliente que são feitos diretamente à *API Gateway*, que depois traduz os diferentes pedidos específicos para os protocolos utilizados nos serviços de que o sistema dispõe. Desta forma, existe um ponto único de entrada no sistema através das funcionalidades disponibilizadas pela *API Gateway* e nunca existe um pedido direto aos serviços do sistema.

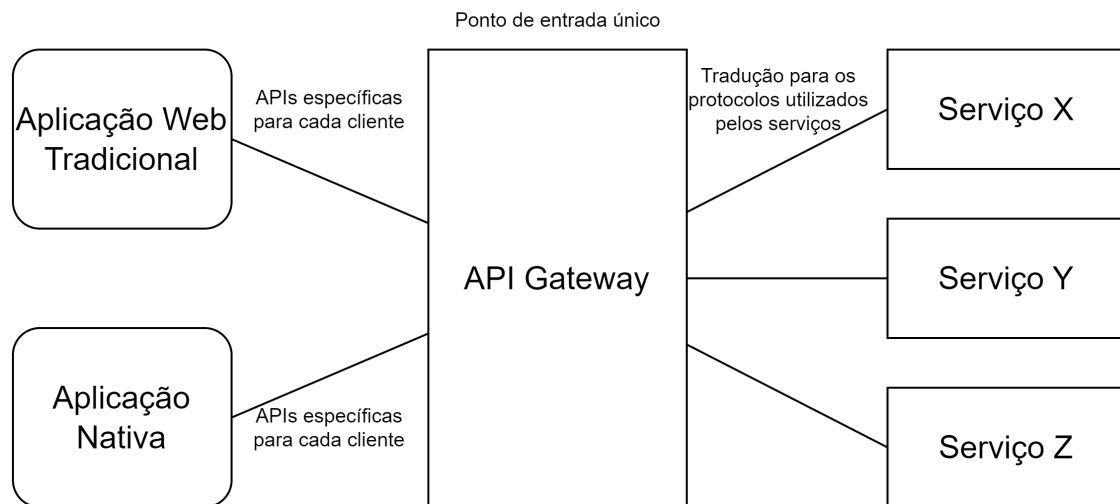


Figura 6.1: Exemplo de uma *API Gateway*

Access Token

O acesso aos diferentes microsserviços deve ser controlado de modo a apenas permitir o envio de informação a clientes que se encontram autorizados para aceder à mesma. Caso seja implementada uma *API Gateway*, esta pode realizar o processo de autenticação e encaminhar os pedidos para os serviços necessários, que em si podem chamar outros serviços^[48]. Neste encaminhamento de pedidos, a *API Gateway* envia também uma *Access Token* que identifica de forma segura cada pedido aos serviços^[48]. Desta forma, a identidade de quem faz o pedido é passada de forma segura ao longo do sistema e os diferentes serviços podem verificar que quem está a fazer o pedido tem a devida autorização para o fazer.

Database Per Service

A adoção de uma arquitetura de microsserviços possibilita a utilização de bases de dados separadas, isto é, cada serviço terá acesso a uma base de dados privada, criando assim uma maior decopulação dos serviços. Desta forma, os dados persistentes de cada serviço são privados a esse serviço e só podem ser acedidos através da API do mesmo. Este padrão também faz com que as transações de cada serviço apenas envolvam a sua base de dados^[50]. A Figura 6.2 exemplifica o conceito de base de dados por serviço. Existindo dois serviços distintos, o “*Order Service*” e “*Customer Service*”, cada um deles tem acesso à sua própria base de dados independente da base de dados do outro serviço.

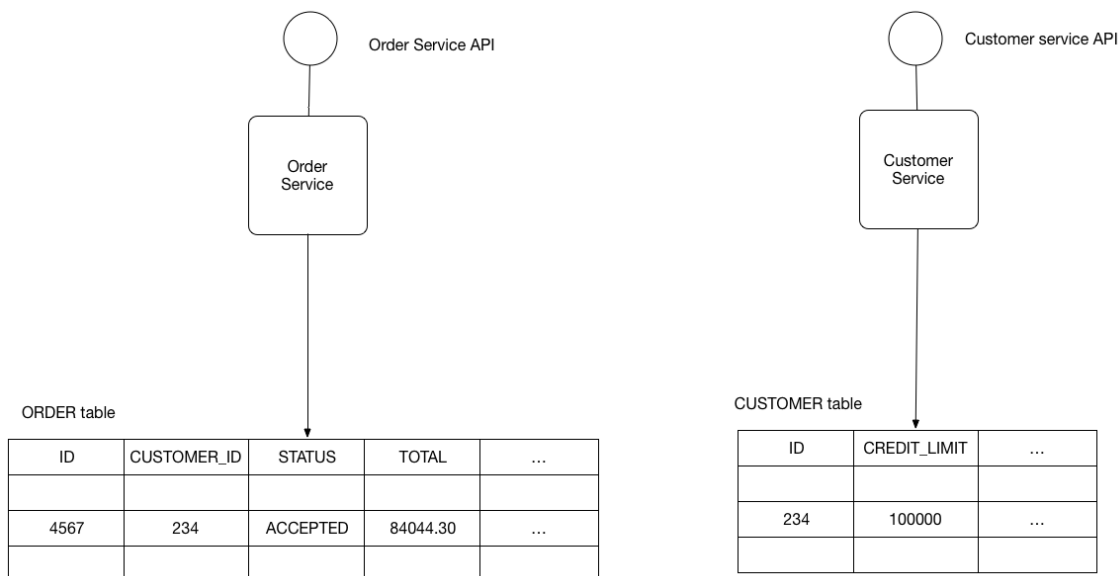


Figura 6.2: Exemplo de dois serviços com uma base de dados privada para cada serviço^[50]

6.2 Arquitetura a Utilizar

No desenvolvimento deste projeto será adotada a arquitetura de microsserviços com recurso aos padrões apresentados na Secção 6.1.1. Esta será a arquitetura a utilizar devido às vantagens apresentadas, nomeadamente: flexibilidade no desenvolvimento, facilidade de manutenção dos serviços separados, diminuição da complexidade de compreensão do sistema como um todo, decopulação dos diferentes serviços dentro do sistema, implementação de mudanças e entregas rápidas, eliminação do *single point of failure* e a possibilidade de escalar cada serviço como for necessário ao longo do tempo de vida do produto.

Estas vantagens também se mostram de elevada importância devido aos planos futuros que a empresa tem para a plataforma. A dimensão atual da empresa pode não justificar a adoção de uma arquitetura desta complexidade mas a mesma acredita que o seu crescimento futuro, e até mesmo a possibilidade de implementação da plataforma em outras empresas de maior dimensão e com necessidades semelhantes, dita a necessidade de criar desde já uma estrutura que consiga responder a um maior número de utilizadores. Como mencionado na introdução deste documento, atualmente a Grama conta com cerca de 30 colaboradores e, deste modo, nesta fase atual nunca existirão mais do que 30 utilizadores em simultâneo na plataforma, no entanto, a empresa quer que a plataforma tenha capacidade para suportar mais pedidos em simultâneo.

Como mencionado na Secção 1.2, a Ferramenta Interna da Grama é composta por quatro módulos principais onde estarão presentes as funcionalidades desejadas para esta primeira implementação da plataforma, sendo que também já se encontram pensados possíveis módulos futuros para a mesma. O uso deste tipo de arquitetura ajuda à separação do código destes diferentes módulos e uma maior facilidade de integrar novos módulos no sistema assim como facilitar o trabalho

simultâneo no desenvolvimento dos quatro módulos atuais em conjunto com o outro estagiário.

Esta decisão de utilizar uma arquitetura de microsserviços não foi tomada sem antes discutir as vantagens e desvantagens assim como o objetivo final da plataforma juntamente com a empresa e com o meu colega Francisco Bugalho que se encontra a desenvolver os outros dois módulos desta versão inicial da Ferramenta Interna da Grama. Cada estagiário realizou a sua análise independente das vantagens e desvantagens dos diferentes tipos de arquitetura e existiu também a sugestão por parte dos orientadores da empresa para que esta fosse baseada em microsserviços, sendo que esta não foi uma restrição para a implementação da plataforma. Tendo em conta o que foi estudado por cada estagiário, a sugestão da empresa e também os requisitos funcionais e não funcionais identificados, fez sentido a escolha de uma arquitetura de microsserviços para a implementação desta plataforma. Nas próximas secções será discutido em maior detalhe o processo de decisão para cada elemento da arquitetura que seja relevante detalhar.

6.2.1 Resposta da Arquitetura aos Requisitos Identificados

A decisão da arquitetura também teve em consideração a resposta da mesma aos requisitos não funcionais apresentados na Secção 4.4. De seguida, será apresentada brevemente a forma como a decisão da arquitetura procura responder a cada requisito não funcional identificado.

Disponibilidade

A adoção de uma arquitetura de microsserviços permite diminuir o tempo em que o sistema se encontra indisponível cada vez que existe um *deploy* pois cada microsserviço pode ser colocado em produção independentemente, o que não aconteceria caso se tratasse de um monólito. Desta forma, as restantes funcionalidades do sistema que são independentes ao microsserviço que se está a dar *deploy*, continuarão a funcionar. Existe um ambiente de testes onde é feito o desenvolvimento local sendo que o *deploy* em si deixa o microsserviço que está a ser atualizado indisponível durante poucos minutos até a nova versão se encontrar no ambiente de produção. Este período não deve exceder os 3 minutos de indisponibilidade por cada atualização a um microsserviço que seja realizada. Este processo deve ser feito tendo em conta os períodos em que é mais provável os colaboradores estarem a usar a plataforma ou avisando previamente dessa mesma situação para que não cause transtorno quando é realizada uma atualização para uma nova versão.

Segurança

A existência de uma *API Gateway* onde chegam inicialmente todos os pedidos do exterior do sistema e onde são verificados os certificados desses pedidos pois

chegam através de HTTPS permite responder ao requisito não funcional da segurança. Cada pedido chega com uma *token* JWT que é confirmada junto do serviço AWS Cognito^[6] para validar a identidade do utilizador que realizou o pedido. As comunicações com o exterior do sistema são sempre realizadas através de canais HTTPS para que exista encriptação.

A utilização do AWS Cognito ^[6] é também uma decisão que pretende melhorar a segurança do sistema como um todo. Como mencionado anteriormente, é este serviço que gere a emissão e validação dos *tokens* JWT que são utilizados em cada pedido^[14]. As credenciais de acesso dos utilizadores são guardadas neste serviço^[15] e, sempre que é necessário, o sistema realiza a validação necessária junto do Cognito. Toda esta validação é realizada através de canais HTTPS entre o sistema e o AWS Cognito permitindo assim a concentração dos esforços no desenvolvimento da plataforma em si, procurando responder ao que a empresa deseja, sem ser necessária a criação de raiz de um sistema de validação de credenciais, emissão e revogação de *tokens* assim como a validação das mesmas, esta responsabilidade fica entregue ao serviço externo.

Apenas os utilizadores com credenciais de acesso válidas poderão autenticar-se na plataforma e receber *tokens* JWT emitidos pelo AWS Cognito. Como todos os pedidos realizados ao sistema necessitam de um *token* válido, um utilizador que não esteja autenticado não conseguirá aceder a informação do sistema.

Ainda do ponto de vista da segurança, a validação da *role* do utilizador que faz o pedido em cada *endpoint* exposto pelo *backend* também permite uma filtragem logo à partida dos pedidos realizados. O *endpoint* de apagar um utilizador, por exemplo, apenas deverá poder ser utilizado por administradores da plataforma. No Capítulo 5 a escolha das tecnologias tem isto em conta devido às funcionalidades de disponibilização de *endpoints* autorizados do Java Spring. A *role* do utilizador é guardada localmente no sistema e também no AWS Cognito para facilitar a validação rápida do mesmo tanto no *frontend* como no *backend*.

Desempenho

A arquitetura escolhida prevê a utilização de uma API *gateway* que permite que os pedidos que chegam através de HTTPS sejam logo validados segundo o que foi descrito anteriormente para garantir a segurança. Esta descriptação permite também que os pedidos dentro do sistema, já fora da internet, sejam feitos em HTTP para que a comunicação interna não sofra problemas de desempenho derivado da encriptação e descriptação de pedidos HTTPS, tornando-se, assim, mais simples e rápida.

A própria arquitetura de microsserviços, em junção com a referida API *gateway*, permite que os pedidos cheguem à mesma e sejam distribuídos para os microsserviços responsáveis por tratar dos mesmos. Desta forma, cada microsserviço que se encontra a correr numa máquina diferente, conforme será explicado em detalhe mais à frente na Secção 6.4, poderá tratar dos seus pedidos. Caso se tratasse de uma arquitetura monolítica a correr numa máquina, todos os pedidos teriam de ser processados no mesmo local, podendo existir uma perda de desempenho.

6.3 Seleção das Tecnologias

Do ponto de vista da programação do *backend* da Ferramenta Interna da Grama, será utilizado **Java Spring** devido à sua boa aplicação para microsserviços e pelo facto de ser considerada uma escolha sólida para aplicações mais complexas. A integração com tecnologias de *frontend* também é uma razão para a escolha desta tecnologia. O uso desta *framework* irá mitigar os riscos associados ao desconhecimento e falta de prática com as tecnologias, pois já foi anteriormente utilizada em outros projetos dentro da empresa, o que significa que existe maior suporte por parte de programadores mais experientes, e também em projetos de unidades curriculares durante o percurso académico.

Para o *frontend* será utilizado **React** devido à sua boa documentação, que facilita a aprendizagem da linguagem e a resolução de problemas, bem como à capacidade de reutilização dos componentes, importante neste projeto pois a identidade visual do mesmo fará uso de componentes reutilizáveis. O código de React é também considerado de fácil leitura, o que é uma vantagem quando se trata de um projeto que poderá no futuro ter mais programadores associados. É também uma tecnologia já usada amplamente noutros projetos dentro da empresa pelo que existe apoio e partilha de conhecimento sobre a mesma.

Quanto ao fornecedor do serviço *cloud*, irá utilizar-se *Amazon Web Services* (AWS) pois é o mais utilizado no mercado atualmente^[32] e também aquele que a empresa implementa regularmente. Desta forma, os serviços que necessitem de suporte da *cloud* serão efetuados com recurso a AWS.

6.4 Desenho da Arquitetura

Na secção que agora se introduz será descrita a arquitetura do *software* da Ferramenta Interna da Grama, tendo em conta as decisões tomadas nas secções anteriores.

Amazon Web Services

Dentro dos recursos disponíveis da *Amazon Web Services* (AWS) serão utilizados os seguintes para as diversas funcionalidades necessárias dentro do funcionamento normal da plataforma:

- **S3** - Armazenamento de ficheiros estáticos será feito com o recurso ao *Simple Storage Service* (S3)^[10];
- **SES** - Envio de emails, como é o caso de emails de primeiro início de sessão, será através do *Simple Email Service* (SES)^[11];
- **Cognito** - Gestão de autenticação será feita através do Cognito^[6];
- **RDS** - As bases de dados relacionais irão recorrer ao serviço de *Relational Database Service* (RDS)^[9];

- **EC2** - Os diferentes microsserviços irão correr em máquinas virtuais *Elastic Compute Cloud* (EC2)^[7];
- **ELB** - A gestão de carga será feita através do uso de *Elastic Load Balancer* (ELB) ^[8];
- **CloudFront CDN** - A distribuição de conteúdo estático será feita com recurso ao *AWS CloudFront Content Delivery Network* (CDN)^[5].

Modelo C4

O modelo C4 propõe a criação de um conjunto hierárquico de diagramas que são abstrações dos diferentes níveis de um sistema de *software*^[17]. Desta forma, é possível entender melhor a arquitetura geral do *software* de um sistema mais complexo. O modelo sugere quatro níveis onde muda o ponto de vista sobre os diferentes elementos dos respetivos diagramas^[17]:

- **Context diagram (Nível 1)** - Mostra o sistema de *software* que se está a desenvolver e como este se insere no ambiente à sua volta. Considera as interações com os utilizadores e com outros sistemas externos ao mesmo;
- **Container diagram (Nível 2)** - Este segundo nível aproxima e expande o sistema de *software*, mostrando os diferentes *containers* que o compõem. São estes, por exemplo, aplicações, microsserviços e bases de dados;
- **Component diagram (Nível 3)** - Neste nível são expandidos os *containers* individuais para revelar os componentes que o constituem e como estes interagem entre si e com outros *containers* exteriores a si próprio;
- **Code diagram (Nível 4)** - O nível mais detalhado, o do código, entra em detalhe sobre como cada componente é implementado em termos do código.

A Figura 6.3 apresenta exemplos dos diferentes níveis do modelo C4 e como em cada nível existe um maior detalhe de cada parte do sistema que é apresentada no nível superior.

De seguida será apresentada a arquitetura da Ferramenta Interna da Grama recorrendo aos diferentes níveis dos diagramas do modelo C4. Serão utilizados todos os níveis exceto o último, o nível de "*Code*", pois esse pressupõe que a estrutura do código para cada componente já se encontra bem definida, o que não é o caso nesta fase.

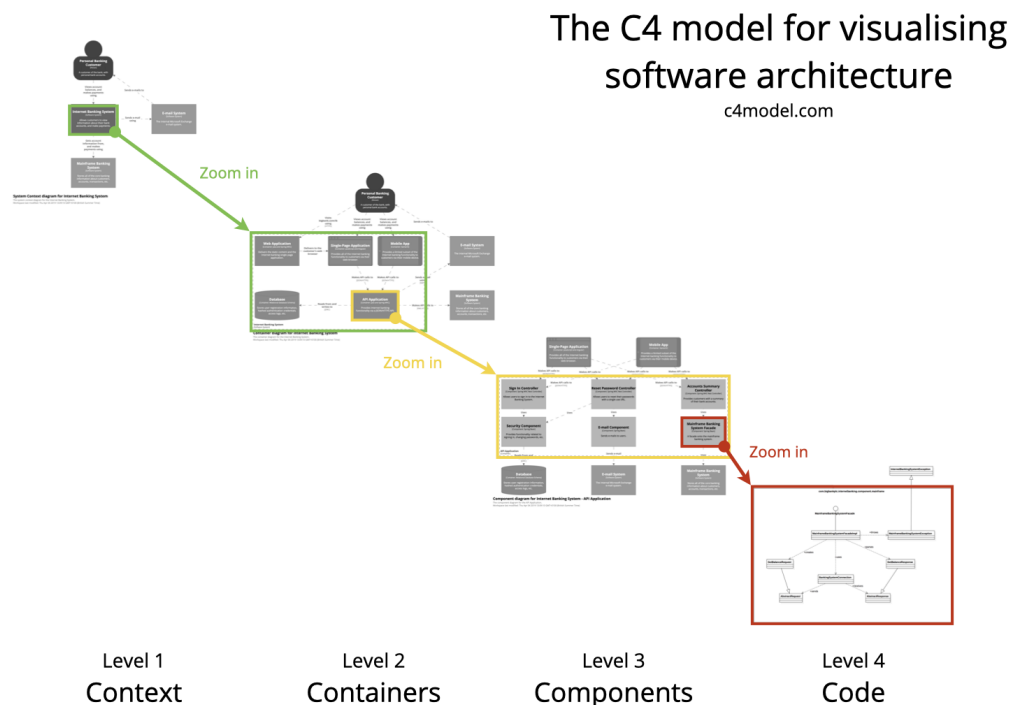


Figura 6.3: Níveis do modelo C4^[17]

Context Diagram

A Figura 6.4 descreve o *Context Diagram* (Diagrama de Contexto) que foi desenhado para a Ferramenta Interna da Grama. Na parte superior é possível ver os três tipos de utilizadores diferentes que irão interagir com a plataforma: Administrador, Gestor e Colaborador. Estes utilizadores usam a Ferramenta Interna da Grama que, neste nível, o mais elevado de abstração, é um sistema de *software* sem entrar em detalhe de como funciona internamente. Este sistema de *software* utiliza os serviços *cloud* da Amazon.

Container Diagram

No *Container Diagram* (Diagrama de Containers) apresentado na Figura 6.5 é agora possível analisar as partes integrantes do sistema de *software* que foi visto no nível anterior. Podemos agora ver com algum detalhe as tecnologias que serão utilizadas em cada *container* e como estes comunicam entre si e com os diferentes serviços necessários.

Neste nível continuamos a ver os diferentes tipos de utilizadores e como eles interagem com o sistema através de uma *Single-Page Application*, isto é, uma plataforma *web*. Dentro do sistema de *software* podemos também observar os diversos serviços da AWS que estão a ser utilizados e como estes se integram com o resto dos *containers* do sistema para que cada um possa realizar a sua função.

As ações realizadas pelos utilizadores na página *web* são traduzidas em pedidos *Hypertext Transfer Protocol Secure* (HTTPS) que são realizados ao *backend*. Numa

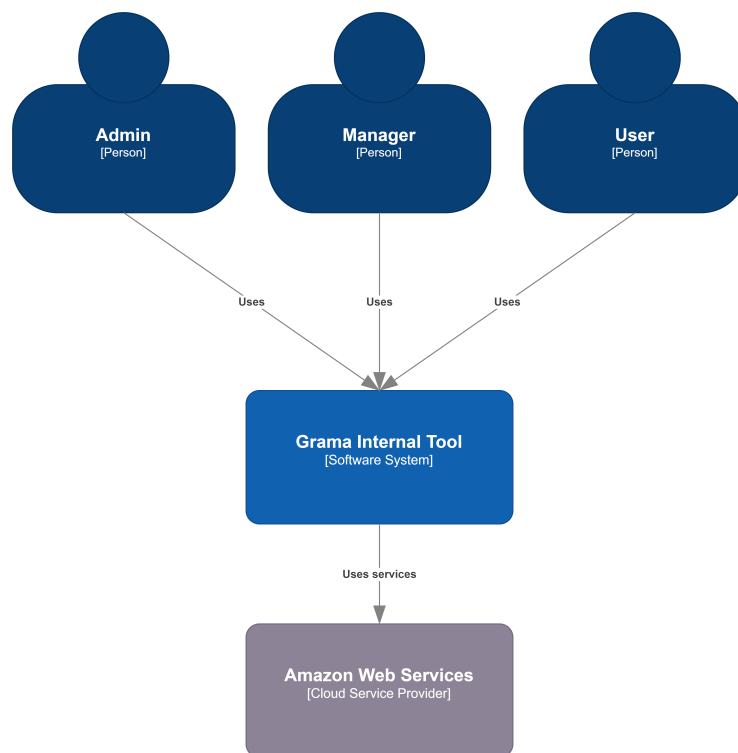


Figura 6.4: Diagrama de Contexto (*Context Diagram*) da Ferramenta Interna da Grama

primeira fase os pedidos chegam ao *External Load Balancer* da AWS que trata da verificação dos certificados dos pedidos HTTPS encriptados e, posteriormente a essa verificação, envia os pedidos para a *API Gateway*, já em *Hypertext Transfer Protocol* (HTTP) sem encriptação. A *API Gateway* recebe os pedidos e autentica os mesmos junto do serviço *Cognito* da AWS^[6]. Após essa verificação, envia os pedidos para os serviços a que estes dizem respeito.

É nesta vista que podemos ver também os diferentes *microserviços* que serão implementados para o correto funcionamento da plataforma, assim como as interações destes com as restantes partes e os serviços da AWS.

Existem dois *microserviços* que irão garantir as diversas funcionalidades da plataforma. Estarão a correr em máquinas *EC2* da AWS^[7] e são os seguintes:

- ***User Management (Gestão de Colaboradores)*** - A gestão das contas dos utilizadores da plataforma assim como de toda a sua informação será gerida por este *microserviço*. É também dentro deste que serão geridas as funcionalidades de gestão de departamentos. Este *microserviço* terá uma base de dados própria que será uma base de dados relacional utilizando o serviço *RDS* da AWS^[9]. Irá também aceder a armazenamento estático na forma de *Bucket S3* da AWS^[10] onde serão armazenados os ficheiros que poderão estar associados aos utilizadores assim como as imagens necessárias;

- **Equipment Management (Gestão de Equipamentos)** - As funcionalidades de gestão dos equipamentos estarão ao encargo deste microsserviço. Inclui a gestão dos equipamentos, gestão de orçamentos dos utilizadores, gestão de pedidos de equipamentos e de reparação dos mesmos. Tem uma base de dados relacional utilizando o serviço RDS da AWS^[9]. Vai buscar informação estática como ficheiros associados aos equipamentos com recurso a *Bucket S3* da AWS^[10]. Também fará uso do serviço de email SES da AWS^[11] para envio de emails informativos sobre os pedidos de equipamentos e de reparação.

No diagrama está representado um *Bucket S3* da AWS^[10] sem qualquer ligação a outros *containers* por razões de facilidade de leitura e interpretação do diagrama. Nos pontos acima, assim como nos diagramas de componentes, é possível perceber por quais elementos este *Bucket* é utilizado e para que fim. Para além dos serviços mencionados acima e que usam um *Bucket S3*, também o *CloudFront* CDN da AWS^[5] faz uso de um para obter o conteúdo estático do *website* que irá distribuir.

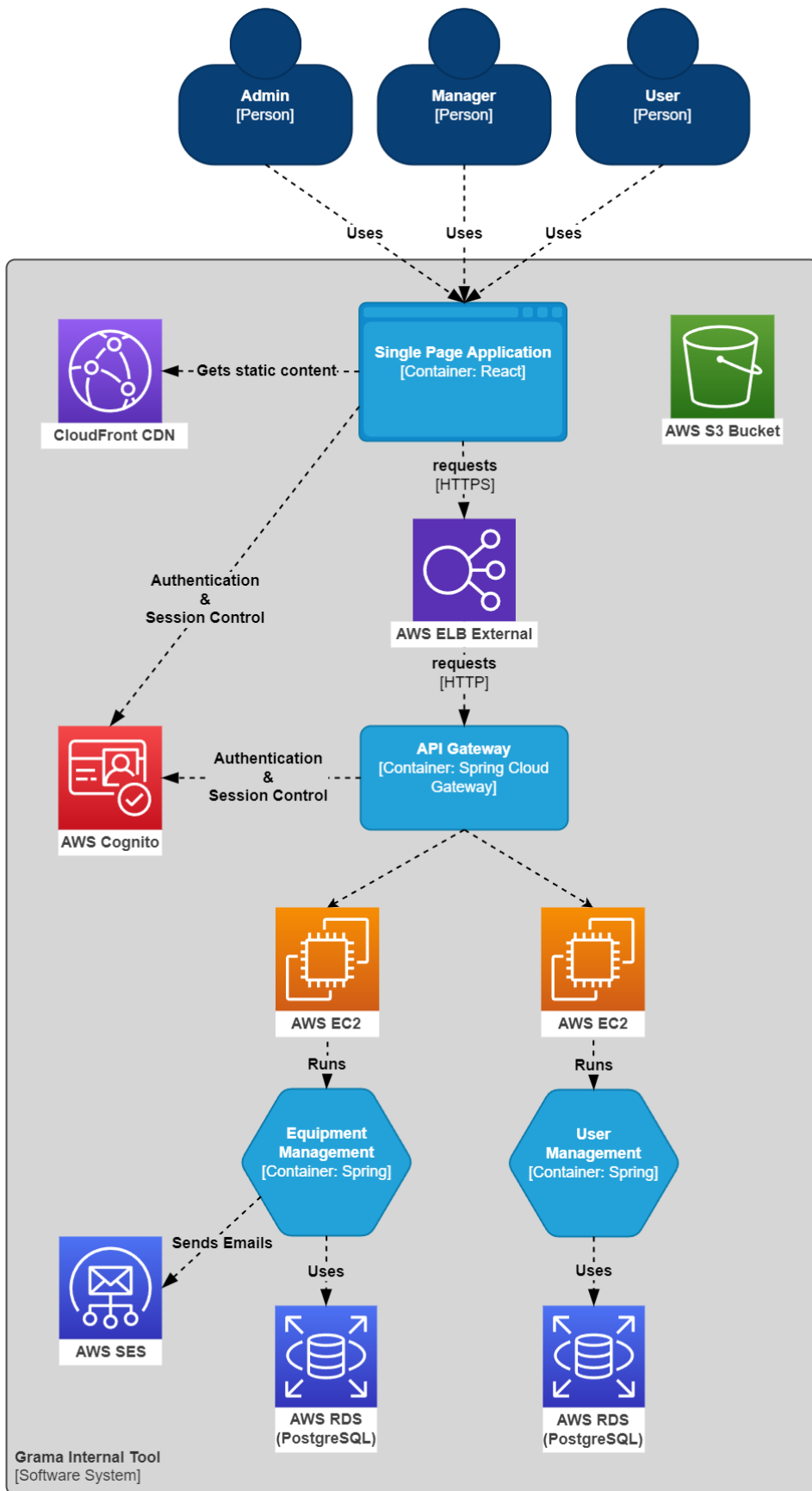


Figura 6.5: Container Diagram da Ferramenta Interna da Grama

Component Diagram

Neste nível de abstração mais detalhado do modelo C4, expandem-se os *containers* do diagrama anterior de modo a ver como funciona cada um deles e quais os componentes que o compõem.

A Figura 6.6 descreve a composição e estrutura interna do *container* que representa a *API Gateway* presente no sistema. Esta recebe, através de HTTP, os pedidos que chegam a partir do *External Load Balancer* da AWS^[8] e realiza a autenticação dos mesmos junto do serviço *Cognito* da AWS^[6]. Estes pedidos já autenticados são posteriormente redirecionados para os serviços que lhes dizem respeito.

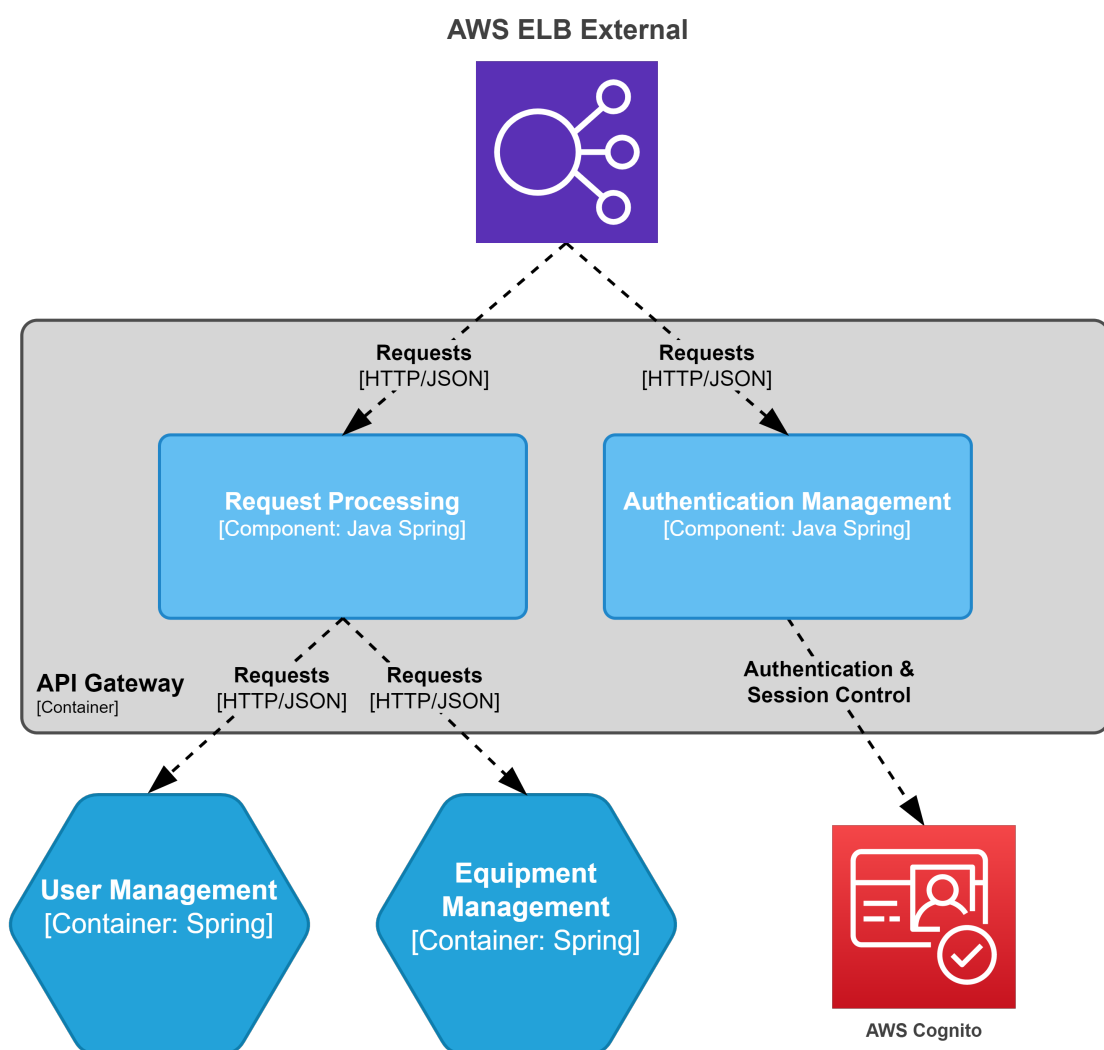


Figura 6.6: *Component Diagram* da *API Gateway*

A Figura 6.7 apresenta os componentes que formam o microserviço de Gestão de Colaboradores e como estes comunicam com os outros serviços presentes no sistema. Dentro deste microserviço existem dois componentes: um que gere todos os utilizadores e outro que gere os departamentos. O primeiro gere toda a informação referente aos utilizadores da plataforma e necessita de comunicar com o

Bucket S3 da AWS^[10] para gerir os ficheiros e fotos de perfil dos utilizadores. Por fim, os dois componentes partilham uma base de dados relacional única onde guardam e gerem toda a informação, que não seja ficheiros ou imagens, referente aos utilizadores.

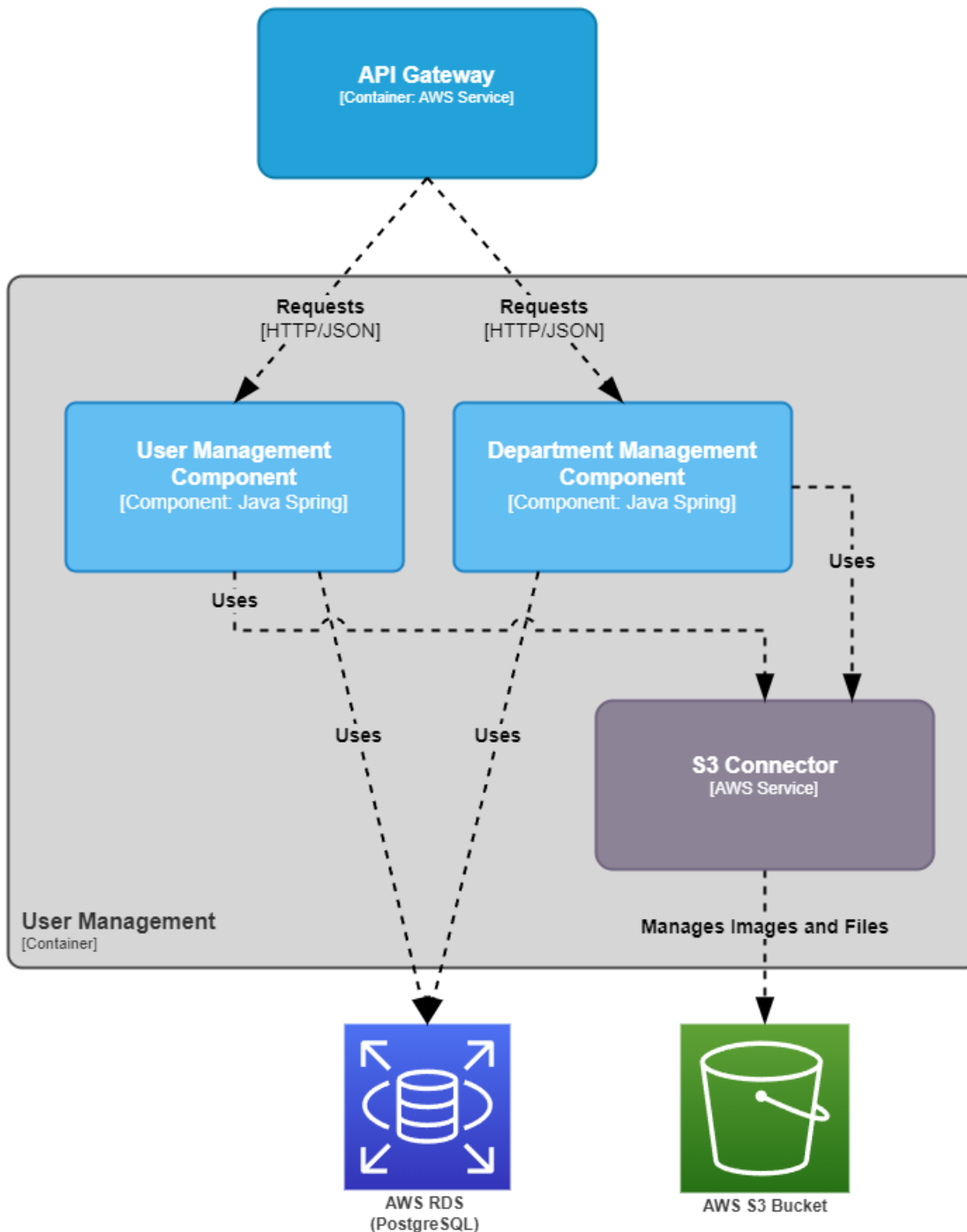


Figura 6.7: *Component Diagram* do microserviço de Gestão de Colaboradores

A Figura 6.8 apresenta os diferentes componentes que compõem o microserviço de gestão de equipamentos. Existe uma base de dados relacional única onde são guardadas todas as informações referentes aos equipamentos da empresa. Os ficheiros associados aos equipamentos encontram-se guardados num *Bucket*

S3 da AWS^[10], pelo que o componente de gestão de equipamentos necessita de comunicar com este. Por fim, utiliza o SES da Amazon^[11] para enviar emails informativos acerca dos pedidos de equipamentos e reparação.

A Figura 6.9 detalha o funcionamento geral da página *web* recorrendo à apresentação dos componentes que a constituem do ponto de vista do *frontend*. Sendo que a tecnologia escolhida é React, os componentes dentro da página *web* podem ser organizados conforme o diagrama. O *Router* encaminha os pedidos para as *Views* que guardam o seu estado nos diversos *States* apresentados. A divisão das *Views* e *States* foi realizada a pensar nos diferentes requisitos e funcionalidades da plataforma. Os pedidos são enviados através de HTTPS para o *backend* para serem tratados. É de notar que o serviço Cognito da AWS^[6] fornece funcionalidades que permitem, a partir deste componente do *frontend*, realizar pedidos ao mesmo sem ter que comunicar diretamente com o *backend*, de modo a gerir a sessão.

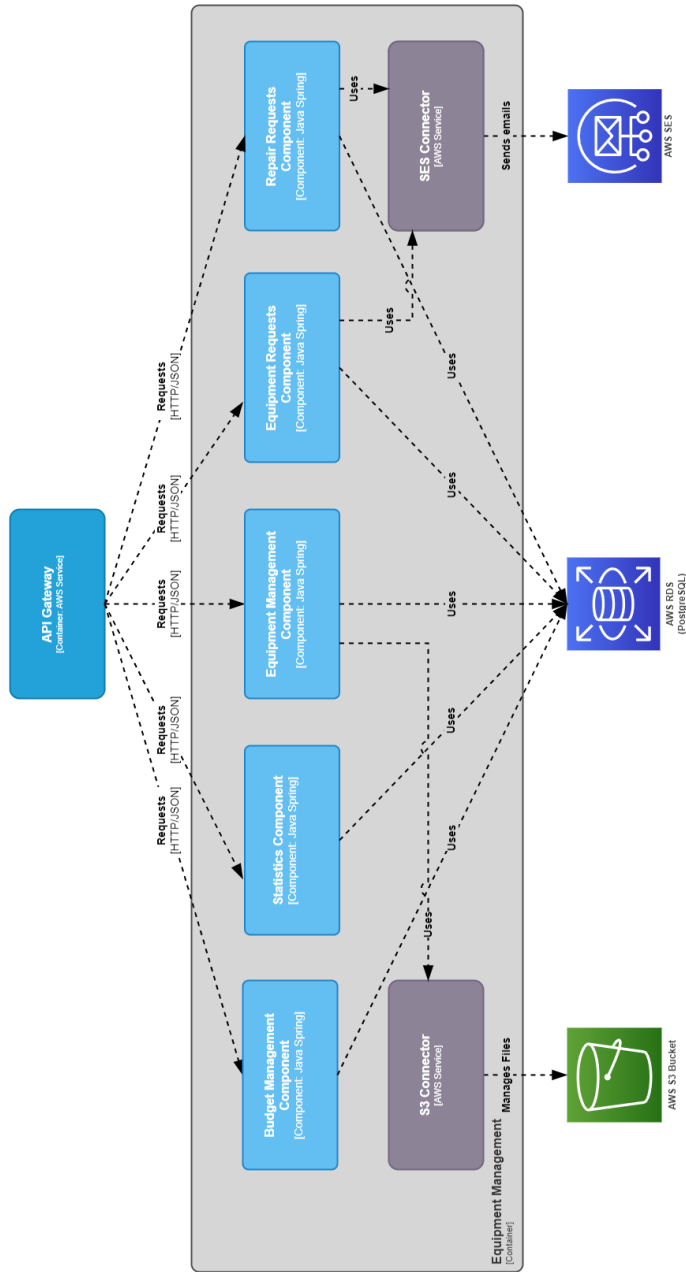


Figura 6.8: Component Diagram do microserviço de Gestão de Equipamentos

6.5 Diagrama Entidade-Relacionamento

A elaboração de um diagrama Entidade-Relacionamento (E-R) ajuda a visualizar e perceber quais as diversas entidades que fazem parte do problema e como elas se relacionam entre si. É também possível ver quais as propriedades de cada entidade de modo a entender melhor qual a informação que vai ser guardada nas bases de dados do sistema.

Os diagramas E-R foram elaborados com recurso à ferramenta ONDA (*Online Database Architect*)^[21] do Departamento de Engenharia Informática (DEI) onde é possível criar o modelo conceptual da base de dados e, de forma automática, gerar o diagrama físico onde é possível visualizar as chaves estrangeiras e tabelas adicionais que são necessárias.

Todas as entidades serão acompanhadas de mais seis propriedades que não se encontram representadas nos diagramas para facilidade de leitura: *created_by*, *created_at*, *updated_by*, *updated_at*, *deleted_by*, *deleted_at*. Estas propriedades vão ajudar na capacidade de auditoria do sistema pois desta forma é possível saber toda a informação sobre quem criou, editou, e removeu certos dados e o momento exato dessa operação.

É de notar que a operação de remover uma entrada presente numa base de dados do sistema nunca levará à sua remoção total, a não ser que seja explicitamente necessário fazê-lo. Qualquer outra operação de remoção irá ser feita como "*soft delete*", ou seja, existirá uma propriedade (e.g. *deleted*) que indica se essa entrada foi apagada para que, caso seja necessário, seja possível recuperar a informação ao ativar de novo a mesma. As duas principais entidades que encontramos nesta plataforma são os Utilizadores (*Users*) e os Equipamentos (*Equipments*). A par destas, existem outras entidades fundamentais e que se relacionam com estas duas como se representa nos diagramas que se seguem.

A Figura 6.10 apresenta o diagrama de base de dados conceptual onde estão representadas todas as entidades que são necessárias para o cumprimento dos requisitos e funcionamento correto da plataforma e também as relações entre as mesmas através de setas que representam a sua cardinalidade. Para além das propriedades que possui, um *User* também pode ter associado a si o histórico de bónus e de salários atribuídos que são criados apenas por Administradores. Os departamentos são compostos por utilizadores normais que são membros do departamento (com permissões de *User*) e também por gestores (que podem ser *Managers* ou *Administrators*).

Um utilizador tem também associado a si um histórico do seu orçamento que se vai alterando ao longo do tempo conforme vai recebendo valor ou conforme este vai sendo retirado quando pede um novo equipamento e este é comprado. Um utilizador pode criar pedidos de novos equipamentos que terão que ser aprovados por outros utilizadores com permissões de *Administrator*. Os pedidos de reparação, encontram-se associados a um equipamento específico já existente. Os equipamentos em si, para além da informação associada a cada um, podem estar alocados a um utilizador e podem estar também associados a pedidos de repa-

ração. Existem tabelas estáticas para os campos que tomam sempre o mesmo conjunto de valores de modo a que estes sejam coerentes entre as diversas entidades.

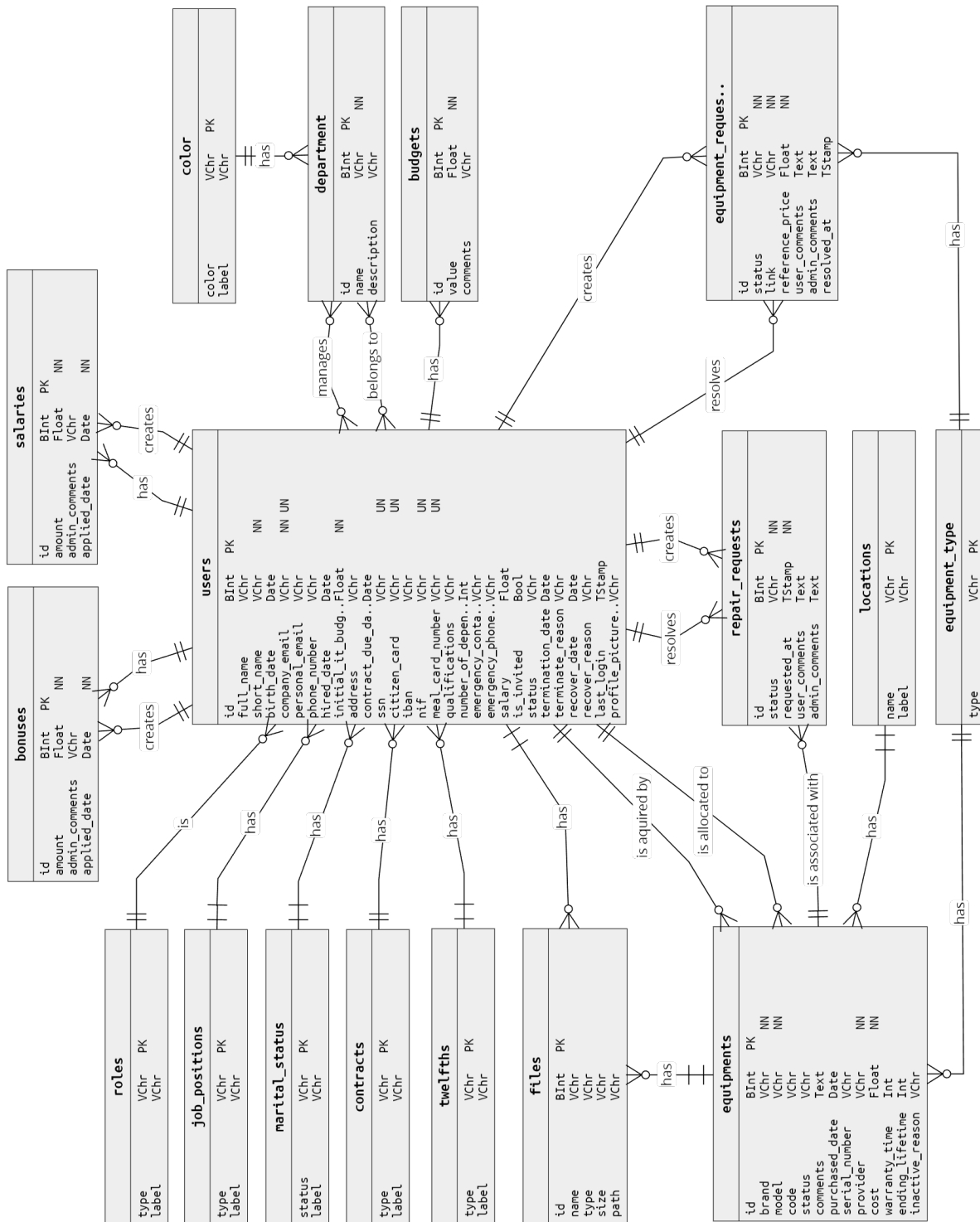


Figura 6.10: Diagrama E-R - Modelo Conceptual

A Figura 6.11 apresenta o diagrama físico que foi automaticamente gerado pelo ONDA^[21] após o desenho do diagrama conceptual (Figura 6.10). Este diagrama físico contém já as chaves estrangeiras que são necessárias incluir em cada tabela de modo a executar corretamente a cardinalidade entre entidades, representada no diagrama conceptual. Inclui também as tabelas resultantes das relações *Many-to-Many* (M-N) de modo a representar estas mesmas relações.

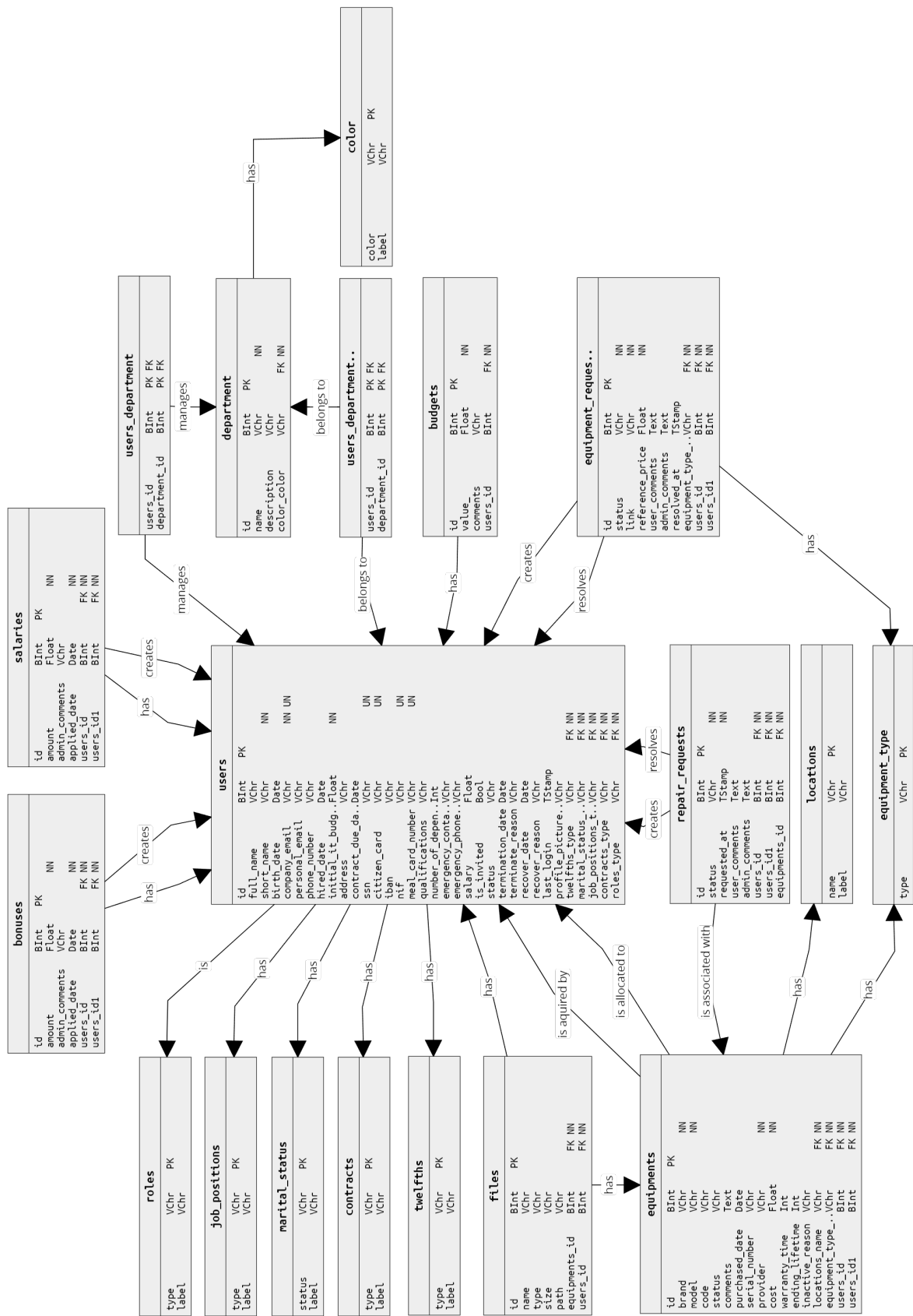


Figura 6.11: Diagrama E-R - Modelo Físico

Capítulo 7

Desenvolvimento

Neste capítulo será apresentado o processo de desenvolvimento da Ferramenta Interna da Grama que foi levado a cabo durante o período do segundo semestre do estágio. Primeiro será apresentado em detalhe a organização do desenvolvimento assim como a equipa de desenvolvimento e como esta se organizou. De seguida, na Secção 7.2 serão analisadas as funcionalidades que foram concluídas à luz dos requisitos funcionais definidos durante o primeiro semestre que se encontram presentes na Secção 4.3.

Na Secção 7.3 serão apresentadas evidências da implementação das funcionalidades mais relevantes da plataforma através da inclusão de imagens da mesma durante o seu funcionamento. Na Secção 7.4 realiza-se uma pequena reflexão sobre os riscos identificados antes do início do desenvolvimento, agora após o mesmo. Por fim, na Secção 7.5 é apresentado o trabalho futuro para a plataforma.

7.1 Processo de Desenvolvimento

O desenvolvimento da Ferramenta Interna da Grama, como mencionado na Secção 3.2, realizou-se aplicando uma metodologia ágil baseada em *Scrum* sendo que a equipa de desenvolvimento foi composta pelo designer da empresa alocado ao projeto e os dois estagiários. Adicionalmente, os orientadores de cada estagiário fizeram parte das diversas cerimónias que foram sendo realizadas ao longo do período de desenvolvimento.

O processo de desenvolvimento começou mesmo antes da realização do estágio através da criação de esboços por parte do designer que serviram de base para o levantamento de requisitos realizado no primeiro semestre, conforme exposto na Secção 4.2. Desta forma, os programadores tinham já *mockups* de alta fidelidade disponíveis aquando do início do desenvolvimento, *mockups* esses que serviram de guia para a construção dos diversos componentes e ecrãs da plataforma *web*. Esses *mockups* foram recebendo a revisão e aprovação da *Product Owner* e do *Scrum Master* antes do início do desenvolvimento por parte dos alunos para que estivessem prontos antes de iniciar o segundo semestre.

O desenvolvimento no segundo semestre começou pela implementação dos diversos componentes de *frontend* comuns a toda a plataforma como é o caso do layout base, botões, caixas de texto, entre outros. Nesta fase foi necessária uma maior coordenação de tarefas entre cada aluno pois estes componentes viriam a ser utilizados por ambos nos seus módulos específicos. Após a implementação de todos os elementos comuns, cada aluno focou-se inteiramente na implementação dos seus módulos. À medida de cada tarefa era cumprida, esta era submetida a uma revisão do código por parte do *scrum master* e outro programador experiente da empresa para minimizar os erros cometidos devido à inexperiência dos alunos e também aproveitando para guiar os mesmos através do ensino de boas práticas nas linguagens utilizadas.

No desenvolvimento dos módulos deste estágio, iniciava-se a implementação de uma determinada funcionalidade pelo desenvolvimento do *backend* e posteriormente a criação das respetivas páginas que habilitavam essa funcionalidade no *frontend*.

Ao longo de todo o processo de desenvolvimento, iam sendo criados testes unitários no *backend* com o objetivo de confirmar o correto comportamento de cada pequena funcionalidade implementada como, por exemplo, confirmar se um certo *endpoint* de criação realmente criava e guardava a entidade desejada, com os campos desejados, de forma correta. Para confirmar o comportamento no *frontend*, ao longo do desenvolvimento, as funcionalidades iam sendo testadas com dados representativos de uma utilização normal da plataforma.

7.1.1 Organização das Tarefas

As tarefas dos diversos *sprints* encontravam-se estruturadas e organizadas com recurso a ferramenta *Linear*^[40] que é uma ferramenta criada para facilitar o trabalho em equipa e a organização de tarefas. Foram criados três quadros distintos para melhor diferenciar o trabalho de cada aluno e separar as tarefas que dizem respeito a cada um.

- ***Internal Tool Common*** tarefas comuns a todo o projeto da Ferramenta Interna da Grama. Exemplo: componentes de *frontend* comuns, configuração do serviço S3 da Amazon, entre outros;
- ***Users and IT Management*** tarefas exclusivas dos módulos de Gestão de Colaboradores e Gestão de Equipamentos;
- ***Time-off and Recruitment*** tarefas exclusivas dos módulos de Gestão de Ausências e Gestão de Recrutamento.

Em cada um dos quadros referidos existe um *backlog* com todas as tarefas por desenvolver. Cada quadro tem uma vista com seis colunas indicando o estado de cada tarefa ativa:

- **Todo:** tarefas a desenvolver durante o *sprint* atual;
- **In progress :** tarefas em desenvolvimento;
- **In Review:** tarefas em *code review*;
- **Ready to test:** tarefas aprovadas e prontas para teste;
- **Done:** tarefas dadas como concluídas;
- **Canceled:** tarefas canceladas.

Na Figura 7.1 podemos observar o exemplo de um dos quadros, neste caso o de *Users and IT Management* e a distribuição das tarefas pelas diversas colunas. É possível observar que cada tarefa tem uma pessoa alocada assim como alguns identificadores que ajudam na organização das mesmas.

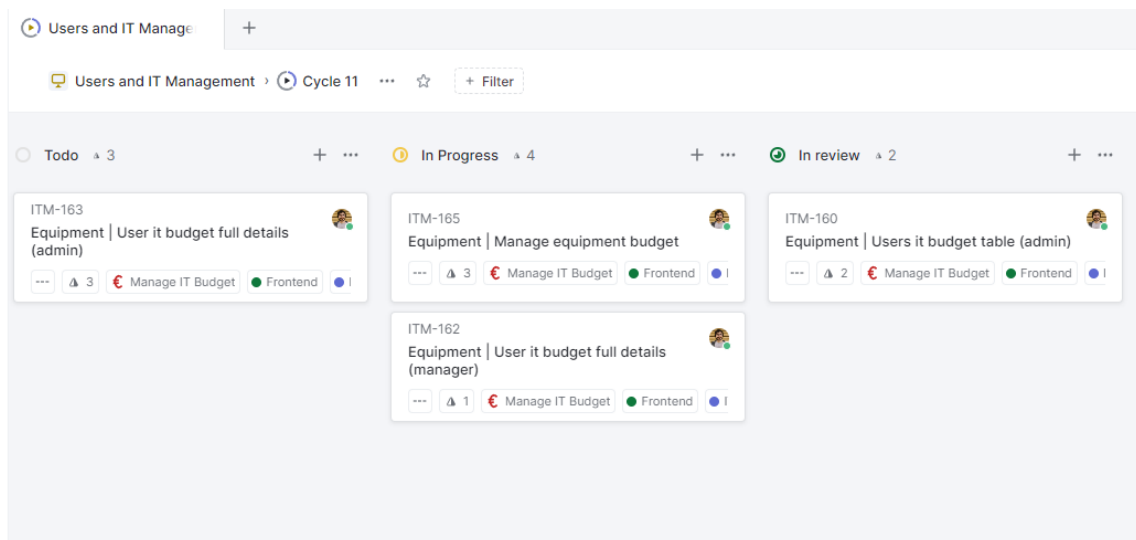


Figura 7.1: Exemplo de um quadro e distribuição de tarefas no *Linear*

7.2 Funcionalidades Desenvolvidas

De seguida apresentam-se em diversas tabelas quais dos requisitos funcionais definidos na Secção 4.3 foram concluídos, ou não, durante a fase de desenvolvimento do estágio.

Módulo de Autenticação

Este primeiro módulo diz respeito a toda a lógica de autenticação e sessão que foi necessário implementar para a Ferramenta Interna da Grama com recurso ao uso do serviço Cognito da AWS. No *frontend* foi utilizado o AWS Amplify^[13] que oferece métodos já criados de integração com o Cognito da AWS para realizar os diversos *flows* de autenticação necessários.

É possível observar na Tabela 7.1 que todos estes requisitos de autenticação foram implementados.

Área	ID	Requisito	Prioridade	Implementado
Autenticação	RF-1	Primeiro início de sessão	M	✓
	RF-2	Iniciar sessão	M	✓
	RF-3	Terminar sessão	M	✓
	RF-4	Recuperar palavra-passe	M	✓

Tabela 7.1: Requisitos Funcionais Implementados - Módulo de Autenticação

Módulo de Gestão de Colaboradores

Neste módulo são realizadas todas as ações que dizem respeito à gestão dos utilizadores, da sua informação, estado das contas e organização dos departamentos.

É possível verificar que este módulo viu todos os seus requisitos com prioridade *Must* serem implementados durante o estágio. No entanto é possível observar que alguns requisitos priorizados como *Should* e *Could* não foram implementados.

Na Tabela 7.3 pode verificar-se que os filtros da lista de utilizadores não foram de todo implementados. Optou-se, em reunião com o resto da equipa, por implementar outros requisitos de prioridades *Should* e *Could* que foram considerados mais interessantes de incluir na plataforma. As listas de utilizadores são paginadas e é possível realizar a ordenação das mesmas por cada coluna, pelo que se decidiu que os filtros das mesmas apenas seriam implementados neste estágio caso existisse tempo extra.

Tabela 7.2: Requisitos Funcionais Implementados - Módulo de Gestão de Colaboradores - Área de Gestão de Contas dos Utilizadores

Área	ID	Requisito	Prioridade	Implementado
Gestão de contas de utilizadores	RF-33	Adicionar e convidar um utilizador para a plataforma	M	✓
	RF-34	Reenviar convite para a plataforma	C	✓
	RF-35	Desativar a conta de um colaborador	S	✓
	RF-36	Reativar a conta de um colaborador	S	✓

Tabela 7.3: Requisitos Funcionais Implementados - Módulo de Gestão de Colaboradores - Área de Gestão de Informações dos Utilizadores

Área	ID	Requisito	Prioridade	Implementado	
Gestão de informações dos utilizadores	RF-5	Visualizar lista de utilizadores	conta ativa	M	✓
	RF-6		conta desativada	S	✓
	RF-7	Editar informação de utilizador(es) através da lista		C	X
	RF-8	Filtrar lista de utilizadores	por nome	C	X
	RF-9		por departamento	S	X
	RF-10		por role	S	X
	RF-11		por job position	C	X
	RF-12	Pré-visualizar informação do utilizador		W	
	RF-13	Vista detalhada do perfil completo	de todos os utilizadores	M	✓
	RF-14		de utilizadores do(s) seu(s) departamento(s)	M	✓
	RF-15		do próprio utilizador	M	✓
	RF-16	Editar informações pessoais	de todos os utilizadores	M	✓
	RF-17		do próprio utilizador	M	✓
	RF-18	Visualizar histórico salarial	de todos os utilizadores	M	✓
	RF-19		de utilizadores do(s) seu(s) departamento(s)	M	✓
	RF-20		do próprio utilizador	M	✓
	RF-21	Adicionar e remover salários		M	✓
	RF-22	Editar salários		S	✓
	RF-23	Visualizar histórico de bônus atribuídos	de todos os utilizadores	M	✓
	RF-24		de utilizadores do(s) seu(s) departamento(s)	M	✓
	RF-25		do próprio utilizador	M	✓
	RF-26	Adicionar e remover bônus		S	✓
	RF-27	Editar bônus		C	✓
	RF-28	Editar foto de perfil	de todos os utilizadores	S	✓
	RF-29		do próprio utilizador	S	✓
	RF-30	Visualizar documentos		S	✓
	RF-31	Adicionar e remover documentos		S	✓
	RF-32	Transferir documentos		S	✓

Tabela 7.4: Requisitos Funcionais Implementados - Módulo de Gestão de Colaboradores - Área de Gestão de Departamentos

Área	ID	Requisito	Prioridade	Implementado
Gestão de departamentos	RF-37	Visualizar lista de departamentos ativos	M	✓
	RF-38	Adicionar um novo departamento	M	✓
	RF-39	Editar um departamento existente	M	✓
	RF-40	Eliminar um departamento	M	✓

Módulo de Gestão de Equipamentos

O módulo que agora se apresenta é onde toda a gestão de equipamentos, pedidos e orçamentos dos utilizadores é realizada.

Nas áreas deste módulo, podemos constatar nas tabelas que todos os requisitos priorizados como *Must* foram cumpridos à exceção da totalidade da área de Gestão de Reparação de Equipamentos. A não implementação desta área foi decidida em conjunto com o orientador da empresa e com a restante equipa, incluindo a *Product Owner*, devido à falta de tempo para implementar a mesma.

Ao retirar totalmente esta área foi possível implementar noutras áreas da plataforma alguns requisitos priorizados como *Should* e *Could* para que essas mesmas áreas ficassem com um acabamento mais detalhado no que toca às suas funcionalidades. Considerou-se mais benéfico para o aluno a implementação de funcionalidades mais desafiadoras noutras áreas do que a implementação da Gestão de Pedidos de Reparação de Equipamentos visto que em termos de complexidade e lógica de programação seria bastante semelhante à Gestão de Pedidos de Equipamentos, área essa que se encontra implementada.

Na área de *Dashboard*, cujos requisitos e sua implementação se encontram na Tabela 7.5, nenhuma das funcionalidades foi implementada. Tal é justificado pelas prioridades dos requisitos que não eram *Must* e também pelo baixo valor acrescentado à plataforma. Foi decidido em equipa que esta área de implementação ficaria para o final da implementação caso existisse tempo visto que não acrescenta nenhuma funcionalidade estrutural à plataforma, apenas recolhe e apresenta informação em gráficos e tabelas.

À semelhança dos filtros das listas de utilizadores, também os filtros das listas presentes nas diversas áreas destes módulos não chegaram a ser implementados. Tal aconteceu devido à falta de tempo e pela maior priorização de outras funcionalidades e tarefas que foram consideradas pela equipa como sendo mais benéficas para o desenvolvimento. Também as ações rápidas realizadas diretamente através das listas não foram implementadas visto que as mesmas ações podem ser efetuadas a partir de outros ecrãs, pelo que não acrescentavam tanto valor à plataforma quanto outras funcionalidades.

Tabela 7.5: Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de *Dashboard*

Área	ID	Requisito	Prioridade	Implementado
Dashboard	RF-41	Visualizar informação agregada sobre equipamentos	S	X
	RF-42	Visualizar informação agregada sobre orçamentos	S	X
	RF-43	Visualizar informação agregada sobre pedidos	S	X

Tabela 7.6: Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Equipamentos

Área	ID	Requisito	Prioridade	Implementado	
Gestão de equipamentos	RF-44	Visualizar lista de equipamentos ativos	ativos	M	✓
	RF-45		inativos	M	✓
	RF-46		obsoletos	S	✓
	RF-47	Filtrar as listas de equipamentos por	colaborador a quem está alocado	S	X
	RF-48		adquirido por	S	X
	RF-49		tipo de equipamento	S	X
	RF-50		marca	C	X
	RF-51		fornecedor	C	X
	RF-52	estado	S	X	
	RF-53	Pesquisar na lista de equipamentos (através de nome, id, etc.)		S	X
	RF-54	Editar informações do equipamento		M	✓
	RF-55	Editar a informação de equipamento(s) através da lista		C	X
	RF-56	Adicionar um novo equipamento		M	✓
	RF-57	Desativar equipamento		M	✓
	RF-58	Desativar equipamento através da lista		C	X
	RF-59	Reativar equipamento		S	✓
	RF-60	Reativar equipamento através da lista		C	X
	RF-61	Alterar a que utilizador o equipamento está alocado		M	✓
	RF-62	Alocar equipamento inativo através da lista		C	X
	RF-63	Visualizar todas as informações de	qualquer equipamento	M	✓
	RF-64		equipamentos alocados a membros da(s) sua(s) equipa(s)	M	✓
	RF-65	Adicionar e remover documento(s) associado(s) ao equipamento		S	✓
	RF-66	Descarregar documento(s) associado(s) ao equipamento		C	✓
RF-67	Visualizar lista de equipamentos pessoais		M	✓	

Tabela 7.7: Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Orçamentos dos Utilizadores

Área	ID	Requisito	Prioridade	Implementado	
Gestão de equipamentos	RF-44	Visualizar lista de equipamentos ativos	ativos	M	✓
	RF-45		inativos	M	✓
	RF-46		obsoletos	S	✓
	RF-47	Filtrar as listas de equipamentos por	colaborador a quem está alocado	S	X
	RF-48		adquirido por	S	X
	RF-49		tipo de equipamento	S	X
	RF-50		marca	C	X
	RF-51		fornecedor	C	X
	RF-52		estado	S	X
	RF-53	Pesquisar na lista de equipamentos (através de nome, id, etc.)		S	X
	RF-54	Editar informações do equipamento		M	✓
	RF-55	Editar a informação de equipamento(s) através da lista		C	X
	RF-56	Adicionar um novo equipamento		M	✓
	RF-57	Desativar equipamento		M	✓
	RF-58	Desativar equipamento através da lista		C	X
	RF-59	Reativar equipamento		S	✓
	RF-60	Reativar equipamento através da lista		C	X
	RF-61	Alterar a que utilizador o equipamento está alocado		M	✓
	RF-62	Alocar equipamento inativo através da lista		C	X
	RF-63	Visualizar todas as informações de	qualquer equipamento	M	✓
RF-64	equipamentos alocados a membros da(s) sua(s) equipa(s)		M	✓	
RF-65	Adicionar e remover documento(s) associado(s) ao equipamento		S	✓	
RF-66	Descarregar documento(s) associado(s) ao equipamento		C	✓	
RF-67	Visualizar lista de equipamentos pessoais		M	✓	

Tabela 7.8: Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Pedidos de Equipamentos

Área	ID	Requisito	Prioridade	Implementado	
Gestão de pedidos de equipamento	RF-95	Visualizar informações dos pedidos de equipamentos	de todos os utilizadores	M	✓
	RF-96		de utilizadores do(s) seu(s) departamento(s)	M	✓
	RF-97		do próprio utilizador	M	✓
	RF-98	Aceitar ou recusar pedido de equipamento		M	✓
	RF-99	Adicionar próprio pedido de equipamento		M	✓
	RF-100	Editar próprio pedido de equipamento		C	X
	RF-101	Cancelar próprio pedido de equipamento		S	X

Tabela 7.9: Requisitos Funcionais Implementados - Módulo de Gestão de Equipamentos - Área de Gestão de Reparação de Equipamentos

Área	ID	Requisito	Prioridade	Implementado	
Gestão de reparação de equipamentos	RF-102	Visualizar lista de pedidos de reparação de equipamentos	por todos os utilizadores	M	X
	RF-103		por utilizadores do(s) seu(s) departamento(s)	M	X
	RF-104		por próprio utilizador	M	X
	RF-105	Criar um novo pedido de reparação de um equipamento		M	X
	RF-106	Alterar, no pedido de reparação, o estado de um equipamento		M	X
	RF-107	Editar um pedido de reparação	de qualquer utilizador	M	X
	RF-108		do próprio utilizador	M	X
	RF-109	Adicionar ficheiro(s) a um pedido de reparação		C	X
	RF-110	Filtrar lista de pedidos de reparação de equipamentos por	Utilizador	S	X
	RF-111		Tipo de equipamento	S	X
	RF-112		Período do pedido	S	X

7.3 Evidências

Nesta secção serão expostas algumas das principais funcionalidades através de capturas de ecrã da plataforma desenvolvida. Todas as mais relevantes encontram-se nesta secção sendo que o resto das funcionalidades principais podem ser analisadas no Apêndice E.

É importante referir que a maioria das funcionalidades que se encontram apresentadas são do ponto de vista de um administrador da plataforma pois é o que tem acesso a mais informação, como explicado na Secção 4.1. As diferenças principais passam pela barra lateral que é diferente nos colaboradores e nos gestores, não existindo a separação da área de administração, e também na informação a que cada um tem acesso. Analisando os ecrãs dos administradores é possível ver o máximo de funcionalidades da plataforma como por exemplo os diversos botões de ações administrativas (desativar utilizador, desativar equipamento ou aceitar pedidos). Nas páginas dos colaboradores e gestores esses botões não estão presentes.

Módulo de Autenticação

Início de Sessão

Um novo utilizador da plataforma é convidado por um administrador e recebe um email com uma palavra passe temporária para que possa realizar o seu primeiro início de sessão na plataforma. Através do ecrã de início de sessão (Figura 7.2) deve inserir o seu email e password temporária e será redirecionado para o ecrã de primeiro início de sessão (Figura 7.3) onde será obrigado a redefinir a sua palavra passe. Após esta primeira redefinição, poderá iniciar sessão na plataforma com a palavra passe definitiva e será redirecionado de imediato para as páginas normais da plataforma.

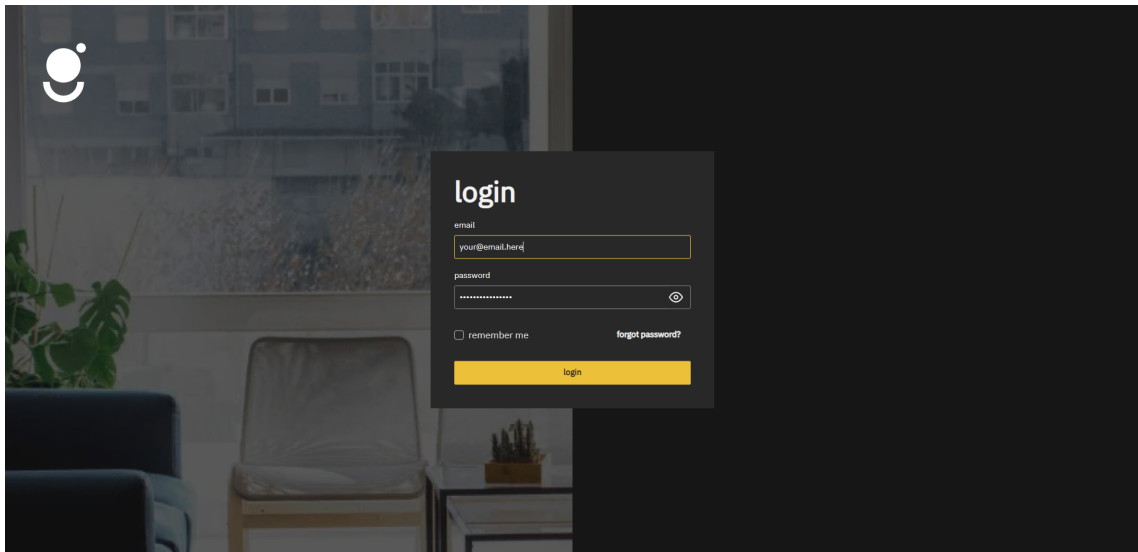


Figura 7.2: Ecrã de Início de Sessão

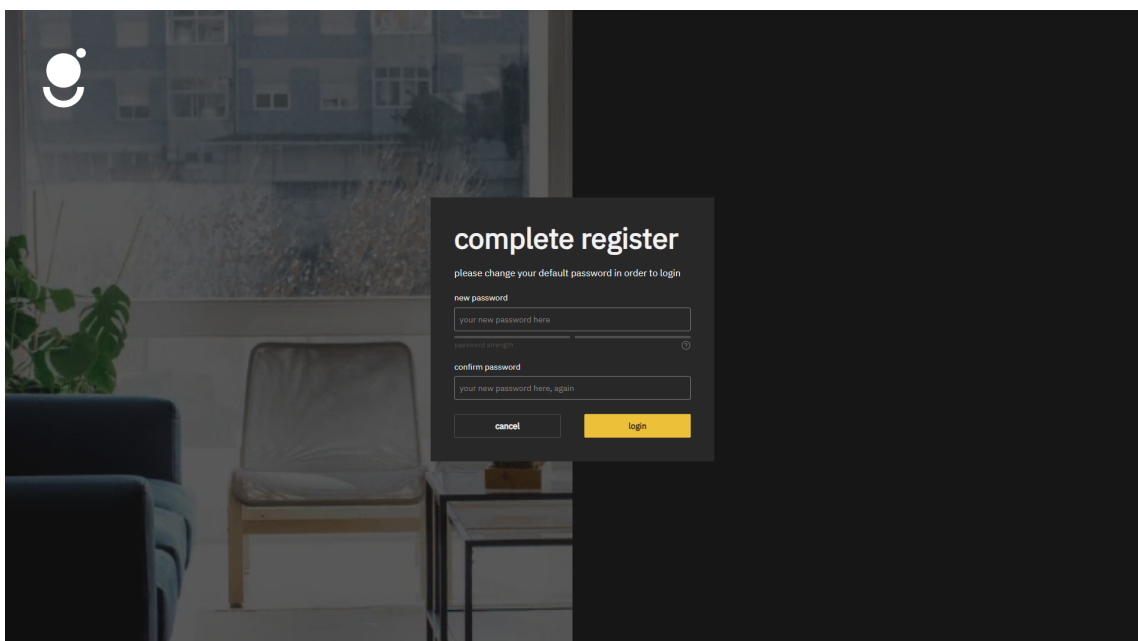


Figura 7.3: Ecrã de Primeiro Início de Sessão

Módulo de Gestão de Colaboradores

Lista de Utilizadores

Os administradores da plataforma têm acesso a listas com todos os utilizadores da plataforma, divididos pelos que têm a sua conta ativa (Figura 7.4) e pelos que tiveram a sua conta terminada (Figura E.3). Também os gestores têm acesso a listas de utilizadores mas, neste caso, apenas podem ver os utilizadores que fazem parte dos seus departamentos e que, naturalmente, têm a sua conta ativa.

The screenshot shows a web interface for managing users. The main content area is titled 'USERS' and has tabs for 'active users' (selected) and 'terminated users'. There are buttons for 'terminate', 'filter by', and 'add user'. Below the tabs is a table with the following data:

name	role	phone number	department	job position	hired date	last login
André Belo belo@empresa.com	user	+351 918520010	Test Department	senior software engineer	09/04/2020	n/a
António Cruz toni@empresa.com	user	+351 917654321	Test Department	senior software engineer	01/01/2022	23/04/2023
Almádo Moura almado@empresa.com	manager	+351 924879651	not allocated users	senior software engineer	01/06/2019	n/a
Francisco Bugalho fbugalho@hotmail.com	administrator	+351 968745136	Test manager	trainee	19/09/2022	n/a
Gonçalo São Marcos goncalo@empresa.com	administrator	+351 926539407	Test manager	intern	29/09/2022	yesterday, 20:50
Joana Boavida joana.boavida@empresa.com	user	+351 910320057	Golden Goal	ui/ux designer	13/06/2021	n/a
João Esperança joao.esperanca@gmail.com	user	+351 916548327	Test manager	intern	26/06/2023	n/a
Marcos Mendes marco.mendes@empresa.com	user	+351 914787002	Internal Tool	ui/ux designer	24/11/2021	n/a
Marcos Calvo marcos@empresa.com	manager	+351 912678964	not allocated users	senior software engineer	01/05/2019	n/a
Margarida Vasconcelos margarida@empresa.com	user	+351 914777885	Human Resources	operations assistant	04/08/2022	n/a

At the bottom of the table, it says 'page 1 of 2' and '10 rows'.

Figura 7.4: Lista de Utilizadores Com Conta Ativa

Vista Detalhada do Perfil de um Utilizador

Todos os utilizadores da plataforma podem aceder à vista detalhada do perfil de um utilizador sendo que um administrador acede à vista detalhada de qualquer utilizador, um gestor apenas dos utilizadores que fazem parte dos seus departamentos e o colaborador apenas pode ver o seu próprio perfil. Apenas o administrador tem acesso aos botões de edição de informações e término de conta do utilizador.

A vista detalhada encontra-se dividida em quatro separadores: informação pessoal (Figura 7.5), contratual (Figura 7.6), pagamento e da conta (Figuras E.4 e E.5). O separador da informação contratual é o que concentra as funcionalidades de adição e remoção de ficheiros (Figura E.7), assim como os históricos de salários e bónus, sendo possível realizar ações sobre os mesmos (Figura E.8).

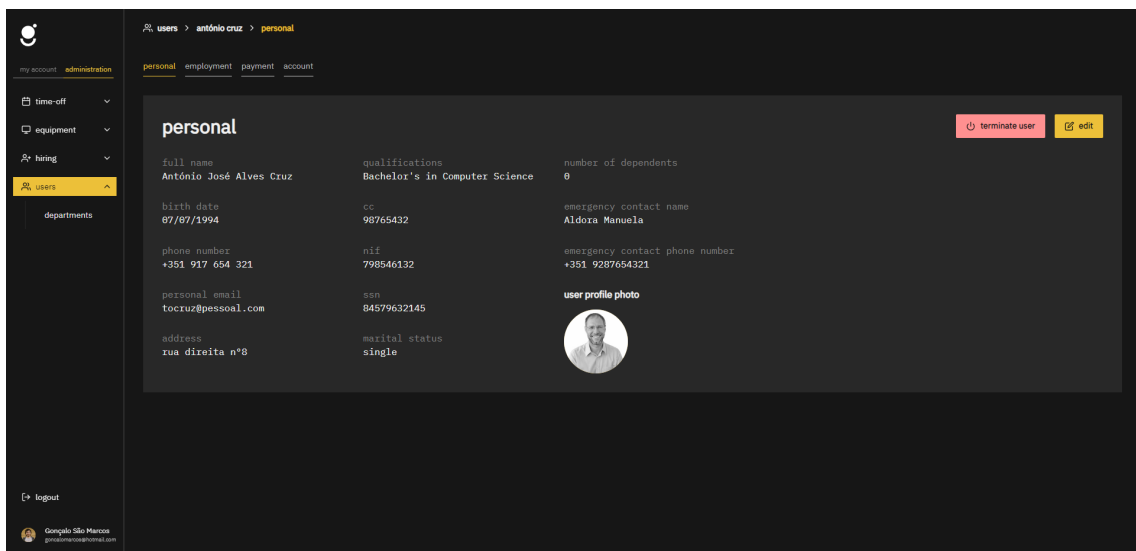


Figura 7.5: Vista Detalhada do Perfil de um Utilizador - Separador da Informação Pessoal

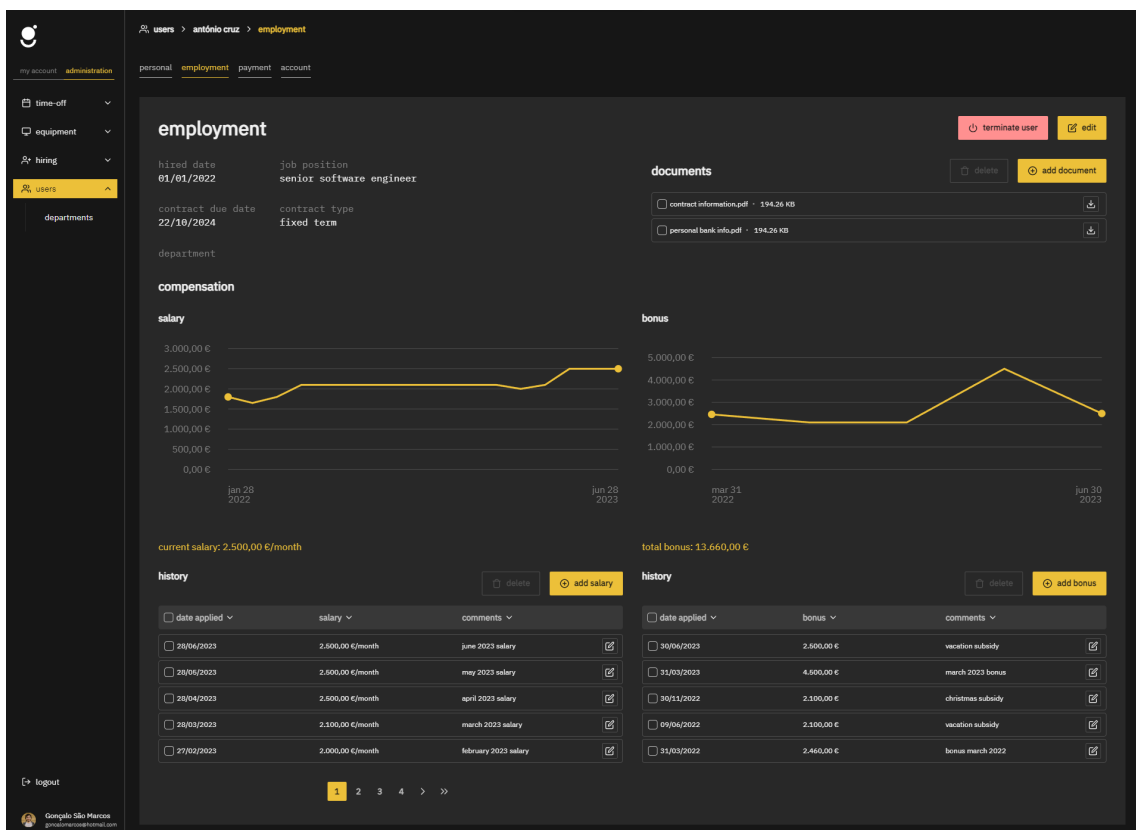


Figura 7.6: Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual

Adicionar Novo Utilizador à Plataforma

Durante a adição de um novo utilizador à plataforma por parte de um administrador é possível definir todos os campos de informação incluindo o seu *Role* na plataforma. No final do preenchimento é possível guardar o utilizador escolhendo se é pretendido enviar de imediato o convite para primeiro início de sessão na plataforma. Na Figura 7.7 é possível ver um dos separadores da adição de um utilizador, os restantes encontram-se no Apêndice E.

The screenshot shows a web application interface for adding a new user. The main content area is titled "add new user" and features four tabs: "personal", "employment", "payment", and "account". The "personal" tab is selected. The form includes the following fields:

- full name: your full name here
- qualifications: your qualifications here
- number of dependents: 0
- birth date: your birth date
- cc: your cc here
- emergency contact name: your emergency contact name here
- country: PT
- phone number: your phone number
- nif: your nif here
- emergency phone number: your emergency phone number
- personal email: your personal email here
- ssn: your ssn here
- address: your address here
- marital status: choose a marital status

At the bottom right of the form, there are three buttons: "cancel", "save", and "save and invite user".

Figura 7.7: Adicionar Novo Utilizador - Separador de Informação Pessoal

Gestão de Departamentos

Também neste módulo é possível realizar a gestão dos departamentos sendo que essa gestão passa pela consulta das informações de cada um através da lista de departamentos (Figura 7.8), pela edição das mesmas informações (Figura E.14) e também pela adição de novos departamentos (Figura E.13).

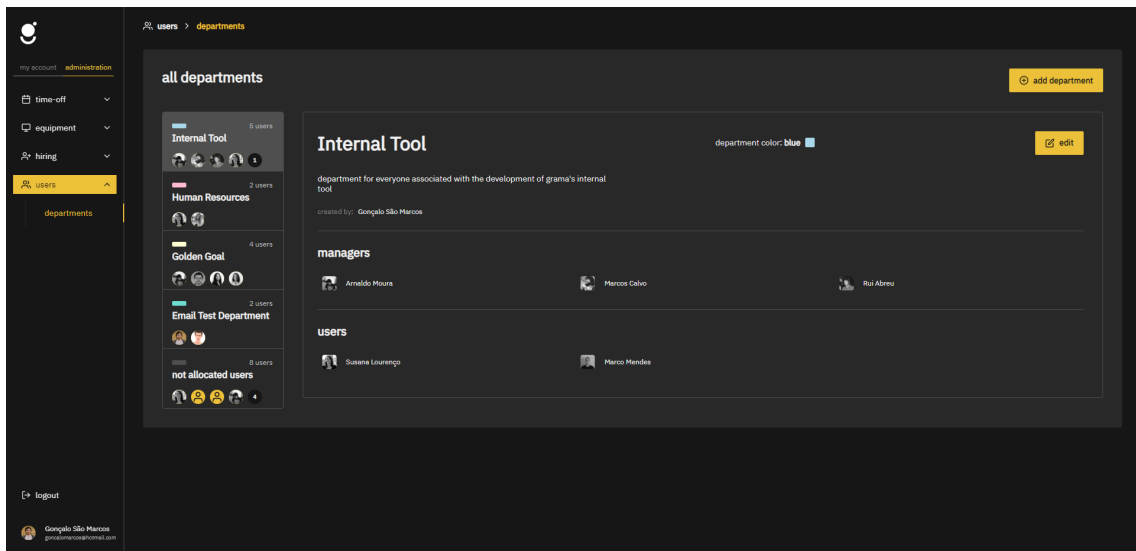


Figura 7.8: Lista de Departamentos

Módulo de Gestão de Equipamentos

Lista de Equipamentos

Os administradores da plataforma têm acesso a listas com todos os equipamentos da plataforma, divididos pelo seu estado: ativo (Figura 7.9, inativo ou obsoleto (Figuras E.15 e E.16). Também os gestores têm acesso a listas de equipamentos mas, neste caso, apenas podem ver os equipamentos alocados aos utilizadores que fazem parte dos seus departamentos.

code	type	model	provider	cost	purchased on	acquired by	allocated to	status
GRAMA-PC-0011	pc	Zenbook 14	Worten	1.500,00 €	05/04/2023	Susana Lourenço	Arnaldo Hours	in use
GRAMA-PC-0017	pc	MX Master 3	Worten	100,00 €	04/04/2023	Gongalo São Marcos	Gongalo São Marcos	in use
GRAMA-PERIPHERAL-0018	peripheral	QC35II	FNAC	300,00 €	04/04/2023	User Gongalo	User Gongalo	in use
GRAMA-OTHER-0019	other	Playstation 5	FNAC	699,99 €	14/10/2022	Susana Lourenço	Grana	in use
GRAMA-SCREEN-0020	screen	4K TV	Worten	1.499,99 €	12/10/2022	Margarida Vasconcelos	Grana	in use
GRAMA-OTHER-0022	other	PlayStation 5	FNAC	599,99 €	19/04/2023	Gongalo São Marcos	João Esperança	in use
GRAMA-SCREEN-0026	screen	QE50Q66B	Worten	499,99 €	25/04/2023	Susana Lourenço	Gongalo São Marcos	in use
GRAMA-PC-0027	pc	MacBook Air M2	Worten	1.146,00 €	26/04/2023	Susana Lourenço	Francisco Bugalho	in use

Figura 7.9: Lista de Equipamentos Ativos

Vista Detalhada de um Equipamento

Todos os utilizadores da plataforma podem aceder à vista detalhada de um equipamento sendo que um administrador acede à vista detalhada de qualquer equipamento, um gestor apenas aos equipamentos que estão alocados aos utilizadores que fazem parte dos seus departamentos e o colaborador apenas pode ver os equipamentos que lhe estão alocados. Apenas o administrador tem acesso aos botões de edição de informações e mudança de estado dos equipamentos.

Um exemplo da vista detalhada de um equipamento pode ser observado da Figura 7.10.

Adicionar Novo Equipamento à Plataforma

É possível os administradores adicionarem novos equipamentos à plataforma. Durante a adição de um novo equipamento podem definir-se todos os campos de informação do mesmo incluindo por quem foi comprado e a quem este é alocado de imediato após a criação.

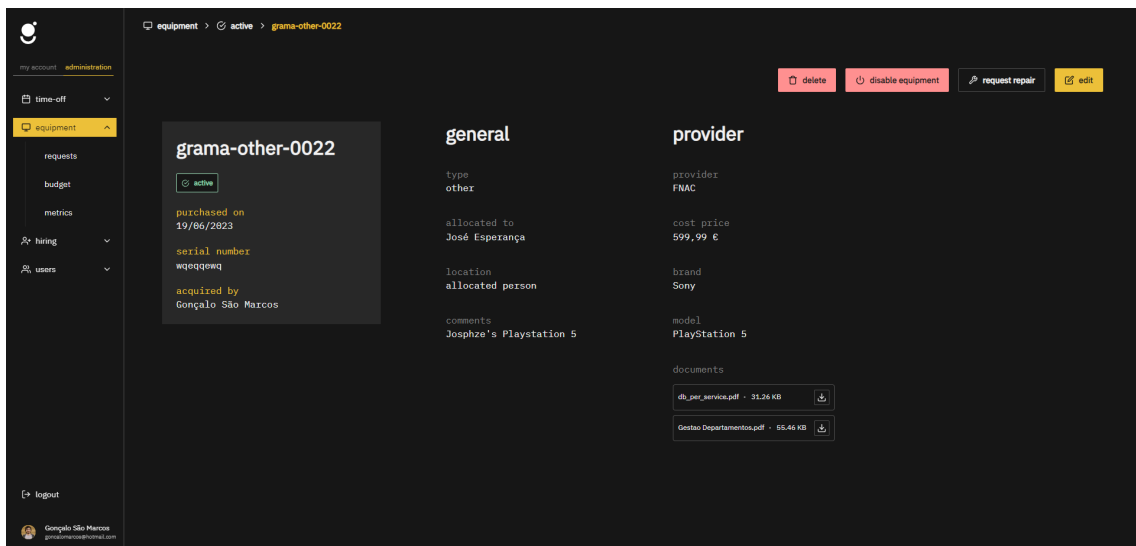


Figura 7.10: Vista Detalhada de um Equipamento

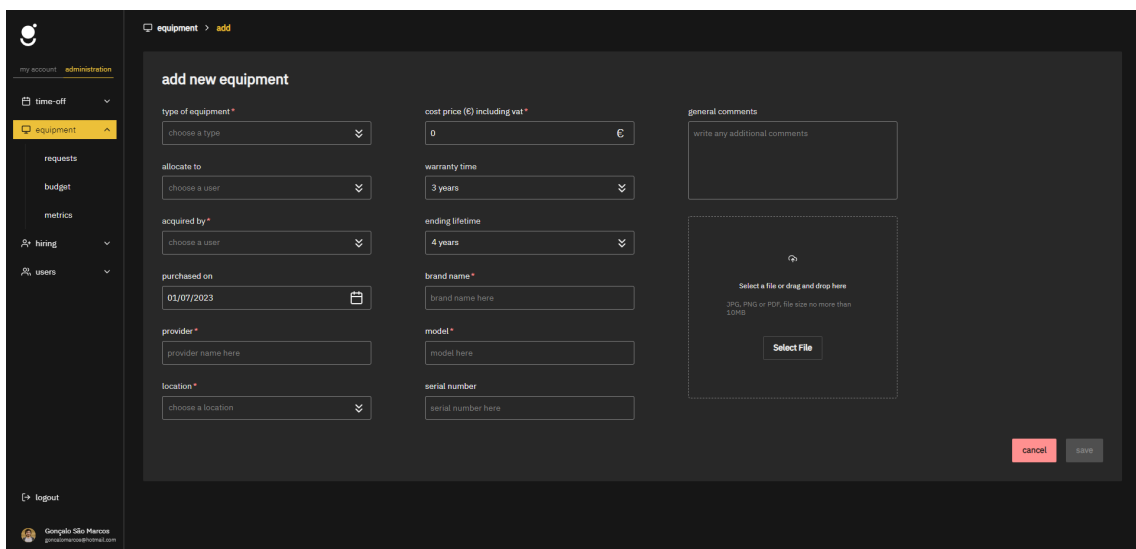


Figura 7.11: Adição de um Novo Equipamento

Pedidos de Equipamentos

Qualquer utilizador da plataforma pode realizar um pedido de equipamento. Este pedido contém informação sobre um equipamento que o utilizador pretende que seja comprado para uso na empresa, através do seu orçamento pessoal para equipamentos.

Os administradores da plataforma têm acesso a listas com todos os pedidos de equipamentos da plataforma, divididos pelo seu estado: por aprovar (Figura 7.12, aprovados ou recusados (Figuras E.20 e E.21). Também os gestores têm acesso a listas de pedidos de equipamentos mas, neste caso, apenas podem ver os pedidos de equipamentos realizados pelos utilizadores que fazem parte dos seus departamentos.

Na Figura 7.13 é possível observar a vista detalhada de um pedido de equipa-

mento que se encontra por aprovar. Nesta vista, com uma conta de administrador, é possível ver os botões com que se aprova ou recusa o pedido em causa.

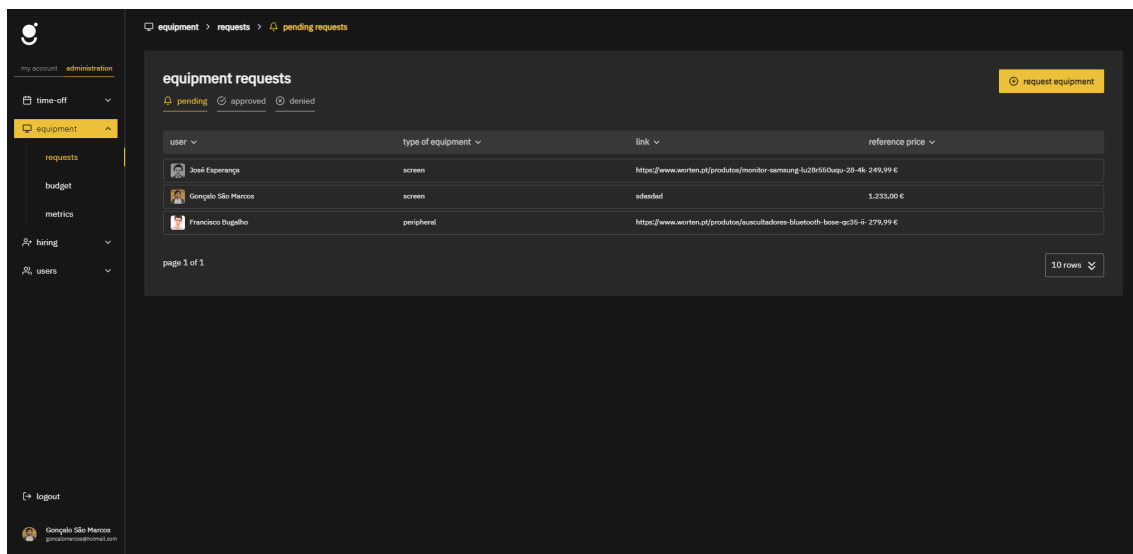


Figura 7.12: Lista de Pedidos de Equipamentos Por Aprovar

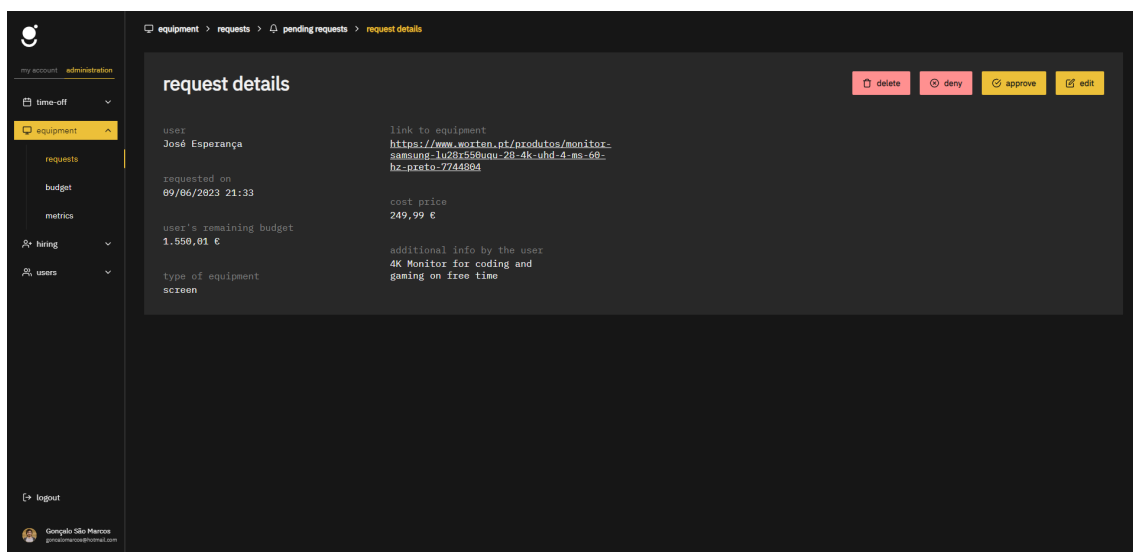


Figura 7.13: Vista Detalhada de um Pedido de Equipamento Por Aprovar

Orçamentos dos Utilizadores

Quando um utilizador é adicionado à plataforma é possível definir o seu orçamento inicial para equipamentos. Os seguintes ecrãs permitem a consulta e gestão dos orçamentos dos utilizadores. Os administradores da plataforma têm acesso total aos orçamentos dos utilizadores. Os gestores têm acesso apenas aos orçamentos dos utilizadores que fazem parte dos seus departamentos, sem capacidade de edição dos mesmos.

Na Figura 7.14 é possível ver a lista dos orçamentos dos utilizadores. Na Figura 7.15 é possível observar a vista detalhada do orçamento de um utilizador.

user	updated on	initial budget	used budget	remaining budget	entry date
Marcos Celso	23/06/2023	1.800,00 €	0,00 €	1.800,00 €	01/06/2019
Sora Vilela	23/06/2023	1.800,00 €	0,00 €	1.800,00 €	07/06/2021
Gonçalo São Marcos	26/06/2023	1.800,00 €	499,99 €	1.300,01 €	29/09/2022
User Gonçalo	23/06/2023	1.800,00 €	0,00 €	1.800,00 €	06/06/2023

Figura 7.14: Lista de Orçamentos dos Utilizadores

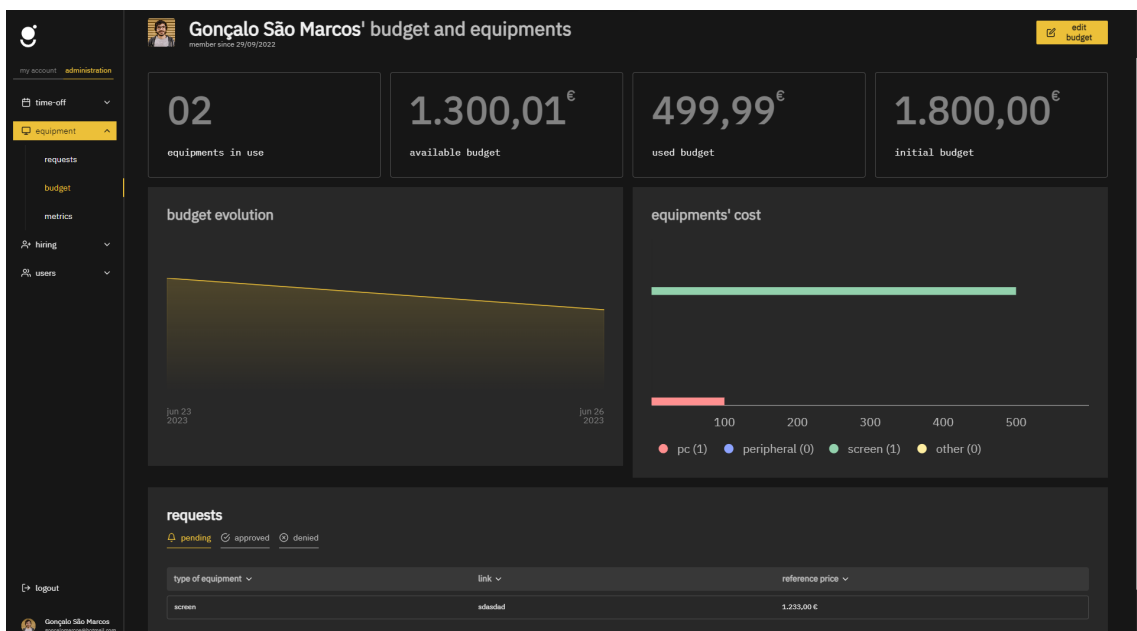


Figura 7.15: Vista Detalhada do Orçamento de um Utilizador

7.4 Reflexão Sobre os Riscos Identificados

No Apêndice A, correspondente à análise de riscos realizada antes do início da implementação da plataforma, foram identificados alguns riscos referentes ao desenvolvimento da Ferramenta Interna da Grama. De seguida, são refletidos os mesmos à luz do que foi sentido ao longo da implementação da mesma.

- **R-1: Falta de experiência nas tecnologias** - Imediatamente antes do período de implementação começar, a empresa forneceu ao aluno o acesso a cursos online na plataforma Udemy^[63] sobre as tecnologias que tinham sido escolhidas para serem utilizadas (Java *Spring* e *React*). Durante o período da implementação, qualquer dúvida técnica que ia surgindo por parte do aluno era questionada junto do orientador da empresa ou de outros programadores da empresa com maior experiência na tecnologia de modo a que esta fosse esclarecida. A curva de aprendizagem inicial foi desafiante mas a escolha da ordem das tarefas ajudou na adaptação. Tal aconteceu para a implementação do *frontend* onde se começou por implementar componentes comuns a toda a plataforma, permitindo assim ao aluno aprender o funcionamento dos componentes de *React* antes de passar para o desenvolvimento das páginas em si. No *backend*, as tarefas iniciais passaram pela criação de *endpoints* mais simples como por exemplo os de criação e leitura de uma certa entidade, permitindo também aprender como funcionavam as diversas funcionalidades da *Spring Framework* ao longo do tempo;
- **R-2: Dimensão do trabalho demasiado ambiciosa** - O atual estágio não foi o primeiro a ser disponibilizado pela empresa pelo que a preparação da proposta teve decerto em conta as experiências passadas com outros estágios no que toca à dimensão de trabalho do mesmo. No entanto, a dimensão total é muito difícil de prever com exatidão e, como tal, o que foi idealizado para ser implementado acabou por não o ser na sua totalidade. De certa forma este risco passou a problema pois quando se concluiu que não seria possível implementar todas as funcionalidades pensadas, houve necessidade de tomar a decisão de quais funcionalidades priorizar de modo a que o resultado final fosse de encontro ao esperado. Foi decidido que não se iria implementar a área de gestão de pedidos de reparação na sua totalidade assim como os *dashboards* onde seriam agregadas algumas informações da plataforma, como mencionado anteriormente na Secção 7.2;
- **R-3: Trabalho conjunto** - Desde o primeiro semestre do estágio foi mantida uma comunicação abundante com o outro estagiário, Francisco Bugalho, de modo a que o planeamento e desenvolvimento da Ferramenta Interna da Grama corresse de forma eficaz e sem problemas derivados de uma possível falta de trabalho conjunto. O trabalho conjunto surgiu apenas em algumas situações como no caso de algumas decisões arquiteturais e de tecnologias assim como na implementação inicial do *frontend* da plataforma onde existiu uma divisão das tarefas pelos dois estagiários, tornando assim o desenvolvimento dessa parte mais célere. Desde o início, o Francisco demonstrou sempre disponibilidade e vontade de conversar sobre os pontos comuns e

também trabalhar nos mesmos pelo que este risco nunca esteve perto de se tornar um problema durante todo o período do estágio.

7.5 Trabalho Futuro

Olhando para o trabalho futuro a realizar na Ferramenta Interna da Grama à luz do trabalho que foi realizado durante o período deste estágio, considera-se de maior importância terminar a implementação de todos os requisitos funcionais definidos, nomeadamente a área de Gestão de Pedidos de Reparação de Equipamentos e também a área de *Dashboard*. Outro ponto importante será garantir que a plataforma responde aos requisitos não funcionais identificados, como por exemplo o desempenho da mesma que, para esta primeira versão, ficou aquém do que era expectável como será analisado na Secção 8.2.

Como foi idealizado desde o início, a plataforma foi pensada e criada com vista a ser bastante flexível no que toca à adição de novas funcionalidades, quer seja em termos de atualizações aos módulos já existentes, quer seja na adição de novos módulos com funcionalidades distintas das atuais. Como mencionado inicialmente na Secção 1.2, a empresa tem ideias adicionais de módulos a implementar no futuro da Ferramenta Interna da Grama como os de gestão de formações e de partilha de conhecimentos.

Capítulo 8

Testes

No seguimento da implementação realizada ao longo do segundo semestre deste estágio, foram realizados testes à Plataforma Interna da Grama. Este capítulo apresentará todos os testes realizados, começando na Secção 8.1 onde se apresentam os testes às funcionalidades da plataforma. Na Secção 8.2 é possível encontrar os testes aos requisitos não funcionais identificados na Secção 4.4.

8.1 Testes Funcionais

Ao longo de toda a implementação do *backend*, foram criados testes unitários a cada *endpoint* de forma a identificar potenciais problemas nos mesmos. Naturalmente é impossível testar todos os casos possíveis para a funcionalidade disponibilizada num certo *endpoint*, portanto procurou-se testar de forma representativa com dados prováveis no uso normal da plataforma mas tendo sempre em conta os casos mais complexos (*edge cases*, por exemplo). Foi utilizada a ferramenta JUNIT^[37] para a realização dos mesmos.

No código abaixo é possível visualizar um exemplo de teste unitário realizado. No caso apresentado é feito um teste ao *endpoint* de criação de um novo utilizador da plataforma.

Teste do *endpoint* POST user

```
@Test
@DisplayName("Create_User")
public void testCreateUser(@Autowired
                           WebTestClient webTestClient) {
    UserRequest userRequest=/* Create a user request
with mock information fields */

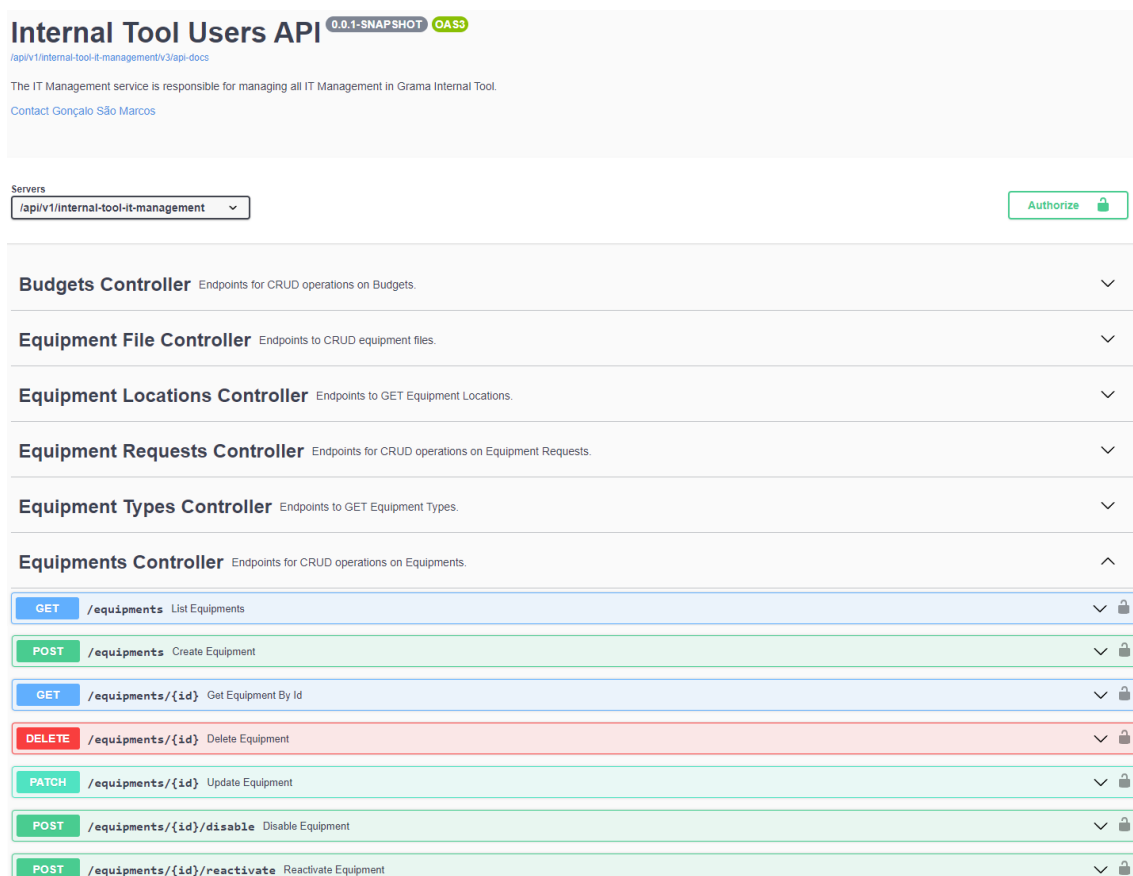
    UserResponse response=webTestClient.post()
        .uri("/users/")
        .header("Authorization", getToken())
        .bodyValue(userRequest)
```

```
.exchange ()
.expectStatus ().isCreated ()
.expectBody (/* Request Type */)
.returnResult ()
.getResponseBody ();

assertNotNull (response );
assertEquals (response .getCompanyEmail (),
              userRequest .getCompanyEmail ());
/* More asserts to check if all the
information was saved successfully */
}
```

Foi também utilizado durante o desenvolvimento e está incluído no código do *backend* de cada microsserviço um *endpoint* para realização de testes manuais à plataforma recorrendo à ferramenta *Swagger*^[59].

Na Figura 8.1 é possível ver alguns dos controladores disponíveis e os seus *endpoints* para o caso do microsserviço da gestão de equipamentos.



Internal Tool Users API 0.0.1-SNAPSHOT OAS3
/api/v1/internal-tool-it-management/v3/api-docs

The IT Management service is responsible for managing all IT Management in Grama Internal Tool.
Contact Gonçalo São Marcos

Servers
/api/v1/internal-tool-it-management Authorize

- Budgets Controller** Endpoints for CRUD operations on Budgets.
- Equipment File Controller** Endpoints to CRUD equipment files.
- Equipment Locations Controller** Endpoints to GET Equipment Locations.
- Equipment Requests Controller** Endpoints for CRUD operations on Equipment Requests.
- Equipment Types Controller** Endpoints to GET Equipment Types.
- Equipments Controller** Endpoints for CRUD operations on Equipments.
 - GET** /equipments List Equipments
 - POST** /equipments Create Equipment
 - GET** /equipments/{id} Get Equipment By Id
 - DELETE** /equipments/{id} Delete Equipment
 - PATCH** /equipments/{id} Update Equipment
 - POST** /equipments/{id}/disable Disable Equipment
 - POST** /equipments/{id}/reactivate Reactivate Equipment

Figura 8.1: Exemplo do *Swagger* no microsserviço de Gestão de Equipamentos

No fim do período de implementação das funcionalidades, foram levados a cabo testes de integração de modo a confirmar o correto funcionamento da plataforma durante a sua utilização normal, confirmando assim a articulação entre o *backend*

e a página *web* do *frontend*. Desta forma, foram revisitadas as *User Stories* de modo a confirmar que era possível realizar todas as tarefas descritas pelas mesmas.

Na impossibilidade de incluir todos estes testes no corpo do documentos, apresenta-se na Tabela 8.1 um exemplo do planeamento de testes realizado para a área específica de autenticação. As restantes tabelas com os testes às outras áreas encontram-se no apêndice F.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
1	1	Utilizador inicia sessão	- Aceder ao site da plataforma - Inserir email e palavra-passe válidos	O utilizador entra na plataforma	O utilizador entrou na plataforma	Passou
2	2	Utilizador pretende recuperar a sua palavra-passe	- Aceder ao site da plataforma - Selecionar o link "forgot password" - Colocar o email da conta que pretende recuperar - Receber um email com um código de recuperação - Colocar o código, a nova palavra-passe e confirmar a palavra-passe	A palavra-passe é alterada com sucesso e o utilizador entra na plataforma	A palavra-passe foi alterada com sucesso e o utilizador entrou na plataforma	Passou
3	3	Utilizador termina sessão	- Clicar no botão "logout"	O utilizador é redirecionado para a página de início de sessão	O utilizador foi redirecionado para a página de início de sessão	Passou
4	4	Utilizador efetua o seu primeiro início de sessão	- Receber o convite e palavra passe temporária - Aceder ao site da plataforma pela primeira vez - Inserir email e palavra-passe temporária - Redefinir palavra-passe	O utilizador entra na plataforma	O utilizador entrou na plataforma	Passou

Tabela 8.1: Testes Realizados - Módulo de Autenticação

É de notar que algumas das US foram agrupadas nas linhas das tabelas dado que certas funcionalidades são muito semelhantes, apenas mudando o *role* do utilizador que interage com a plataforma. Lembra-se que um Colaborador tem acesso apenas à sua informação, e um Gestor tem acesso a alguma informação extra, nomeadamente dos utilizadores pertencentes ao(s) seu(s) departamento(s).

8.2 Testes aos Requisitos Não Funcionais

Na presente secção o sistema será testado à luz dos requisitos não funcionais que foram levantados durante o primeiro semestre do estágio e que se encontram descritos na Secção 4.4.

Disponibilidade

Como referido na Secção 4.4 onde se começou por apresentar este requisito não funcional, grande parte da responsabilidade deste requisito não é controlável pois está dependente da disponibilidade dos serviços *cloud* da Amazon. No entanto, foi referido que existem pontos possíveis de testar e que podem influenciar a experiência do utilizador final. Um desses casos é a atualização de um micro-serviço da plataforma que leva a que a mesma fique indisponível durante alguns momentos.

É importante referir que a indisponibilidade da plataforma para receber pedidos dos utilizadores depende de qual microserviço é desligado. Se o microserviço

de gestão de equipamentos for desligado, apenas as informações relativas aos mesmos ficarão indisponíveis, sendo que o resto da plataforma continua a funcionar, sendo possível, por exemplo, visualizar dados pessoais dos utilizadores ou obter informação de outros microsserviços que não dependam do que está a ser atualizado. Por outro lado, se for desligada a *API Gateway* para uma atualização, todos os pedidos deixam de funcionar durante o período de atualização.

Com a plataforma *deployed* na AWS segundo a arquitetura apresentada na Secção 6.4 é possível testar quanto tempo a plataforma fica indisponível quando se realiza o *deploy* de uma nova versão de um microsserviço (ou mais microsserviços, caso a atualização seja em simultâneo). Foram realizados testes para confirmar a resposta do sistema durante atualizações. Foram realizadas dez medições sendo que cada tempo é contado a partir do momento em que a plataforma deixa de responder a certos pedidos do utilizador, ou seja, quando um dos microsserviços é desligado, até ao momento em que é possível realizar pedidos ao microsserviço que foi atualizado.

O processo de atualização de um microsserviço passa por compilar o projeto de Java *Spring* para um ficheiro *JAR* que posteriormente é carregado para a máquina EC2 da AWS. O tempo de carregamento desse ficheiro nunca demorou mais de 30 segundos durante os testes realizados. Este tempo depende sempre da conexão à internet pois tem cerca de 70MB de tamanho (para o microsserviço de Gestão de Utilizadores). O carregamento do mesmo pode ser realizado antes de desligar o microsserviço de forma a reduzir o tempo de indisponibilidade.

A média das 10 medições foi de 1 minuto e 29 segundos, confirmando assim que durante uma atualização o sistema fica total ou parcialmente indisponível durante menos de 3 minutos. No apêndice G, na Tabela G.1 pode-se observar a tabela de registo de tempos onde se separam os tempos de inicialização da máquina EC2 da AWS (após *reboot*) e os tempos de inicialização do microsserviço em si.

Segurança

A segurança na Plataforma Interna da Grama foi considerada como sendo de prioridade alta, como explicado na Secção 4.4, devido aos dados com que a mesma opera, desde dados pessoais dos colaboradores até informações da empresa e o seu inventário de equipamentos. Na decisão da arquitetura do sistema, exposta na Secção 6.4, foi desde logo tida em conta a questão da segurança e optou-se por usar o serviço Cognito da AWS para gerir a autenticação dos utilizadores na plataforma. Através do uso deste serviço é possível a emissão de *tokens* e também realizar a sua validação junto do Cognito em cada pedido recebido pelo *backend* de modo a verificar a validade do *token* e também para obter informação sobre, por exemplo, o *role* do utilizador na plataforma.

O *role* do utilizador toma um lugar importante na segurança dos *endpoints* expostos pelo *backend*. Após o *token* ser validado, é verificado o *role* do utilizador e cada *endpoint* tem regras de quais *roles* podem aceder ao mesmo. O *endpoint* de criação de um utilizador, por exemplo, apenas pode ser acedido por utilizadores com o

role de administrador.

Foram definidos dois cenários de segurança: um que considera utilizadores não autenticados e outro que considera utilizadores não autorizados. Partindo pela análise do cenário de segurança considerando utilizadores não autenticados, este não tem permissão para aceder a qualquer informação do sistema nem a qualquer página da plataforma. Este utilizador não autenticado apenas pode ver as páginas de autenticação como a página de início de sessão.

Na Tabela 8.2 podem-se observar os testes que foram realizados de modo a confirmar que um utilizador não autenticado não terá acesso a qualquer informação da plataforma. Apenas com um *token* JWT válido é que é possível obter uma resposta com estado "OK" do *backend*. Sem *token*, ou com um inválido ou expirado, o resultado é uma resposta de erro com o estado "*forbidden*".

ID	Cenário	Resultado Esperado	Resultado Obtido	Conclusões
1	Pedido sem <i>token</i> JWT	Resposta com erro 403	Resposta com erro 403	Passou
2	Pedido com <i>token</i> JWT inválido	Resposta com erro 403	Resposta com erro 403	Passou
3	Pedido com <i>token</i> JWT expirado	Resposta com erro 403	Resposta com erro 403	Passou
4	Pedido com <i>token</i> JWT válido	Resposta com código 200	Resposta com código 200	Passou
5	Utilizador sem registo tenta aceder a qualquer página da plataforma	Redirecionado para página de <i>login</i>	Redirecionado para página de <i>login</i>	Passou

Tabela 8.2: Testes de Segurança - Utilizador Não Autenticado

O segundo cenário diz respeito a utilizadores sem autorização, ou seja, utilizadores que se encontram autenticados na plataforma mas que tentam aceder a informação à qual não têm permissão para aceder. Como mencionado anteriormente, o *role* do utilizador é o que dita qual informação este pode aceder e, portanto, é o primeiro ponto de validação. Abaixo pode ver-se um exemplo de dois *endpoints* do microsserviço de utilizadores em que se observa que cada um realiza uma verificação do *role* do utilizador que realiza o pedido. Apenas os *roles* enumerados na anotação *@PreAuthorize* é que podem realizar pedidos ao *endpoint*, sendo que os restantes são rejeitados. A *role* é obtida por confirmação junto do serviço Cognito da AWS, após a validação do *token* JWT. Todos os *endpoints* de todos os microsserviços da plataforma têm esta anotação e as *roles* que constam na mesma vão de encontro ao que cada uma pode aceder.

Exemplos de *endpoints* do microsserviço de utilizadores

```

    @Operation(summary = "Create_User" ,
description="Create_a_new_user . ")
    @PreAuthorize("hasRole('ROLE_ADMINISTRATOR')")
    @PostMapping
    ResponseEntity<UserResponse>
    createUser(@RequestBody @Validated UserRequest user);

    @Operation(summary="Get_User_By_Id" ,
description="Get_user_by_id . ")
    @PreAuthorize("hasRole('ROLE_ADMINISTRATOR')
|| hasRole('ROLE_MANAGER') || hasRole('ROLE_USER')")
    @GetMapping("/{id}")
    ResponseEntity<UserResponse>
    getUserById(@PathVariable UUID id);
}

```

No entanto alguns *endpoints* têm obrigatoriamente de aceitar pedidos de utilizadores com qualquer *role* entre os três definidos na plataforma mas apenas podem devolver uma resposta com o que foi requisitado se o utilizador em causa tiver permissões para tal. Um exemplo é o *endpoint* mostrado anteriormente de obter um utilizador pelo seu identificador, *getUserById*). Neste *endpoint* qualquer tipo de *role* pode realizar um pedido mas os colaboradores apenas podem ver a sua própria informação, os gestores apenas podem ver as informações dos utilizadores que pertencem aos seus departamentos e os administradores podem ver a informação de qualquer utilizador. Desta forma, é posteriormente realizada outra verificação onde, dependendo do *role*, se devolve a informação pedida ou se devolve uma resposta com erro, evitando assim mostrar informação sensível a quem não tem permissão para a ver.

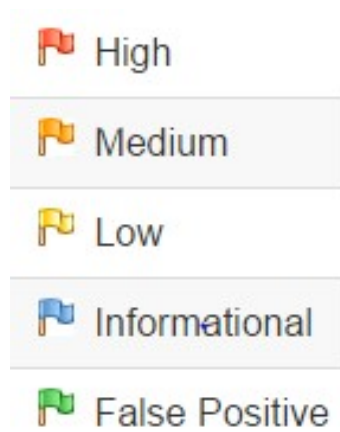
Na Tabela 8.3 é possível ver alguns testes simples que foram realizados manualmente, quer localmente, quer na versão *deployed* na AWS, para garantir que as verificações realizadas nos pedidos ao *backend* surtiam efeito no que o utilizador da plataforma conseguia ou não aceder. No *frontend* da plataforma foi tido em conta a possibilidade de um utilizador editar o URL de forma a tentar aceder a outras páginas, por exemplo de administração (teste com ID 7 da Tabela 8.3). Nesse caso o mesmo é redirecionado para uma página de "*unauthorized*". No teste com ID 8 da tabela alterou-se o URL, tentando aceder à vista detalhada de outro utilizador existente na plataforma. O resultado obtido foi o que se encontra apresentado na tabela. Os testes unitários no *backend* também procuram testar alguns destes casos de não autorização de acesso a certas informações. Os restantes testes de segurança de utilizador não autorizado encontram-se no apêndice G.

Com a plataforma *deployed* na AWS, conforme a arquitetura apresentada na Secção 6.4, foi também realizado um teste automatizado recorrendo à ferramenta OWASP ZAP (Zed Attack Proxy)^[44] que procura testar os pontos mais comuns de segurança das plataformas *web*. Os resultados obtidos são apresentados de

ID	Cenário	Resultado Esperado	Resultado Obtido	Conclusões
6	Colaborador ou Gestor tenta abrir uma página reservada a administradores	Redirecionado para página "unauthorized"	Redirecionado para página "unauthorized"	Passou
7	Colaborador tenta abrir página de perfil pessoal	Página é mostrada corretamente	Página é mostrada corretamente	Passou
8	Colaborador tenta aceder à informação de outro utilizador	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
9	Gestor tenta aceder à informação de um utilizador que não faz parte dos seus departamentos	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
10	Administrador tenta aceder à informação de qualquer utilizador	Página é mostrada corretamente	Página é mostrada corretamente	Passou

Tabela 8.3: Testes de Segurança - Utilizador Não Autorizado

modo a poderem ser analisados e potencialmente corrigidos, aumentando assim a segurança da plataforma contra o tipo de ataques testados. A sinalética de alertas segue os símbolos apresentados na Figura 8.2 sendo que, de cima para baixo na figura, o nível dos alertas pode ser: *alto*, *médio*, *baixo*, *informativo* ou *falso positivo*.

Figura 8.2: Sinalética dos Alertas da Ferramenta OWASP ZAP^{[44][43]}

Ao realizar um "Automated Scan" com o URL da plataforma *deployed*, a ferramenta OWASP ZAP retornou **seis** vulnerabilidades identificadas. Nenhum dos alertas identificados é de risco *alto* sendo que dois são de risco *médio*, dois de risco *baixo* e dois *informativos*. Na Figura 8.3 é possível observar o resultado obtido com a ferramenta OWASP ZAP.

De seguida analisam-se os alertas um a um de modo a perceber melhor o risco que cada um deles representa e quais seriam as possíveis soluções para eliminar

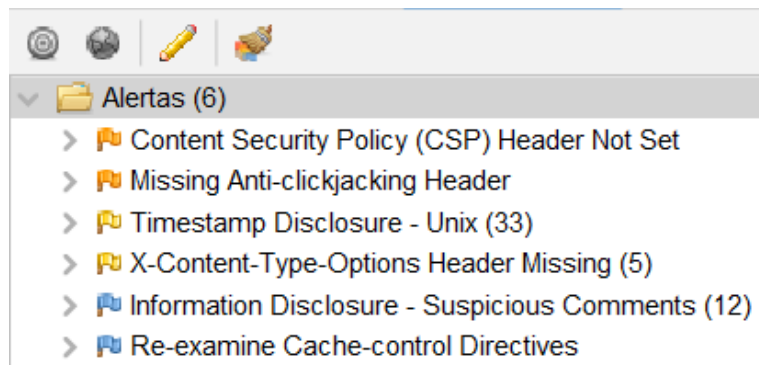


Figura 8.3: Alertas da Ferramenta OWASP ZAP

estes alertas e riscos. É de notar que durante a implementação da Ferramenta Interna da Grama não existiu tempo de abordar a correção destes alertas. Desta forma, os alertas devem ser futuramente considerados e solucionados de modo a tornar a plataforma mais segura e resistente ao tipo de ataques a que se encontra vulnerável de momento, identificados por estes alertas da ferramenta OWASP ZAP.

Content Security Policy CSP Header Not Set

O *Content Security Policy* (CSP) é uma camada de segurança que ajuda a detetar e mitigar alguns ataques como por exemplo os ataques de *Cross-Site Scripting* (XSS)^[41]. Este tipo de ataques podem ser usados de forma maliciosa para obter informação privada ou até mesmo distribuir programas maliciosos.

Uma possível solução para evitar a presença deste alerta e mitigar os riscos seria a correta implementação no servidor *web* de forma a retornar o *header* HTTP *Content-Security-Policy* corretamente configurado^[34] para que a que apenas o conteúdo permitido fosse carregado a partir das fontes permitidas.

Missing Anti-clickjacking Header

O conceito de *clickjacking* passa por utilizar truques como várias camadas transparentes ou opacas para enganar o utilizador de forma a este clicar em botões ou hiperligações de outras páginas sem se aperceber pois julga estar a clicar na página correta^[25].

Através de uma pesquisa é possível encontrar algumas soluções para prevenir este tipo de ataques sendo que uma delas passa por corretamente configurar o *header* HTTP *Content-Security-Policy* mencionado previamente. Uma das diretivas que este *header* permite configurar é a de *frame ancestors*^[42] que indicam ao navegador quais *parents* de HTTP podem incluir outras páginas dentro de si. Os *headers* HTTP *X-Frame-Options* são também uma possível solução com maior suporte em navegadores mais antigos.

Timestamp Disclosure - Unix

Este tipo de alerta, de nível baixo, indica que existem *timestamps* que são expostos pelo sistema. À primeira vista pode não parecer uma vulnerabilidade mas caso esses *timestamps* sejam utilizados pelo sistema para algum tipo de criptografia ou *hashing*, um atacante poderia utilizar os mesmos para mais rapidamente reproduzir o resultado do *hashing*, comprometendo assim a informação do sistema^[52].

Realizando uma pesquisa sobre o alerta em causa, conclui-se que cada alerta deste tipo deve ser manualmente verificado para garantir que o *timestamp* exposto pelo sistema não causa problemas de segurança, ou seja, não pode ser utilizado para obter informações sensíveis. Garantindo que se tratam de *timestamps* que não são utilizados para operações importantes dentro do sistema, pode-se descartar este alerta como sendo falso-positivo.

X-Content-Type-Options Header Missing

O alerta de nível baixo indica a falta de um *header* que pode causar problemas de segurança para a plataforma *web*. Para solucionar este risco é necessário configurar o *header X-Content-Type-Options* para "*nosniff*"^[26]. Esta configuração é recomendada para evitar *MIME-sniffing* que consiste no navegador interpretar o tipo conteúdo que recebe, podendo levar a que este seja interpretado como sendo diferente daquele que vem descrito no *Content-Type header*.

Information Disclosure - Suspicious Comments

Este alerta informativo indica que são expostos publicamente comentários no código que podem comprometer a segurança da plataforma. Caso estes comentários tenham informação sensível, um atacante pode usar a mesma para obter mais informação sobre como comprometer a plataforma.

Após uma rápida análise dos comentários suspeitos identificados pela ferramenta OWASP ZAP, foi possível verificar que estes se tratam de comentários em ficheiros de licenças do *React*, *React-Router* e *AWS Amplify*, pelo que não são informação sensível que possa comprometer a segurança.

Re-examine Cache-control Directives

Este alerta informativo diz respeito à não definição do *header* de controlo de *cache*. Para algum tipo de informação estática, a falta deste *header* pode ser intencional, no entanto, para informações sensíveis este *header* deve ser corretamente configurado para evitar que o navegador guarde essa mesma informação em cache e que esta possa ser posteriormente acedida e utilizada em ataques maliciosos^[24].

Uma forma de solucionar este alerta seria definir o *header* de *cache-control* como "*no-cache, no-store, must-revalidate*"^{textsuperscript[24]} nos pedidos que contenham informação sensível de modo a garantir que esta não é guardada pelo navegador.

Desempenho

O requisito não funcional do desempenho foi definido na Secção 4.4 como sendo de prioridade média pelas razões previamente expostas nessa mesma secção. Foi definido um cenário para este requisito onde se estabeleceu o limite máximo de carregamento das páginas *web* como 2.5 segundos. Desta forma, é possível testar se as diversas páginas da plataforma cumprem este limite de desempenho considerado.

Para a realização destes testes considerou-se a plataforma *deployed* em ambiente *cloud* conforme a arquitetura apresentada na Secção 6.4. Testes realizados em ambiente local não seriam representativos do comportamento real da plataforma visto que a capacidade de processamento da máquina local, em comparação com as máquinas EC2 da AWS utilizadas, iria afetar o desempenho dos testes. Os testes de desempenho foram realizados sem qualquer tráfego adicional (*load*) no sistema pelo que se tratam do melhor caso possível para a realização destes testes. Adicionalmente deveriam ser realizados testes com maior *load* no sistema de modo a compreender como este se comportaria numa situação de maior tráfego e de mais pedidos. Tal não foi possível realizar durante o período de testes à plataforma.

Para medição dos tempos de carregamento das páginas *web* da plataforma foram utilizadas as Ferramentas de Desenvolvimento do navegador Google Chrome que registam o tempo de carregamento total da página. Foi considerado o tempo "*finish*" destas ferramentas pois é o que representa o carregamento total da página com todos os pedidos realizados a receberem a respetiva resposta. As medições foram realizadas tendo efetuado início de sessão com uma conta de administrador pois é o tipo de utilizador que obtém mais informação e, portanto, será o que irá experienciar tempos de carregamento maiores.

Foram testadas algumas das páginas principais da plataforma, como se pode observar na Tabela 8.4 que se encontra dividida por módulos e páginas. A coluna da média de tempo diz respeito à média das 20 medições para cada página.

Como se pode observar, apenas a página de *login* da aplicação obteve um tempo médio de carregamento total abaixo dos 2.5 segundos. As restantes páginas todas obtiveram resultados em média cerca de 24% acima do esperado. Através das mesmas ferramentas de desenvolvimento do navegador, foi possível observar as principais razões pelo tempo mais demorado no carregamento das páginas. Tal deve-se à realização de pedidos ao *backend* que possivelmente são desnecessários para a informação presente no ecrã. Ao realizar estes pedidos adicionais, é criado um *overhead* de pedidos por mais informação do que a estritamente necessária, atrasando o carregamento da página. Tal comportamento não se deve à escolha da arquitetura ou tecnologias mas sim à otimização do código do *frontend* onde são realizados os pedidos.

Como mencionado anteriormente, este requisito foi classificado como sendo de prioridade média, no entanto, considera-se que futuramente se deve visitar a questão do desempenho da plataforma de forma a que a experiência dos utilizadores da mesma não seja afetada negativamente devido aos tempos de carre-

gamento das páginas. É, portanto, importante refinar futuramente a lógica de *frontend* da plataforma que realiza os pedidos ao *backend* para que os tempos de carregamento possam ser inferiores aos 2.5 segundos.

Módulo	Página	Média (em segundos)	Conclusões
Autenticação	Login	0.663	Passou
Gestão de Colaboradores	Lista de Utilizadores Ativos	3.096	Falhou
	Vista Detalhada de um Utilizador	2.928	Falhou
	Adicionar Novo Utilizador	3.384	Falhou
	Lista de Departamentos	2.889	Falhou
	Adicionar Departamento	3.263	Falhou
Gestão de Equipamentos	Lista de Equipamentos Ativos	2.882	Falhou
	Vista Detalhada de um Equipamento	3.413	Falhou
	Lista de Pedidos de Equipamentos Por Aprovar	2.841	Falhou
	Vista Detalhada de um Pedido de Equipamento	3.112	Falhou
	Lista de Orçamentos	2.755	Falhou
	Vista Detalhada do Orçamento de um Utilizador	3.525	Falhou

Tabela 8.4: Testes ao Desempenho das Páginas Web

Capítulo 9

Conclusão

O desenvolvimento do presente documento, no âmbito da realização do estágio curricular, proporcionou ao estagiário um contacto mais próximo com a realidade de uma empresa de desenvolvimento de *software* assim como a oportunidade de participação no processo de planeamento e pesquisa que está envolvido como fase inicial de um projeto de engenharia de *software* e também no processo de desenvolvimento de um projeto de *software* com uma escala considerável.

Foi possível, para o estagiário, aprender com profissionais na área da engenharia de *software* e aplicar os conhecimentos que foram sendo aprofundados ao longo do Mestrado em Engenharia Informática no ramo da Engenharia de *Software*. O período total do estágio foi dividido em dois momentos: uma primeira fase de preparação para o desenvolvimento e a posterior implementação da Ferramenta Interna da Grama. Cada fase correspondeu a um dos semestres do ano letivo.

Inicialmente foi realizada uma análise das plataformas concorrentes de modo a perceber melhor as soluções existentes de ferramentas de gestão de equipamentos e de colaboradores. Desta forma, e realizando uma síntese das funcionalidades e das vantagens e desvantagens de cada uma, foi possível perceber a importância e a necessidade do desenvolvimento de uma plataforma própria para a empresa Grama que respondesse às necessidades da mesma.

Após o estudo das plataformas concorrentes, foi realizada uma análise dos riscos associados com o estágio de modo a compreender melhor quais os obstáculos e pontos que poderiam influenciar a realização deste estágio curricular e do desenvolvimento da plataforma sugerida.

De seguida, foram exploradas algumas das tecnologias atualmente presentes no mercado de desenvolvimento de *software*, mais especificamente para o desenvolvimento de plataformas web, sintetizando as vantagens e desvantagens de cada uma das tecnologias estudadas. Foram analisadas tecnologias de *backend*, *frontend* e também os fornecedores de serviços *cloud* que atualmente se usam em larga escala. Foi também nesta fase que se começaram a estudar as metodologias de desenvolvimento e se definiu a metodologia a aplicar durante a realização do presente estágio. Foi definida a adoção de uma metodologia baseada em *Scrum*.

Seguidamente, foi realizado um levantamento dos requisitos da plataforma a im-

plementar. O constante *feedback* por parte da empresa e da *Product Owner* assim como a análise de esboços iniciais da ferramenta, permitiu a elaboração de *User Stories* de onde foram posteriormente extraídos os requisitos funcionais da plataforma, existindo também um processo de priorização dos mesmos. Também foram definidos os requisitos não funcionais para a plataforma.

Por fim, foram analisados os tipos de arquitetura de *software* e foi decidido implementar uma arquitetura baseada em microsserviços, utilizando as tecnologias *Java Spring* para *backend*, *React* para *frontend* e AWS como fornecedor de serviços *cloud*. Foi também feito um desenho da arquitetura a implementar que respondesse às necessidades da plataforma. Adicionalmente, foi desenvolvido um diagrama Entidade-Relacionamento (E-R) que reflete a estrutura dos dados e os seus relacionamentos que estarão presentes nas bases de dados do sistema.

Após todo o processo de estudo e planeamento, o aluno passou para a implementação da Ferramenta Interna da Grama tendo por base todo o trabalho elaborado ao longo do primeiro semestre. No início da implementação começou por existir algum trabalho em conjunto com o outro estagiário pois as primeiras tarefas passaram pela implementação de componentes comuns a toda a plataforma e a integração do serviço Cognito da AWS que viria a ser utilizado e acedido por ambos. Após a criação dos componentes comuns, passou-se à implementação dos microsserviços, começando pelo de Gestão de Colaboradores, havendo um equilíbrio entre o que ia sendo desenvolvido no *backend*, em *Java Spring*, e as páginas que iam sendo criadas no *frontend*, em *React*.

Ao longo do desenvolvimento a equipa apercebeu-se que não existiria tempo suficiente para implementar todos os requisitos funcionais que foram inicialmente planeados. Desta forma, o aluno implementou os requisitos que foram considerados de maior prioridade, sempre em concordância com a empresa e a *Product Owner* e tendo em conta a carga de trabalho e o tempo disponível para o término do estágio.

No final do desenvolvimento foram realizados testes aos requisitos funcionais e não funcionais da plataforma para confirmar a correta implementação dos mesmos. Foi também feito *deploy* da plataforma conforme a arquitetura pensada para realizar certos testes num ambiente mais próximo do real.

Concluindo, considera-se necessário uma futura melhoria ao nível do desempenho e responsividade das diversas páginas da plataforma, assim como a conclusão das funcionalidades que não foram implementadas durante o estágio. No entanto, acredita-se que os principais objetivos do estágio foram cumpridos com sucesso, tendo sido implementados todos os requisitos de maior prioridade e a plataforma encontra-se preparada para a futura implementação de novos módulos. O estágio permitiu ao aluno ter um contacto próximo com o desenvolvimento de *software* em mundo real, interagindo com a equipa multi disciplinar e experiente da Grama, tendo sempre o apoio necessário mas também a liberdade que é necessária numa nova experiência desta magnitude. Esta versão inicial da Ferramenta Interna da Grama foi de encontro às expectativas da empresa mesmo com as dificuldades enfrentadas, pelo que o balanço final do estágio é bastante positivo.

Referências

- [1] Omar Al-Debagy and Peter Martinek. A comparative review of microservices and monolithic architectures. In *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 000149–000154. IEEE, 2018.
- [2] Altexsoft. "Pros and Cons of Angular Development Framework", 2022. <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/>. Acedido a 11 de novembro de 2022.
- [3] Altexsoft. "Non-functional requirements", 2022. <https://www.altexsoft.com/blog/non-functional-requirements/>. Acedido a 6 de dezembro de 2022.
- [4] Altexsoft. "The Pros and Cons of Vue.js", 2022. <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>. Acedido a 10 de novembro de 2022.
- [5] Amazon. "Low-Latency Content Delivery Network (CDN)", 2022. <https://aws.amazon.com/cloudfront/>. Acedido a 21 de dezembro de 2022.
- [6] Amazon. "Customer Identity and Access Management", 2022. <https://aws.amazon.com/cognito/>. Acedido a 16 de dezembro de 2022.
- [7] Amazon. "Amazon EC2", 2022. <https://aws.amazon.com/ec2/>. Acedido a 6 de dezembro de 2022.
- [8] Amazon. "Elastic Load Balancing", 2022. <https://aws.amazon.com/elasticloadbalancing/>. Acedido a 16 de dezembro de 2022.
- [9] Amazon. "Fully Managed Relational Database", 2022. <https://aws.amazon.com/rds/>. Acedido a 16 de dezembro de 2022.
- [10] Amazon. "Cloud Object Storage - Amazon S3", 2022. <https://aws.amazon.com/s3/>. Acedido a 16 de dezembro de 2022.
- [11] Amazon. "Cloud Email Service", 2022. <https://aws.amazon.com/ses/>. Acedido a 16 de dezembro de 2022.
- [12] Amazon. "Amazon Compute Service Level Agreement", 2022. <https://aws.amazon.com/compute/sla/>. Acedido a 6 de dezembro de 2022.

- [13] Amazon. *"App Authentication and Web App Authorization - AWS Amplify - AWS"*, 2023. <https://aws.amazon.com/amplify/authentication/>. Acedido a 29 de junho de 2023.
- [14] Amazon. *"Verifying a JSON Web Token - Amazon Cognito"*, 2023. <https://docs.aws.amazon.com/cognito/latest/developerguide/amazon-cognito-user-pools-using-tokens-verifying-a-jwt.html>. Acedido a 20 de junho de 2023.
- [15] Amazon. *"Security in Amazon Cognito - Amazon Cognito"*, 2023. <https://docs.aws.amazon.com/cognito/latest/developerguide/security.html>. Acedido a 20 de junho de 2023.
- [16] Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice*. Addison-Wesley Professional, 2003.
- [17] Simon Brown. *"The C4 model for visualizing software architecture"*, 2022. <https://c4model.com/>. Acedido a 19 de dezembro de 2022.
- [18] Jane Cleland-Huang. Quality requirements and their role in successful products. In *15th IEEE International Requirements Engineering Conference (RE 2007)*, pages 361–361. IEEE Computer Society, 2007.
- [19] Agile Business Consortium. *Chapter 10: MoSCoW Prioritisation*, 2022. <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html>. Acedido a 1 de dezembro de 2022.
- [20] Statistics & Data. *"Most popular backend frameworks 2012-2022"*, 2022. <https://statisticsanddata.org/data/most-popular-backend-frameworks-2012-2022/>. Acedido a 10 de novembro de 2022.
- [21] DEI. *"ONDA - Online Database Architect"*, 2022. <http://onda.dei.uc.pt/v4/>.
- [22] Kelly Dent. *"AWS vs Azure vs Google"*, 2022. <https://www.jeffersonfrank.com/insights/aws-vs-azure-vs-google-cloud-provider-comparison>. Acedido a 18 de novembro de 2022.
- [23] Data Flair. *"Django Advantages and Disadvantage"*, 2022. <https://data-flair.training/blogs/django-advantages-and-disadvantages/>. Acedido a 8 de novembro de 2022.
- [24] OWASP Foundation. *'OWASP ZAP - Re-examine Cache-control Directives'*, 2023. <https://www.zaproxy.org/docs/alerts/10015/>. Acedido a 3 de julho de 2023.
- [25] OWASP Foundation. *'Clickjacking | OWASP Foundation'*, 2023. <https://owasp.org/www-community/attacks/Clickjacking>. Acedido a 3 de julho de 2023.

-
- [26] OWASP Foundation. 'OWASP ZAP - X-Content-Type-Options Header Missing', 2023. <https://www.zaproxy.org/docs/alerts/10021/>. Acedido a 3 de julho de 2023.
- [27] Martin Fowler, Jim Highsmith, et al. The agile manifesto. *Software development*, 9(8):28–35, 2001.
- [28] Walter Gonzáles. "When to use AlpineJS", 2022. <https://lightit.io/blog/when-to-use-alpine-js/>. Acedido a 11 de novembro de 2022.
- [29] Sacha Greif. "The State of JS", 2022. <https://2021.stateofjs.com/en-US>. Acedido a 8 de novembro de 2022.
- [30] A Guide. Project management body of knowledge (pmbok® guide). In *Project Management Institute*, volume 11, pages 7–8, 2001.
- [31] Ankit Gupta. "Why AWS?? ...Advantages and Disadvantage", 2022. <https://medium.com/analytics-vidhya/why-aws-advantages-and-disadvantages-f0e666b869b3>. Acedido a 18 de novembro de 2022.
- [32] Mark Haranas. "Top 5 Cloud Market-Share Leaders: AWS, Microsoft Google", 2022. <https://www.crn.com/news/cloud/top-5-cloud-market-share-leaders-aws-microsoft-google-in-q3-2022>. Acedido a 18 de novembro de 2022.
- [33] Red Hat. "Quarkus vs Spring Boot", 2022. <https://www.redhat.com/en/topics/cloud-native-apps/what-is-quarkus>. Acedido a 8 de novembro de 2022.
- [34] Foundeo Inc. 'Content-Security-Policy Header -> CSP Reference & Examples', 2023. <https://content-security-policy.com/>. Acedido a 3 de julho de 2023.
- [35] inteliPaat. "Top 7 Benefits of AWS", 2022. <https://intellipaate.com/blog/aws-benefits-and-drawbacks/#no9>. Acedido a 18 de novembro de 2022.
- [36] JavaTPoint. "Pros and Cons of ReactJS", 2022. <https://www.javatpoint.com/pros-and-cons-of-react>. Acedido a 8 de novembro de 2022.
- [37] JUNIT. "JUnit 5", 2023. <https://junit.org/junit5/>. Acedido a 30 de junho de 2023.
- [38] Anton Kharenko. *Monolithic vs. Microservices Architecture*, 2022. <https://articles.microservices.com/monolithic-vs-microservices-architecture-5c4848858f59>. Acedido a 10 de novembro de 2022.
- [39] Nimrod Kramer. "Svelte pros and cons, ecosystem overview and top resources", 2022. <https://daily.dev/blog/building-with-svelte-all-you-need-to-know-before-you-start>. Acedido a 8 de novembro de 2022.

- [40] Linear. *"Linear"*, 2023. <https://linear.app/>. Acedido a 26 de junho de 2023.
- [41] Developer Mozilla Org. *'Content Security Policy (CSP) - HTTP | MDN'*, 2023. <https://www.zaproxy.org/getting-started/>. Acedido a 3 de julho de 2023.
- [42] Developer Mozilla Org. *'CSP: frame-ancestors - HTTP | MDN'*, 2023. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors>. Acedido a 3 de julho de 2023.
- [43] OWASP. *'OWASP ZAP - Getting Started'*, 2023. <https://www.zaproxy.org/getting-started/>. Acedido a 3 de julho de 2023.
- [44] OWASP. *'OWASP ZAP'*, 2023. <https://www.zaproxy.org/>. Acedido a 3 de julho de 2023.
- [45] Jeel Patel. *"Vue vs Angular"*, 2022. <https://www.monocubed.com/blog/vue-vs-angular/>. Acedido a 11 de novembro de 2022.
- [46] Max Rehkopf. *"User Stories"*, 2022. <https://www.atlassian.com/agile/project-management/user-stories>. Acedido a 11 de novembro de 2022.
- [47] Mark Richards and Neal Ford. *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly Media, 2020.
- [48] Chris Richardson. *Access token*, 2022. <https://microservices.io/patterns/security/access-token.html>. Acedido a 12 de novembro de 2022.
- [49] Chris Richardson. *Api Gateway pattern*, 2022. <https://microservices.io/patterns/apigateway.html>. Acedido a 12 de novembro de 2022.
- [50] Chris Richardson. *Database per service*, 2022. <https://microservices.io/patterns/data/database-per-service.html>. Acedido a 12 de novembro de 2022.
- [51] Chris Richardson. *Microservice Architecture pattern*, 2022. <https://microservices.io/patterns/microservices.html>. Acedido a 12 de novembro de 2022.
- [52] ScanRepeat. *'Timestamp Disclosure - Unix | ScanRepeat'*, 2023. <https://scanrepeat.com/web-security-knowledge-base/timestamp-disclosure---unix>. Acedido a 3 de julho de 2023.
- [53] Ken Schwaber and Jeff Sutherland. The scrum guide. *Scrum Alliance*, 21(19): 1, 2011.
- [54] Scrum.org. *"What is Scrum? Scrum.org"*, 2023. <https://www.scrum.org/resources/what-is-scrum>. Acedido a 2 de janeiro de 2023.
- [55] Sheetal Sharma, Darothi Sarkar, and Divya Gupta. Agile processes and methodologies: A conceptual study. *International journal on computer science and Engineering*, 4(5):892, 2012.

-
- [56] Utkarsh Sidana. "Advantages and Disadvantages of Angular", 2022. <https://www.edureka.co/blog/advantages-and-disadvantages-of-angular/>. Acedido a 11 de novembro de 2022.
- [57] Spark. "Spark Framework: An expressive web framework for Kotlin and Java", 2022. <https://sparkjava.com/>. Acedido a 8 de novembro de 2022.
- [58] Marian Stoica, Marinela Mircea, and Bogdan Ghilic-Micu. Software development: agile vs. traditional. *Informatica Economica*, 17(4), 2013.
- [59] Swagger. 'API Documentation & Design Tools For Teams | Swagger ', 2023. <https://swagger.io/>. Acedido a 30 de junho de 2023.
- [60] Rollbar Editorial Team. "What is Quarkus?", 2022. <https://rollbar.com/blog/quarkus-vs-spring-boot/>. Acedido a 10 de novembro de 2022.
- [61] TeamGantt. "TeamGantt", 2022. <https://app.teamgantt.com/>.
- [62] Besant Technologies. "What is Express.js", 2022. <https://www.besanttechnologies.com/what-is-expressjs>. Acedido a 10 de novembro de 2022.
- [63] Udemy. 'Online Courses - Learn Anything, On Your Schedule | Udemy', 2023. <https://www.udemy.com/>. Acedido a 30 de junho de 2023.
- [64] Mario Villamizar, Oscar Garcés, Harold Castro, Mauricio Verano, Lorena Salamanca, Rubby Casallas, and Santiago Gil. Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. In *2015 10th Computing Colombian Conference (10CCC)*, pages 583–590. IEEE, 2015.

Apêndices

Apêndice A

Análise de Riscos

Antes de começar a implementação do projeto, é importante abordar os riscos que podem afetar o seu desenvolvimento. Analisar um risco passa por avaliar a incerteza de algo que pode acontecer durante a realização do projeto e o seu impacto no mesmo. Deste modo, é importante o trabalho de análise de riscos antes do desenvolvimento para aplicar estratégias que possam reduzir ou, idealmente, eliminar por completo o impacto dos riscos e fazer com que o projeto não seja afetado.

Cada risco será caracterizado através dos seguintes campos: título, descrição, impacto, probabilidade de ocorrência, espaço de tempo, consequências e plano de mitigação. A escolha destes campos e dos níveis pelos quais serão divididos os mesmos foram escolhidos tendo em conta o contexto do problema, da equipa e do espaço temporal para a realização do projeto.

O impacto pode ser baixo, médio, alto ou crítico. Um impacto baixo significa que o projeto praticamente não será afetado caso o risco se torne realidade, enquanto que um impacto crítico impossibilita a sua realização por completo. Foram escolhidos quatro níveis de impacto, um número par de níveis, para evitar a tendência da classificação de cada risco assumir mais vezes o valor central, desta forma, há uma obrigatoriedade de diferenciar de forma mais clara o impacto de cada risco.

A probabilidade pode ser baixa, média, alta ou extrema, sendo que cada nível corresponde a um intervalo de percentagem de probabilidade desse risco acontecer: até 20%, 21 a 60%, 61 a 80% e superior a 80%, respetivamente. Estes valores foram adaptados da matriz de probabilidade e impacto em "*A Guide to the Project Management Body of Knowledge (PMBOK Guide)*" (2009)^[30], agregando alguns dos níveis da escala numérica para se adaptarem a quatro níveis de probabilidade.

O espaço de tempo pode ser classificado em curto, médio ou longo prazo dependendo de quando o risco é identificado, e até quando terá que se lidar com o mesmo, tendo em conta a duração total do projeto de estágio. Um curto espaço de tempo considera-se como sendo de 3 semanas. No caso de médio prazo considera-se de 3 semanas a 2 meses. Longo prazo corresponde a mais de 2 meses. Os valores temporais foram escolhidos tendo em conta a duração total da fase de implementação do projeto e a metodologia de desenvolvimento escolhida.

Risco #1

Identificador: R-1

Título: Falta de experiência nas tecnologias

Descrição: As tecnologias escolhidas pelo aluno podem ser totalmente desconhecidas pelo mesmo, levando a curva de aprendizagem seja maior do que o expectável

Impacto: Alto

Probabilidade: Alta (61 a 80%)

Espaço de Tempo: Médio Prazo

Consequências: Falha no desenvolvimento e implementação das funcionalidades do projeto

Plano de Mitigação: A decisão das tecnologias a usar deve ser tomada com base nas que o aluno já se sente mais à vontade, que fazem sentido usar no projeto e que o orientador da Grama domine. Neste último caso, o aluno terá sempre apoio de um especialista na tecnologia, mitigando o risco. Formações sobre a tecnologia também podem ajudar a mais rapidamente aprender as tecnologias.

Risco #2

Identificador: R-2

Título: Dimensão do trabalho demasiado ambiciosa

Descrição: A dimensão total do projeto de estágio sugerido pela empresa pode levar a que exista uma lista demasiado extensa de requisitos a implementar no tempo disponível

Impacto: Alto

Probabilidade: Alta (61 a 80%)

Espaço de Tempo: Médio Prazo

Consequências: Falha no desenvolvimento e implementação das funcionalidades planeadas para o projeto

Plano de Mitigação: Priorização dos requisitos identificados de modo a perceber quais devem ser realmente implementados. Redefinição das prioridades dos requisitos caso não seja possível terminar todas as funcionalidades dentro do tempo disponível.

Risco #3**Identificador:** R-3**Título:** Trabalho conjunto

Descrição: A plataforma consiste na implementação de 4 módulos, 2 são objeto de estudo deste estágio e outros 2 de outro estágio. Tem, portanto, que existir colaboração como outro estagiário. As funcionalidades comuns aos quatro módulos podem ser afetadas caso o outro estagiário não desempenhe a sua função.

Impacto: Médio**Probabilidade:** Baixa (até 20%)**Espaço de Tempo:** Longo Prazo

Consequências: O trabalho que deveria ser realizado em conjunto terá que ser realizado por apenas uma pessoa, elevando a carga de trabalho acima do esperado.

Plano de Mitigação: Coordenar o trabalho com o outro estagiário, dividir as tarefas conjuntas.

Matriz de Riscos

A Tabela A.1 apresenta uma matriz de riscos onde as colunas representam o impacto do risco, dividido nos diferentes níveis definidos, e as linhas representam a probabilidade desse mesmo risco ocorrer, segundo os níveis apresentados no início da presente secção.

Tabela A.1: Matriz de Riscos

		Impacto			
		Baixo	Médio	Alto	Crítico
Prob.	Extrema				
	Alta			R-2, R-1	
	Média				
	Baixa		R-3		

Apêndice B

Epic Stories e User Stories

Módulo Autenticação

Módulo que gere a autenticação dos utilizadores da Ferramenta Interna da Grama.

Autenticação

ES-1: Como utilizador da plataforma não autenticado, **eu quero** autenticar-me na plataforma, **para que** consiga aceder às diversas funcionalidades da mesma.

US-1: Iniciar sessão

- (a) **Como** administrador/gestor/colaborador, **eu quero** iniciar sessão na plataforma, **para que** consiga aceder às suas funcionalidades. Quero introduzir os campos descritos abaixo para que a minha identidade seja validada no acesso à plataforma:

- Email*;
- Palavra-passe*.
(* campo obrigatório)

US-2: Recuperar palavra-passe

- (a) **Como** administrador/gestor/colaborador, **eu quero** poder recuperar a minha palavra-passe, **para que** consiga voltar a autenticar-me na plataforma. Quero introduzir os dados abaixo descritos para que possa receber o email de recuperação de palavra passe:

- Email*;
(* campo obrigatório)

Após inserir o meu email, quero receber um email com um código para recuperar o acesso à conta. Deve ser necessário definir uma nova palavra-passe para conseguir autenticar-me novamente.

US-3: Terminar sessão

- (a) **Como** administrador/gestor/colaborador autenticado na plataforma, **eu quero** poder terminar a sessão na plataforma, **para que** consiga sair de forma segura da mesma.

US-4: Primeiro início de sessão

- (a) **Como** administrador/gestor/colaborador que ainda não aceitou o convite enviado por um administrador e realizar o meu primeiro início de sessão, **eu quero** aceitar o convite enviado por um administrador e realizar o meu primeiro início de sessão, **para que** consiga começar a utilizar a plataforma e as suas funcionalidades. Neste primeiro acesso à plataforma devo ser obrigado a definir a minha palavra-passe para que possa aceder à plataforma de forma segura.

- Palavra passe temporária recebida no email*;
 - Nova palavra passe*;
 - Repetição da nova palavra passe*.
- (* campo obrigatório)

Módulo Gestão de Colaboradores

Este módulo tem como responsabilidade fornecer todas as funcionalidades de gestão de colaboradores associadas com a Ferramenta Interna da Grama.

Gestão de informações dos utilizadores

ES-2: **Como** um administrador/gestor da plataforma, **eu quero** aceder à informação dos utilizadores, **para que** consiga corretamente gerir as informações de cada um.

US-5: Visualizar lista de utilizadores da plataforma com conta ativa

- (a) **Como** administrador, **eu quero** visualizar a lista de todos os utilizadores da plataforma com conta ativa, **para que** consiga ter acesso a todos os utilizadores e às suas informações mais relevantes.
- (b) **Como** gestor, **eu quero** visualizar a lista dos utilizadores da plataforma que fazem parte do(s) meu(s) departamento(s) com conta ativa, **para que** consiga ter acesso aos mesmos e às suas informações mais relevantes.
- (c) A informação deve ser apresentada na forma de tabela com as seguintes colunas:
- Nome;
 - *Role*;
 - Número de telemóvel;
 - Departamento;
 - Posição/Profissão;
 - Data de contratação;
 - Último Login.

US-6: Visualizar lista de utilizadores da plataforma com conta desativada

- (a) **Como** administrador, **eu quero** visualizar a lista de todos os utilizadores da plataforma com conta desativada, **para que**, caso seja necessário, possa reativar as mesmas.
- (b) A informação deve ser apresentada na forma de tabela com as seguintes colunas:
 - Nome;
 - *Role*;
 - Número de telemóvel;
 - Departamento;
 - Cargo na empresa;
 - Data de contratação;
 - Data de desativação da conta.

US-7: Editar informação de utilizador(es) através da lista

- (a) **Como** administrador, **eu quero** editar através da lista as informações mais relevantes de um ou mais utilizadores, **para que** consiga alterar e corrigir informações sem entrar na vista detalhada de cada utilizador.

US-8: Filtrar lista de utilizadores

- (a) **Como** administrador, **eu quero** aplicar filtros na lista de todos os utilizadores, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero filtrar através das seguintes propriedades:
 - Nome;
 - Departamento;
 - Tipo de utilizador (administrador, gestor, colaborador);
 - Cargo na empresa.
- (b) **Como** gestor, **eu quero** aplicar filtros na lista dos utilizadores que fazem parte do(s) meu(s) departamento(s), **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero filtrar através das seguintes propriedades:
 - Nome;
 - Departamento;
 - Tipo de utilizador (administrador, gestor, colaborador);
 - Cargo na empresa.
- (c) **Como** administrador/gestor, **eu quero** efetuar uma pesquisa nas listas a que tenho acesso através das propriedades dos filtros, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar.

US-9: Aceder à vista detalhada de um utilizador

- (a) **Como** administrador, **eu quero** aceder à vista detalhada de um utilizador a partir da lista, **para que** consiga visualizar toda a informação desse mesmo utilizador.
- (b) **Como** gestor, **eu quero** aceder à vista detalhada de um utilizador que faça parte do(s) meu(s) departamento(s) a partir da lista, **para que** consiga visualizar toda a informação desse mesmo utilizador.

ES-3: **Como** um administrador da plataforma, **eu quero** visualizar todas as informações de um utilizador através da sua vista detalhada, **para que** consiga ver e editar alguma informação que seja necessária.

US-10: Visualizar perfil completo

- (a) **Como** administrador, **eu quero** visualizar todas as informações pessoais de qualquer utilizador, **para que** consiga ver qualquer informação que seja necessária. A informação disponível deve estar organizada da seguinte forma:
 - Informação Pessoal:
 - Nome completo;
 - Data de nascimento;
 - Número de telemóvel;
 - Email pessoal;
 - Morada;
 - Habilitações;
 - Número do cartão de cidadão;
 - Número de identificação fiscal;
 - Número da segurança social;
 - Estado civil;
 - Número de dependentes;
 - Nome do contacto de emergência;
 - Número de telemóvel do contacto de emergência;
 - Foto de perfil;
 - Informação Contratual:
 - Data de contratação;
 - Data de término do contrato;
 - Departamento(s) onde está inserido;
 - Cargo na empresa;
 - Tipo de contrato;
 - Documentos associados (com possibilidade de descarregamento);
 - Histórico de Salários em formato de gráfico e em tabela;
 - Histórico de Bónus em formato de gráfico e em tabela;
 - Informação Pagamento:
 - IBAN;
 - Número do cartão de alimentação;
 - Tipo de duodécimos;

- Informação da Conta:
 - Último início de sessão;
 - Nome curto;
 - Email da empresa;
 - Orçamento inicial para equipamentos.

US-11: Visualizar histórico salarial

- (a) **Como** administrador, **eu quero** visualizar o histórico salarial de qualquer utilizador, **para que** consiga saber a sua evolução salarial ao longo do contrato. A informação deve ser exposta da seguinte forma:
- Gráfico com evolução salarial ao longo do tempo;
 - Tabela com os diversos salários inseridos.

US-12: Visualizar histórico de bónus atribuídos

- (a) **Como** administrador, **eu quero** visualizar o histórico de bónus atribuídos a qualquer utilizador, **para que** consiga saber quais os bónus que este recebeu e quando ao longo do contrato. A informação deve ser exposta da seguinte forma:
- Gráfico com evolução dos bónus atribuídos ao longo do tempo;
 - Tabela com os diversos bónus inseridos.

US-13: Editar informações pessoais

- (a) **Como** administrador, **eu quero** editar as informações pessoais de um utilizador, **para que** consiga adicionar, editar ou eliminar o que for necessário. Poderei editar qualquer um dos campos descritos na US-10, excepto o email da empresa e o orçamento inicial para equipamentos.

US-14: Adicionar salários

- (a) **Como** administrador, **eu quero** adicionar salários a um utilizador, **para que** consiga atualizar o salário de um utilizador conforme o necessário. Poderei preencher os seguintes campos:
- Valor do salário*;
 - Data de aplicação (deve ser alterável mas pré-preenchida com a data atual);
 - Comentários.
(* campo obrigatório)

US-15: Editar salários

- (a) **Como** administrador, **eu quero** editar os salários de um utilizador, **para que** consiga editar um salário previamente introduzido de um utilizador conforme o necessário. Poderei realizar alterações aos seguintes campos:
- Valor do salário*;
 - Data de aplicação;
 - Comentários.
- (* campo obrigatório)

US-16: Remover salários

- (a) **Como** administrador, **eu quero** remover salários de um utilizador, **para que** consiga remover um salário previamente introduzido de um utilizador conforme o necessário.

US-17: Adicionar bónus

- (a) **Como** administrador, **eu quero** adicionar bónus a um utilizador, **para que** consiga introduzir os bónus atribuídos a um utilizador. Poderei preencher os seguintes campos:
- Valor do bónus*;
 - Data de aplicação (deve ser alterável mas pré-preenchida com a data atual);
 - Comentários.
- (* campo obrigatório)

US-18: Editar bónus

- (a) **Como** administrador, **eu quero** editar bónus de um utilizador, **para que** consiga editar os bónus que um utilizador recebeu e que foram previamente introduzidos. Poderei realizar alterações aos seguintes campos:
- Valor do bónus*;
 - Data de aplicação;
 - Comentários.
- (* campo obrigatório)

US-19: Remover bónus

- (a) **Como** administrador, **eu quero** remover bónus de um utilizador, **para que** consiga remover os bónus que um utilizador recebeu e que foram previamente introduzidos.

US-20: Adicionar foto de perfil

- (a) **Como** administrador, **eu quero** ter possibilidade de adicionar uma foto de perfil a qualquer utilizador, **para que** consiga corrigir a mesma caso necessário.

US-21: Remover foto de perfil

- (a) **Como** administrador, **eu quero** ter possibilidade de remover a foto de perfil de qualquer utilizador, **para que** esta deixe de aparecer na plataforma.

US-22: Editar foto de perfil

- (a) **Como** administrador, **eu quero** editar a foto de perfil de qualquer plataforma, **para que** consiga corrigir a mesma caso necessário.

US-23: Visualizar e transferir documentos

- (a) **Como** administrador, **eu quero** visualizar e transferir os documentos de todos os utilizadores, **para que** possa aceder aos mesmos.

US-24: Adicionar e remover documentos

- (a) **Como** administrador, **eu quero** conseguir adicionar e remover documentos a todos os utilizadores, **para que** possa gerir os mesmos.

ES-4: **Como** um gestor da plataforma, **eu quero** visualizar todas as informações de cada utilizador que pertença ao(s) meu(s) departamento(s) através da sua vista detalhada, **para que** consiga ver alguma informação que seja necessária, nomeadamente, a sua informação pessoal, documentos contratuais, histórico salarial e histórico de bónus.

US-25: Visualizar perfil completo

- (a) **Como** gestor, **eu quero** visualizar todas as informações pessoais do utilizador que pertence ao meu departamento na mesma página, **para que** consiga ver alguma informação que seja necessária. As informações disponíveis para visualização devem ser as mesmas descritas na US-10 mas sem possibilidade de edição das mesmas.

US-26: Visualizar histórico salarial

- (a) **Como** gestor, **eu quero** visualizar o histórico salarial de um utilizador que pertence ao meu departamento, **para que** consiga saber a sua evolução salarial ao longo do contrato. As informações disponíveis para visualização devem ser as mesmas descritas na US-11.

US-27: Visualizar histórico de bônus atribuídos

- (a) **Como** gestor, **eu quero** visualizar o histórico de bônus atribuídos a um utilizador que pertence ao meu departamento, **para que** consiga saber quais os bônus que este recebeu e quando. As informações disponíveis para visualização devem ser as mesmas descritas na US-12.

US-28: Visualizar e transferir documentos

- (a) **Como** gestor, **eu quero** visualizar e transferir os documentos dos utilizadores que fazem parte do(s) meu(s) departamento(s), **para que** possa aceder aos mesmos.

ES-5: **Como** um administrador/gestor/colaborador da plataforma, **eu quero** visualizar todas as minhas informações através da minha vista detalhada, **para que** consiga ver alguma informação que seja necessária, nomeadamente, a minha informação pessoal, documentos contratuais, histórico salarial e histórico de bônus.

US-29: Visualizar perfil completo

- (a) **Como** administrador/gestor/colaborador, **eu quero** visualizar todas as minhas informações pessoais, **para que** consiga ver alguma informação que seja necessária. As informações disponíveis para visualização devem ser as mesmas descritas na US-10.

US-30: Visualizar histórico salarial

- (a) **Como** administrador/gestor/colaborador, **eu quero** visualizar o meu histórico salarial, **para que** consiga saber a minha evolução salarial ao longo do contrato. As informações disponíveis para visualização devem ser as mesmas descritas na US-11.

US-31: Visualizar histórico de bônus atribuídos

- (a) **Como** administrador/gestor/colaborador, **eu quero** visualizar o meu histórico de bônus atribuídos, **para que** consiga saber quais os bônus que recebi e quando ao longo do contrato. As informações disponíveis para visualização devem ser as mesmas descritas na US-12.

US-32: Editar informações pessoais

- (a) **Como** administrador/gestor/colaborador, **eu quero** editar as minhas próprias informações pessoais, **para que** consiga adicionar, editar ou eliminar o que for necessário. As informações editáveis devem ser as mesmas descritas na US-13.

US-33: Adicionar foto de perfil

- (a) **Como** administrador/gestor/colaborador, **eu quero** adicionar a minha própria foto de perfil, **para que** reflita a foto que eu quero ter na plataforma.

US-34: Remover foto de perfil

- (a) **Como** administrador/gestor/colaborador, **eu quero** remover a minha própria foto de perfil, **para que** esta deixe de aparecer na plataforma.

US-35: Editar foto de perfil

- (a) **Como** administrador/gestor/colaborador, **eu quero** editar a minha própria foto de perfil, **para que** reflita a foto que eu quero ter na plataforma.

US-36: Visualizar e transferir documentos

- (a) **Como** administrador/gestor/colaborador eu quero conseguir visualizar e transferir os documentos que me estão associados, **para que** possa aceder aos mesmos.

Gestão de contas de utilizadores

ES-6: **Como** um administrador da plataforma, **eu quero** gerir as contas de todos os utilizadores da plataforma, **para que** consiga adicionar novos utilizadores, assim como desativar e reativar contas.

US-37: Adicionar e convidar um novo utilizador para a plataforma

- (a) **Como** administrador, **eu quero** adicionar e convidar um novo utilizador para a plataforma, **para que** um novo colaborador da empresa consiga utilizar a mesma. Quero definir as seguintes propriedades de um novo utilizador:

- Informação Pessoal:
 - Nome completo;
 - Data de nascimento;
 - Número de telemóvel;
 - Email pessoal;
 - Morada;
 - Habilitações;
 - Número do cartão de cidadão;
 - Número de identificação fiscal;
 - Número da segurança social;
 - Estado civil;
 - Número de dependentes;
 - Nome do contacto de emergência;

- Número de telemóvel do contacto de emergência;
- Informação Contratual:
 - Data de contratação;
 - Data de término do contrato;
 - Departamento onde deverá ser inserido;
 - Cargo na empresa;
 - Tipo de contrato;
- Informação Pagamento:
 - IBAN;
 - Número do cartão de alimentação;
 - Tipo de duodécimos;
- Informação da Conta:
 - *Role**;
 - Nome curto*;
 - Email da empresa*;
 - Orçamento inicial para equipamentos*.
(* campo obrigatório)

(b) Quero também que seja enviado automaticamente um convite ao novo utilizador de modo a que ele consiga realizar o seu primeiro início de sessão na plataforma.

US-38: Reenviar convite para a plataforma

(a) **Como** administrador, **eu quero** reenviar o convite para a plataforma a um utilizador, **para que** este possa realizar o seu primeiro início de sessão na eventualidade de ter perdido o(s) outro(s) convite(s).

US-39: Desativar a conta de um utilizador

(a) **Como** administrador, **eu quero** conseguir desativar a conta de um utilizador, **para que** este passe para a lista de utilizadores desativados e já não apareça na utilização normal da plataforma nem consiga iniciar sessão na mesma. Poderei preencher os seguintes campos:

- Razão da desativação*;
- Data da desativação (deve ser alterável mas pré-preenchida com a data atual).
(* campo obrigatório)

US-40: Reativar a conta de um utilizador

- (a) **Como** administrador, **eu quero** conseguir reativar a conta de um utilizador, **para que** este volte a ser um utilizador ativo na plataforma. Ao reativar o utilizador, este deve receber um novo convite para a plataforma. Poderei preencher os seguintes campos:
- Razão da reativação*;
 - Data da reativação (deve ser alterável mas pré-preenchida com a data atual).
(* campo obrigatório)

Gestão de departamentos

ES-7: **Como** um administrador da plataforma, **eu quero** gerir todos os departamentos e as pessoas atribuídas a cada um, **para que** consiga ver e editar alguma informação que seja necessária.

US-41: Visualizar lista de departamentos ativos

- (a) **Como** administrador, **eu quero** visualizar a lista de todos os departamentos ativos, **para que** consiga aceder a todos e visualizar ou editar a informação necessária. Cada departamento deverá ter a seguinte informação disponível para visualização:
- Nome;
 - Descrição;
 - Cor identificativa;
 - Gestores;
 - Membros.

US-42: Adicionar um novo departamento

- (a) **Como** administrador, **eu quero** conseguir adicionar novos departamentos e definir os seus membros e gestor, **para que** consiga gerir os diversos departamentos da empresa. Quero adicionar, num novo departamento, a seguinte informação:
- Nome;
 - Descrição;
 - Cor identificativa;
 - Gestores;
 - Membros.

US-43: Editar um departamento existente

- (a) **Como** administrador, **eu quero** conseguir editar departamentos existentes, **para que** consiga alterar membros e gestor(es) dos diversos departamentos da empresa. Quero editar, num departamento existente as mesmas propriedades descritas na US-42.

US-44: Remover um departamento

- (a) **Como** administrador, **eu quero** conseguir remover departamentos existentes, **para que** consiga gerir os diversos departamentos da empresa.

ES-8: **Como** um gestor da plataforma, **eu quero** ver os meus departamentos e as pessoas atribuídas a cada um, **para que** consiga visualizar alguma informação que seja necessária.

US-45: Visualizar lista dos meus departamentos

- (a) **Como** gestor, **eu quero** visualizar a lista dos meus departamentos, **para que** consiga aceder aos mesmos e visualizar a informação necessária.

Módulo Gestão de Equipamentos

Toda a gestão dos equipamentos da empresa é feita através deste módulo da Ferramenta Interna da Grama.

Dashboard

ES-9: **Como** um administrador da plataforma, **eu quero** conseguir visualizar, numa página, as informações mais relevantes em relação à gestão de equipamentos assim como lista de equipamentos e utilizadores com informação relevante, **para que** consiga visualizar um resumo das estatísticas mais importantes sem ter que aceder a diversas páginas diferentes.

US-46: Visualizar, num *dashboard*, a informação agregada sobre equipamentos, orçamentos e pedidos.

- (a) **Como** administrador, **eu quero** visualizar um resumo de todas as informações mais relevantes acerca da gestão de equipamentos, **para que** consiga ver numa só página toda a informação importante. Quero visualizar na mesma página a seguinte informação:
- Gráficos com informação sobre distribuição do orçamento ao longo do tempo;
 - Listas de equipamentos com informação mais relevante;
 - Listas de utilizadores com informação mais relevante.
- (b) **Como** administrador, **eu quero** visualizar listas de equipamentos com informação mais relevante, **para que** consiga ver numa só página toda a informação importante. Quero visualizar listas referentes aos equipamentos:
- Totais;
 - Em uso;
 - Alocados a um utilizador;
 - Não alocados a um utilizador;
 - A terminar a garantia;
 - A terminar tempo de vida útil.

- (c) **Como** administrador, **eu quero** visualizar listas de utilizadores com informação mais relevante, **para que** consiga ver numa só página toda a informação importante. Quero visualizar listas referentes aos utilizadores:
- Cujo orçamento será acrescido;
 - Com maior/menor orçamento disponível.

Gestão de equipamentos

ES-10: **Como** um administrador da plataforma, **eu quero** conseguir visualizar a lista de todos os equipamentos ativos, inativos e desativados, **para que** consiga ver quais equipamentos que existem na empresa, qual o estado dos mesmos, a quem estão associados e outras informações relevantes.

US-47: Visualizar listas de equipamentos ativos/inativos/obsoletos

- (a) **Como** administrador, **eu quero** visualizar listas com todos os equipamentos ativos/inativos/obsoletos, **para que** consiga através das listas perceber quais equipamentos estão alocados e a que utilizador assim como quais é que não estão alocados e quais estão desativados
- (b) A informação deve ser apresentada na forma de tabela com as seguintes colunas:
- Código do equipamento;
 - Tipo;
 - Modelo;
 - Fornecedor;
 - Preço;
 - Data de compra;
 - Utilizador que realizou o processo de compra;
 - Utilizador a quem está alocado (não existente na tabela de equipamentos obsoletos);
 - Estado / Data desde que é obsoleto (caso seja na tabela de equipamentos obsoletos);

US-48: Filtrar lista de equipamentos

- (a) **Como** administrador, **eu quero** aplicar filtros nas listas de equipamentos, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero filtrar através das seguintes propriedades:
- Colaborador a quem está associado;
 - Por quem foi adquirido;
 - Tipo de equipamento;
 - Marca;
 - Fornecedor;
 - Estado.

(b) **Como** administrador, **eu quero** pesquisar nas listas de equipamentos, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero pesquisar, por exemplo, através do:

- Nome;
- ID.

US-49: Editar informação de equipamento(s) através da lista

(a) **Como** administrador, **eu quero** editar a informação de um ou mais equipamentos através da lista, **para que** consiga alterar certas informações sem ter que aceder à vista detalhada de um equipamento.

US-50: Adicionar um novo equipamento

(a) **Como** administrador, **eu quero** poder adicionar um novo equipamento, **para que** este possa ser alocado a um utilizador através da plataforma. Quero definir as seguintes propriedades de um novo equipamento:

- Tipo;
- Estado;
- Alocado a;
- Comprado por;
- Comprado em;
- Fornecedor;
- Preço;
- Tempo de garantia;
- Tempo de vida útil;
- Marca;
- Modelo
- Número de série;
- Localização;
- Documentos associados;
- Comentários gerais.

(b) A cada equipamento criado deve ser atribuído automaticamente um código único que permita identificar o mesmo. Exemplo: "grama-pc-0001".

ES-11: **Como** um administrador da plataforma, **eu quero** visualizar todas as informações de um equipamento através da sua vista detalhada, **para que** consiga ver e editar alguma informação que seja necessária, descarregar documentos associados ao equipamento, desativar o equipamento ou alterar o utilizador ao qual está alocado.

US-51: Visualizar todas as informações de um equipamento

- (a) **Como** administrador, **eu quero** visualizar na mesma página todas as informações de um equipamento, **para que** consiga ver alguma informação que seja necessária. A informação disponível deve ser a seguinte:
- Código do equipamento;
 - Estado;
 - Data de compra;
 - Número de série;
 - Utilizador que realizou o processo de compra;
 - Tipo;
 - Utilizador a quem se encontra alocado;
 - Localização;
 - Comentários;
 - Fornecedor;
 - Preço;
 - Marca;
 - Modelo;
 - Documentos associados.

US-52: Adicionar documentos a um equipamento

- (a) **Como** administrador, **eu quero** adicionar documentos a um equipamento, **para que** fiquem disponíveis para outros utilizadores da plataforma que possam necessitar dos mesmos.

US-53: Descarregar documentos associados a um equipamento

- (a) **Como** administrador, **eu quero** descarregar os documentos que estão associados a um equipamento, **para que** consiga aceder à informação dos mesmos como é o caso da fatura.

US-54: Editar as informações de um equipamento

- (a) **Como** administrador, **eu quero** editar as informações de um equipamento, **para que** consiga adicionar, atualizar ou remover alguma informação que seja necessária. Quero que seja possível editar toda as informações descritas na US-51 excepto o código do equipamento, a data de compra, o número de série e o utilizador que realizou o processo de compra do equipamento.

US-55: Desativar um equipamento

- (a) **Como** administrador, **eu quero** conseguir desativar um equipamento (torná-lo obsoleto), **para que** deixe de estar disponível para ser alocado a utilizadores.

US-56: Alterar a que colaborador está alocado um equipamento

- (a) **Como** administrador, **eu quero** conseguir alterar o utilizador a quem está alocado o equipamento, **para que** esteja corretamente definida a alocação do mesmo.

ES-12: **Como** um utilizador da plataforma, **eu quero** visualizar todas as informações de um equipamento através da sua vista detalhada, **para que** consiga ver alguma informação que seja necessária ou descarregar documentos associados ao equipamento.

US-57: Visualizar todas as informações de um equipamento

- (a) **Como** gestor, **eu quero** visualizar na mesma página todas as informações de um equipamento que esteja alocado a um membro do meu departamento, **para que** consiga ver alguma informação que seja necessária.
- (b) **Como** administrador/gestor/colaborador, **eu quero** visualizar na mesma página todas as informações de um equipamento que me esteja alocado, **para que** consiga ver alguma informação que seja necessária.
- (c) A informação disponibilizada deve ser a mesma descrita na US-51.

US-58: Descarregar documentos associados a um equipamento

- (a) **Como** gestor, **eu quero** descarregar os documentos que estão associados a um equipamento que esteja alocado a um membro do meu departamento, **para que** consiga aceder à informação dos mesmos como é o caso da fatura.
- (b) **Como** administrador/gestor/colaborador, **eu quero** descarregar os documentos que estão associados a um equipamento que me está alocado, **para que** consiga aceder à informação dos mesmos como é o caso da fatura.

Gestão de orçamentos dos utilizadores

ES-13: **Como** um administrador da plataforma, **eu quero** conseguir gerir o orçamento dos utilizadores da plataforma, **para que** consiga adicionar ou remover orçamento ou aceder aos orçamentos de cada um dos utilizadores e aceitar/recusar pedidos de equipamentos.

US-59: Visualizar lista com todos os orçamentos de todos os colaboradores

- (a) **Como** administrador, **eu quero** visualizar uma lista com todos os orçamentos de todos os utilizadores e outras informações relevantes, **para que** possa ter uma visão geral do estado dos orçamentos de cada utilizador. A informação deve ser apresentada na forma de tabela com as seguintes colunas:
- Nome do utilizador;
 - Data de última alteração do orçamento;

- Orçamento inicial para equipamentos;
- Orçamento utilizado;
- Orçamento disponível;
- Data de compra;
- Data de entrada na empresa.

US-60: Editar o orçamento de um dos utilizadores através da lista

- (a) **Como** administrador, **eu quero** conseguir editar o orçamento de um utilizador através da lista, **para que** possa adicionar ou remover um valor ao orçamento.

US-61: Visualizar a lista com todos os pedidos pendentes de equipamentos

- (a) **Como** administrador, **eu quero** visualizar uma lista com todos os pedidos pendentes de equipamentos realizados pelos utilizadores, **para que** possa aceitar ou recusar os mesmos. A informação deve ser apresentada na forma de tabela com as seguintes colunas:

- Nome do utilizador;
- Tipo de equipamento;
- *Link* para o equipamento;
- Preço de referência.

US-62: Analisar pedido pendente de equipamento

- (a) **Como** administrador, **eu quero** conseguir analisar o pedido pendente de um equipamento por parte de um utilizador, **para que** possa aceitar ou recusar o mesmo e este saiba se o seu pedido foi aceite ou não.

US-63: Visualizar listas com todos os pedidos aceites

- (a) **Como** administrador, **eu quero** visualizar uma lista com todos os pedidos que foram aceites, **para que** consiga saber quais foram os pedidos de equipamento a serem aceites. A informação deve ser apresentada na forma de tabela com as seguintes colunas:

- Nome do utilizador;
- Tipo de equipamento;
- *Link* para o equipamento;
- Preço de referência;
- Data de aprovação;
- Administrador que aprovou o pedido.

US-64: Visualizar listas com todos os pedidos recusados

- (a) **Como** administrador, **eu quero** visualizar uma lista com todos os pedidos que foram recusados, **para que** consiga saber quais foram os pedidos de equipamento a serem recusados. A informação deve ser apresentada na forma de tabela com as seguintes colunas:
- Nome do utilizador;
 - Tipo de equipamento;
 - *Link* para o equipamento;
 - Preço de referência;
 - Data de recusa;
 - Administrador que recusou o pedido.

US-65: Filtrar lista de orçamentos dos utilizadores

- (a) **Como** administrador, **eu quero** aplicar filtros na lista de orçamentos dos utilizadores, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero filtrar através das seguintes propriedades:
- Colaborador;
 - Tipo de equipamento;
 - Período do pedido;
 - Orçamento restante do utilizador.
- (b) **Como** administrador, **eu quero** pesquisar sobre a lista de orçamentos dos utilizadores, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar.

ES-14: **Como** um administrador da plataforma, **eu quero** visualizar os detalhes do orçamento de um utilizador, **para que**, através de gráficos com histórico assim como os pedidos de novos equipamentos e pedidos de reparação por parte do utilizador, a escolha de aceitar ou recusar um pedido de equipamento seja mais informada.

US-66: Visualizar histórico do orçamento do utilizador

- (a) **Como** administrador, **eu quero** ver um histórico com a evolução do orçamento do utilizador, **para que** possa ter uma visão geral da evolução do orçamento do utilizador.

US-67: Visualizar lista de equipamentos associados ao utilizador

- (a) **Como** administrador, **eu quero** ver uma lista dos equipamentos associados ao utilizador, **para que** consiga saber quais os equipamentos que este tem alocados a si.

US-68: Visualizar pedidos de novos equipamentos realizados pelo utilizador

- (a) **Como** administrador, **eu quero** ver uma lista dos pedidos de novos equipamentos realizados pelo utilizador, **para que** consiga saber quais são os pedidos pendentes que o utilizador realizou.

US-69: Visualizar pedidos de reparação realizados pelo utilizador

- (a) **Como** administrador, **eu quero** ver uma lista dos pedidos de reparação realizados pelo utilizador, **para que** consiga saber quais são os equipamentos do utilizador que necessitam reparação.

Gestão de pedidos de equipamento

ES-15: **Como** um administrador da plataforma, **eu quero** visualizar os detalhes de um pedido de equipamento assim como a informação do utilizador que realizou o pedido, **para que** consiga aceitar ou recusar o mesmo.

US-70: Visualizar todas as informações de um pedido de equipamento

- (a) **Como** administrador, **eu quero** visualizar na mesma página todas as informações de um pedido de equipamento assim como a informação de quem fez o pedido, **para que** consiga tomar a decisão de aceitar ou recusar o pedido baseado no orçamento disponível e outras informações do utilizador. As informações apresentadas devem ser as seguintes:
- Data de realização do pedido;
 - Nome do utilizador;
 - Orçamento do utilizador restante caso o equipamento seja comprado;
 - *Link* para o equipamento;
 - Preço de referência;
 - Informação adicional fornecida pelo utilizador que realizou o pedido.

US-71: Aceitar ou recusar um pedido de equipamento

- (a) **Como** administrador, **eu quero** aceitar ou recusar um pedido de equipamento, **para que** o utilizador que realizou o pedido obtenha a sua resposta.

ES-16: **Como** um administrador/gestor/colaborador, **eu quero** gerir os meus pedidos de equipamentos, **para que** consiga realizar, editar ou cancelar os meus pedidos de equipamentos.

US-72: Visualizar lista de pedidos de reparação de equipamentos

- (a) **Como** administrador/gestor/colaborador, **eu quero** ver uma lista dos meus pedidos de reparação de equipamentos, **para que** consiga saber quais são os pedidos que eu realizei.

US-73: Criar um novo pedido de equipamento

- (a) **Como** administrador/gestor/colaborador, **eu quero** conseguir criar um novo pedido de equipamento, **para que** este possa ser analisado por um administrador e este possa aceitar ou recusar o mesmo. Quero definir os seguintes campos:
- Tipo de equipamento;
 - *Link* para o equipamento;
 - Preço de referência;
 - Informação adicional.

US-74: Filtrar lista de pedidos de equipamento

- (a) **Como** administrador/gestor/colaborador, **eu quero** aplicar filtros na lista dos meus pedidos de equipamentos, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero filtrar através das seguintes propriedades:
- Colaborador;
 - Tipo de equipamento;
 - Período do pedido.
- (b) **Como** administrador/gestor/colaborador, **eu quero** pesquisar na minha lista de pedidos de equipamentos, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar.

Gestão de reparação de equipamentos

ES-17: **Como** um administrador da plataforma, **eu quero** gerir os pedidos de reparação de equipamentos, **para que**, consiga ter a possibilidade de alterar o estado dos mesmos, aceitar pedidos de reparação ou criar novos pedidos.

US-75: Visualizar lista de pedidos de reparação de equipamentos

- (a) **Como** administrador, **eu quero** ver uma lista de todos os pedidos de reparação pendentes, **para que** consiga saber quais são os equipamentos que necessitam reparação e os utilizadores que necessitam dos seus equipamentos reparados.

US-76: Alterar, no pedido de reparação, o estado de um equipamento

- (a) **Como** administrador, **eu quero** alterar o estado de um equipamento diretamente no pedido de reparação na lista de pedidos de reparação, **para que** consiga alterar o estado sem entrar na vista detalhada do pedido de reparação.

US-77: Adicionar comentário a um pedido de reparação

- (a) **Como** administrador, **eu quero** conseguir adicionar um comentário informativo a um pedido de reparação, **para que** o utilizador que realizou o mesmo possa ser informado nos detalhes acerca do pedido de reparação.

US-78: Adicionar ficheiro(s) a um pedido de reparação

- (a) **Como** administrador, **eu quero** conseguir adicionar um ou mais ficheiros a um pedido de reparação, **para que** o mesmo fique completo com toda a informação necessária para reparar o equipamento.

US-79: Filtrar lista de pedidos de reparação

- (a) **Como** administrador, **eu quero** aplicar filtros na lista de pedidos de reparação, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero filtrar através das seguintes propriedades:
- Colaborador;
 - Tipo de equipamento;
 - Período do pedido.
- (b) **Como** administrador, **eu quero** pesquisar na lista de pedidos de reparação, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar.

US-80: Criar um novo pedido de reparação de equipamento

- (a) **Como** administrador, **eu quero** conseguir criar um novo pedido de reparação de equipamento, **para que** o processo de reparação possa começar e, eventualmente, o equipamento ser reparado e voltar a estar em utilização. Quero definir os seguintes campos:
- Equipamento que necessita de reparação;
 - Descrição do problema do equipamento;
 - Informação adicional.

ES-18: **Como** um administrador/gestor/colaborador, **eu quero** gerir os meus pedidos de reparação de equipamentos, **para que** consiga realizar, editar ou cancelar pedidos de reparação dos meus equipamentos.

US-81: Visualizar lista de pedidos de reparação de equipamentos

- (a) **Como** administrador/gestor/colaborador, **eu quero** ver uma lista dos meus pedidos de reparação de equipamentos, **para que** consiga saber quais são os pedidos que eu realizei. A informação deve ser apresentada na forma de tabela com as seguintes colunas:

- Nome do utilizador que realizou o pedido;
- Código da reparação;
- Tipo;
- Código do equipamento;
- Data do pedido de reparação;
- Descrição do problema.

US-82: Criar um novo pedido de reparação de equipamento

- (a) **Como** administrador/gestor/colaborador, **eu quero** conseguir criar um novo pedido de reparação de equipamento, **para que** o processo de reparação possa começar e, eventualmente, o equipamento ser reparado e voltar a estar em utilização. Os campos a definir devem ser os mesmos da US-80.

US-83: Filtrar lista de pedidos de reparação

- (a) **Como** administrador/gestor/colaborador, **eu quero** aplicar filtros na lista dos meus pedidos de reparação, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar. Quero filtrar através das seguintes propriedades:
- Colaborador;
 - Tipo de equipamento;
 - Período do pedido.
- (b) **Como** administrador/gestor/colaborador, **eu quero** pesquisar na minha lista de pedidos de reparação, **para que** consiga visualizar apenas aqueles que necessito para a tarefa que estou a realizar.

Apêndice C

Tabelas de Requisitos Funcionais

Módulo de Autenticação

Tabela C.1: Tabela de Requisitos Funcionais - Módulo de Autenticação

Área	ID	Requisito	Administrado	Gestor	Colaborador	Prioridade	Dependência(s)
Autenticação	RF-1	Primeiro início de sessão	✓	✓	✓	M	
	RF-2	Iniciar sessão	✓	✓	✓	M	RF-1
	RF-3	Terminar sessão	✓	✓	✓	M	RF-2
	RF-4	Recuperar palavra-passe	✓	✓	✓	M	RF-1

Módulo de Gestão de Colaboradores

Tabela C.2: Tabela de Requisitos Funcionais - Módulo de Gestão de Colaboradores - Área de Gestão de Informações dos Utilizadores

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)	
Gestão de informações dos utilizadores	RF-5	Visualizar lista de utilizadores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2	
	RF-6	conta ativa	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2	
	RF-7	conta desativada	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2	
	RF-7	Editar informação de utilizador(es) através da lista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-1, RF-5, RF-6	
	RF-8	por nome	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	C	RF-1, RF-5, RF-6	
	RF-9	por departamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-1, RF-5, RF-6	
	RF-10	por role	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-1, RF-5, RF-6	
	RF-11	por job position	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	C	RF-1, RF-5, RF-6	
	RF-12	Pré-visualizar informação do utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	W		
	RF-13	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2	
	RF-14	Vista detalhada do perfil completo	de utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-15	do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2	
	RF-16	Editar informações pessoais	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-17	do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2	
	RF-18	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-13	
	RF-19	Visualizar histórico salarial	de utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-14
	RF-20	do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-15	
	RF-21	Adicionar e remover salários	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-18	
	RF-22	Editar salários	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-18	
	RF-23	Visualizar histórico de bônus atribuídos	de todos os utilizadores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-13
	RF-24	de utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-14	
	RF-25	do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-15	
	RF-26	Adicionar e remover bônus	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-18	
	RF-27	Editar bônus	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-2, RF-18	
	RF-28	Editar foto de perfil	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-13
	RF-29	do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-15	
	RF-30	Visualizar documentos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-13, RF-14, RF-15	
	RF-31	Adicionar e remover documentos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-13	
RF-32	Transferir documentos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-13, RF-14, RF-15		

Tabela C.3: Tabela de Requisitos Funcionais - Módulo de Gestão de Colaboradores - Área de Gestão de Contas dos Utilizadores

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)
Gestão de contas de utilizadores	RF-33	Adicionar e convidar um utilizador para a plataforma	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-34	Reenviar convite para a plataforma	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-2, RF-33
	RF-35	Desativar a conta de um colaborador	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-33
	RF-36	Reativar a conta de um colaborador	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-6

Tabela C.4: Tabela de Requisitos Funcionais - Módulo de Gestão de Colaboradores - Área de Gestão de Departamentos

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)
Gestão de departamentos	RF-37	Visualizar lista de departamentos ativos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-38	Adicionar um novo departamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-39	Editar um departamento existente	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-38
	RF-40	Eliminar um departamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-38

Módulo de Gestão de Equipamentos

Tabela C.5: Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de *Dashboard*

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)
Dashboard	RF-41	Visualizar informação agregada sobre equipamentos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2
	RF-42	Visualizar informação agregada sobre orçamentos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2
	RF-43	Visualizar informação agregada sobre pedidos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2

Tabela C.6: Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Equipamentos

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)	
Gestão de equipamentos	RF-44	Visualizar lista de equipamentos ativos	ativos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-56
	RF-45		inativos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-56
	RF-46		obsoletos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-56
	RF-47	Filtrar as listas de equipamentos por	colaborador a quem está alocado	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-56
	RF-48		adquirido por	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-56
	RF-49		tipo de equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-56
	RF-50		marca	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	C	RF-2, RF-56
	RF-51		fornecedor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	C	RF-2, RF-56
	RF-52		estado	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-56
	RF-53		Pesquisar na lista de equipamentos (através de nome, id, etc.)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-56
	RF-54	Editar informações do equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-56	
	RF-55	Editar a informação de equipamento(s) através da lista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-2, RF-56	
	RF-56	Adicionar um novo equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2	
	RF-57	Desativar equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-44, RF-56	
	RF-58	Desativar equipamento através da lista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-2, RF-44, RF-56	
	RF-59	Reativar equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-46, RF-56, RF-57	
	RF-60	Reativar equipamento através da lista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-2, RF-46, RF-56, RF-57	
	RF-61	Alterar a que utilizador o equipamento está alocado	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-56	
	RF-62	Alocar equipamento inativo através da lista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-2, RF-45	
	RF-63	Visualizar todas as informações de	qualquer equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-64		equipamentos alocados a membros da(s) sua(s) equipa(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-65	Adicionar e remover documento(s) associado(s) ao equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-63	
	RF-66	Descarregar documento(s) associado(s) ao equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	C	RF-2	
	RF-67	Visualizar lista de equipamentos pessoais	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2	

Tabela C.7: Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Orçamentos dos Utilizadores

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)	
Gestão de orçamentos dos utilizadores	RF-68	Visualizar lista com os orçamentos de todos os utilizadores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2	
	RF-69	Editar o orçamento de um dos utilizadores (adicionar/remover)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-68	
	RF-70	Visualizar lista com todos os pedidos pendentes de equipamentos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-99	
	RF-71	Visualizar lista com todos os pedidos de equipamentos aceites	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-98	
	RF-72	Visualizar lista com todos os pedidos de equipamentos recusados	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-98	
	RF-73	Filtrar lista de orçamentos dos colaboradores por	colaborador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-68
	RF-74		tipo de equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-68
	RF-75		período do pedido	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-68
	RF-76		orçamento restante do colaborador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-68
	RF-77	Visualizar orçamento disponível	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-78		de utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2
	RF-79		do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2
	RF-80	Visualizar histórico do orçamento	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2
	RF-81		de utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2
	RF-82		do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2
	RF-83	Auditar alterações de orçamento	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2
	RF-84		de utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2
	RF-85		do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2
	RF-86	Visualizar lista de equipamentos alocados	a todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-61
	RF-87		a utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-61
	RF-88		ao próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-61
	RF-89	Visualizar pedidos de novos equipamentos realizados	por todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-99
	RF-90		por utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-99
	RF-91		por próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-99
RF-92	Visualizar pedidos de reparação de equipamentos realizados	por todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-105	
RF-93		por utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-105	
RF-94		por próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-105	

Tabela C.8: Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Pedidos de Equipamentos

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)	
Gestão de pedidos de equipamento	RF-95	Visualizar informações dos pedidos de equipamentos	de todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-99
	RF-96		de utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-99
	RF-97		do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-99
	RF-98	Aceitar ou recusar pedido de equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-99	
	RF-99	Adicionar próprio pedido de equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2	
	RF-100	Editar próprio pedido de equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	C	RF-2, RF-99	
	RF-101	Cancelar próprio pedido de equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-99	

Tabela C.9: Tabela de Requisitos Funcionais - Módulo de Gestão de Equipamentos - Área de Gestão de Reparação de Equipamentos

Área	ID	Requisito	Administrador	Gestor	Colaborador	Prioridade	Dependência(s)
Gestão de reparação de equipamentos	RF-102	Visualizar lista de pedidos de reparação de equipamentos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-105
	RF-103	por todos os utilizadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-105
	RF-104	por utilizadores do(s) seu(s) departamento(s)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-105
	RF-105	por próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-105
	RF-106	Criar um novo pedido de reparação de um equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2
	RF-107	Alterar, no pedido de reparação, o estado de um equipamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-105
	RF-108	Editar um pedido de reparação	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-105
	RF-109	de qualquer utilizador	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	M	RF-2, RF-105
	RF-110	do próprio utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	M	RF-2, RF-105
	RF-111	Adicionar ficheiro(s) a um pedido de reparação	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	C	RF-2, RF-105
	RF-112	Filtrar lista de pedidos de reparação de equipamentos por	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-105
		Utilizador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	S	RF-2, RF-105
		Tipo de equipamento	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-105
		Período do pedido	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S	RF-2, RF-105

Apêndice D

Vantagens e Desvantagens das Tecnologias Analisadas

Tabela D.1: Vantagens e Desvantagens das Tecnologias - Backend

Framework	Express.js	Django	Quarkus	Spring	Spark
Ano de Lançamento	2010	2005	2019	2003	2011
Linguagem	JavaScript	Python	Java/Kotlin	Java/Kotlin	Java/Kotlin
Popularidade no GitHub	58.8 mil	67.2 mil	11 mil	49.7 mil	9.4 mil
Vantagens	<ul style="list-style-type: none">- Fácil aprendizagem por ser baseado em Javascript- Roteamento de pedidos de clientes- Simplicidade, flexibilidade, eficiência, minimalismo, escalabilidade- Desenvolvimento mais rápido do lado do servidor- Middleware- Mesma linguagem para o backend e frontend (se este for javascript, mais comum)	<ul style="list-style-type: none">- Implementado em python- Muito código já feito para desenvolvimento web- Escalável- Segurança	<ul style="list-style-type: none">- Documentação detalhada- Boot time da aplicação mais rápido que Spring- Contentores Kubernetes- Eficaz para integração em microsserviços	<ul style="list-style-type: none">- Documentação detalhada- Comunidade grande com muitas soluções para problemas comuns- Operações de I/O mais rápidas em comparação a Quarkus- Melhor segurança comparado com Quarkus	<ul style="list-style-type: none">- Integração com microsserviços- Acesso total à JVM- Flexibilidade de implementação
Desvantagens	<ul style="list-style-type: none">- Baseado em Node.js significa que é single-threaded, código com muitos callbacks- Não inclui predefinições sobre como implementar a segurança	<ul style="list-style-type: none">- Arquitetura Monolítica- Pesado em termos de recursos, não é ideal para projetos pequenos	<ul style="list-style-type: none">- Complicado de configurar- Poucas soluções para problemas devido à pequena comunidade- Mais difícil para iniciantes	<ul style="list-style-type: none">- Tempos de inicialização maiores que Quarkus- Contém muitas dependências- Mais recursos de memória do que Quarkus	<ul style="list-style-type: none">- Comunidade pequena- Pouco código já existente

Tabela D.2: Vantagens e Desvantagens das Tecnologias - Frontend

Framework	Vue.js	Angular	React	Svelte	Alpine
Ano de Lançamento	2014	2016	2013	2016	2019
Popularidade no GitHub	201 mil	84.8 mil	197 mil	63.1 mil	22.1 mil
Vantagens	<ul style="list-style-type: none"> - Uso da DOM Virtual - Reusabilidade de componentes - Código de fácil leitura - Fácil aprendizagem - Documentação concisa - Comunidade grande 	<ul style="list-style-type: none"> - Performance - CLI - Bom suporte para desenvolvimento mobile - Reusabilidade de componentes - Fácil leitura do código - Unit-testing simplificado - Boa documentação - Suporte da Google - Ecossistema 	<ul style="list-style-type: none"> - Fácil aprendizagem e uso - Bom para aplicações web dinâmicas - Reusabilidade de componentes - Uso da DOM Virtual - Aplicações fáceis de testar 	<ul style="list-style-type: none"> - Compila o código para Javascript nativo e eficiente - Reduz <i>overhead</i> da framework - Código de fácil leitura 	<ul style="list-style-type: none"> - Minimalista - Simples e eficiente - Fácil aprendizagem
Desvantagens	<ul style="list-style-type: none"> - Barreira de linguagem, muita informação online encontra-se em Chinês - Menor popularidade que React ou Angular 	<ul style="list-style-type: none"> - Aprendizagem inicial difícil - Gestão dos componentes requer muitos ficheiros (até 5 por componente) 	<ul style="list-style-type: none"> - Constantes updates levam a uma documentação menos rica para novas funcionalidades - Apenas para UI - Aprendizagem de JSX pode ser uma barreira para iniciantes 	<ul style="list-style-type: none"> - Poucas ferramentas dev disponíveis - Comunidade ainda em crescimento 	<ul style="list-style-type: none"> - Recomendado para websites simples e mais estáticos - Poucas funcionalidades - Comunidade pequena

Tabela D.3: Vantagens e Desvantagens das Tecnologias - Cloud Service Provider

Service Provider	Google	Microsoft	Amazon
Nome da plataforma	Google Cloud Services	Microsoft Azure	Amazon Web Services (AWS)
Fatia de mercado	~11%	~21%	~34%
Vantagens	<ul style="list-style-type: none"> - Integração com outros serviços da Google - I/O rápido - Análise de dados e armazenamento - Facilidade de colaboração - Desenhado para negócios baseados na cloud - Portabilidade e suporte Open Source 	<ul style="list-style-type: none"> - Disponibilidade alta - Foco grande na segurança - Escalabilidade - Opções IaaS e PaaS competitivas - Suporte para Open Source - Cloud Híbrida 	<ul style="list-style-type: none"> - Maior presença no mercado (popularidade) - Maturidade e dominância - Flexibilidade - Muitas aplicações - Updates e manutenção frequentes - Versão grátis disponível - Bom controlo sobre a segurança - Escalabilidade - Rápido deployment - Suporte para empresas de grande dimensão
Desvantagens	<ul style="list-style-type: none"> - Baseado em proprietary tech da Google, pouca gestão sobre as VMs - Escolha limitada de linguagens de programação - Menos funcionalidades/serviços que outras ofertas - Menos data centers globais 	<ul style="list-style-type: none"> - Necessita de mais gestão - Necessita de um conhecimento mais detalhado - Menos possibilidades de retro-compatibilidade - Mais caro para opções pay-as-you-go - Serviço de apoio 	<ul style="list-style-type: none"> - A grande oferta de escolhas pode ser um fator de desencorajamento para empresas menos habituadas à Cloud - Existem limites impostos a recursos dependendo da regiões - Downtime dos servidores - Custos variáveis

Apêndice E

Evidências

O presente apêndice mostra as restantes funcionalidades principais da Ferramenta Interna da Grama que foram desenvolvidas que não foram incluídas no corpo do documento para facilitar a sua leitura.

Módulo de Autenticação

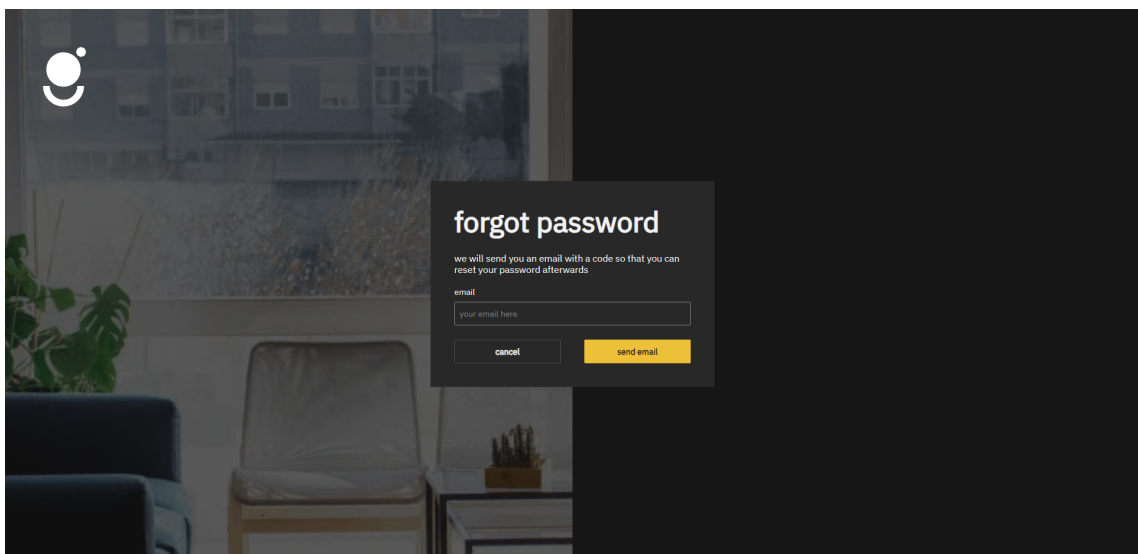


Figura E.1: Ecrã de recuperar palavra passe

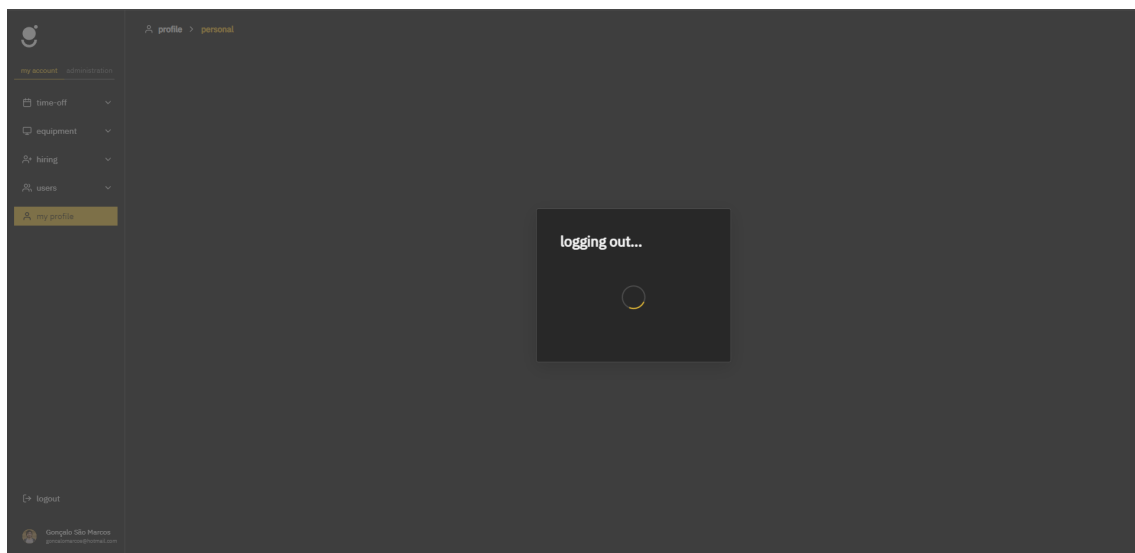


Figura E.2: *Modal* de terminar sessão

Módulo de Gestão de Colaboradores

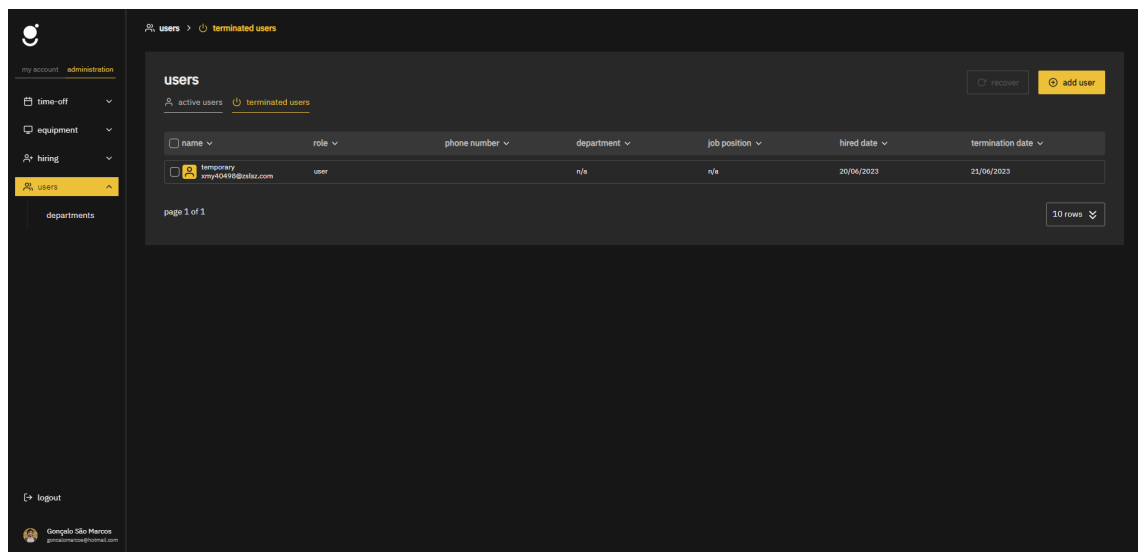


Figura E.3: Lista de Utilizadores Com Conta Desativada

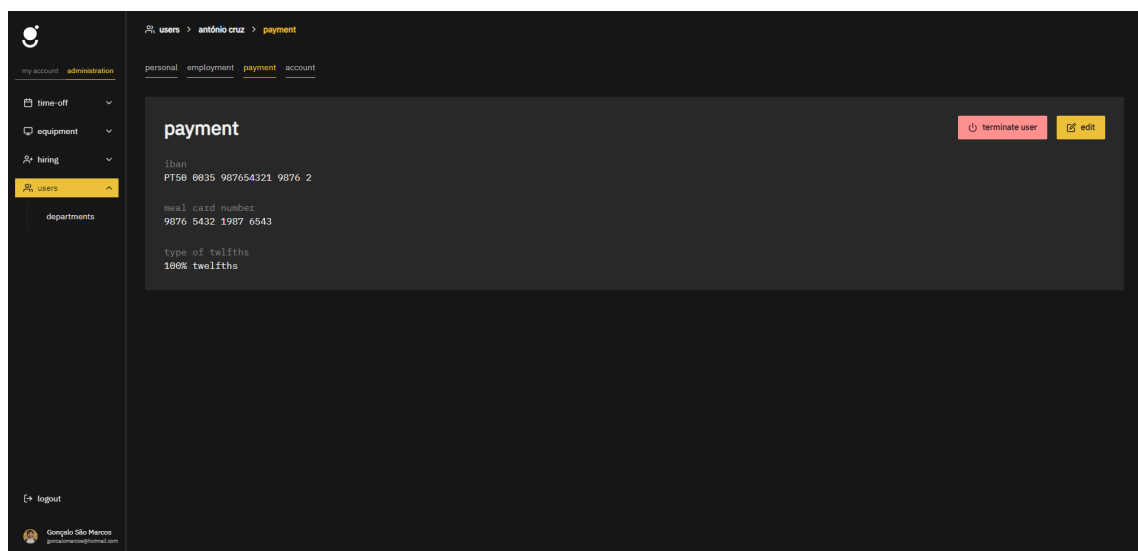


Figura E.4: Vista Detalhada do Perfil de um Utilizador - Separador da Informação de Pagamento

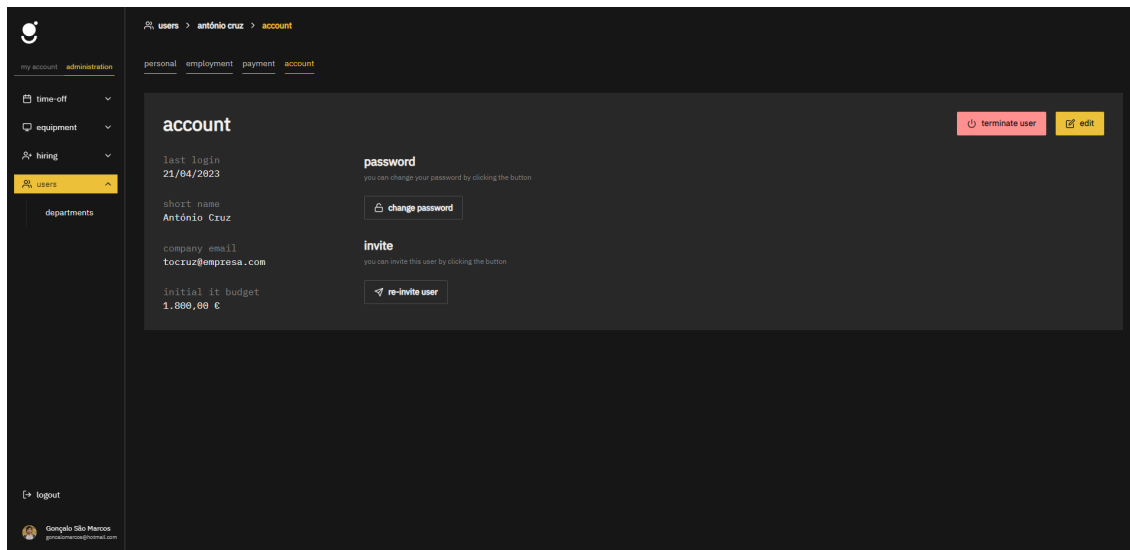


Figura E.5: Vista Detalhada do Perfil de um Utilizador - Separador da Informação da Conta

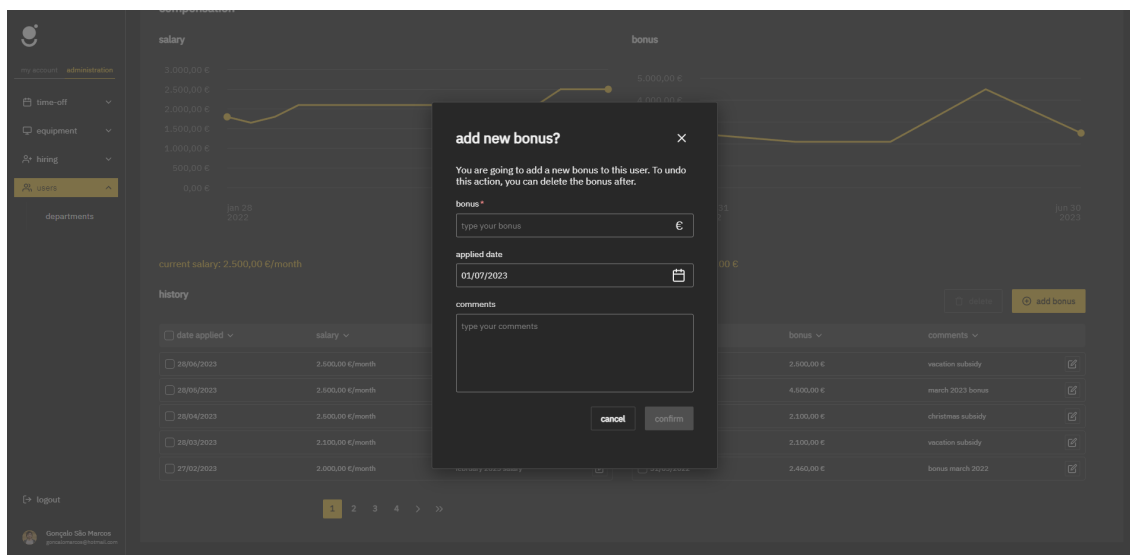


Figura E.6: Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual - Modal de Adicionar Bónus

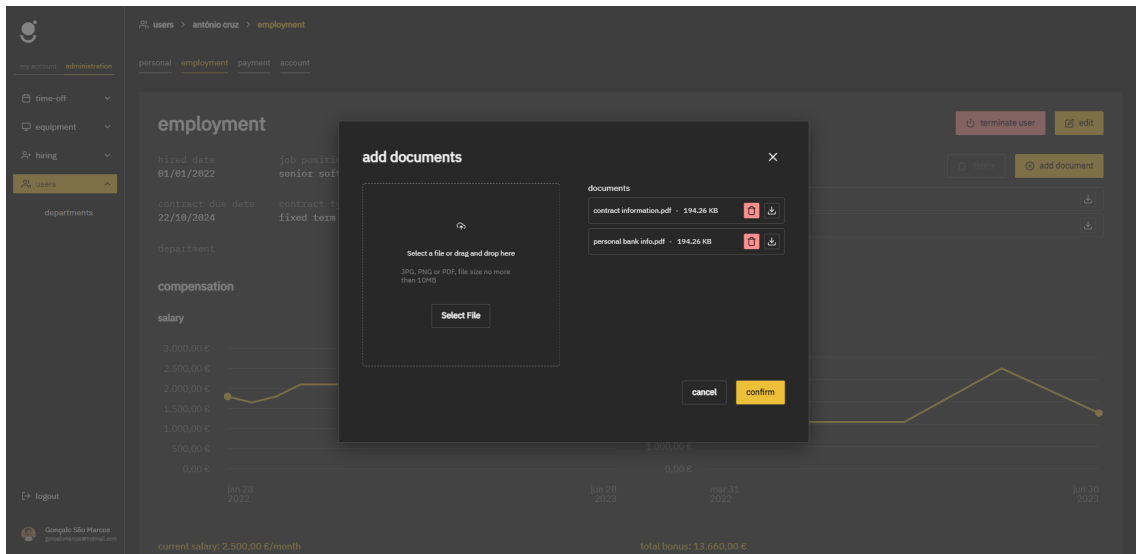


Figura E.7: Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual - *Modal* de Adicionar Ficheiros

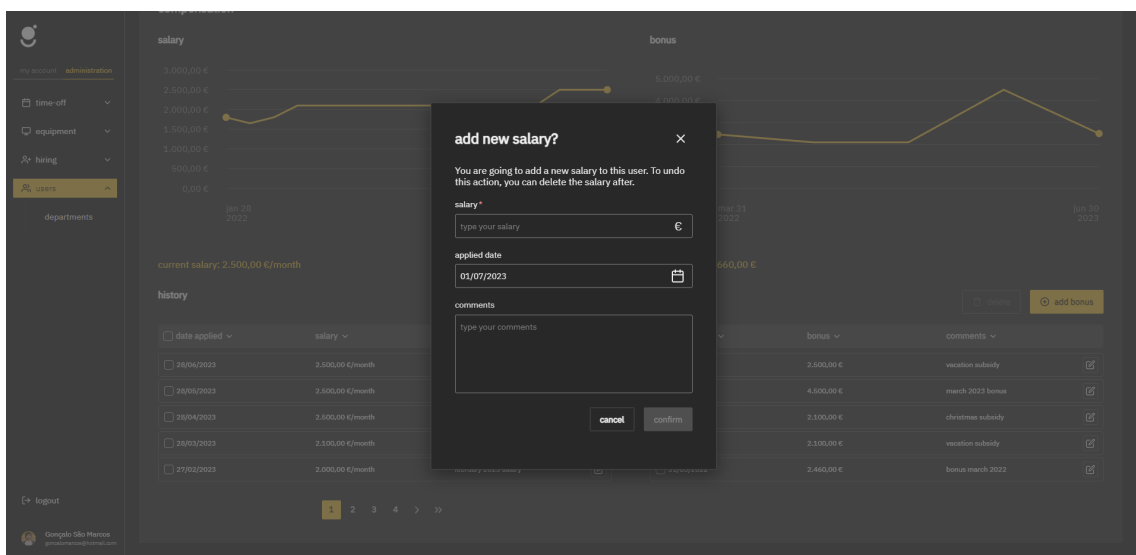


Figura E.8: Vista Detalhada do Perfil de um Utilizador - Separador da Informação Contratual - *Modal* de Adicionar Salário

The screenshot shows a web application interface for adding a new user. The main content area is titled "add new user" and has four tabs: "personal", "employment", "payment", and "account". The "employment" tab is currently selected. The form contains the following fields:

- hired date:** A date input field with the value "29/06/2023" and a calendar icon.
- contract due date:** A date input field with the placeholder text "your contract due date" and a calendar icon.
- department:** A dropdown menu with the placeholder text "choose a department".
- job position:** A dropdown menu with the placeholder text "choose a job position".
- contract type:** A dropdown menu with the placeholder text "choose a contract type".
- salary:** A text input field with the value "0".

At the bottom right of the form, there are three buttons: "cancel" (in red), "save", and "save and invite user".

Figura E.9: Adicionar Novo Utilizador - Separador de Informação Contratual

The screenshot shows the same "add new user" form, but with the "payment" tab selected. The form contains the following fields:

- iban:** A text input field with the placeholder text "your iban here".
- meal card number:** A text input field with the placeholder text "your meal card number here".
- type of twelfths:** A dropdown menu with the placeholder text "choose a type of twelfths".

At the bottom right of the form, there are three buttons: "cancel" (in red), "save", and "save and invite user".

Figura E.10: Adicionar Novo Utilizador - Separador de Pagamento

add new user

personal employment payment account

short name *

your short name here

role *

user

company email *

your company email here

initial it budget *

1700 €

cancel save save and invite user

logout

Gongalo São Marcos
gongalo.sao.marcos@pessoal.com

Figura E.11: Adicionar Novo Utilizador - Separador de Informação da Conta

personal

cancel save

full name

António José Alves Cruz

qualifications

Bachelor's in Computer Science

number of dependents

0

birth date

07/07/1994

cc

98765432

emergency contact name

Aldora Manuela

country

PT

phone number

917654321

nif

798546132

emergency phone number

PT

9287654321

personal email

tocruz@pessoal.com

ssn

84579632145

address

rua direita nº9

marital status

single

user profile photo

upload new photo delete

add your new photo, the file type should be .png or .jpg, the recommended size is 256 x 256px

logout

Gongalo São Marcos
gongalo.sao.marcos@pessoal.com

Figura E.12: Vista Detalhada do Perfil de um Utilizador - Modo de Edição do Separador da Informação Pessoal

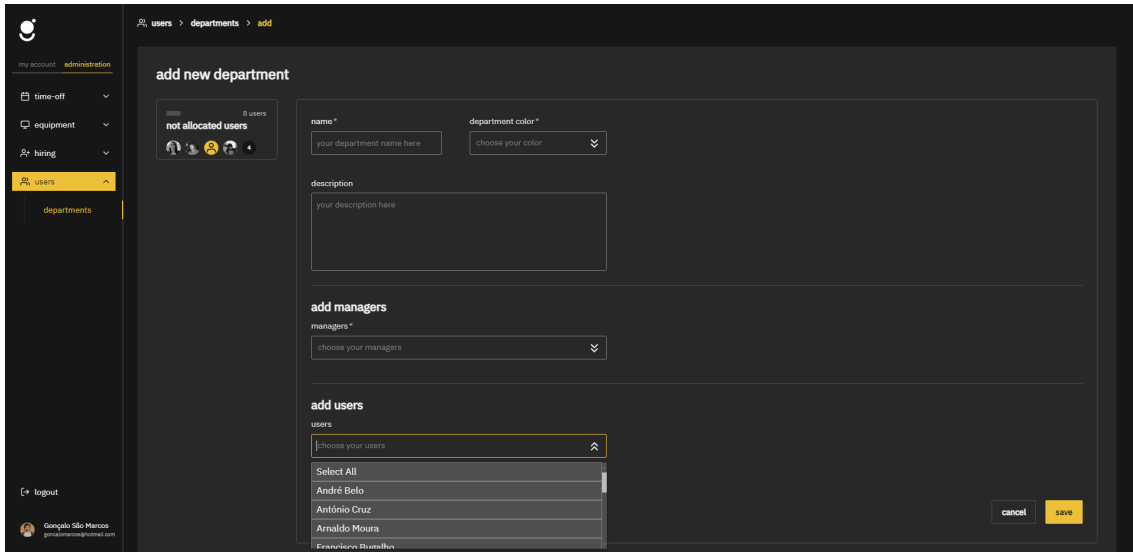


Figura E.13: Adicionar Novo Departamento

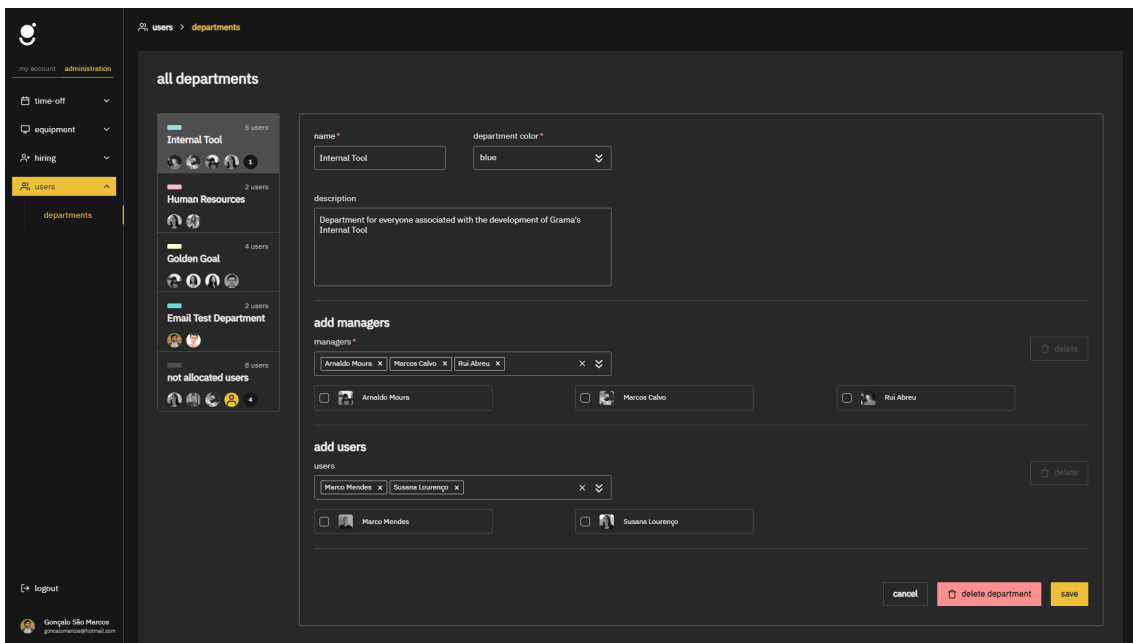


Figura E.14: Editar Departamento

Módulo de Gestão de Equipamentos

equipment > inactive

equipments

active inactive obsolete

code	type	model	provider	cost	purchased on	acquired by	allocated to	status
GRAMA-PERIPHERAL-0013	peripheral	WH10000M4	FNAC	400,00 €	05/06/2023	Araldo Moura	Gramma	inactive
GRAMA-PERIPHERAL-0015	peripheral	DeathStalker V2	Worten	200,00 €	05/06/2023	André Belo	unallocated	unallocated

page 1 of 1

10 rows

Gonçalo Silo Marcos
goncalosilomarcos@hotmail.com

Figura E.15: Lista de Equipamentos Inativos

equipment > obsolete

equipments

active inactive obsolete

code	type	model	provider	cost	purchased on	acquired by	obsolete since
GRAMA-PERIPHERAL-0012	peripheral	Cloud 2	Worten	100,00 €	05/06/2023	Margarida Vasconcelos	06/06/2023

page 1 of 1

10 rows

Gonçalo Silo Marcos
goncalosilomarcos@hotmail.com

Figura E.16: Lista de Equipamentos Obsoletos

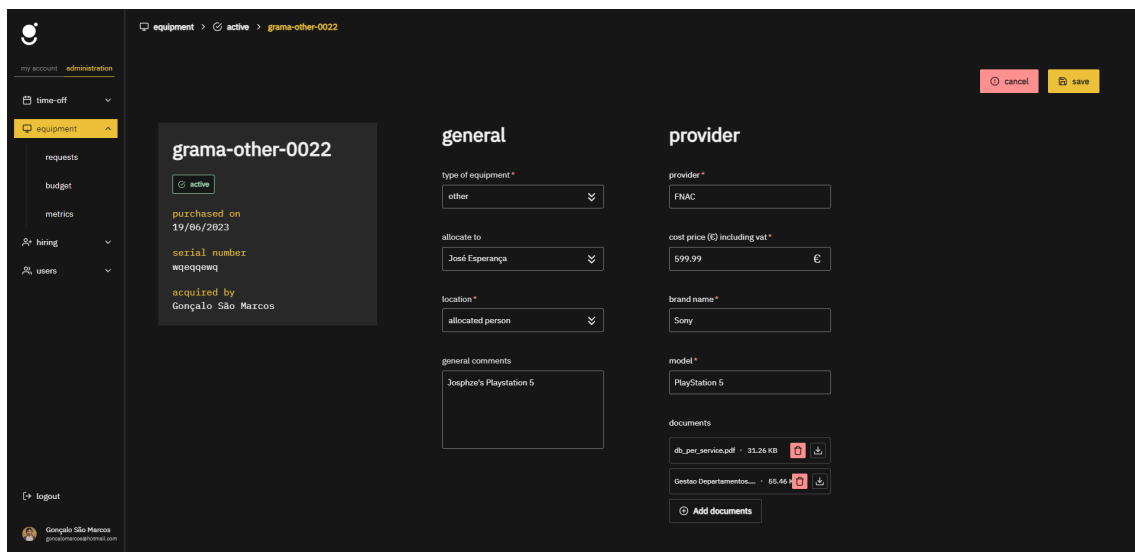


Figura E.17: Vista Detalhada de um Equipamento - Modo de Edição

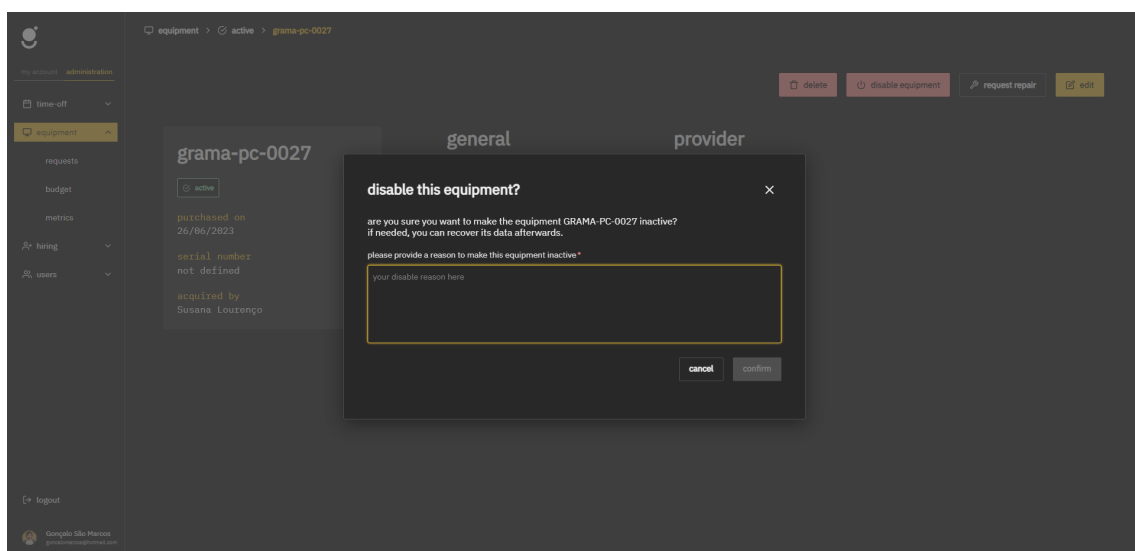


Figura E.18: Desativar um Equipamento

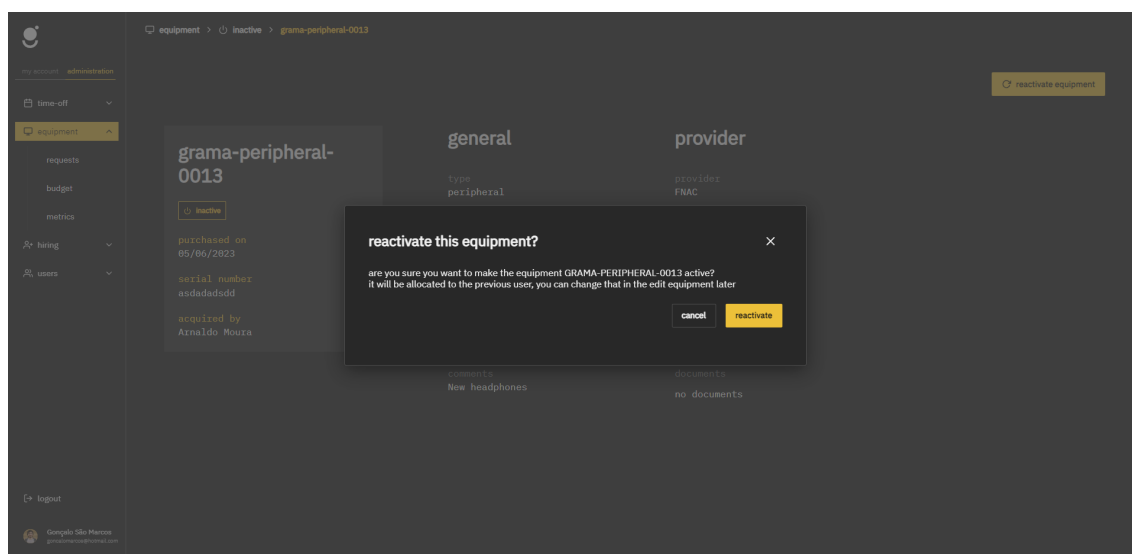


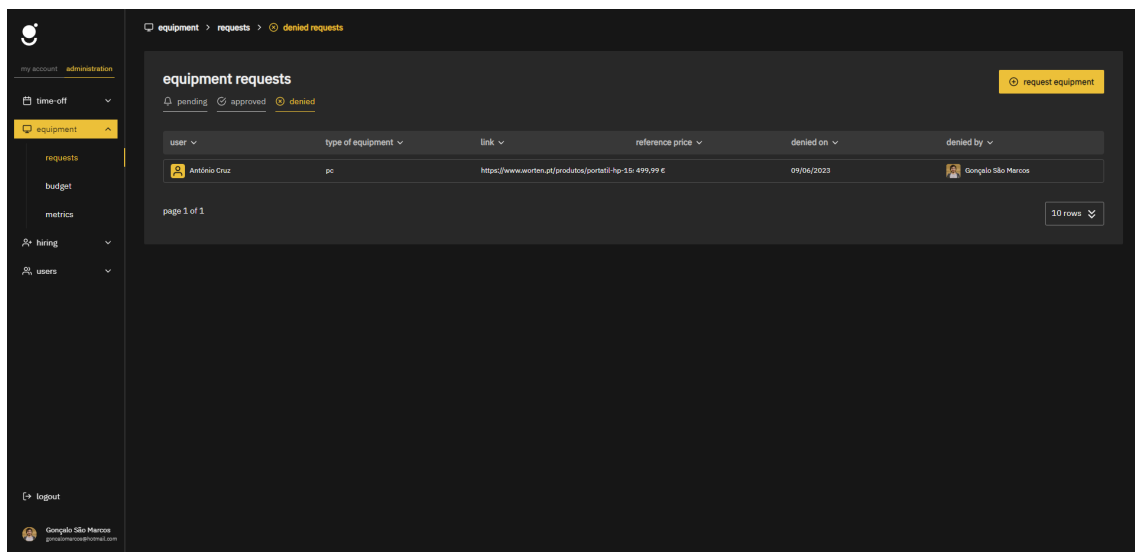
Figura E.19: Reativar um Equipamento

The screenshot shows the 'equipment requests' page, specifically the 'approved requests' tab. The page displays a table of approved requests with the following data:

user	type of equipment	link	reference price	approved on	approved by
Francisco Bugalho	peripheral	https://www.worten.pt/produto/novo-logitech-4	99,99 €	08/06/2023	Gongalo Sfo Marcos
Francisco Bugalho	screen	https://www.worten.pt/produto/monitor-40c-2	99,99 €	20/06/2023	Gongalo Sfo Marcos
Francisco Bugalho	pc	https://www.worten.pt/produto/macbook-air-4	979,99 €	20/06/2023	Gongalo Sfo Marcos

The page also includes a sidebar with navigation options like 'my account', 'administration', 'time-off', 'equipment', 'requests', 'budget', 'metrics', 'hiring', and 'users'. The 'requests' section is currently selected. At the bottom of the table, it shows 'page 1 of 1' and '10 rows'.

Figura E.20: Lista de Pedidos de Equipamentos Aprovados

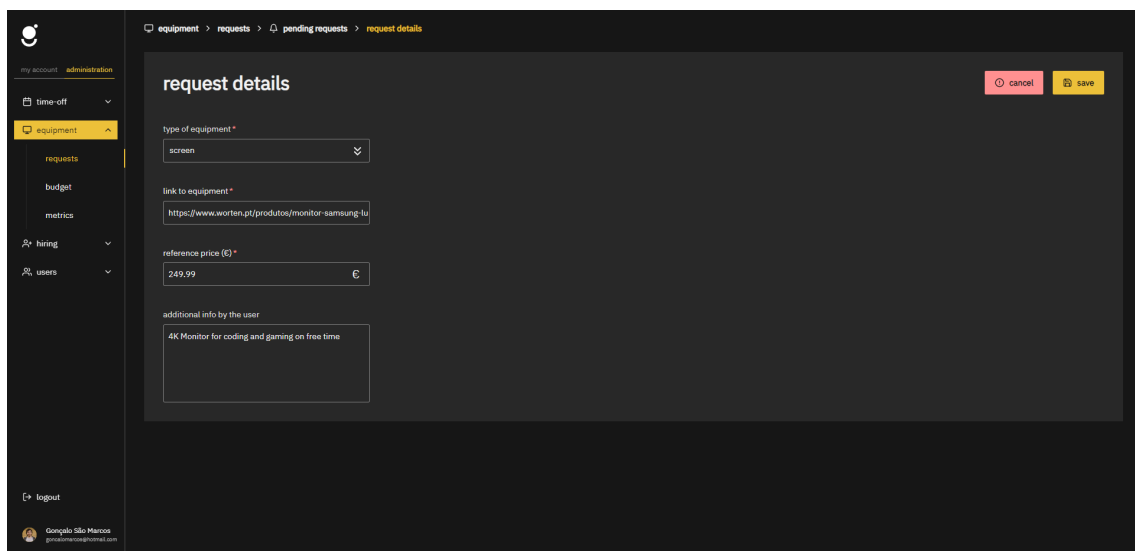


The screenshot shows a web application interface for managing equipment requests. The main content area is titled 'equipment requests' and has a sub-header 'denied requests'. A table lists the details of a denied request:

user	type of equipment	link	reference price	denied on	denied by
António Cruz	pc	https://www.worten.pt/producao/portal-hp-16	499,99 €	09/04/2023	Gonçalo Silo Marcos

Below the table, it indicates 'page 1 of 1' and a '10 rows' dropdown menu. A 'request equipment' button is visible in the top right corner.

Figura E.21: Lista de Pedidos de Equipamentos Recusados



The screenshot shows the 'request details' page in edit mode. The form contains the following fields:

- type of equipment ***: dropdown menu with 'screen' selected.
- link to equipment ***: text input field containing 'https://www.worten.pt/producao/monitor-samsung-lu'.
- reference price (€) ***: text input field containing '249,99' and a currency symbol '€'.
- additional info by the user**: text area containing '4K Monitor for coding and gaming on free time'.

At the top right of the form, there are 'cancel' and 'save' buttons. The breadcrumb navigation shows 'equipment > requests > pending requests > request details'.

Figura E.22: Vista Detalhada - Pedido de Equipamento Por Aprovar - Modo de Edição

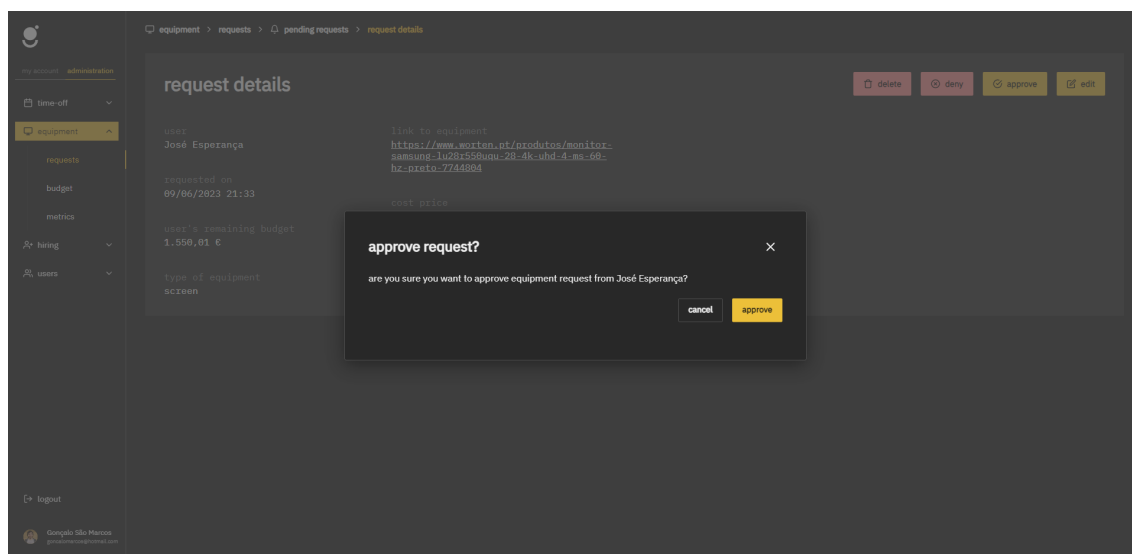


Figura E.23: Vista Detalhada - Aprovar Pedido de Equipamento

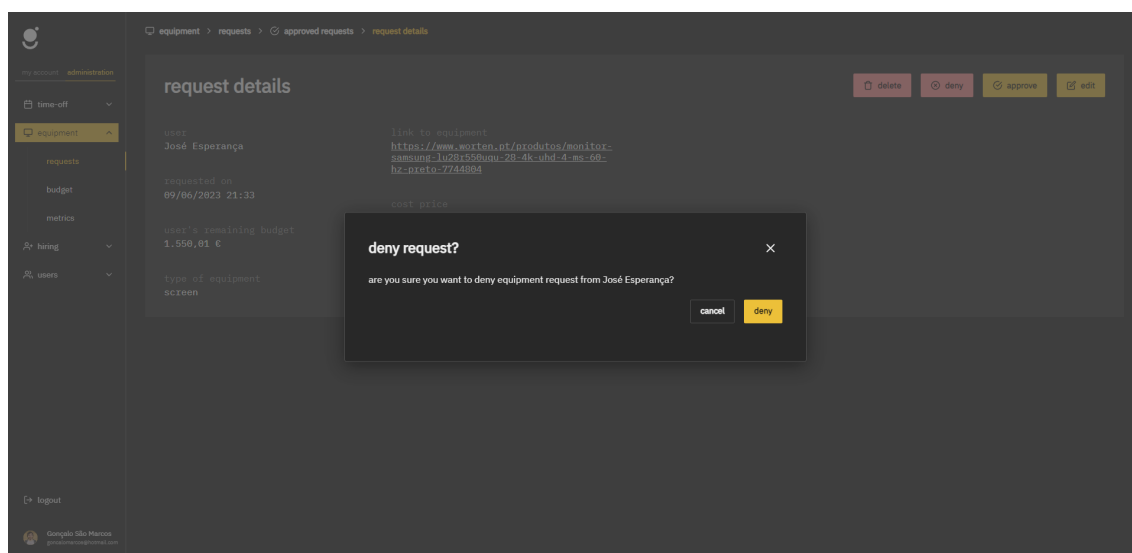


Figura E.24: Vista Detalhada - Recusar Pedido de Equipamento

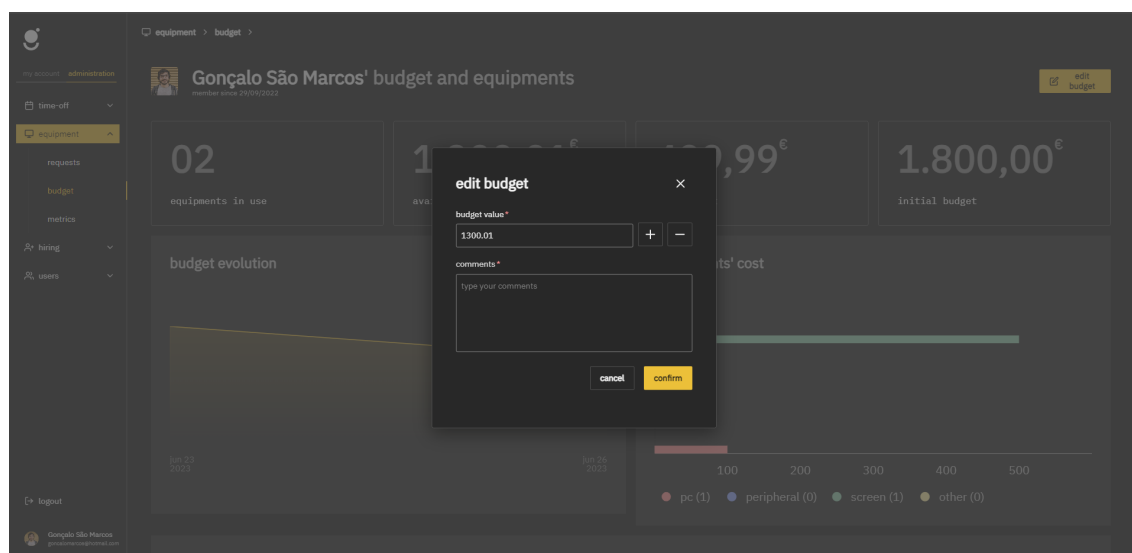


Figura E.25: Edição do Orçamento de um Utilizador

Apêndice F

Testes Funcionais

Tabela F.1: Testes Realizados - Módulo de Gestão de Colaboradores

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
5	5	Visualizar lista de utilizadores com conta ativa	- Aceder à página "users"	Listagem dos utilizadores da plataforma conforme <i>role</i> do utilizador com sessão iniciada (administrador (a) ou gestor (b))	Listagem dos utilizadores da plataforma conforme <i>role</i> do utilizador com sessão iniciada (administrador (a) ou gestor (b))	Passou
6	6	Visualizar lista de utilizadores com conta desativada	- Aceder à página "users" - Clicar no separador "terminated users"	Listagem dos utilizadores da plataforma com conta desativada	Listagem dos utilizadores da plataforma com conta desativada	Passou
7	-	Editar informação de utilizador (es) através da lista	Não desenvolvida			
8	-	Filtrar lista de utilizadores	Não desenvolvida			
9	7	Aceder à vista detalhada de um utilizador	- Aceder à página "users" - Clicar num dos utilizadores da lista	Utilizador é redirecionado para a vista detalhada do utilizador que clicou	Utilizador foi redirecionado para a vista detalhada do utilizador que clicou	Passou
10, 11, 12, 25, 26 e 27	8	Visualizar perfil completo de um utilizador	- Aceder à página "users" - Clicar num dos utilizadores da lista - Ser redirecionado para a vista detalhada do perfil de um utilizador	Utilizador consegue ver toda a informação detalhada nas US-10, 11 e 12, organizada em separadores conforme descrito na mesma (informação pessoal, contratual, de pagamento, da conta)	Utilizador vê toda a informação detalhada na US-10, 11 e 12, organizada em separadores conforme descrito na mesma (informação pessoal, contratual, de pagamento, da conta)	Passou
13	9	Editar informações pessoais	- Aceder à página "users" - Clicar num dos utilizadores da lista - Ser redirecionado para a vista detalhada do perfil de um utilizador - Clicar no botão de "edit" - Editar os campos, respeitando a obrigatoriedade de alguns dos mesmos	Utilizador tem a possibilidade de editar os campos (conforme a US-13) que pretende. Informações são editadas	Utilizador teve a possibilidade de editar os campos (conforme a US-13) que pretendia. Informações foram editadas	Passou
14	10	Adicionar salários	- Aceder à página "users" - Clicar num dos utilizadores da lista - Ser redirecionado para a vista detalhada do perfil de um utilizador - Clicar no separador "employment" - Clicar no botão "add salary"	Utilizador pode criar um novo salário definindo os campos apresentados na US-14	Utilizador conseguiu criar um novo salário definindo os campos apresentados na US-14	Passou
15	11	Editar salários	- Aceder à página "users" - Clicar num dos utilizadores da lista - Ser redirecionado para a vista detalhada do perfil de um utilizador - Clicar no separador "employment" - A partir da lista de salários, clicar no botão de "edit" de uma das entradas da lista - Editar os campos, respeitando a obrigatoriedade de alguns dos mesmos	Utilizador tem a possibilidade de editar os campos (conforme a US-15) do salário selecionado. Informações são editadas	Utilizador teve a possibilidade de editar os campos (conforme a US-15) do salário selecionado. Informações foram editadas	Passou
16	12	Remover salários	- Aceder à página "users" - Clicar num dos utilizadores da lista - Ser redirecionado para a vista detalhada do perfil de um utilizador - Clicar no separador "employment" - A partir da lista de salários, clicar no botão de "edit" de uma das entradas da lista - Clicar na(s) "checkbox(es)" do(s) salário(s) que se pretende(m) remover. - Clicar no botão de "remove" para remover o(s) salário(s) selecionado(s) - Clicar em "confirm"	Salário(s) é(são) removido(s) do histórico	Salário(s) foi(foram) removido(s) do histórico	Passou

Tabela F.2: Testes Realizados - Módulo de Gestão de Equipamentos

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
46	-	Visualizar informação agregada num <i>dashboard</i>		Não desenvolvida		
47	35	Visualizar listas de equipamentos ativos/inativos/obsoletos	- Aceder à página "equipments" - Selecionar entre os três separadores "active", "inactive" e "obsolete"	É apresentada a listagem dos equipamentos conforme o separador selecionado	Foi apresentada a listagem dos equipamentos conforme o separador selecionado	Passou
48	-	Filtrar listas de equipamentos		Não desenvolvida		
49	-	Editar informação de equipamento(s) através da lista		Não desenvolvida		
50	36	Adicionar um novo equipamento à plataforma	- Aceder à página "equipments" - Clicar em "add equipment"	Utilizador pode preencher as informações detalhadas na US-50. O novo equipamento é criado	Utilizador preencheu as informações detalhadas na US-50. O novo equipamento foi criado	Passou
51, 57	37	Visualizar todas as informações de um equipamento	- Aceder à página "equipments" - Clicar num dos equipamentos da lista - Ser redirecionado para a vista detalhada do equipamento	Utilizador consegue ver toda a informação detalhada na US-51	Utilizador conseguiu ver toda a informação detalhada na US-51	Passou
52	38	Adicionar documentos a um equipamento	- Aceder à página "equipments" - Clicar num dos equipamentos da lista - Ser redirecionado para a vista detalhada do equipamento - Clicar no botão "edit" - Clicar no botão de adicionar documento - Selecionar documentos a adicionar	Os novos documentos são adicionados ao equipamento	Os novos documentos foram adicionados ao equipamento	Passou
53, 58	39	Descarregar documentos associados a um equipamento	- Aceder à página "equipments" - Clicar num dos equipamentos da lista - Ser redirecionado para a vista detalhada do equipamento - Clicar no botão de descarregar documento	Documentos são apresentados. Ao carregar no botão de transferência, o documento é transferido e guardado no dispositivo	Documentos foram apresentados. Ao carregar no botão de transferência, o documento foi transferido e guardado no dispositivo	Passou
54	40	Editar informação de um equipamento	- Aceder à página "equipments" - Clicar num dos equipamentos da lista - Ser redirecionado para a vista detalhada do equipamento - Clicar no botão "edit" - Editar os campos, respeitando a obrigatoriedade de alguns dos mesmos	Utilizador tem a possibilidade de editar os campos (conforme a US-54) do equipamento. Informações são editadas	Utilizador teve a possibilidade de editar os campos (conforme a US-54) do equipamento. Informações foram editadas	Passou
55	41	Desativar um equipamento	- Aceder à página "equipments" - Clicar num dos equipamentos da lista - Ser redirecionado para a vista detalhada do equipamento - Clicar no botão "disable equipment"	O equipamento passa a ter o estado "inativo"	O equipamento passou a ter o estado "inativo"	Passou
56	42	Editar a que colaborador está alocado um equipamento	- Aceder à página "equipments" - Clicar num dos equipamentos da lista - Ser redirecionado para a vista detalhada do equipamento - Clicar no botão "edit" - Editar o campo "allocated to"	Informação é editada	Informação foi editada	Passou

Apêndice G

Testes aos Requisitos Não Funcionais

Tabela G.1: Teste de Indisponibilidade da Plataforma Durante Atualização de Microserviço

	Tempo de <i>Reboot</i> da Máquina EC2	Tempo de inicialização do microserviço	Total
Tempo (mm:ss)	1:04	0:27	1:31
	1:03	0:27	1:30
	1:00	0:27	1:27
	0:55	0:26	1:21
	1:01	0:27	1:28
	1:06	0:26	1:32
	1:00	0:27	1:27
	0:58	0:28	1:26
	1:03	0:27	1:30
1:06	0:27	1:33	
Média	1:01.6	0:26.9	1:28.5

Tabela G.2: Testes de Segurança - Utilizador Não Autorizado (Cont.)

ID	Cenário	Resultado Esperado	Resultado Obtido	Conclusões
11	Colaborador tenta aceder a um equipamento que não está alocado a ele próprio	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
12	Gestor tenta aceder a um equipamento que não está alocado a ele próprio nem a nenhum dos utilizadores que fazem parte dos seus departamentos	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
13	Administrador tenta aceder a qualquer equipamento	Página é mostrada corretamente	Página é mostrada corretamente	Passou
14	Colaborador tenta aceder a um pedido de equipamento que não foi ele próprio a realizar	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
15	Gestor tenta aceder a um pedido de equipamento que não foi ele próprio a realizar nem nenhum dos utilizadores que fazem parte dos seus departamentos	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
16	Administrador tenta aceder a qualquer pedido equipamento	Página é mostrada corretamente	Página é mostrada corretamente	Passou
17	Colaborador tenta aceder ao orçamento de outro utilizador	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
18	Gestor tenta aceder a um orçamento que não é dele próprio nem de nenhum dos utilizadores que fazem parte dos seus departamentos	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Toast de erro é mostrado. <i>Backend</i> retorna erro 404	Passou
19	Administrador tenta aceder a qualquer orçamento	Página é mostrada corretamente	Página é mostrada corretamente	Passou