



UNIVERSIDADE D
COIMBRA

José Francisco de Jesus Bugalho

PORTAL DE GESTÃO DE RECURSOS
HUMANOS

Dissertação no âmbito do Mestrado em Engenharia Informática,
especialização em Engenharia de Software, orientada pela Professora
Doutora Naghmeh Ramezani Ivaki e pelo Engenheiro Marcos Calvo e
apresentada ao Departamento de Engenharia Informática da Faculdade
de Ciências e Tecnologia da Universidade de Coimbra.

Julho de 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

José Francisco de Jesus Bugalho

Human Resources Management Portal

Time-off and Recruitment modules

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Prof. Naghmeh Ramezani Ivaki and Eng. Marcos Calvo and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

José Francisco de Jesus Bugalho

Portal de Gestão de Recursos Humanos

Módulo de Ausências e Recrutamento

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software, orientada pela Professora Doutora Naghmeh Ramezani Ivaki e pelo Engenheiro Marcos Calvo e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Julho de 2023

Agradecimentos

Em primeiro lugar quero agradecer aos meus orientadores de estágio Marcos Calvo e Naghmeh Ivaki por toda a ajuda e apoio que me forneceram ao longo da realização do estágio. Quero também agradecer à Susana Lourenço e a todos os elementos da Grama pela forma como me acolheram, pela ajuda e disponibilidade prestada sempre que necessário. Em adição, uma nota de agradecimento aos meus dois colegas estagiários Gonçalo São Marcos e José Esperança que, desde o primeiro momento, se mostraram disponíveis para debater e auxiliar sempre que havia necessidade.

De seguida, quero agradecer à minha família por todo o apoio incondicional ao longo dos anos, em especial aos meus pais que tiveram um papel muito importante em todo o meu percurso académico.

Por fim, mas não menos importante, um agradecimento a todos os meus amigos que para além de excelentes colegas, me acompanharam ao longo deste percurso.

Abstract

This internship aims to add two modules to the Grama Internal Tool addressing the following needs: absence of employees and recruitment management.

Time-off and recruitment management have significant impacts on the field of human resource management within a company. A well-designed time-off policy provides employees with opportunities for rest, relaxation, and personal fulfillment. Furthermore, a thorough time-off policy shows the company's commitment to work-life balance and employee well-being, fostering a positive and supportive work environment. The recruitment management is of utmost importance for a company as it ensures the process of evaluating and selecting qualified candidates while building a skilled and competent workforce, essential for the company's success and growth.

The time-off module provides an efficient solution for managing and tracking employee time-off requests and balances. With this module, the company may eliminate the need of a third-party software and focus all the information in one place. Through a user-friendly interface, employees can conveniently request time-offs, view their available balances, and track their time-off history. Streamlined approval workflows empower administrators to promptly review and respond to requests, while also offering the flexibility to configure and customize various time-off policies to align with company guidelines.

The recruitment module focuses the information in a centralized solution that enhances the managing and tracking candidate applications, interviews, and hiring decisions, eliminating the issue of scattered information among different platforms. The primary goal of this module is to optimize the recruitment process by providing the human resources department with a customized method for displaying candidates throughout the various phases of the recruitment process. With these features, the recruitment management module enables the company to efficiently evaluate and select the most suitable candidates.

Keywords

Absences Management, Recruitment Management, Amazon Web Services, Microservices

Resumo

O presente estágio tem como objetivo adicionar os seguintes dois módulos à Plataforma Interna da Grama: gestão de ausências e gestão de recrutamento.

A gestão de ausências e de recrutamento são áreas com um impacto significativo na gestão de recursos humanos de uma empresa. Uma política de ausências bem trabalhada oferece aos colaboradores oportunidades de descanso e realização pessoal. Além disso, demonstra o compromisso da empresa entre vida profissional e pessoal dos colaboradores e o bem-estar dos mesmos, promovendo um ambiente de trabalho positivo e cooperativo. A gestão do recrutamento é também de muita importância para uma empresa, pois implica o processo de avaliação e seleção de candidatos qualificados e a construção de uma equipa competente, que é um dos pontos essenciais para o sucesso e crescimento de uma empresa.

O módulo de ausências apresenta uma solução eficiente para gerir e controlar os pedidos de ausência dos colaboradores e o seu balanço. Com este módulo, a empresa pode eliminar a necessidade de uma ferramenta externa e concentrar todas as informações apenas numa plataforma. Através de uma interface simples, os colaboradores podem requisitar as suas ausências, visualizar o seu balanço e acompanhar o seu histórico de ausências. Os fluxos de aprovação permitem aos administradores analisar e responder prontamente aos pedidos e, ao mesmo tempo, oferecem flexibilidade para configurar e personalizar as várias políticas de ausência adaptando-as aos princípios da empresa.

O módulo de recrutamento concentra todas as informações numa solução que auxilia na gestão e controlo dos candidatos, das suas entrevistas e observações delas retiradas, eliminando assim o problema da dispersão de informação em diferentes plataformas. O principal objetivo deste módulo é otimizar o processo de recrutamento, garantindo ao departamento de recursos humanos um método personalizado de apresentação de candidatos ao longo das várias fases do processo de recrutamento. Através destas funcionalidades, o módulo de gestão de recrutamento permite que a empresa possa avaliar e selecionar de forma eficiente os candidatos mais adequados.

Palavras-Chave

Gestão de Ausências, Gestão de Recrutamento, Amazon Web Services, Microsserviços

Conteúdo

1	Introdução	1
1.1	Contexto e Motivação	1
1.2	Objetivos	2
1.3	Estrutura	3
2	Background e Estado da Arte	5
2.1	<i>Background</i>	5
2.2	Análise de Concorrentes	7
2.2.1	Concorrentes Diretos	8
2.2.2	Concorrentes Indiretos	12
3	Requisitos	15
3.1	Atores	15
3.2	<i>User Stories</i>	16
3.3	Requisitos Funcionais	16
3.4	Diagrama de Casos de Uso	22
3.5	Requisitos Não Funcionais	28
3.5.1	Robustez	29
3.5.2	<i>Performance</i>	29
3.5.3	Segurança e Privacidade	30
4	Tecnologias	33
4.1	<i>Frontend</i>	33
4.1.1	Angular	34
4.1.2	React	34
4.1.3	Svelte	34
4.1.4	Vue.js	35
4.1.5	Comparação das tecnologias	35
4.1.6	Tomada de decisão	35
4.2	<i>Backend</i>	36
4.2.1	Django	36
4.2.2	Spring Boot	36
4.2.3	Quarkus	37
4.2.4	Express.js	37
4.2.5	Comparação das tecnologias	38
4.2.6	Tomada de decisão	38
5	Arquitetura	39
5.1	Arquitetura Monolítica vs Microsserviços	39

5.2	Arquitetura baseada em Microsserviços	40
5.2.1	Base de dados por serviço	41
5.2.2	Base de dados partilhada	42
5.2.3	API Gateway	43
5.2.4	Token de acesso	43
5.3	Cloud computing	44
5.3.1	Amazon Web Services	44
5.3.2	Microsoft Azure	45
5.3.3	Google Cloud	45
5.3.4	Comparação entre as plataformas	45
5.3.5	Tomada de decisão	47
5.4	Desenho da arquitetura	47
5.4.1	Amazon Web Services	47
5.4.2	Definição da arquitetura	48
5.5	Diagrama Entidade-Relacionamento	56
6	Metodologia e Planeamento	59
6.1	Metodologia	59
6.2	Planeamento	62
6.2.1	1º Semestre	62
6.2.2	2º Semestre	67
6.3	Riscos	72
6.3.1	Síntese dos riscos	75
6.3.2	Matriz de Riscos	76
7	Desenvolvimento	77
7.1	Processo de desenvolvimento	77
7.2	Repriorização dos requisitos	79
7.3	Requisitos funcionais desenvolvidos	80
7.4	Principais funcionalidades	84
7.5	Análises dos riscos levantados	94
7.6	Trabalho Futuro	94
8	Testagem	97
8.1	Testes funcionais	97
8.2	Testes não funcionais	99
9	Conclusão	107
	Apêndice A Epic e User stories	119
	Apêndice B Funcionalidades Desenvolvidas	149
	Apêndice C Plano de Testes de Integração	159
	Apêndice D Plano de Testes de Aceitação	173

Acrónimos

ACID *Atomicity, Consistency, Isolation e Durability.*

API *Application Programming Interface.*

AWS *Amazon Web Services.*

CORS *Cross-Origin Resource Sharing.*

CQRS *Command Query Responsibility Segregation.*

CRUD *Create, Read, Update e Delete.*

DOM *Document Object Model.*

EC2 *Amazon Elastic Compute Cloud.*

ELB *Elastic Load Balancing.*

ER *Entidade-Relacionamento.*

ES *Epic Story.*

HTTP *Hypertext Transfer Protocol.*

ID *Identificador.*

MVP *Minimum Viable Product.*

MVT *Model View Template.*

ORM *Object Relational Mapper.*

PII *Personally Identifiable Information.*

RDS *Amazon Relational Database Service.*

RF *Requisito Funcional.*

RGPD *Regulamento Geral sobre a Proteção de Dados.*

RNF *Requisito Não Funcional.*

S3 *Amazon Simple Storage Service.*

SES *Amazon Simple Email Service.*

UML *Unified Modeling Language.*

US *User Story.*

Lista de Figuras

2.1	Exemplo do calendário na ferramenta PurelyHR [3].	6
2.2	Exemplo do ficheiro da <i>sourcing list</i>	6
2.3	Exemplo do ficheiro da <i>pipeline</i> de recrutamento.	7
2.4	Exemplo de um <i>template</i> de marcação de ausências [27].	13
2.5	Exemplo de um <i>template</i> de gestão de recrutamento [28].	13
3.1	Diagrama de contexto.	22
3.2	Diagrama de casos de uso o módulo de autenticação.	23
3.3	Diagrama de casos de uso para o administrador no módulo de ausências.	24
3.4	Diagrama de casos de uso para o gestor no módulo de ausências.	25
3.5	Diagrama de casos de uso para o colaborador no módulo de ausências.	26
3.6	Diagrama de casos de uso para o administrador no módulo de recrutamento.	27
3.7	Diagrama de casos de uso para o gestor no módulo de recrutamento.	28
3.8	Cenário do requisito não funcional robustez.	29
3.9	Exemplos de tempos de carregamento (adaptado de [35]).	29
3.10	Cenário do requisito não funcional <i>performance</i>	30
3.11	Cenário do requisito não funcional segurança, representado o acesso a uma página por um utilizador não autenticado.	30
3.12	Cenário do requisito não funcional segurança, representado o acesso a uma página por um utilizador não autorizado.	31
4.1	Tendências no <i>Stack Overflow</i> [37].	34
5.1	Representação esquemática do padrão base de dados por serviço.	42
5.2	Representação esquemática do padrão base de dados partilhada.	42
5.3	Representação esquemática do padrão <i>API Gateway</i>	43
5.4	Representação esquemática do padrão <i>token</i> de acesso.	43
5.5	Crescimento no mercado dos provedores de serviços na nuvem ao longo do tempo [85].	46
5.6	Exemplo do modelo C4 [93].	48
5.7	Diagrama de contexto da Plataforma Interna da Grama.	49
5.8	Diagrama de <i>containers</i> da Plataforma Interna da Grama.	51
5.9	Diagrama de componentes relativo à <i>API Gateway</i>	52
5.10	Diagrama de componentes relativo ao microserviço de ausências.	53
5.11	Diagrama de componentes relativo ao microserviço de recrutamento.	54

5.12	Diagrama de componentes relativo ao <i>frontend</i> da aplicação.	55
5.13	Diagrama ER relativo ao módulo de ausências.	56
5.14	Diagrama ER relativo ao módulo de recrutamento.	57
6.1	Metodologia ágil, <i>Scrum</i> [96].	62
6.2	Planeamento de temporal do primeiro semestre.	64
6.3	Diagrama de <i>Gantt</i> das tarefas concluídas ao longo do primeiro semestre.	66
6.4	Planeamento de temporal do segundo semestre.	68
6.5	Diagrama de <i>Gantt</i> que representa o planeamento real do segundo semestre.	70
6.6	Tarefas realizadas durante os <i>sprints</i> no quadro de tarefas comuns (esquerda) e no quadro de tarefas dos microserviços de ausências e recrutamento (direita).	71
6.7	Matriz de riscos.	76
7.1	Exemplo da organização das tarefas na ferramenta Linear.	79
7.2	Página de início de sessão.	84
7.3	Página de primeiro início de sessão.	85
7.4	Exemplo de um <i>email</i> de convite para a plataforma.	85
7.5	Página de vista de um mês em calendário (para administradores).	86
7.6	Página de vista de um mês em calendário (para gestores).	86
7.7	Página de vista de um mês em calendário (para colaboradores).	87
7.8	Página de requisição de uma ausência.	88
7.9	Página de requisição de uma ausência (para administradores).	88
7.10	Exemplo de um <i>email</i> de um pedido de ausência.	89
7.11	Página de métricas sobre as ausências (para administradores).	90
7.12	Página da <i>pipeline</i> de recrutamento.	91
7.13	Modal de adição de um candidato a uma coluna da <i>pipeline</i>	91
7.14	Lista com todos os candidatos não arquivados.	92
7.15	Detalhes gerais de um candidato.	93
7.16	Interações de um candidato.	93
8.1	Exemplo do <i>endpoint</i> do <i>Swagger</i> para o microserviço de ausências.	98
8.2	Exemplo dos alertas no OWASP ZAP [103].	104
8.3	Vulnerabilidades encontradas após o teste com a ferramenta OWASP ZAP.	104
B.1	Ecrã de recuperação da palavra-passe.	149
B.2	Modal de término de sessão.	150
B.3	Vista de uma semana em calendário.	150
B.4	Vista de um dia em calendário.	151
B.5	Lista dos pedidos de ausência.	151
B.6	Detalhes de um pedido de ausência.	152
B.7	Modal para aprovar um pedido de ausência (igual para cancelar e apagar).	152
B.8	Modo de edição de um pedido de ausência.	153
B.9	Lista do balanço dos utilizadores.	153
B.10	Detalhes do balanço de um utilizador.	154

B.11	Modo de edição do balanço de um utilizador.	154
B.12	Modo de edição da <i>pipeline</i> de recrutamento.	155
B.13	Modal de filtros a aplicar na <i>pipeline</i> de recrutamento.	155
B.14	Adição de um candidato com todos os detalhes.	156
B.15	Lista os candidatos arquivados.	156
B.16	Edição dos detalhes gerais de um candidato.	157
B.17	Edição das interações de um candidato.	157
B.18	Modal para arquivar um candidato (igual para recuperá-lo).	158

Lista de Tabelas

2.1	Tabela de comparação dos concorrentes diretos.	12
2.2	Vantagens e desvantagens da ferramenta Excel [29].	14
3.1	Requisitos funcionais para o módulo de autenticação.	17
3.2	Requisitos funcionais para o módulo de ausências.	18
3.3	Requisitos funcionais para o módulo de recrutamento.	20
4.1	Tabela comparativa das tecnologias analisadas para o <i>frontend</i>	35
4.2	Tabela comparativa das tecnologias analisadas para o <i>backend</i>	38
5.1	Tabela comparativa dos provedores de serviços na nuvem.	46
6.1	Tabela sumária dos riscos levantados.	75
7.1	Requisitos funcionais desenvolvidos para o módulo de autenticação.	80
7.2	Requisitos funcionais desenvolvidos para o módulo de ausências.	81
7.3	Requisitos funcionais desenvolvidos para o módulo de recrutamento.	83
8.1	Plano de testes para a área Gerir Quadro de Recrutamento.	99
8.2	Testes de aceitação para a adição e edição de colunas do quadro de recrutamento.	100
8.3	Tempos médios de carregamento das principais páginas nos módulos de ausências e recrutamento.	101
8.4	Testes ao requisito não funcional de segurança, não autenticação.	102
8.5	Testes ao requisito não funcional de segurança, não autorização.	103
C.1	Plano de testes para o módulo de autenticação.	159
C.2	Plano de testes para a área <i>Dashboard</i>	160
C.3	Plano de testes para a área <i>Dashboard</i> para colaboradores.	160
C.4	Plano de testes para a área Gerir Calendário.	161
C.5	Plano de testes para a área Gerir Pedidos.	162
C.6	Plano de testes para a área Gerir Pedidos para colaboradores.	163
C.7	Plano de testes para a área Gerir Ausências.	163
C.8	Plano de testes para a área Requisitar Ausências.	164
C.9	Plano de testes para a área Requisitar Ausências para colaboradores.	165
C.10	Plano de testes para a área Notificar.	165
C.11	Plano de testes para a área Notificar para colaboradores.	166
C.12	Plano de testes para a área Gerir Tipos de Ausências.	166
C.13	Plano de testes para a área Gerir Balanço.	166

C.14 Plano de testes para a área Gerir Eventos Especiais.	167
C.15 Plano de testes para a área Gerir Quadro de Recrutamento.	168
C.16 Plano de testes para a área Customizar Quadro de Recrutamento. . .	169
C.17 Plano de testes para a área Lista de Candidatos.	170
C.18 Plano de testes para a área Analisar Candidatos.	171
D.1 Testes de aceitação para adição e edição de ausências.	174
D.2 Testes de aceitação para a adição e edição de candidatos.	175
D.3 Testes de aceitação para a adição e edição de candidatos (cont.). . .	176
D.4 Testes de aceitação para a adição e edição de interações.	177
D.5 Testes de aceitação para a adição e edição de colunas do quadro de recrutamento.	177

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 na realização de dois módulos para a Plataforma Interna da Grama. Está a ser acompanhado pela Professora Doutora Naghmeh Ivaki, do Departamento de Engenharia Informática e pelo Engenheiro Marcos Calvo, por parte da empresa Grama¹.

1.1 Contexto e Motivação

O estágio curricular decorre na empresa Grama e teve a duração de um ano curricular. A Grama, fundada em 2017 e com sede em Coimbra, é uma empresa de desenvolvimento de *software* que fornece um amplo conjunto de serviços em qualquer fase do ciclo de vida do desenvolvimento do *software*, isto é, desde gestão de projetos, *design* e desenvolvimento de produtos. Conta com uma equipa com vasta experiência no desenvolvimento e entrega de projetos para um largo número de clientes, desde pequenas *startups* até grandes empresas de telecomunicações.

Atualmente, a empresa enfrenta alguns desafios na gestão de ausências, recursos humanos, equipamentos e recrutamento, devido à utilização de diferentes ferramentas como o PurelyHR [1] e ficheiros Excel [2]. Existe uma descentralização de informação cujas consequências podem afetar a integridade dos dados.

A utilização de pastas partilhadas com ficheiros Excel aumenta consideravelmente a probabilidade de ocorrência de erros humanos, como alterações incorretas de células ou a modificação simultânea dos mesmos dados, resultando na potencial perda de informações. Além disso, a necessidade de interagir e editar múltiplas plataformas resulta na perda de tempo e aumenta também o risco de erros humanos devido ao esquecimento de atualizar informações relevantes em todas as plataformas.

¹<https://www.grama.io/>

Para superar os desafios enumerados e modernizar o processo de gestão, a empresa propõe o desenvolvimento de uma solução modular e customizável, a Plataforma Interna da Grama. Esta pretende centralizar a gestão de recursos humanos, ausências, equipamentos e recrutamento, substituindo as múltiplas ferramentas atualmente utilizadas. Pretende reduzir a probabilidade de incongruências e erros na inserção/edição de informação, para além de aumentar a eficiência na gestão de informação.

A necessidade de modernização surge da constatação de que as soluções existentes no mercado são, na sua maioria, direcionadas a grandes empresas e exigem configurações complexas para se adaptarem às necessidades específicas da empresa. A Plataforma Interna da Grama procura preencher essa lacuna através da criação de uma solução personalizável, adequada ao contexto e às necessidades reais da empresa, sem a dependência de várias plataformas distintas e ficheiros Excel dispersos. Poderá ainda ser produtizável e comercializada para empresas com dimensões e realidades semelhantes às da Grama.

A Plataforma Interna da Grama, como foi pensada numa versão inicial, consiste em quatro módulos distintos:

- **Gestão de colaboradores** da empresa;
- **Gestão de equipamentos** alocados aos vários colaboradores e escritórios possibilitando a sua inventariação, requisição e aquisição com a devida aprovação, bem como gestão de avarias e garantias;
- **Gestão de ausências** dos colaboradores oferecendo funcionalidades de ciclo de requisição e aprovação, calendários pessoais, por departamentos ou gerais, e também serviço de notificações;
- **Gestão de recrutamento** de novos colaboradores recorrendo a uma *pipeline* customizável representando as várias fases do processo, com ciclos de *feedback* entre os gestores envolvidos em cada processo.

De realçar que, no futuro, poderão existir outros módulos na plataforma como, por exemplo, módulos de gestão de formações e de partilha de conhecimentos, que poderão complementar esta ferramenta e auxiliar a gestão operacional da empresa.

1.2 Objetivos

Os objetivos do estágio curricular correspondem ao desenvolvimento dos módulos de gestão de ausências e gestão de recrutamento da Plataforma Interna da Grama. Sendo que os restantes dois módulos foram realizados, em paralelo, pelo estagiário Gonçalo São Marcos, também aluno do Mestrado em Engenharia Informática.

O **módulo de ausências** da Plataforma Interna da Grama pretende substituir as funcionalidades atuais cumpridas pelo PurelyHR [1]. Este módulo destaca-se

pela simplicidade de configuração de aspetos como o número de dias de ausência por colaborador ou os tipos de ausências disponíveis e também pelo seu *design* apelativo.

Já o **módulo de gestão de recrutamento** da Plataforma Interna da Grama, pretende agregar todo o conteúdo relacionado ao recrutamento na mesma plataforma. Existindo duas disposições de informação diferentes influenciadas pelo estado de cada candidato, ou seja:

- Uma lista onde são apresentados todos os candidatos;
- A *pipeline* de recrutamento, isto é, uma tabela cujas colunas representam as várias fases do processo de recrutamento e onde os candidatos (com estado "ativo") são distribuídos consoante a fase onde se encontram.

Os estados são definidos pela empresa sendo que qualquer candidato com o estado "ativo" é apresentado na *pipeline* de recrutamento. Isto significa que este poderá ser movido entre colunas à medida que o seu processo de recrutamento avança. Esta *pipeline* é customizável às necessidades e aos processos da empresa em cada momento podendo-se adicionar ou remover colunas.

A título sumário, os **cinco** principais objetivos do estágio curricular são os seguintes

- Gestão de ausências dos colaboradores;
- Notificação e aprovação de ausências;
- Visualização dos dados em *dashboards* e calendários;
- Gestão de recrutamento de novos colaboradores;
- Visualização da *pipeline* dos vários processos em curso.

1.3 Estrutura

O presente documento é dividido em **9 capítulos** distintos organizados do seguinte modo:

- No **capítulo 1** é apresentado o contexto, a motivação e os objetivos da realização do estágio curricular;
- No **capítulo 2** é descrito como a empresa atualmente faz a gestão da informação. De seguida, é analisada e comparada a concorrência direta e indireta, de forma a compreender o mercado atual e onde a plataforma a desenvolver se insere;
- O **capítulo 3** apresenta o levantamento de requisitos da plataforma. Começando pelos requisitos funcionais baseados em *user stories* e respetivos casos de uso. Os requisitos não funcionais são descritos em formato de atributo de qualidade;

- Ao longo do **capítulo 4** são analisadas as tecnologias estudadas para a implementação da plataforma apresentado, no final, qual a tecnologia escolhida para o desenvolvimento do *frontend* e do *backend*;
- No **capítulo 5** é apresentada a decisão do tipo de arquitetura para a realização da plataforma, bem como uma descrição mais aprofundada da mesma. Segue-se o desenho da arquitetura utilizando o modelo C4 e, por fim, diagrama Entidade-Relacionamento (ER) para as bases de dados;
- O **capítulo 6** tem como objetivo apresentar a metodologia a utilizar ao longo do estágio curricular, assim como o planeamento de ambos os semestres. São também descritos os riscos associados à realização do estágio;
- No **capítulo 7** é apresentado o processo de desenvolvimento, ou seja, a organização da equipa e das tarefas ao longo do desenvolvimento da plataforma. Depois, são apresentados os requisitos funcionais desenvolvidos, seguidos de exemplos das principais funcionalidades. Este capítulo termina com a análise aos riscos levantados no primeiro semestre e com algumas notas de trabalho futuro;
- O **capítulo 8** descreve qual foi o processo de realização dos testes para os requisitos funcionais e não funcionais, e algumas conclusões como resultado dos mesmos.
- Por fim, o **capítulo 9** apresenta uma breve conclusão do presente documento.

Capítulo 2

Background e Estado da Arte

Ao longo deste capítulo, será descrito como a empresa Grama efetua a gestão de ausências e de recrutamento e como os módulos a realizar neste estágio pretendem solucionar o problema atual. De seguida, é realizada a análise de plataformas concorrentes e efetuada a sua comparação.

2.1 *Background*

Sabendo que os principais objetivos do presente estágio são a realização de dois módulos para a Plataforma Interna da Grama será necessário analisar quais os métodos atuais utilizados pela empresa para essa gestão. Depois, é apresentado como os módulos pretendem solucionar os problemas atuais de gestão de informação.

Relativamente à gestão das ausências é utilizada uma plataforma externa, o PurelyHR [1] (Figura 2.1). A ferramenta tem como objetivo apresentar num calendário os colaboradores ausentes nos respetivos dias. Deste modo, sempre que um colaborador necessita de tirar um ou vários dias de ausência este deverá aceder a esta plataforma. Após um colaborador requisitar uma ausência, terá de esperar a aprovação de um dos gestores, posto isto, os dias serão descontados ao seu balanço e receberá uma notificação via *email* a informar que a ausência foi aprovada. Caso contrário, será apenas informado que o seu pedido foi cancelado.

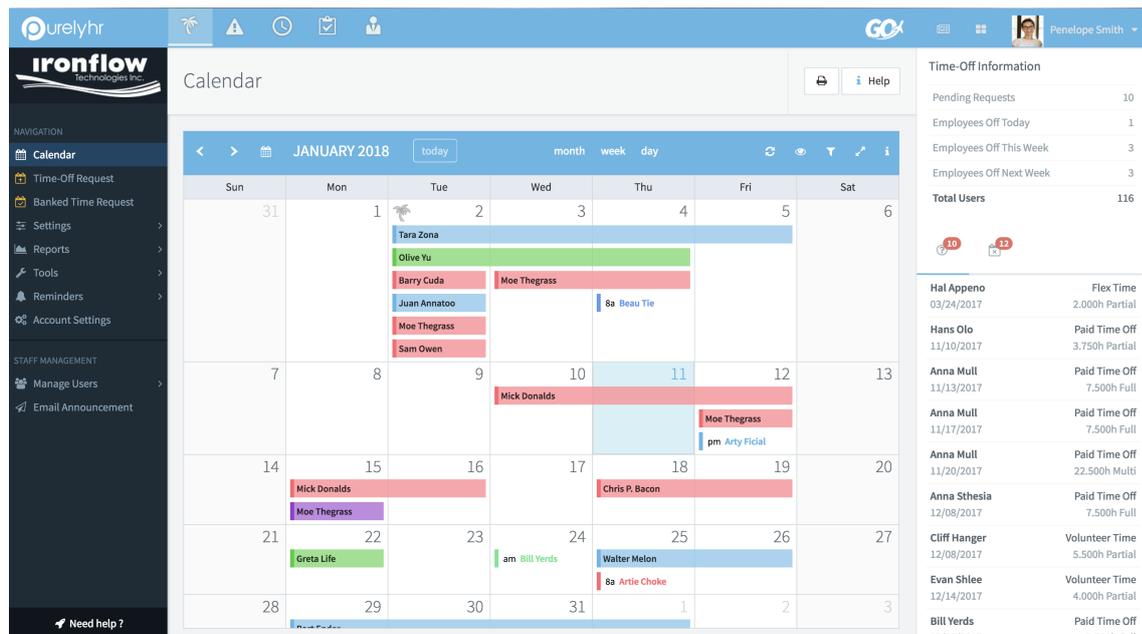


Figura 2.1: Exemplo do calendário na ferramenta PurelyHR [3].

Para efetuar a gestão de processos de recrutamento a empresa utiliza dois ficheiros Excel. O primeiro (Figura 2.2) serve para a gestão dos candidatos onde serão indicados as suas informações (por exemplo, nome e contacto) e, o segundo representa a *pipeline* de recrutamento (Figura 2.3). Isto é, o ficheiro está subdividido em colunas que representam cada uma das fases do processo de recrutamento definidas pela empresa e, em cada linha dessas colunas, serão colocados os nomes dos candidatos nessa fase. Note-se que nem todos os candidatos presentes no primeiro ficheiro constam no segundo, ou seja, apenas os candidatos cuja empresa está interessada ou em conversações estão colocados no segundo ficheiro. Sempre que um candidato transita para a *pipeline* ou avança/recua numa das fases do processo de recrutamento este tem de ser movido à mão, o que não é um processo simples, é moroso e também suscetível a erros.

Além dos ficheiros Excel utilizados, são realizadas várias trocas de *emails* com informação relativa aos candidatos. Ou seja, são enviados *currículos vitae* ou outros documentos relevantes, bem como todas as notas referentes às interações (entrevistas) com os candidatos até ao momento.

	A	B	C	D
1	Name	Status	Profile	Contact
2	José Matos	Active	Full-Stack	josematos@email.com
3	Nuno Cunha	On-Hold	Frontend	nunocunha@email.com
4	Eduarda Faria	Active	Quality Assurance	edufar98@email.com
5	Miguel Campos	Lead	Frontend	campos.miguel@email.com

Figura 2.2: Exemplo do ficheiro da *sourcing list*.

	A	B	C	D	E	F
1	Applied		Interview		Hired	
2						
3	José Matos, Full-Stack		Eduarda Faria, Quality Assurance		Pedro Monteiro, Full-Stack	
4						
5	Tiago Marques, Frontend				Bruno Pereira, Backend	
6						
7	Marta Dias, Full-Stack					
8						

Figura 2.3: Exemplo do ficheiro da *pipeline* de recrutamento.

Deste modo, a empresa decidiu agregar não só estas duas funcionalidades, mas também as que estão a ser realizadas em paralelo pelo outro estagiário para a mesma plataforma. Assim, os colaboradores da empresa apenas necessitarão de uma conta caso queiram aceder às suas ausências e os gestores da empresa agregam as funcionalidades de recrutamento numa ferramenta ajustável às suas necessidades, estando menos suscetíveis a erros e agilizando o processo de recrutamento. Futuramente, espera-se que a plataforma possa adquirir novos módulos, por exemplo, um módulo de partilha de conhecimentos.

O principal objetivo do módulo de ausências é substituir as funcionalidades fornecidas pelo PurelyHR utilizadas pela Grama. Permitindo aos colaboradores visualizarem as suas ausências em vários formatos, bem como aos gestores e administradores personalizarem os tipos de dias de ausência e o balanço dos seus colaboradores. Com esta solução, não só se reduzirá toda a necessidade de configuração do PurelyHR, mas também será apresentada uma ferramenta alinhada com a identidade da Grama.

Já o módulo de recrutamento tem como objetivo informatizar o processo permitindo à empresa agregar os candidatos e as suas informações numa plataforma. A informação (seja ela ficheiros ou notas das entrevistas) fica associada ao respetivo candidato, não se perdendo com o tempo nos *emails*. Este módulo pretende também disponibilizar uma *pipeline* de recrutamento customizável.

2.2 Análise de Concorrentes

A análise de concorrentes permite observar os competidores e recolher informações sobre os seus produtos. Através desta análise é possível ajustar um produto ao mercado em que este se vai inserir.

Para iniciar a análise é importante definir quais os principais pontos a analisar de forma a encontrar os concorrentes. Neste caso, serão analisadas plataformas que contenham funcionalidades como marcação e gestão de ausências, bem como gestão de recrutamento e da sua *pipeline*.

A concorrência pode ser dividida em dois grupos: concorrentes diretos e indiretos. Os **concorrentes diretos** referem-se a empresas cuja plataforma se destine ao mesmo público-alvo e contenha as mesmas funcionalidades que a plataforma que se pretende desenvolver. Enquanto que, os **concorrentes indiretos** são empresas que fornecem produtos semelhantes, mas que visam um mercado ou segmento

de clientes diferente.

2.2.1 Concorrentes Diretos

De seguida, são analisados os concorrentes diretos da Plataforma Interna da Grama. São considerados concorrentes diretos as plataformas que cumpram os objetivos acima enumerados, isto é, apenas gestão de ausências, apenas gestão de recrutamento ou, principalmente, ambas as funcionalidades.

Absence.io

Esta aplicação permite aos colaboradores selecionar os dias de ausência desejados. Após a requisição, os gestores recebem uma notificação por *email* podendo aceitar ou recusar o pedido. Tem calendários pessoais e para a equipa onde, em ambos, é possível visualizar toda a informação necessária, não só os dias de ausência mas também horários de trabalho e feriados. É ainda possível exportar a informação para um ficheiro Excel [4].

BambooHR

A BambooHR é uma empresa que desenvolve vários tipos de produtos com o foco principal no controlo e gestão de recursos humanos [5].

A *Applicant Tracking System* é a plataforma de recrutamento onde os anunciantes são notificados após a submissão de cada proposta. As equipas podem avaliar os candidatos (avaliação com base em estrelas), deixar comentários, enviar *emails* e alterar o estado do candidato. Conseguem ainda acompanhar o progresso de um candidato durante o processo de recrutamento. Existe também uma aplicação móvel com o mesmo propósito [6].

Time-Off Management Software é a funcionalidade através da qual os colaboradores de uma empresa podem requisitar ausências identificando a respetiva razão (férias, folga e/ou doença). Os gestores devem aprovar ou rejeitar os pedidos e, de seguida, é enviada uma notificação e/ou *email* para informar o colaborador. Após a aprovação de um pedido, este é automaticamente adicionado ao calendário [7].

Calamari

Calamari é uma aplicação *web* que permite aos utilizadores requisitarem ausências (férias, folgas e trabalho remoto). Esses pedidos têm de ser aprovados, podendo ainda ser definida uma ordem de aprovação dos mesmos tendo em conta a estrutura da empresa. Suporta vários pedidos de licença (por exemplo, licença médica ou maternidade), permitindo anexar documentos aos pedidos. As ausências dos colaboradores são apresentadas num calendário, onde é possível visualizar que colaborador está ausente e em que dia(s). Adicionalmente, o sistema

permite a integração com outras aplicações como, por exemplo, o calendário da Google [8].

ClanHR

O *software* apresentado pela ClanHR é focado nos recursos humanos e na gestão da empresa. Este *software* permite que um colaborador visualize e contabilize o número de dias disponíveis para ausências, bem como pedir dias de ausência. Após a aprovação por parte de um gestor, esses dias serão adicionados a um calendário (ou cronograma ou lista) que mostrará as ausências de todos os colaboradores da empresa. É possível aplicar filtros de pesquisa e, se necessário, anexar uma justificativa para uma falta (por exemplo, um relatório médico). O *software* informa sobre possíveis sobreposições de ausências e permite também a exportação para ficheiros Excel [9].

Greenhouse

Um dos produtos fornecidos pela Greenhouse é um *software* para gestão de recrutamento. Os candidatos podem concorrer às várias propostas das empresas fornecendo todas as informações relevantes. As equipas de recrutamento conseguem analisar e avaliar todos os candidatos, controlar os estados dos processos de candidatura e ainda obter algumas estatísticas relacionadas ao processo de recrutamento [10].

JazzHR

A JazzHR é uma plataforma que permite aos utilizadores publicar várias propostas de emprego em várias plataformas *online*, dependendo do plano adquirido. Fornece formas de visualizar e avaliar as propostas recebidas, assim como um método para organizar as várias entrevistas de emprego. Permite sincronizar as entrevistas com o calendário e controlar todos os estados do processo de recrutamento. É possível criar e preencher documentos contratuais com assinatura eletrónica e obter relatórios das contratações [11].

PurelyHR

A aplicação *web* PurelyHR permite, entre outras funcionalidades, a requisição de ausências por parte dos utilizadores. Cada pedido passará por um processo de aprovação, no qual um gestor avaliará e decidirá se este deve ou não ser aprovado. Após a aprovação, o pedido será automaticamente adicionado ao calendário da equipa e o colaborador será notificado por *email*. Permite também exportar o calendário para outros calendários como Google e Outlook [1].

Teamtaylor

A Teamtaylor é uma plataforma onde as empresas se podem registar e publicar as propostas de emprego, sendo essas propostas publicadas em vários outros *websites*. Os candidatos podem concorrer às vagas disponíveis e conectar-se com a empresa comunicando através de *email* ou de um *chat*. A equipa de recrutamento consegue avaliar os candidatos, controlar o estado do processo de recrutamento e adicionar estados personalizáveis, além de agendar entrevistas que ficam registadas no calendário da aplicação. Contém ainda algumas estatísticas com base no processo de recrutamento [12].

Workable

Através da Workable é possível publicar propostas de emprego em múltiplos *websites* e pesquisar especificamente por candidatos por todo o mundo. Permite à equipa de recrutamento analisar e avaliar os candidatos, fazer entrevistas por videochamada e ainda controlar os vários estados de uma candidatura. Possui integração com o Gmail e o Outlook permitindo, assim, agendar entrevistas. Fornece relatórios sobre o processo de recrutamento e existe também uma aplicação móvel com algumas das funcionalidades para o recrutamento [13].

Zoho

A Zoho é uma empresa que fornece aos seus clientes vários tipos de *software* tendo em conta as necessidades dos mesmos. Para algumas das plataformas abaixo seria necessário utilizar outras ferramentas disponibilizadas pela Zoho com o objetivo de obter a melhor experiência dos produtos [14]. Alguns exemplos de outras ferramentas extra: Zoho Analytics [15], Zoho Payroll [16], Zoho Sign [17] e Zoho Workerly [18].

Zoho People

Através da plataforma Zoho People, de entre as várias funcionalidades, os utilizadores podem marcar os seus dias de ausência, estes serão marcados num calendário após a aprovação por parte de um gestor. Permite ainda obter relatórios referentes aos dias de presença dos trabalhadores. Podem-se utilizar estas funcionalidades através de uma aplicação móvel [19].

Zoho Recruit

A Zoho Recruit é uma plataforma onde as empresas podem adicionar propostas de emprego, visualizar a informação dos candidatos, criar hierarquias de recrutamento e também aplicar filtros para facilitar a pesquisa, por exemplo, localização e/ou habilitações. Tem funcionalidades como análise dos candidatos, agendamento, realização de entrevistas (por vídeo) e controlo do estado de cada candi-

data. Além disso, permite gerar relatórios e contém cartas personalizadas com assinatura eletrônica, o que facilita a aceitação dos candidatos [20].

Zoho Shifts

A Zoho Shifts é uma ferramenta focada no controle dos turnos dos colaboradores da empresa. Permite configurar as agendas dos colaboradores criando turnos sem sobreposição com os dias de ausência, notificando os colaboradores sempre que existem alterações. Os gestores recebem uma notificação para aprovação sempre que um colaborador marcar os seus dias de ausência. Contém um método de relógio de ponto e uma possível integração de folhas de pagamento [21].

Comparação entre concorrentes diretos

A análise efetuada anteriormente aos concorrentes pode ser sumariada na Tabela 2.1. Como resultado, pode-se, de uma forma mais objetiva, analisar e comparar as funcionalidades entre as várias plataformas concorrentes. Sendo possível concluir que, as plataformas BambooHR e Zoho são as que cumprem todos os objetivos requeridos.

No entanto, a BambooHR tem apenas dois planos disponíveis [22]. O acesso a todas as funcionalidades pretendidas está disponível exclusivamente no plano *advantages*, visto que, o plano mais simples contém somente a funcionalidade de ausências.

Relativamente à ferramenta Zoho, a sua configuração pode ser considerada uma desvantagem, uma vez que os seus módulos necessitam de integração e adaptação para atender às necessidades específicas de cada empresa. O que pode requerer um esforço adicional em termos de configuração e personalização.

Tabela 2.1: Tabela de comparação dos concorrentes diretos.

Objetivos Concorrentes		Gestão de ausências	Notificação e aprovação de ausências	Visualização dos dados das ausências	Gestão de recruta- mento	Visualização da <i>pipeline</i>
Absence.io		✓	✓	✓	✗	✗
BambooHR		✓	✓	✓	✓	✓
Calamari		✓	✓	✓	✗	✗
ClanHR		✓	✓	✓	✗	✗
Greenhouse		✗	✗	✗	✓	✓
JazzHR		✗	✗	✗	✓	✓
PurelyHR		✓	✓	✓	✗	✗
Teamtailor		✗	✗	✗	✓	✓
Workable		✗	✗	✗	✓	✓
Zoho	People	✓	✓	✓	✗	✗
	Recruit	✗	✗	✗	✓	✓
	Shifts	✓	✓	✓	✗	✗

2.2.2 Concorrentes Indiretos

De seguida, é analisado o principal concorrente indireto da Plataforma Interna da Grama. Um concorrente indireto consiste num programa que auxilia as empresas a fazer a sua gestão de informação. Por uma questão de familiaridade as empresas ainda recorrem frequentemente ao uso de folhas de cálculo.

Folhas de cálculo

Um dos métodos mais antigos de realizar o controlo das ausências e também da gestão dos processos de recrutamento é através de folhas de cálculo. O Excel [2], desenvolvido pela Microsoft [23], é um programa que oferece aos utilizadores várias folhas de cálculo para a manipulação de dados através de gráficos, tabelas, fórmulas, entre outras funcionalidades.

Além do Excel, poder-se-iam enumerar outras ferramentas como o Google Sheets [24], Zoho Sheet [25] ou LibreOffice Calc [26]. Todas estas são folhas de cálculo, isto é, programas desenvolvidos com o objetivo de criar e manipular dados em formato tabela(s) e/ou gráfico(s).

Para gerir as ausências (Figura 2.4) ou os processos de recrutamento (Figura 2.5) existem vários *templates* disponíveis *online* (gratuitos ou pagos).

Employee Absence Schedule

Absence Type Key		V	P	S	Custom 1											
		V	P	S	Custom 1											
January		Dates														
Employee Name		Fri 1	Sat 2	Sun 3	Mon 4	Tue 5	Wed 6	Thu 7	Fri 8	Sat 9	Sun 10	Mon 11	Tue 12	Wed 13	Thu 14	Fri 15
Employee 1				V	V	V	V							V		
Employee 2					S	S						P				
Employee 3				P											S	
Employee 4								P								
Employee 5				S	V	V										
January Total				2	2	3	3	1				1		1	1	

Figura 2.4: Exemplo de um *template* de marcação de ausências [27].

New Application	Available Jobs	Recruiter Name	Candidate Name	Application Source	Application Date	Last Recruitment Stage	Final Decision	If Reject Notify Reason	Decision Date	Joining Date	Application Period In Days
RC0001	Recruiter 1	Candidate 1	Glassdoor	25-Dec-20	HR Interview	Rejected			30-Dec-20	5-Jan-21	5
RC0002	Recruiter 2	Candidate 2	Linkedin	7-Dec-20	Hired	Accepted		Culture	29-Dec-20	7-Jan-21	22
RC0002	Recruiter 2	Candidate 3	Agency	5-Dec-20	Onsite Interview	Rejected	Technical		12-Dec-20	20-Dec-20	7
RC0003	Recruiter 1	Candidate 4	Facebook	30-Nov-20	Application	Shortlisted					60
RC0003	Recruiter 3	Candidate 5	Company Website	1-Dec-20	Offer	Accepted			25-Dec-20	1-Jan-21	24
RC0005	Recruiter 4	Candidate 6	Linkedin	30-Mar-20	Onsite Interview	In-Process					325
RC0007	Recruiter 2	Candidate 7	Agency	15-Jan-20	Manager Interview	Rejected	Technical		10-Feb-20		26
RC0005	Recruiter 4	Candidate 8	Facebook	12-Apr-20	Application	In-Process					312
RC0008	Recruiter 5	Candidate 9	Company Website	3-Dec-20	Phone Screen	Rejected	Salary		3-Dec-20		0
RC0011	Recruiter 6	Candidate 10	Linkedin	7-Aug-19	Hired	Accepted			9-Sep-19		33

Figura 2.5: Exemplo de um *template* de gestão de recrutamento [28].

A utilização de uma ferramenta como o Excel para efetuar quer a gestão de ausências, quer a gestão de recrutamento acarreta mais desvantagens do que vantagens como se pode concluir da Tabela 2.2.

Tabela 2.2: Vantagens e desvantagens da ferramenta Excel [29].

Vantagens	Desvantagens
Ferramenta familiar a todos	Propenso a erros
Rápido de usar pois existem vários <i>templates</i> disponíveis	Pode-se tornar confuso e complexo rapidamente
De customização fácil às necessidades reais da empresa	Aceitação/Confirmação dos pedidos morosa pois implica troca de <i>emails</i>
	Não escalável, visualização difícil com o aumento da folha
	Sincronização dos dados tem de ser efetuada através da partilha do ficheiro
	Perigo de perda dos dados ou até mesmo o próprio ficheiro

Capítulo 3

Requisitos

Este capítulo documenta os atores e as funcionalidades a serem desenvolvidas. Os requisitos funcionais foram descritos em formato de *user stories*, priorizados segundo o método de *MoSCoW* e representados em diagramas de casos de uso. Enquanto que os requisitos não funcionais serão apresentados em formato de atributo de qualidade.

3.1 Atores

A plataforma terá **três tipos de atores**: administradores, gestores e colaboradores. Estes terão de iniciar sessão sempre que queiram aceder a qualquer conteúdo da plataforma. Após a sessão iniciada:

- Um **colaborador** tem acesso a um calendário onde pode consultar as ausências de todos os colaboradores. Pode ainda efetuar pedidos de ausências que terão de ser posteriormente aprovados por um gestor ou administrador;
- Os **gestores** têm acesso às mesmas funcionalidades que os colaboradores e, adicionalmente, conseguem gerir as ausências dos colaboradores da(s) sua(s) equipa(s). Têm acesso à *pipeline* de recrutamento onde visualizam os candidatos com os quais já interagiram, isto é, os candidatos que entrevistaram;
- Os **administradores**, para além das funcionalidades dos gestores, conseguem visualizar métricas relacionadas com as ausências, efetuar a gestão das ausências de toda a empresa, bem como atualizar configurações das ausências. Na *pipeline* de recrutamento podem fazer uma gestão global de todos os candidatos e customizar toda a *pipeline*.

Os três tipos de utilizadores receberão notificações por *email* informando se os seus pedidos de ausência foram ou não aprovados. Por fim, como podemos inferir pela descrição acima, existe uma **herança de funcionalidades**, ou seja, um administrador terá acesso às funcionalidades de gestor e colaborador, enquanto um gestor terá apenas acesso às funcionalidades de colaborador.

3.2 User Stories

Uma *Epic Story* (ES) representa uma grande funcionalidade do produto que pode ser subdividida em tarefas menores, denominadas *user stories*. As *user stories* são uma forma informal de descrever funcionalidades da perspectiva do utilizador [30].

Uma *User Story* (US) é descrita numa frase curta onde um tipo de utilizador demonstra intenção de realizar uma determinada ação por uma razão. Deste modo, as *user stories* são estruturadas da seguinte forma [31]:

"**Como** [tipo de utilizador] **quero** [ação] **de modo a** [razão]."

Todas as *user stories* desenvolvidas foram incluídas no Anexo A deste documento. A seguir, é apresentada, de forma exemplificativa para este capítulo, uma *epic* e *user story* do módulo de autenticação.

Autenticação

ES-1: Como administrador autenticado **quero** enviar um convite **de modo a** adicionar um utilizador à plataforma.

US-1: Enviar convite para a plataforma

- (a) **Como** administrador autenticado **quero** enviar um convite para a plataforma **de modo a** que o utilizador possa efetuar o seu primeiro registo.
- (b) **Como** administrador autenticado **quero** poder reenviar um convite para a plataforma **de modo a** que o utilizador possa efetuar o seu primeiro registo.
- (c) **Como** administrador autenticado **quero** poder reenviar um convite para a plataforma **de modo a** que o utilizador possa recuperar o seu acesso.

3.3 Requisitos Funcionais

Um Requisito Funcional (RF) corresponde a uma descrição de uma funcionalidade que a plataforma deverá ter. Os requisitos funcionais apresentados são priorizados segundo o método de *MoSCoW* (*Must Have, Should Have, Could Have, Will Not Have*). As categorias são objetivas e representam a prioridade do requisito [32].

- **Must:** requisito crítico para o correto funcionamento da plataforma, logo, deve estar presente no *Minimum Viable Product* (MVP);

- **Should:** requisito não prioritário para o sucesso da plataforma, contudo, adiciona valor substancial;
- **Could:** a implementação deste requisito não é importante para o funcionamento da plataforma e só deve ser implementado caso exista tempo extra;
- **Will Not:** requisito que não será implementado mas será levantado para uma futura implementação.

De seguida, serão apresentados os requisitos funcionais devidamente separados por área para os três tipos de utilizador. Os requisitos funcionais estão também priorizados e apresentam um Identificador (ID) único. Sendo assim, a estrutura das tabelas abaixo é a seguinte: área, identificador do requisito, descrição do requisito, os três tipos de utilizador e, por fim, a sua prioridade.

Módulo de Autenticação

O módulo de autenticação (Tabela 3.1) abrange todas as funcionalidades relacionadas à autenticação do utilizador na plataforma. Incluindo o envio de um convite ao utilizador, o seu primeiro início de sessão, para além do início e término de sessão, assim como a recuperação da palavra-passe. Embora este não seja um dos dois módulos definidos nos objetivos do estágio, o principal requisito para o uso da plataforma é a autenticação dos utilizadores pelo que a sua realização é essencial.

Tabela 3.1: Requisitos funcionais para o módulo de autenticação.

Área	ID	Requisito	Admin.	Gestor	Colab.	Prioridade
Autenticação	RF-1	Enviar convite para a plataforma	✓	✗	✗	M
	RF-2	Efetuar primeiro início de sessão	✓	✓	✓	M
	RF-3	Iniciar sessão	✓	✓	✓	M
	RF-4	Recuperar palavra-passe	✓	✓	✓	M
	RF-5	Terminar sessão	✓	✓	✓	M

Módulo de Ausências

O módulo de ausências (Tabela 3.2) representa todas as funcionalidades associadas à visualização e requisição de ausências. Ou seja, desde o pedido de ausência de um colaborador até à sua aprovação ou rejeição por parte de um gestor ou administrador. Engloba também as notificações via *email* e funcionalidades de gestão de tipos de ausência ou eventos especiais aos quais apenas o administrador terá acesso.

Tabela 3.2: Requisitos funcionais para o módulo de ausências.

Área	ID	Requisito	Admin.	Gestor	Colab.	Prioridade
Dashboard	RF-6	Visualizar métricas	✓	✓	✓	M
	RF-7	Pesquisar e aplicar filtros à <i>dashboard</i>	✓	✓	✓	S
Gerir calendário	RF-8	Visualizar ausências em calendário - mês	✓	✓	✓	M
	RF-9	Visualizar ausências em calendário - semana	✓	✓	✓	S
	RF-10	Visualizar ausências em calendário - dia	✓	✓	✓	S
	RF-11	Visualizar ausências marcadas em grelha	✓	✓	✓	W
	RF-12	Aplicar filtros ao calendário	✓	✓	✓	S
	RF-13	Visualizar pedidos	✓	✓	✓	M
RF-14	Aprovar pedido	✓	✓	✗	M	
RF-15	Rejeitar pedido	✓	✓	✗	M	
RF-16	Cancelar pedido	✓	✓	✓	M	
RF-17	Editar detalhes do pedido	✓	✓	✓	M	

Gerir pedidos	RF-18	Pesquisar e aplicar filtros à lista de pedidos	✓	✓	✗	S
Gerir ausências	RF-19	Ver detalhes de uma ausência	✓	✗	✗	M
	RF-20	Editar detalhes de uma ausência	✓	✗	✗	M
Requisitar ausência	RF-21	Requisitar ausência - múltiplos dias	✓	✓	✓	M
	RF-22	Requisitar ausência - dia completo	✓	✓	✓	M
	RF-23	Requisitar ausência - dia parcial	✓	✓	✓	M
Notificar	RF-24	Receber <i>email</i> com pedido de aprovação	✓	✓	✗	M
	RF-25	Receber <i>email</i> com o resultado do pedido	✓	✓	✓	M
	RF-26	Enviar <i>email</i> informativo sobre a ausência	✓	✗	✗	M
	RF-27	Receber <i>email</i> informativo sobre a ausência	✓	✓	✓	M
	RF-28	Visualizar tipos disponíveis	✓	✗	✗	M
	RF-29	Adicionar tipo	✓	✗	✗	M

Gerir tipos de ausências	RF-30	Editar tipo	✓	✗	✗	M
	RF-31	Arquivar tipo	✓	✗	✗	M
Gerir balanço	RF-32	Visualizar ausências dos colaboradores ativos	✓	✓	✗	M
	RF-33	Editar balanço	✓	✗	✗	M
	RF-34	Pesquisar e aplicar filtros à lista de colaboradores	✓	✓	✗	S
Gerir eventos especiais	RF-35	Visualizar eventos especiais	✓	✗	✗	C
	RF-36	Adicionar eventos especiais	✓	✗	✗	C

Módulo de Recrutamento

Neste módulo, encontram-se representadas as funcionalidades relativas ao recrutamento de novos colaboradores. Começando pela sua adição ao quadro (*pipeline*) de recrutamento, passando pela gestão de candidatos e de interações com os mesmos. O administrador poderá ainda customizar o quadro de recrutamento para o adaptar ao processo de recrutamento da empresa. Na Tabela 3.3 estão representados apenas dois dos três atores, uma vez que, os colaboradores não terão acesso a este módulo.

Tabela 3.3: Requisitos funcionais para o módulo de recrutamento.

Área	ID	Requisito	Admin.	Gestor	Prioridade
	RF-37	Ver candidatos ativos	✓	✓	M
	RF-38	Adicionar candidato	✓	✗	M
	RF-39	Adicionar candidato a uma coluna específica	✓	✗	S

Gerir quadro de recrutamento	RF-40	Adicionar candidato com informação detalhada	✓	✗	M
	RF-41	Pesquisar e aplicar filtros ao quadro	✓	✓	S
Customizar quadro de recrutamento	RF-42	Adicionar coluna	✓	✗	M
	RF-43	Editar coluna	✓	✗	M
	RF-44	Editar coluna diretamente no quadro	✓	✗	S
	RF-45	Remover coluna	✓	✗	M
Lista de candidatos	RF-46	Editar candidato pela tabela	✓	✗	S
	RF-47	Pré-visualizar informação do candidato	✓	✓	S
	RF-48	Ver candidatos em lista	✓	✓	S
	RF-49	Arquivar candidato	✓	✗	M
	RF-50	Recuperar candidato arquivado	✓	✗	M
	RF-51	Pesquisar e aplicar filtros à lista	✓	✓	S
	Analisar candidato	RF-52	Ver detalhes gerais	✓	✓
RF-53		Adicionar interações	✓	✓	M
RF-54		Editar detalhes via detalhes completos	✓	✗	M
RF-55		Editar interações	✓	✗	M

3.4 Diagrama de Casos de Uso

Um diagrama de casos de uso representa, de forma mais detalhada, a interação dos atores com o sistema (a plataforma). Após a definição e priorização dos requisitos, juntamente com a realização das *user stories*, é possível elaborar diagramas de casos de uso para os módulos do sistema.

Inicialmente, foi criado um diagrama de contexto para demonstrar os atores e suas interações com o sistema. Os atores principais são o colaborador, o gestor e o administrador, e cada um deles realizará várias ações na plataforma. De notar ainda a herança de funcionalidades, em que o gestor herda as funcionalidades do colaborador e o administrador herda as funcionalidades do gestor. O ator secundário é um serviço de *emails*, que auxiliará no envio e recepção de *emails* relacionados com as ausências.

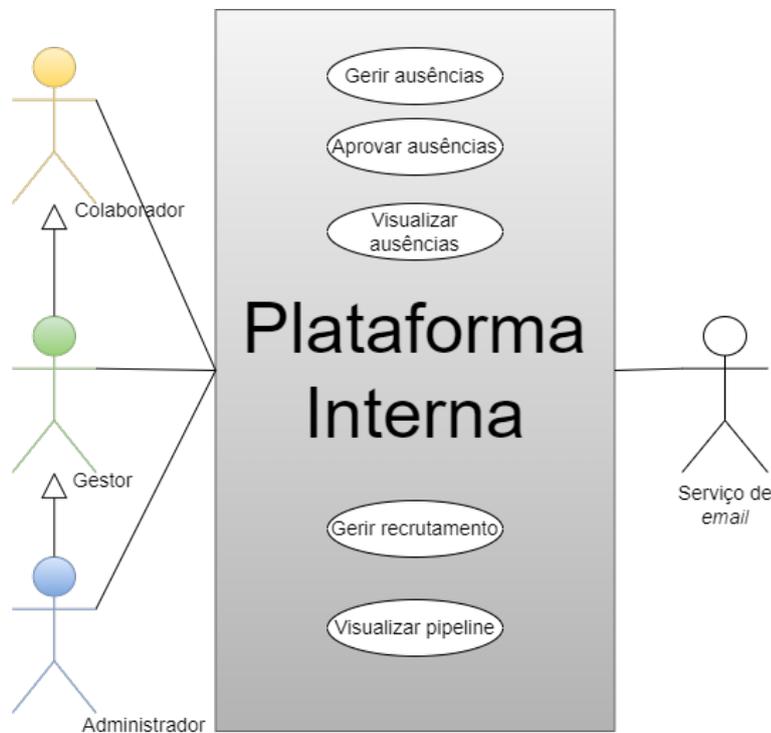


Figura 3.1: Diagrama de contexto.

Módulo Autenticação

Na Figura 3.2, é possível observar quais as ações relacionadas à autenticação dos três tipos de utilizador. Desde efetuar o primeiro início de sessão após a receção de um *email* de convite, até ao frequente inicio e término de sessão.

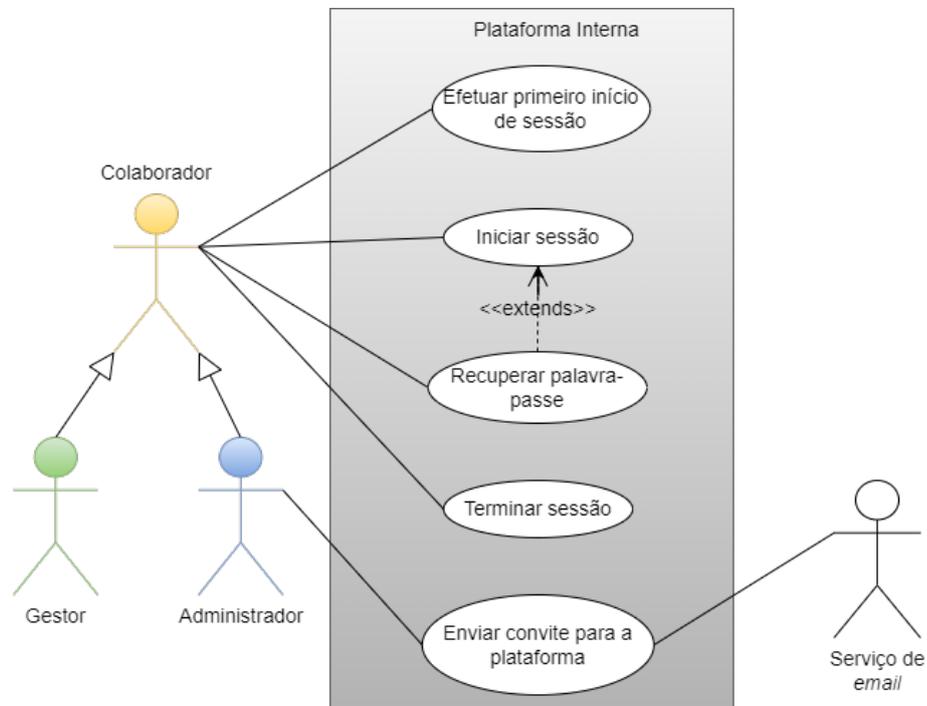


Figura 3.2: Diagrama de casos de uso o módulo de autenticação.

Módulo Ausências

Nas figuras seguintes, é possível verificar todas as possíveis ações dos três tipos de utilizador no módulo de ausências (Figuras 3.3, 3.4, e 3.5, respetivamente). De notar a presença de um ator secundário (serviço de *email*) nas ações relacionadas ao envio e/ou receção de *emails*.

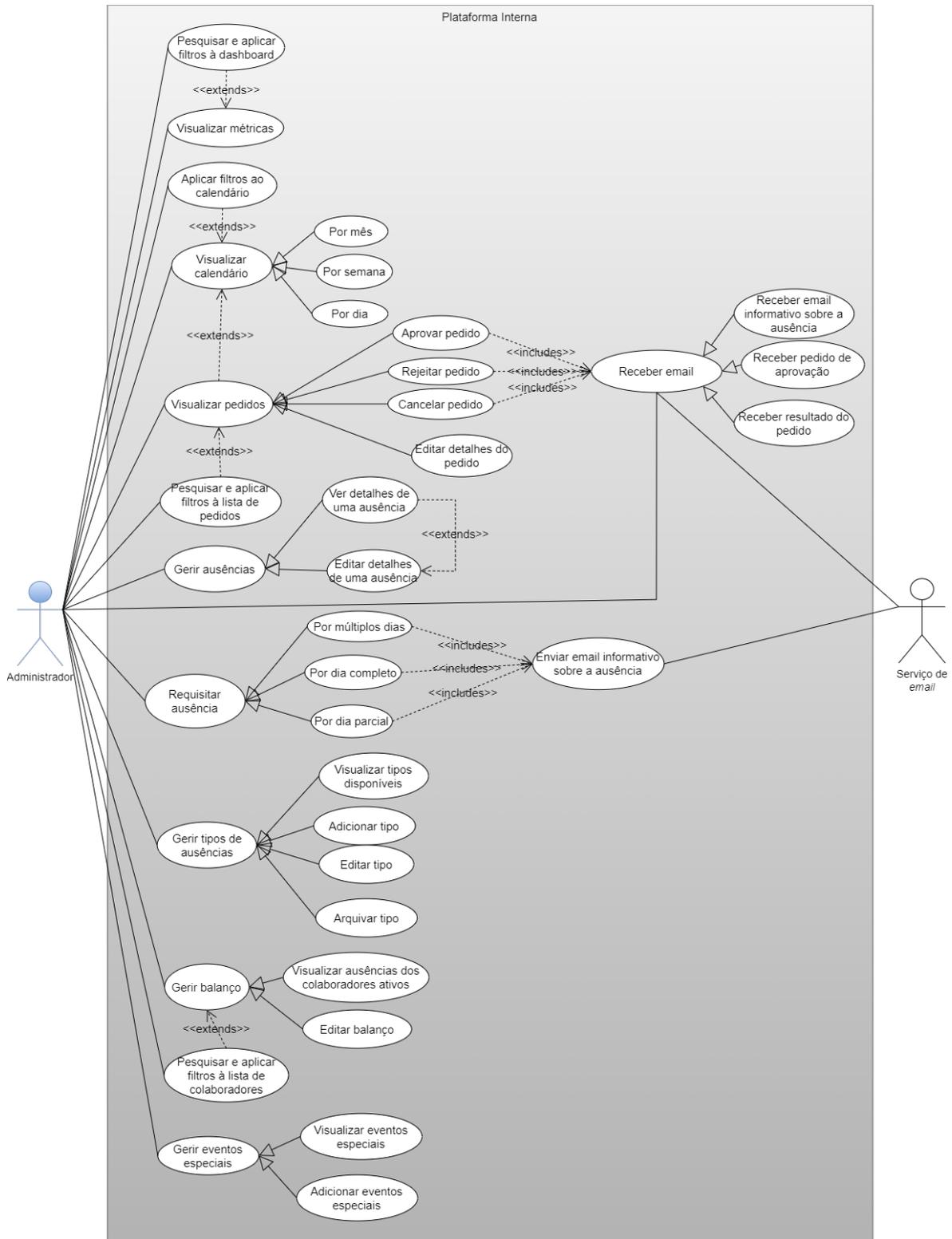


Figura 3.3: Diagrama de casos de uso para o administrador no módulo de ausências.

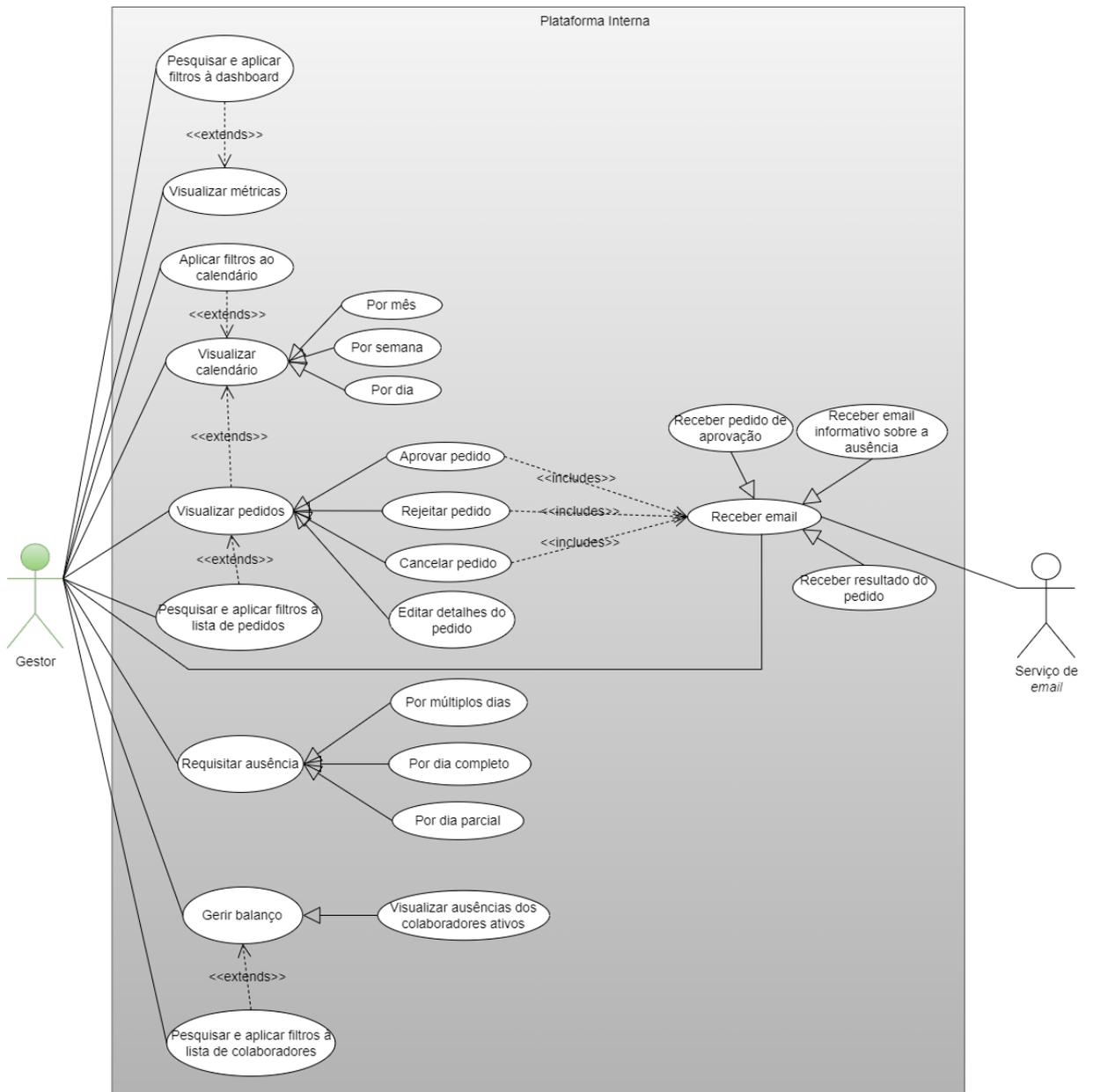


Figura 3.4: Diagrama de casos de uso para o gestor no módulo de ausências.

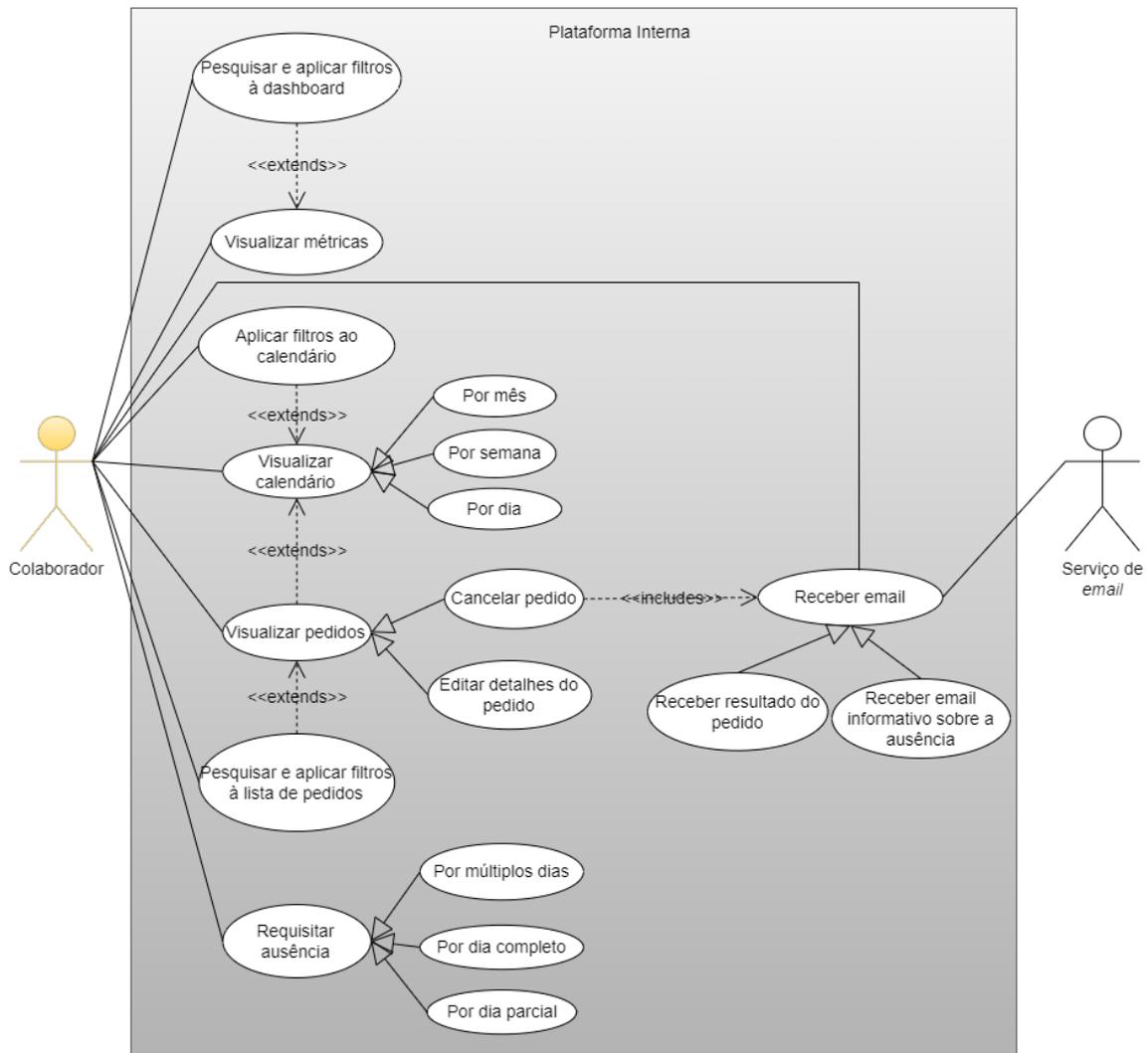


Figura 3.5: Diagrama de casos de uso para o colaborador no módulo de ausências.

Módulo Recrutamento

Visto que um colaborador não terá acesso ao módulo de recrutamento, as figuras seguintes representam apenas as interações de um administrador (Figura 3.6) e de um gestor (Figura 3.7) no módulo de recrutamento.

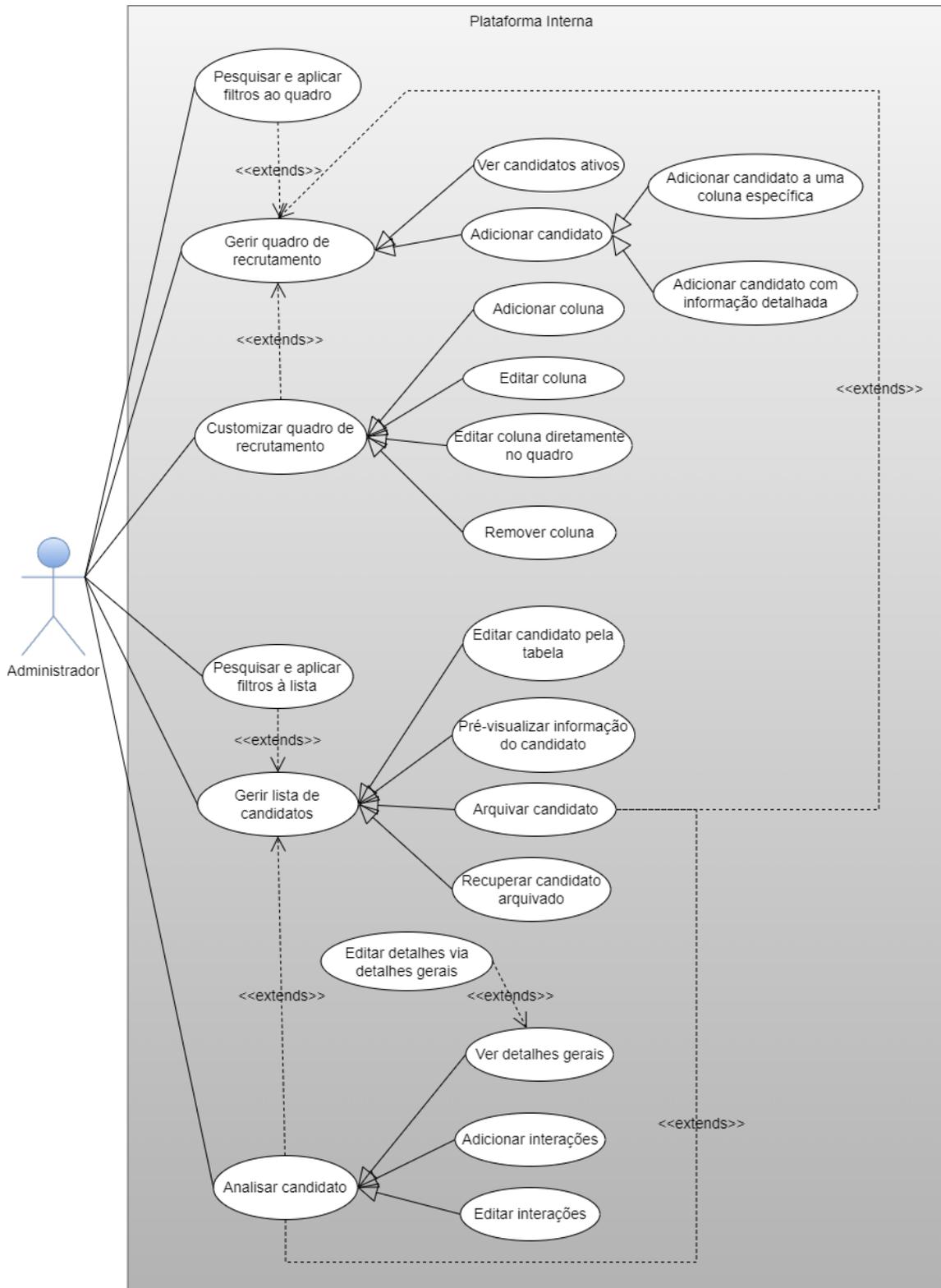


Figura 3.6: Diagrama de casos de uso para o administrador no módulo de recrutamento.

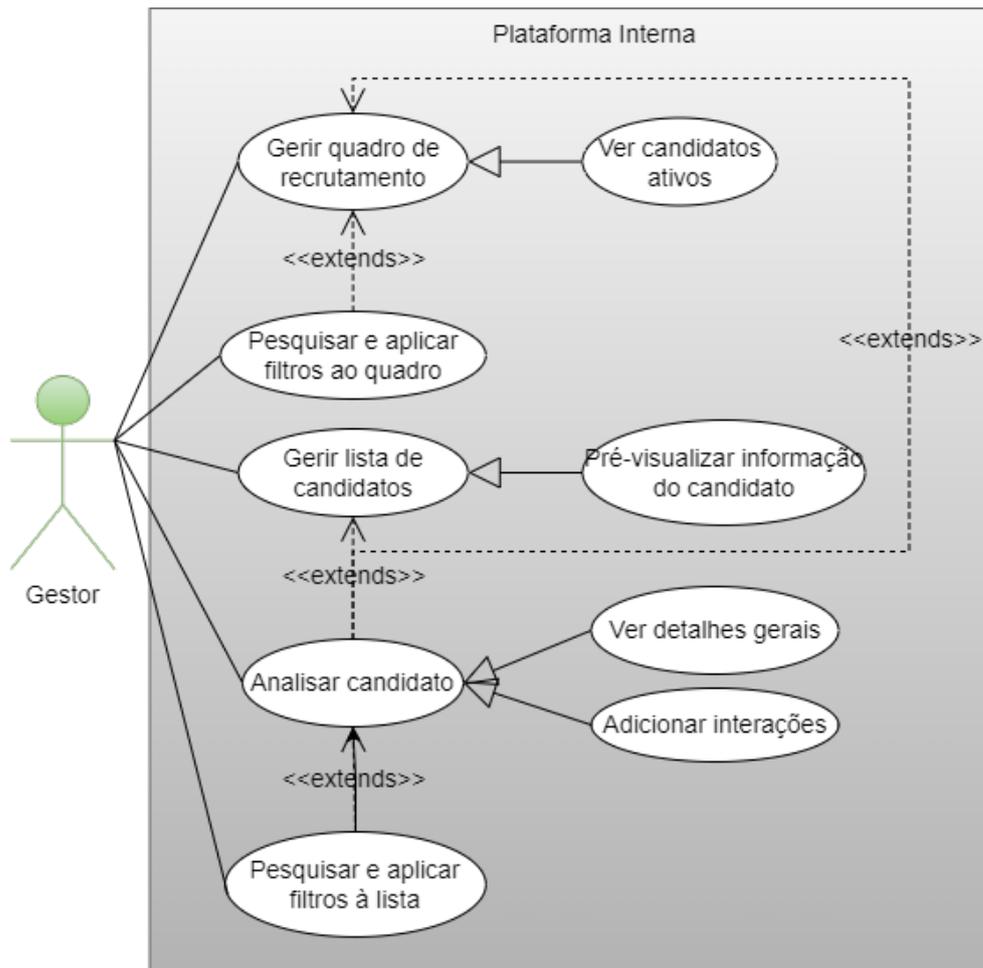


Figura 3.7: Diagrama de casos de uso para o gestor no módulo de recrutamento.

3.5 Requisitos Não Funcionais

Os requisitos não funcionais são um conjunto de especificações que descrevem as capacidades e/ou restrições de operação de um sistema. Os requisitos não funcionais abaixo encontram-se apresentados em formato de atributo de qualidade [33].

Para cada Requisito Não Funcional (RNF), é apresentada uma breve descrição, seguida de um cenário. O cenário é composto por uma fonte (utilizador) que provoca um estímulo aplicado a um artefacto num determinado ambiente, resultando numa resposta que é medida através de uma métrica.

3.5.1 Robustez

A robustez (RNF-1) representa o quão tolerante é um sistema aquando de uma falha. Neste caso, como o *software* se comportará na presença de um *bug*, por exemplo um *input* inválido do utilizador ou uma qualquer exceção.

De seguida, é apresentado um cenário demonstrativo da introdução de um *input* inválido por parte de um utilizador da plataforma (Figura 3.8).

Cenário: O utilizador insere dados inválidos.

Prioridade: Alta

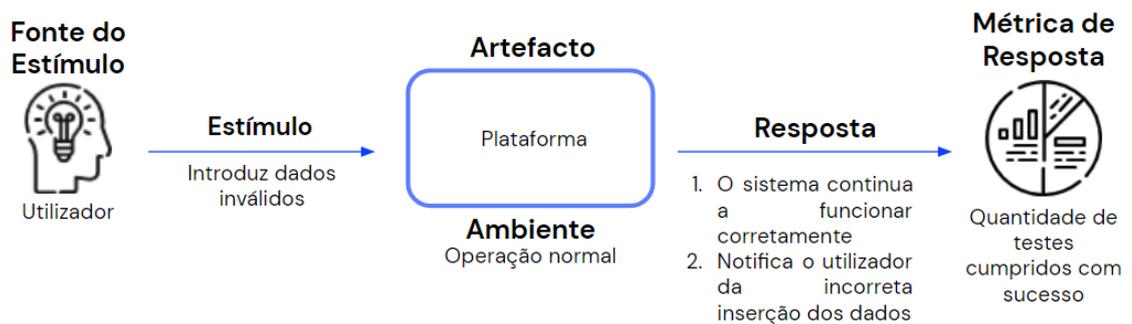


Figura 3.8: Cenário do requisito não funcional robustez.

3.5.2 Performance

A *performance* (RNF-2) trata da capacidade de resposta de um sistema a certas ações num determinado período de tempo. Existem duas formas de medir a *performance*: o tempo gasto na resposta a um evento (latência) e a quantidade de eventos que podem ocorrer numa determinada janela temporal [34].

Pela análise da Figura 3.9, podemos inferir que um tempo de carregamento é considerado bom até 2,5 segundos. Valores compreendidos entre 2,5 e 4 segundos indicam que as páginas precisam de pequenas melhorias para reduzir o tempo de carregamento. Além disso, páginas com tempos de carregamento superiores a 4 segundos devem ser consideradas prioritárias para correção e melhoria do tempo de carregamento, de modo a proporcionar uma melhor experiência ao utilizador. Neste caso pretende-se garantir que as informações são apresentadas no ecrã num curto período de tempo, ou seja, até **2,5 segundos** [35].

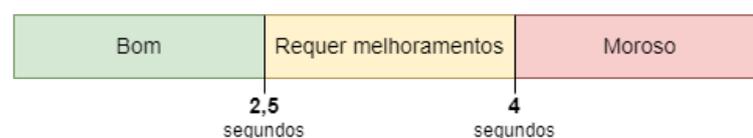


Figura 3.9: Exemplos de tempos de carregamento (adaptado de [35]).

O cenário seguinte representa um utilizador a aceder a um qualquer módulo da plataforma para visualizar uma qualquer informação (Figura 3.10).

Cenário: O utilizador acede a um módulo da plataforma.

Prioridade: Baixa

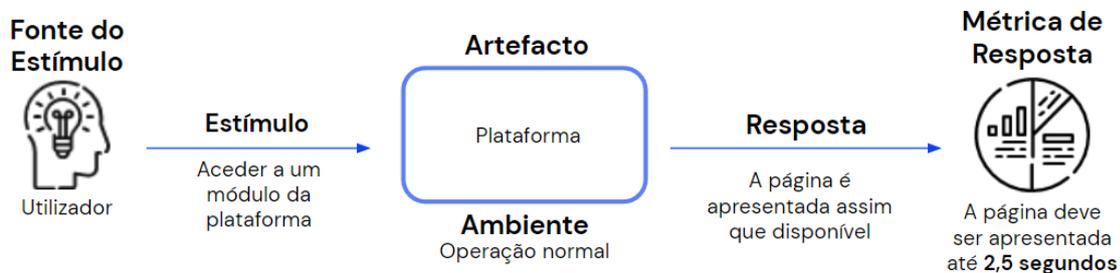


Figura 3.10: Cenário do requisito não funcional *performance*.

3.5.3 Segurança e Privacidade

Segurança (RNF-3) é um atributo de qualidade que mede a capacidade do sistema de proteger dados a acessos de terceiros enquanto garante acesso a utilizadores autorizados [34]. Visto que a plataforma conterà dados pessoais (*Personally Identifiable Information* (PII)) é necessário manter a confidencialidade e integridade dos dados segundo o Regulamento Geral sobre a Proteção de Dados (RGPD) da União Europeia.

Com base no RGPD é importante garantir: o direito de acesso, direito à portabilidade dos dados, direito a corrigir os dados, direito de oposição e direito ao apagamento dos dados (direito a ser esquecido) [36].

Para tal foram elaborados dois cenários: o primeiro para demonstrar a **não autenticação** (Figura 3.11) e o segundo para demonstrar a **não autorização** (Figura 3.12).

Cenário: O utilizador tenta aceder a uma página sem permissão.

Prioridade: Alta

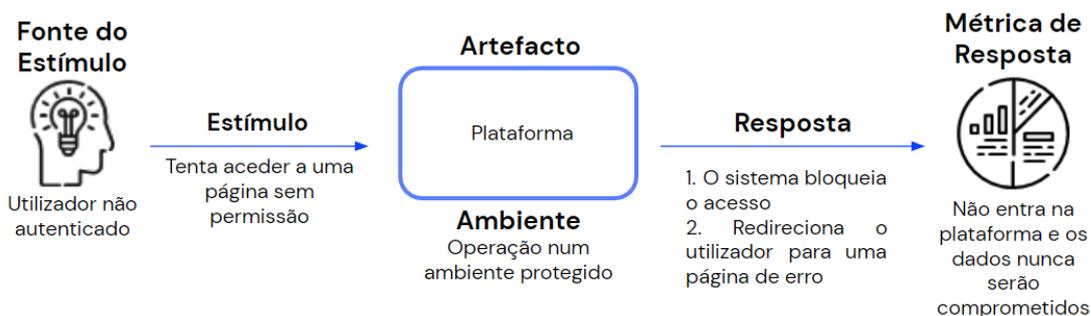


Figura 3.11: Cenário do requisito não funcional segurança, representado o acesso a uma página por um utilizador não autenticado.

Cenário: O utilizador tenta aceder a conteúdos de outros.

Prioridade: Alta



Figura 3.12: Cenário do requisito não funcional segurança, representado o acesso a uma página por um utilizador não autorizado.

Capítulo 4

Tecnologias

O desenvolvimento dos módulos de ausências e recrutamento da Plataforma Interna da Grama exige um estudo e uma tomada de decisão sobre as tecnologias a utilizar para o *frontend* e para *backend* tendo em consideração os requisitos previamente definidos (secção 3).

Embora não tenham sido impostas quaisquer restrições à escolha das tecnologias a analisar, foram sempre consideradas tecnologias já utilizadas em projetos da empresa. Dessa forma, além de permitir um suporte mais técnico por parte do orientador, também existia a possibilidade de obter ajuda de outros colaboradores experientes da empresa, caso necessário.

Nesta secção, será inicialmente apresentado o estudo relativo às tecnologias para o desenvolvimento do *frontend* e, de seguida, para o *backend*. Para cada subsecção, para além da análise das tecnologias individualmente, estas são comparadas entre si, terminando com a apresentação da tecnologia desenvolvida.

4.1 *Frontend*

Para o desenvolvimento do *frontend* existem várias tecnologias como, por exemplo: **Angular**, **Backbone.js**, **Ember.js**, **React**, **Svelte** e **Vue.js**. Através da análise da Figura 4.1, concluí-se que as maiores tendências entre a comunidade são **React**, **Angular**, **Vue.js** e **Svelte**. Algumas destas tecnologias são já adotadas por programadores experientes da Grama, o que possibilitará um maior suporte por parte dos mesmos.

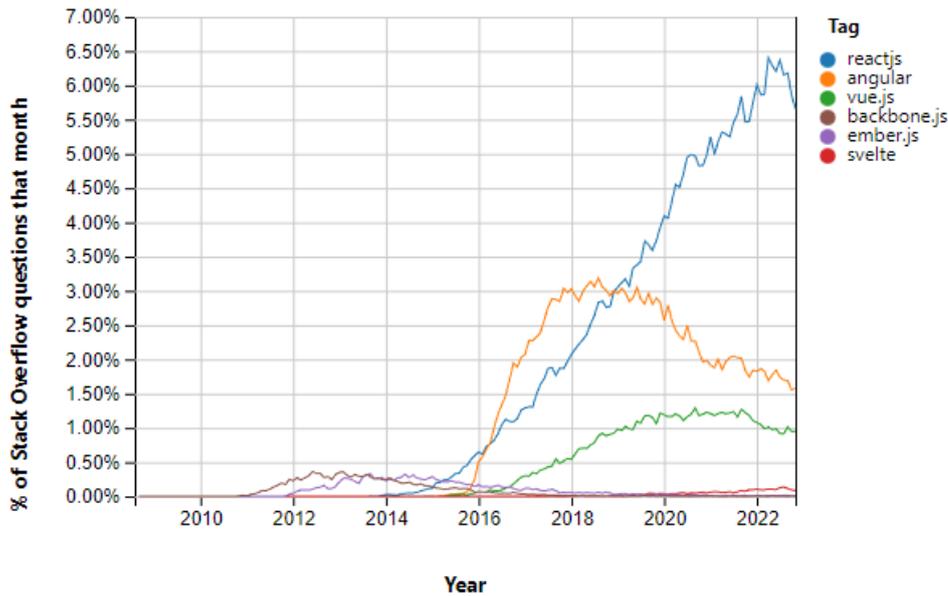


Figura 4.1: Tendências no *Stack Overflow* [37].

4.1.1 Angular

Angular é uma *framework* desenvolvida pela Google e baseada em TypeScript [38]. Destaca-se por uma ligação de dados bidirecional (*two-way binding*), ou seja, existe uma sincronização em tempo real entre o *model* e a *view* [39]. É amplamente utilizada e possui uma comunidade ativa, o que proporciona uma vasta gama de recursos, documentação e suporte disponíveis. Contudo, é uma *framework* com uma curva de aprendizagem elevada, devido à complexidade e detalhe da linguagem [40].

4.1.2 React

React é uma biblioteca JavaScript utilizada para construir interfaces. Desenvolvida pelo Facebook, destaca-se pelo *Document Object Model* (DOM) virtual e *one-way data binding* [41]. Uma das principais vantagens é a reutilização dos componentes. No entanto, como não é uma *framework*, pode necessitar de bibliotecas externas para algumas funcionalidades, o que pode causar um difícil de controlo de dependências e perda de desempenho. Uma vez que está sujeita a atualizações constantes torna-se difícil manter a documentação sempre atualizada [40].

4.1.3 Svelte

Svelte é um compilador que, ao invés de interpretar o código em *runtime* (por exemplo, como React), este permite que os componentes sejam criados e, durante a compilação, converte-os em código JavaScript que integra posteriormente no DOM [42]. Não necessita de um DOM virtual mas, como ainda é uma ferramenta recente, tem pouca documentação e uma comunidade pequena [40].

4.1.4 Vue.js

Vue.js é uma *framework* JavaScript utilizada para desenvolver interfaces. Esta *framework* tenta juntar o melhor de Angular e de React. Tem uma ligação de dados bidirecional como em Angular e um DOM virtual tal como React [43]. As maiores desvantagens são a falta de *plugins* e de suporte, porém, é uma *framework* leve e de fácil aprendizagem [40].

4.1.5 Comparação das tecnologias

A Tabela 4.1 permite comparar as tecnologias para o desenvolvimento do *frontend* da Plataforma Interna da Grama (* - dados do dia 03/11/2022).

Tabela 4.1: Tabela comparativa das tecnologias analisadas para o *frontend*.

	Angular	React	Svelte	Vue.js
Ano de lançamento	2016	2013	2016	2014
Linguagem	TypeScript	JavaScript	JavaScript / TypeScript	JavaScript / TypeScript
Vantagens	Ligação bidirecional; Arquitetura baseada em componentes; Bom suporte da comunidade.	DOM virtual o que permite operações rápidas; Componentes reutilizáveis; Fácil aprendizagem.	Leve; Arquitetura baseada em componentes; Não necessita de um DOM virtual.	Ligação bidirecional; Documentação completa; Sintaxe simples, aprendizagem rápida.
Desvantagens	Complexo e detalhado; Curva de aprendizagem elevada.	Falta de documentação elaborada.	Falta de documentação pois é recente; Comunidade pequena.	Falta de <i>plugins</i> ; Pouco suporte.
Utilização	Aplicações de larga escala	Projetos que requerem vários componentes	Aplicações de pequena dimensão	Indicada para <i>single-page applications</i>
Popularidade* (GitHub)	84.6 mil ★ ^[44]	197 mil ★ ^[45]	63 mil ★ ^[46]	200 mil ★ ^[47]

4.1.6 Tomada de decisão

A tomada de decisão sobre a tecnologia utilizar para a realização do *frontend* da plataforma, após a análise das várias tecnologias (secção 4.1), recai sobre **React**

ou **Vue**. Isto devido à elevada curva de aprendizagem de Angular e à falta de documentação da tecnologia recente Svelte.

Embora Vue seja a tecnologia com mais popularidade (200 mil estrelas no GitHub) em comparação com React (197 mil estrelas), para o desenvolvimento do *frontend* da Plataforma Interna da Grama será utilizado **React**. Vue possui uma documentação completa, mas apresenta escassez de *plugins*. A escolha tem por base o facto de que a plataforma a desenvolver irá requerer alguns componentes específicos como, por exemplo, gráficos e calendários. Portanto, o maior número de *plugins* disponíveis para o React, juntamente com a sua maior comunidade e suporte, será uma vantagem adicional.

A empresa fornece suporte quer em React, quer em Vue, mas como o aluno já teve contacto com React não terá de aprender uma linguagem de raiz. Logo, agilizará o processo de desenvolvimento e mitigará os riscos associados à falta de experiência em novas tecnologias.

4.2 Backend

Tal como para o *frontend*, existem várias tecnologias para desenvolver o *backend* de aplicações, por exemplo: **Django**, **Express.js**, **Laravel**, **Ruby on Rails**, **Spring**, **ASP.NET Core** [48]. Nesta secção serão analisadas algumas das mais utilizadas pela comunidade e pela empresa. São estas **Django**, **Spring Boot**, **Quarkus** e **Express.js**.

4.2.1 Django

Django é uma *framework web open-source* em Python e segue o padrão *Model View Template* (MVT) [49]. É rápido, escalável e, como é uma *framework* popular, conta com uma grande comunidade e boa documentação. Todavia, tem uma curva de aprendizagem elevada, os erros ao nível dos *templates* são difíceis de encontrar e corrigir. O sistema de *Object Relational Mapper* (ORM) tem poucas funcionalidades, visto que não permite aos utilizadores utilizar o conjunto de ferramentas do Python [50].

4.2.2 Spring Boot

Spring Boot é uma ferramenta Java que permite a criação de aplicações *standalone* e *production-grade* utilizando a *framework* Spring [51]. É escalável e, como é baseado em Java, é bastante simples de encontrar suporte e documentação. Contudo, tem a desvantagem de não oferecer uma fácil gestão de dependências, o que pode levar à inclusão de dependências desnecessárias [48].

4.2.3 Quarkus

Quarkus é uma *framework* Java, *full stack* nativa em *Kubernetes*. Desenvolvida pela Red Hat e com o objetivo de tornar Java a plataforma líder em *Kubernetes* permitindo aos *developers* fazer aplicações para correrem na nuvem [52]. Quarkus tem algumas vantagens como muita documentação simples disponível e, comparativamente a Spring Boot, tem tempos de inicialização e *reload* mais rápidos. Contudo, a instalação é um processo complexo [53].

4.2.4 Express.js

Express.js é uma *framework web* que corre em Node.js. É utilizada para desenvolver *single page*, *multipage* e *hybrid web applications* [54]. Uma das vantagens é o facto de utilizar JavaScript/TypeScript, ou seja, é possível partilhar a mesma linguagem com *frontend* simplificando a lógica de negócio. No entanto, tem desvantagens como: baixo poder de computação e não tem mensagens de erro simples, podendo tornar alguns erros morosos de resolver [55].

4.2.5 Comparação das tecnologias

A Tabela 4.2 permite comparar as tecnologias para o desenvolvimento do *backend* da Plataforma Interna da Grama (* - dados do dia 03/11/2022).

Tabela 4.2: Tabela comparativa das tecnologias analisadas para o *backend*.

	Django	Spring	Quarkus	Express.js
Ano de lançamento	2005	2003	2019	2010
Linguagem	Python	Java / Kotlin	Java / Kotlin	JavaScript
Vantagens	Rápido e escalável; Bom suporte da comunidade e boa documentação.	Simple e flexível; Bom suporte da comunidade e boa documentação.	Muito documentado; Rápido tempo de inicialização e de <i>reload</i> .	<i>Performance</i> elevada; Fácil aprendizagem; Bom suporte da comunidade.
Desvantagens	Curva de aprendizagem elevada; Poucos recursos no sistema ORM; Arquitetura muito monolítica.	A criação de um projeto pode incluir dependências extra desnecessárias; Curva de aprendizagem elevada.	Difícil instalação; Pouco suporte da comunidade.	Não é adequado para computação pesada.
Popularidade* (GitHub)	67.1 mil ★ ^[56]	63.9 mil ★ ^[57]	10.9 mil ★ ^[58]	58.8 mil ★ ^[59]

4.2.6 Tomada de decisão

Relativamente ao *backend*, para além das vantagens e desvantagens das tecnologias analisadas (secção 4.2) e do suporte fornecido pela empresa, o aluno já teve contacto com **Django** e **Spring Boot**. Contudo, existe uma maior preferência pela linguagem Java em relação a Python (quer pelo aluno, mas principalmente pela empresa), logo, a tecnologia escolhida para o desenvolvimento do *backend* será **Spring Boot**. Embora o Quarkus apresente melhor *performance*, é relativamente recente e tem pouco suporte da comunidade, daí a escolha final ser **Spring Boot**.

Capítulo 5

Arquitetura

Neste capítulo, será apresentada e detalhada a escolha do tipo de arquitetura a utilizar no desenvolvimento da Plataforma Interna da Grama com base nos requisitos levantados (secção 3). De seguida, será apresentada a arquitetura a usar no desenvolvimento dos módulos de ausências e recrutamento da plataforma e, depois, será representado o desenho da arquitetura utilizando o modelo C4. Por fim, serão definidos os diagramas ER desenhados para as bases de dados da plataforma.

5.1 Arquitetura Monolítica vs Microserviços

Arquiteturar um *software* é um processo importante pois ajuda a definir os componentes e como estes interagem entre si de forma a atingir os atributos de qualidade definidos para o sistema. Como a plataforma será construída de raiz, é importante definir qual o melhor estilo arquitetural para a mesma. Ou seja, seguindo uma abordagem mais monolítica ou fazendo a separação das funcionalidades, seguindo uma arquitetura de microserviços.

Um monólito é uma aplicação de uma camada só onde os vários componentes interagem entre si. De um modo geral, numa **arquitetura monolítica** o código é realizado e mantido numa única *codebase* que, posteriormente, é compilado e executado como um único programa, o que facilita o desenvolvimento, realização de testes e *deployment*. Contudo, com o crescimento da aplicação torna-se difícil de efetuar a manutenção e a implementação de novas funcionalidades. Caso exista um *bug*, este afetará toda a aplicação, ou seja, se um serviço falhar, toda a aplicação irá falhar (*single point of failure*). Por fim, outra desvantagem é o facto de, para se fazer *redploy*, após a introdução de uma nova funcionalidade, é necessário fazer *deploy* de toda a aplicação [60].

A **arquitetura de microserviços** baseia-se na criação de uma aplicação separando-a em vários e pequenos serviços independentes que comunicam entre si através de *Application Programming Interfaces* (APIs). Cada serviço desenvolvido é responsável por uma tarefa específica e pode ser desenvolvido, testado e implementado de forma independente dos restantes serviços. O facto de estes serem

independentes facilita na manutenção e no desenvolvimento contínuo. O *deploy* de um novo serviço é feito isoladamente sem ser necessário fazer *deploy* de toda a aplicação. Outra vantagem é a possibilidade dos serviços serem implementados recorrendo a diferentes tecnologias e, também, a eliminação do *single point of failure*, isto é, se um serviço falhar a aplicação não falha por completo [61].

Das arquiteturas analisadas na anteriormente, a arquitetura monolítica obriga a uma dependência entre as funcionalidades, enquanto que os microsserviços são independentes, e a sua manutenção/atualização pode não afetar toda a aplicação.

Sabendo que estarão dois alunos a desenvolver módulos distintos para a plataforma em paralelo, foi importante ambos fazerem a análise individual dos vários tipos de arquitetura. Antes da tomada de decisão, os alunos debateram as vantagens e desvantagens dos tipos de arquitetura em conjunto com os orientadores da empresa, concluindo que a proposta de solução teria como base uma **arquitetura baseada em microsserviços**. Esta decisão permite a separação dos módulos durante o desenvolvimento da versão inicial da plataforma e, no futuro, ao adicionar novos módulos, estes poderão ser realizados de forma independente, sem afetar toda a aplicação.

5.2 Arquitetura baseada em Microsserviços

Uma vez que a escolha para o desenvolvimento da plataforma foi uma arquitetura de microsserviços, é importante compreender em detalhe esse estilo arquitetural. Trata-se de uma arquitetura onde a aplicação é separada em serviços que comunicam entre si através de APIs. Estes serviços são independentes, pouco acoplados, de fácil manutenção e atualização, nunca pondo em causa o correto funcionamento dos restantes serviços que compõem a aplicação [62].

Para a melhor utilização deste tipo de arquitetura recorrem-se a padrões (*microservice architecture pattern language*), ou seja, como descrito no *site* do *Hillside group*: “cada padrão é uma regra de três partes, que expressa uma relação entre um determinado contexto, um problema e uma solução.” [63]. Sendo assim, um padrão representa a solução a um problema num determinado contexto aquando do *design* de uma arquitetura baseada em microsserviços.

Existem muitos tipos de padrões que podem ser aplicados nas arquiteturas de microsserviços tais como a decomposição da aplicação, gestão dos dados, realização de testes, comunicação entre serviços, segurança ou padrões das interfaces [64].

De seguida, serão apresentados e descritos alguns dos padrões de microsserviços relevantes para o projeto em questão. Estes padrões foram escolhidos por serem os mais relevantes para a implementação de uma plataforma *web* baseada em microsserviços e também por serem os mais utilizados pela empresa em alguns dos seus projetos.

5.2.1 Base de dados por serviço

Neste tipo de arquiteturas os serviços são fracamente acoplados e desenvolvidos de forma independente o que levanta a questão: qual a estrutura da base de dados? Deste modo, pode-se decidir quais as bases de dados mais adequadas para cada serviço e, assim, estes não têm de partilhar uma camada de dados entre eles (Figura 5.1).

Através deste padrão, as alterações na base de dados de cada microsserviço não afetam os restantes microsserviços. As bases de dados individuais não podem ser acedidas diretamente por outros microsserviços e os dados persistentes são acedidos apenas através de APIs. Contudo, existem algumas desvantagens como a gestão entre bases de dados relacionais e não relacionais, e a dificuldade de implementação de transações ou pesquisas que envolvam múltiplos serviços, devendo até ser evitadas. Devem cumprir pelo menos dois dos requisitos do teorema CAP: consistência (*consistency*), disponibilidade (*availability*) e tolerância às partições (*partition tolerance*) [65].

Para a implementação deste padrão existem três possíveis métodos [66]:

- **Tabelas privadas por serviço:** cada serviço tem um conjunto de tabelas que só devem ser acedidas por esse mesmo serviço;
- **Schema por serviço:** cada serviço tem definido um *schema* específico;
- **Base de dados por serviço:** cada serviço tem o seu próprio servidor de base de dados.

A implementação de transações entre serviços pode ser realizada utilizando o padrão *Saga*. Uma *saga* é uma sequência de transações que atualiza cada microsserviço e publica um evento que definirá a próxima transação. Em caso de sucesso efetua-a mas, caso falhe, são efetuadas transações para anular as anteriores [67].

Para a implementação das pesquisas existem dois padrões:

- **API Composition:** utiliza um *API Composer* para implementar uma pesquisa invocando os microsserviços que contêm os dados e, por fim, junta-os em memória [68];
- **Command Query Responsibility Segregation (CQRS):** as operações CRUD (*Create, Read, Update e Delete*) são tratadas de forma distinta das consultas nos microsserviços. Enquanto as operações CRUD lidam diretamente com os dados, as consultas são realizadas utilizando réplicas de leitura [69].

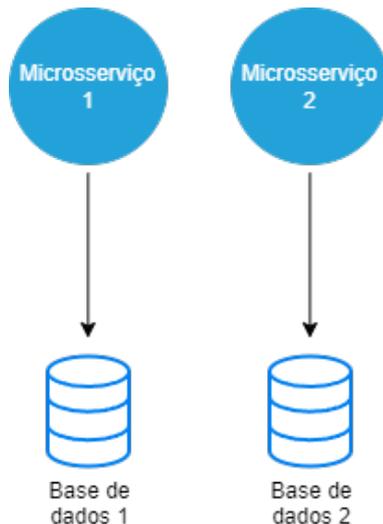


Figura 5.1: Representação esquemática do padrão base de dados por serviço.

5.2.2 Base de dados partilhada

Este padrão, tal como o anterior, procura auxiliar na questão: qual a estrutura da base de dados numa arquitetura de microserviços? Neste caso, a solução apresentada baseia-se na existência de uma base de dados partilhada entre os vários serviços (Figura 5.2). Cada serviço acede à informação através de transações ACID locais [70].

Traz vantagens como a facilidade de utilizar apenas uma base de dados e o uso de transações ACID que ajudam a garantir a consistência dos dados. No entanto, é importante garantir que a base de dados atenda aos requisitos de todos os microserviços. Por outro lado, uma desvantagem dessa abordagem é que, como todos os serviços acedem à mesma base de dados, pode haver bloqueios por transações morosas, o que pode impactar o desempenho e a disponibilidade dos serviços [71].

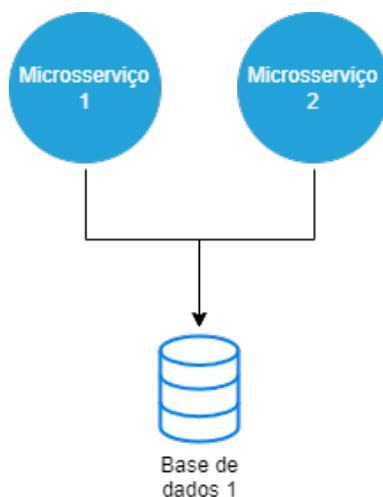


Figura 5.2: Representação esquemática do padrão base de dados partilhada.

5.2.3 API Gateway

Uma preocupação relevante é a comunicação de um cliente com a aplicação ou com um microserviço específico. Implementando este padrão consegue-se obter uma *gateway* acessível através de um *endpoint* único que irá mapear os pedidos dos clientes para os microserviços correspondentes. Esta API deve garantir que o cliente que faz o pedido está autorizado para tal através de um *token* de acesso [72] (Figura 5.3).

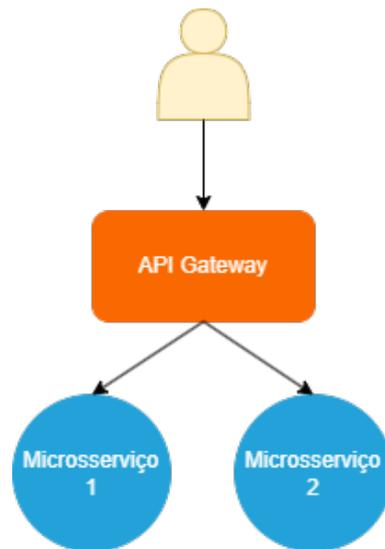


Figura 5.3: Representação esquemática do padrão *API Gateway*.

5.2.4 Token de acesso

Após um pedido ser feito por um cliente, é fundamental garantir que o utilizador esteja autorizado a efetuar essa ação. Para tal, utiliza-se um *token* de acesso (por exemplo, um *JSON Web Token* [73]). A *API Gateway* é responsável por autenticar o pedido e enviar o *token* que será utilizado para identificar quem efetuou o pedido em cada solicitação aos serviços. Esse *token* serve como uma forma de garantir a segurança e autenticidade das interações entre o cliente e os microserviços [74] (Figura 5.4).

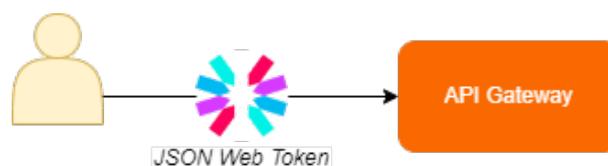


Figura 5.4: Representação esquemática do padrão *token* de acesso.

5.3 *Cloud computing*

Antes da análise das plataformas que oferecem serviços na nuvem, é importante compreender o conceito de computação na nuvem. Isto porque a escolha da arquitetura para o desenvolvimento dos módulos de ausências e recrutamento da Plataforma Interna da Grama será baseada em microsserviços.

A computação na nuvem (*cloud computing*) baseia-se na utilização de vários serviços através da *internet*, nos quais o pagamento é baseado conforme a sua utilização. Traz vantagens como: *performance*, escalabilidade, confiabilidade e o custo, pois não há a necessidade de comprar *hardware* e/ou *software* [75].

A computação na nuvem pode ser dividida em três tipos [76]:

- **Nuvem pública:** os provedores de serviços fornecem os recursos via *internet*;
- **Nuvem privada:** os recursos estão alocados fisicamente nos escritórios da empresa. Embora sendo possível pagar para os provedores de serviços hospedarem nuvens privadas;
- **Nuvem híbrida:** combinação de nuvens públicas e privadas.

Os serviços fornecidos são divididos em três tipos principais [77]:

- **Infraestrutura como um serviço:** corresponde a servidores, máquinas virtuais e serviços de armazenamento;
- **Plataforma como um serviço:** serviços que fornecem ambientes de desenvolvimento, gestão e testes;
- **Software como um serviço:** corresponde a *software* acessível através da *internet*.

A Amazon Web Services (AWS), a Azure e a Google Cloud são plataformas na nuvem que oferecem aos utilizadores vários serviços de acordo com as suas necessidades. No âmbito do estágio é importante analisar as três de forma a perceber qual a mais adequada ao projeto.

5.3.1 Amazon Web Services

A Amazon Web Services é uma plataforma na nuvem com mais de 200 serviços distribuídos pelo mundo. A sua infraestrutura de nuvem é considerada a mais extensa, confiável e segura. O custo dos serviços é determinado pelo seu uso, e varia de acordo com cada serviço oferecido [78].

Apresenta benefícios como a alta disponibilidade através do isolamento em diferentes regiões geográficas e a segurança robusta. Destaca-se ainda pela baixa

latência, escalabilidade e flexibilidade [79]. Todavia, para novos utilizadores, a vasta gama de serviços disponíveis na AWS pode representar um desafio em termos de complexidade. A curva de aprendizagem pode ser íngreme ao explorar e configurar os serviços adequadamente. No entanto, existem algumas limitações em alguns serviços e opções limitadas para estratégias híbridas [80].

5.3.2 Microsoft Azure

Tal como na AWS, a plataforma na nuvem da Azure conta com mais de 200 produtos e serviços. Os preços dos serviços podem ser analisados através de uma calculadora disponível no *website* [81]. A segurança é uma das principais áreas de foco da Azure, garantindo a proteção dos dados e dos sistemas dos clientes. Além disso, possui um bom suporte para a migração de serviços da Microsoft, facilitando a transição para a nuvem. Ou seja, a integração de *software* que não seja da Microsoft pode exigir mais complexidade e esforço. Oferece também recursos para integração com sistemas de nuvens híbridas [80].

5.3.3 Google Cloud

A Google Cloud consiste num conjunto de recursos físicos e virtuais espalhados por todo o mundo nos centros de dados da Google. Tal como nas plataformas anteriores, o pagamento é feito tendo em conta os serviços utilizados [82]. A Google Cloud integra bem os serviços da Google, especializa-se em áreas como *machine learning* e as suas APIs são bem documentadas. Por outro lado, em comparação com as plataformas anteriores, a Google Cloud não tem muitos serviços disponíveis. A necessidade de suporte extra é caro, e o *download* de informação da Google Cloud Storage pode-se tornar dispendioso [83].

5.3.4 Comparação entre as plataformas

Através da Tabela 5.1 podemos analisar de uma forma mais objetiva as três plataformas. A Google fornece também uma tabela [84] através da qual é possível pesquisar serviços entre as três plataformas tendo em conta o objetivo do utilizador.

Tabela 5.1: Tabela comparativa dos provedores de serviços na nuvem.

	AWS	Azure	Google Cloud
Ano de lançamento	2006	2010	2008
Vantagens	Segurança; Disponibilidade; <i>Performance.</i>	De fácil integração com <i>software</i> da Microsoft; Apresenta suporte para sistemas de nuvens híbridas.	Boa integração com serviços da Google; APIs bem documentadas.
Desvantagens	Complexidade devido à quantidade de serviços; Poucas soluções para uma estratégia híbrida.	Integração complexa com <i>software</i> não da Microsoft.	<i>Download</i> de informação da Google Cloud Storage é caro; Suporte técnico extra pode-se tornar dispendioso.

Pela análise da Figura 5.5, é possível analisar o crescimento do mercado dos provedores de serviços na nuvem anteriormente analisados. Embora o recente crescimento da Microsoft Azure, a AWS continua a ser a plataforma mais popular entre a comunidade.

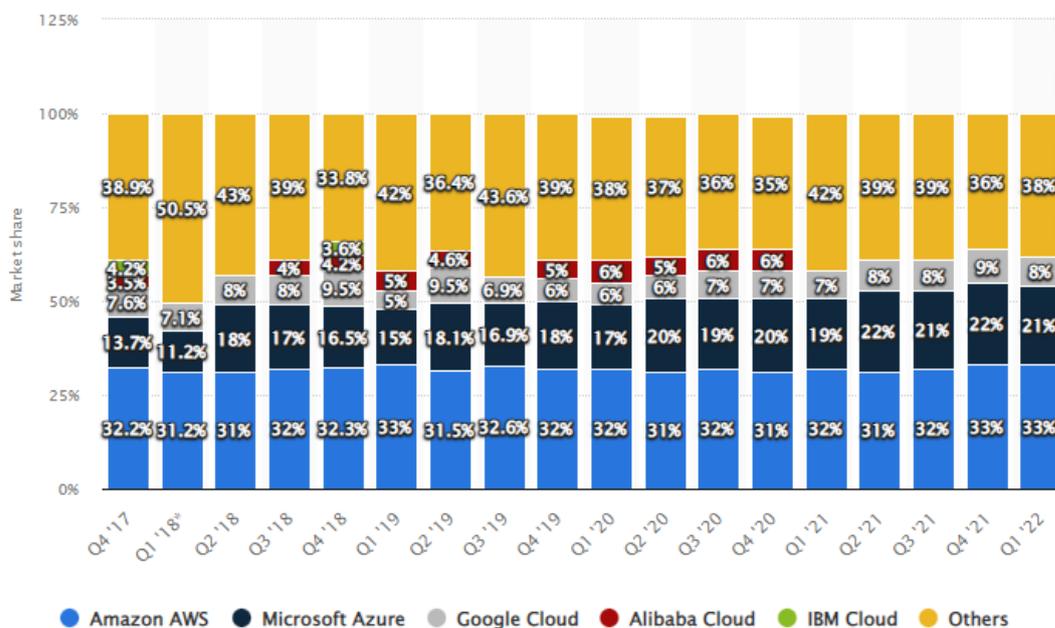


Figura 5.5: Crescimento no mercado dos provedores de serviços na nuvem ao longo do tempo [85].

5.3.5 Tomada de decisão

Após o debate e decisão dos alunos que iriam utilizar uma arquitetura baseada em microsserviços ambos repetiram o processo de pesquisa individual para definir qual o provedor de serviços na nuvem a utilizar. Novamente, os alunos debateram entre eles e com os orientadores da empresa e concluíram que a AWS seria a plataforma a utilizar. A decisão esteve também aliada ao facto de ambos terem alguma experiência com alguns serviços e porque também é a plataforma na qual a Grama tem mais experiência.

5.4 Desenho da arquitetura

Ao longo desta secção, será apresentada e detalhada a arquitetura da Plataforma Interna da Grama, tendo em conta as decisões tomadas nas secções anteriores. A representação da arquitetura será realizada através do Modelo C4.

5.4.1 Amazon Web Services

Dos vários recursos disponíveis na AWS, os seguintes serão incluídos na arquitetura da Plataforma Interna da Grama.

- **Amazon CloudFront:** utilizado para a distribuição do conteúdo estático (HTML, CSS, JavaScript, entre outros) para os utilizadores [86];
- **Amazon Cognito:** serviço de autenticação, autorização e gestão de utilizadores [87];
- **Amazon Elastic Compute Cloud (EC2):** permite instanciar máquinas virtuais para executar aplicações [88];
- **Amazon Relational Database Service (RDS):** utilizado para a criação de bases de dados relacionais [89];
- **Amazon Simple Email Service (SES):** serviço de *emails* da Amazon [90];
- **Amazon Simple Storage Service (S3):** permite o armazenamento de dados na nuvem [91];
- **Elastic Load Balancing (ELB):** serviço que distribui o tráfego de entrada por vários destinos/dispositivos virtuais [92].

5.4.2 Definição da arquitetura

Para a representação da arquitetura da plataforma será utilizado o modelo C4. Este é um método utilizado para representar arquiteturas de forma mais abstrata, começando com uma visão mais ampla e terminando numa visão mais detalhada. Muitas vezes este modelo é apresentado utilizando o exemplo Google Maps, ou seja, quando existe interesse numa determinada área é necessário fazer mais *zoom* para se analisar com mais detalhe.

Embora não seja obrigatória a representação de todos os níveis, este modelo pode-se dividir em 4 níveis (Figura 5.6):

- **Contexto:** representa o contexto do sistema e as relações com os atores e os sistemas externos;
- **Containers:** apresenta a decomposição do sistema em *containers* (por exemplo, serviços ou bases de dados) e as suas interações;
- **Componentes:** apresenta a decomposição dos *containers* em vários componentes relacionando-os entre si;
- **Código:** representa, em formato *Unified Modeling Language* (UML), cada componente.

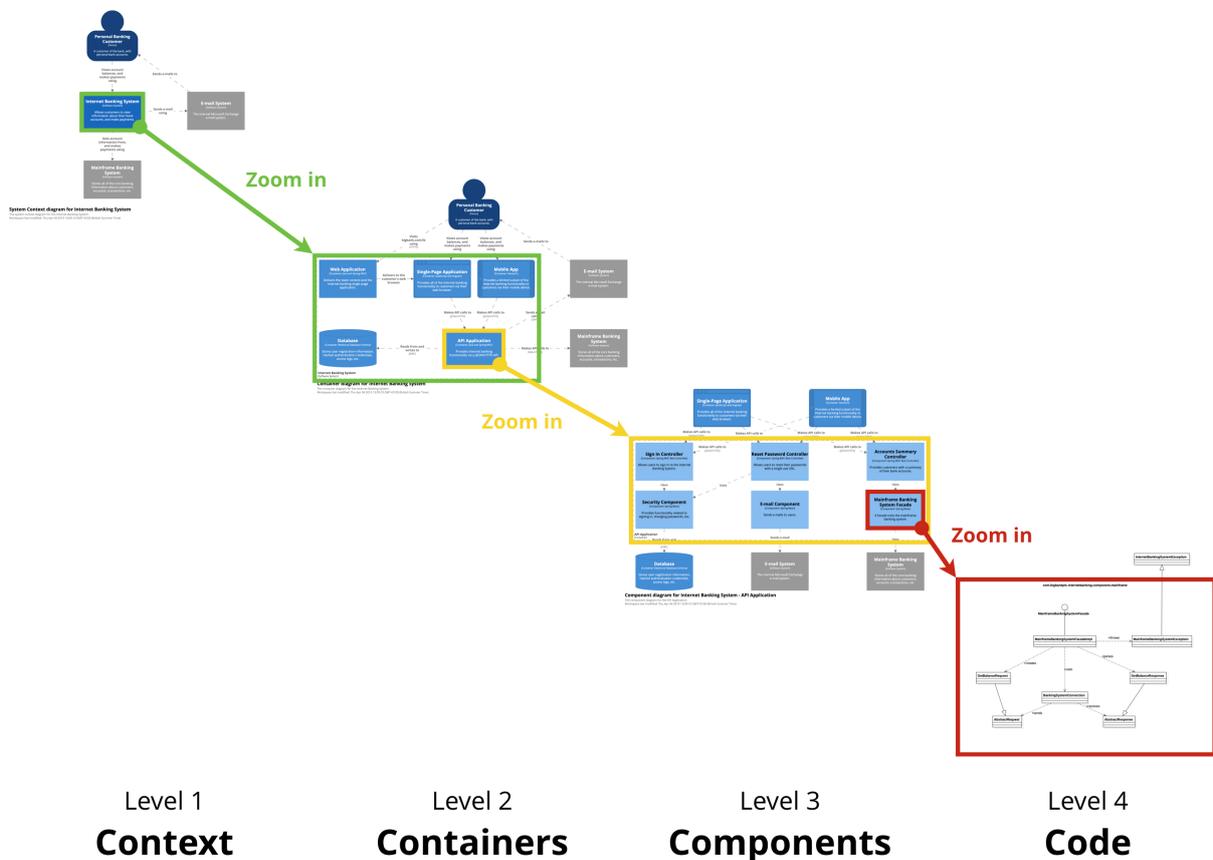


Figura 5.6: Exemplo do modelo C4 [93].

De seguida serão apresentados apenas os diagramas de contexto, *containers* e componentes da plataforma. A representação do nível de código exigiria um grande nível de detalhe de cada componente e, sendo esta opcional, não será incluída.

Contexto

O diagrama de contexto representa a visão mais ampla do sistema, ou seja, são representados os atores da plataforma, bem como os serviços externos que a plataforma utilizará.

Deste modo, o cenário (Figura 5.7) apresenta os **três** atores da plataforma:

- **Colaborador:** efetua e consulta os seus pedidos de ausência;
- **Gestor:** gere as ausências dos colaboradores da(s) sua(s) equipa(s) e visualiza a *pipeline* de recrutamento;
- **Administrador:** gere toda a plataforma, administra as ausências e também a *pipeline* de recrutamento.

Os atores interagem com a Plataforma Interna da Grama e esta utilizará os serviços da AWS para a realização de várias operações.

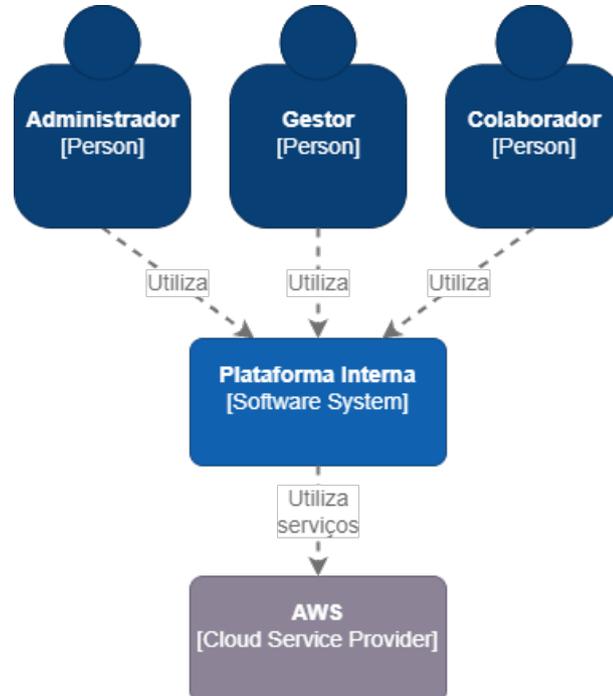


Figura 5.7: Diagrama de contexto da Plataforma Interna da Grama.

Containers

Neste diagrama, o objetivo é decompor o sistema que representa a Plataforma Interna em diversos *containers*. Começando pelo *frontend* da aplicação, o que é apresentado aos utilizadores, é representado pela *Single Page Application* onde o conteúdo é obtido via *Amazon CloudFront*.

Sempre que for efetuado um pedido, este será enviado para um *load balancer* (ELB) externo que verificará os certificados dos pedidos encriptados. Caso estes sejam válidos, enviará o pedido já em *Hypertext Transfer Protocol* (HTTP) para a *API Gateway*. Este é autenticado pelo *Cognito* e, de seguida, redirecionado para o respetivo serviço.

Cada instância EC2 alojará um dos seguintes microsserviços:

- **Ausências:** serviço responsável por gerir os pedidos e as ausências dos colaboradores, bem como apresentar as métricas na *dashboard*. É utilizado o serviço SES da AWS para enviar *emails* relacionados com os pedidos de ausência;
- **Recrutamento:** serviço responsável por gerir a *pipeline* de recrutamento e os novos candidatos a esta adicionados.

A representação do serviço S3 (Figura 5.8) sem ligações serve para facilitar a leitura visto que quer o *CloudFront*, quer os microsserviços de ausências e recrutamento utilizam este serviço. O *Amazon CloudFront* obtém o conteúdo a ser apresentado na *Single Page Application* através de um *bucket* S3, enquanto que os microsserviços de ausências e recrutamento utilizam *buckets* S3 para guardar ficheiros.

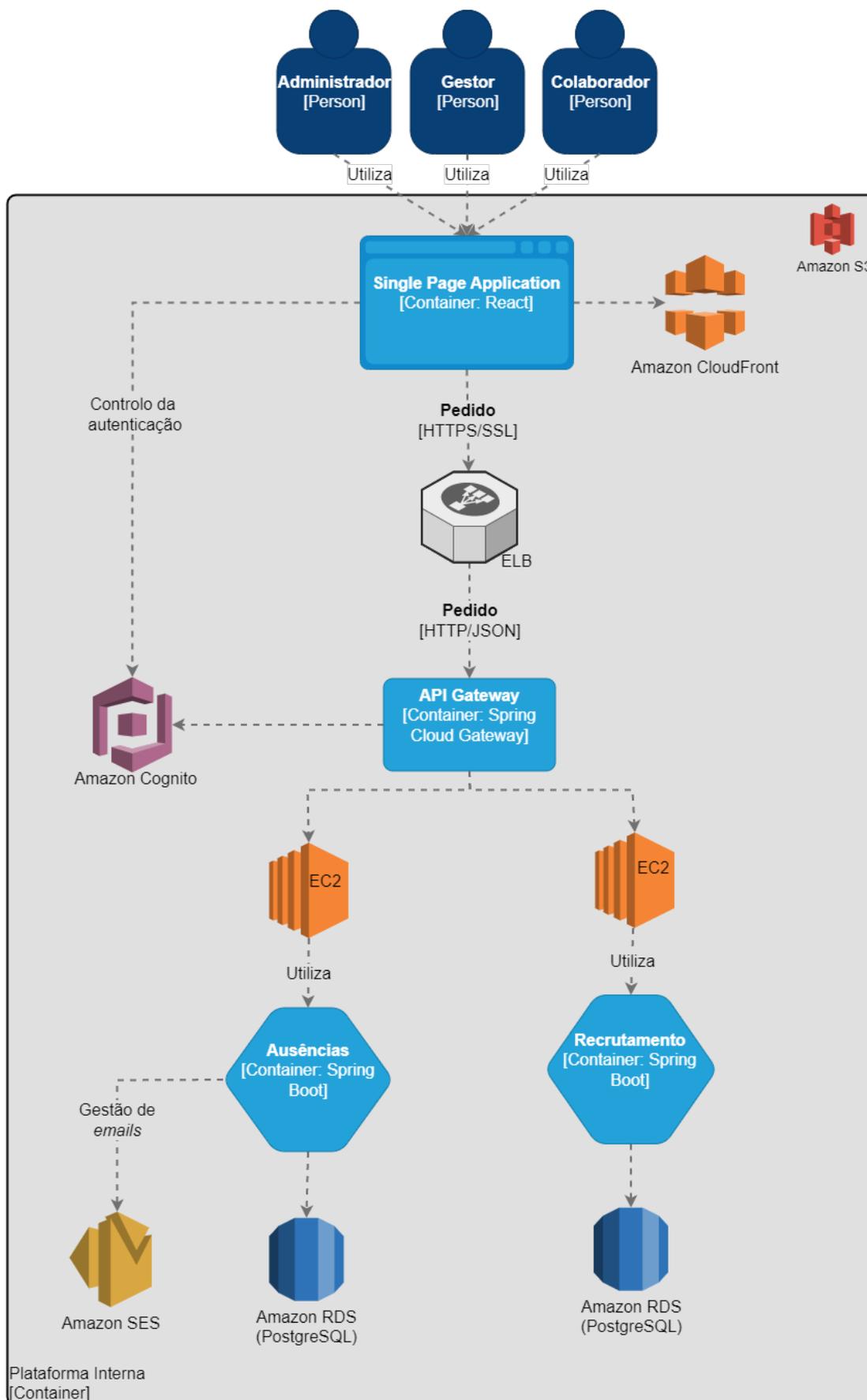


Figura 5.8: Diagrama de *containers* da Plataforma Interna da Grama.

Componentes

O diagrama de componentes corresponde à ampliação e visualização em maior detalhe de cada *container*. Ou seja, serão detalhados: a *API Gateway*, os dois microsserviços anteriormente representados e o *frontend* da plataforma.

A *API Gateway*, Figura 5.9, funcionará como um *proxy* entre o *frontend* e o *backend* da plataforma. Deste modo, os pedidos efetuados no *frontend* serão captados pela *API Gateway*, autenticados através do *Cognito* e, posteriormente, redirecionados para os respetivos serviços (ausências ou recrutamento).

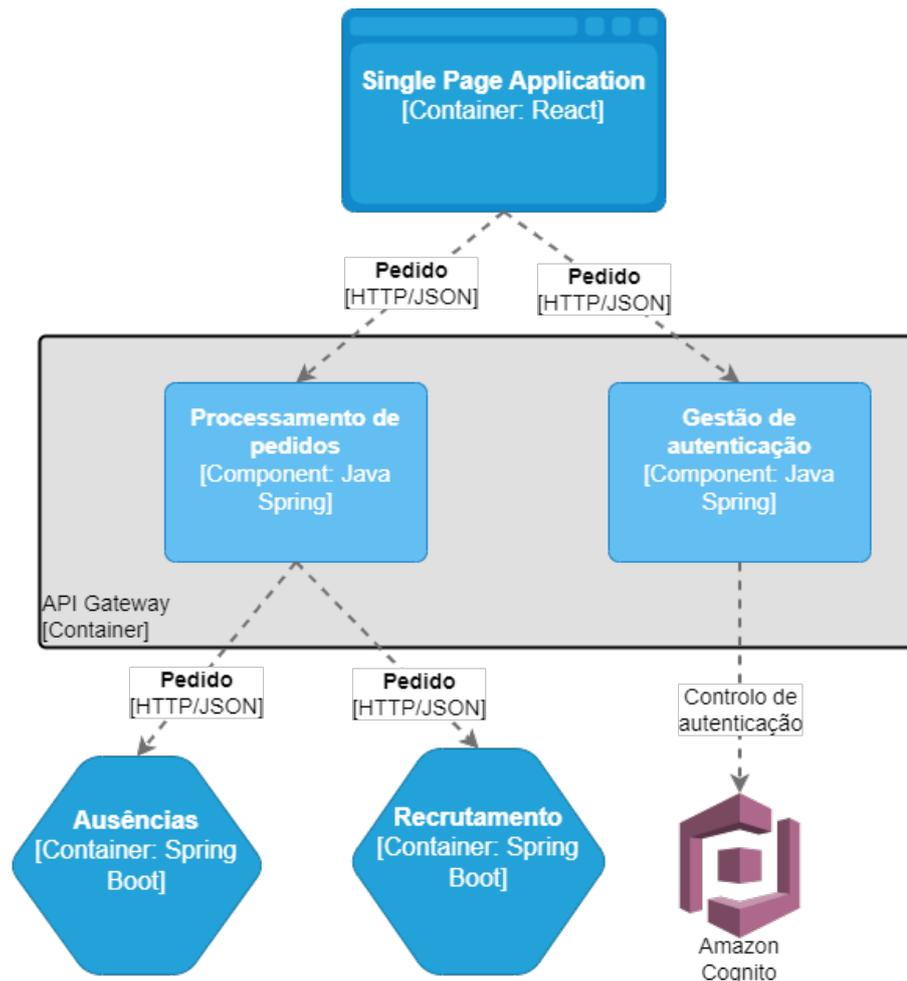


Figura 5.9: Diagrama de componentes relativo à *API Gateway*.

O microsserviço das **ausências** (Figura 5.10) será responsável por gerir os pedidos de ausência (aprovações, rejeições e/ou cancelamentos). Utilizará o Amazon S3 para guardar os anexos associados às ausências e o Amazon SES para enviar os *emails* referentes aos pedidos de ausência. Terá uma componente para definir algumas configurações relacionadas às ausências como, por exemplo, número de ausências dos colaboradores, tipos de pedidos de ausências ou dias de eventos especiais. O terceiro componente servirá para tratar das métricas a apresentar no *dashboard* dos utilizadores, isto é, número de ausências mensal por departamento e números de colaboradores ausentes nos próximos 7 dias. É ainda utilizada uma base de dados relacional (Amazon RDS) para guardar os dados do microsserviço.

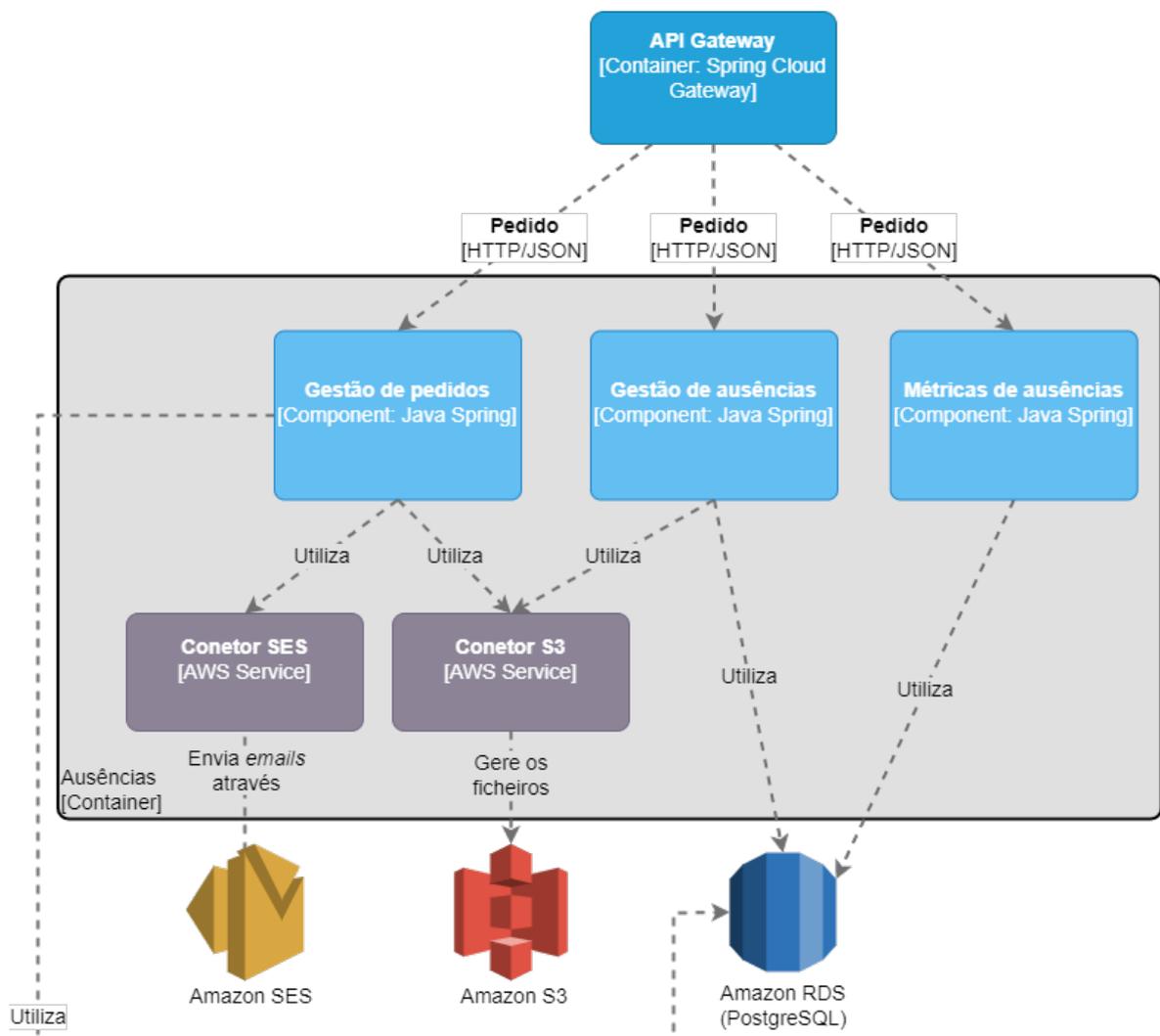


Figura 5.10: Diagrama de componentes relativo ao microserviço de ausências.

De seguida, é apresentado o microserviço responsável pela gestão do **recrutamento** (Figura 5.11), ou seja, a gestão de candidatos em processo de recrutamento como, por exemplo, últimas interações ou a sua fase no processo de recrutamento. Utiliza o Amazon S3 para guardar os ficheiros associados a cada candidato. O outro componente servirá para a customização da *pipeline* de recrutamento, isto é, quais as colunas a apresentar, qual o seu nome e em que ordem serão apresentadas. Ambos os componentes comunicam com uma base de dados relacional (Amazon RDS) onde serão guardadas as informações desde microserviço.

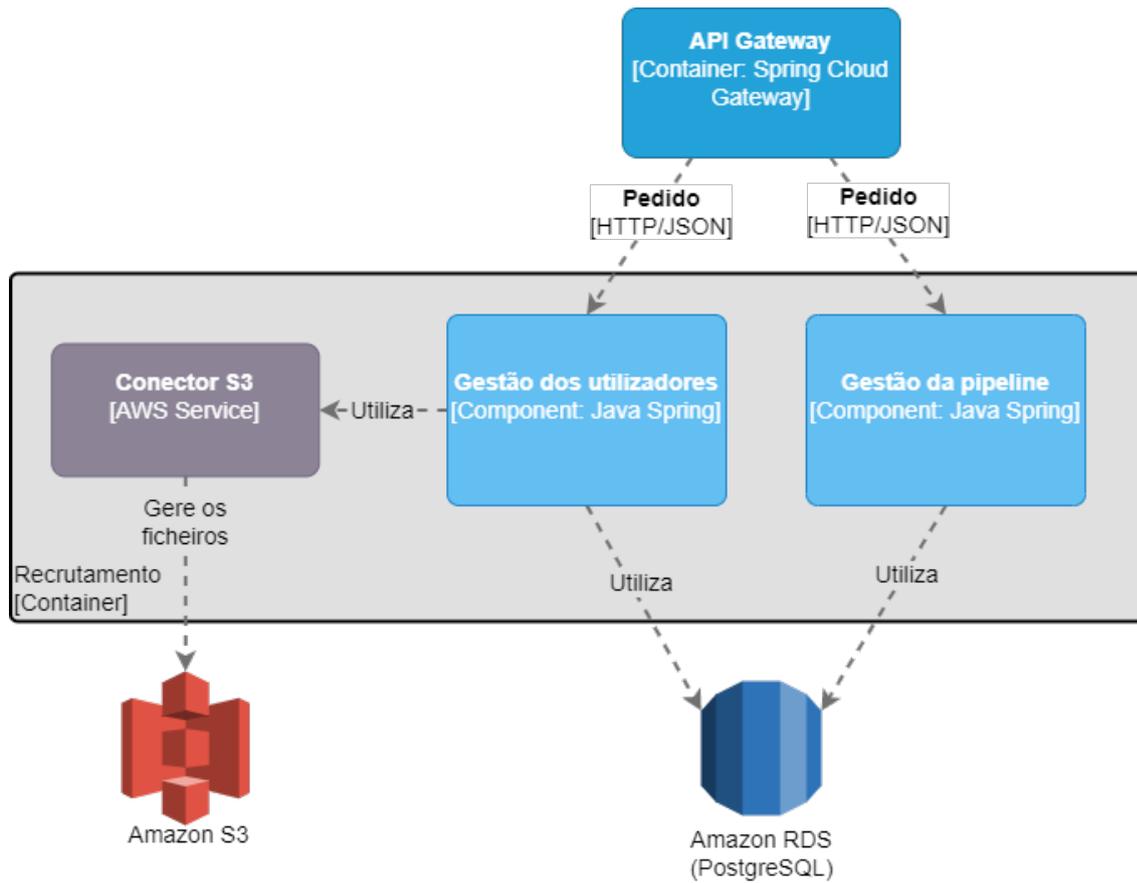


Figura 5.11: Diagrama de componentes relativo ao microserviço de recrutamento.

Por fim, na Figura 5.12, é demonstrado como estarão organizados os componentes que compõem o *frontend* da aplicação. Cada utilizador apenas poderá visualizar as páginas às quais está autorizado a aceder. Essa verificação será feita pelo *Router* e, caso o utilizador tenha permissão para ver a página, este será mapeado para a mesma que efetuará os devidos pedidos ao *backend* sendo os dados guardados no *state* da página.

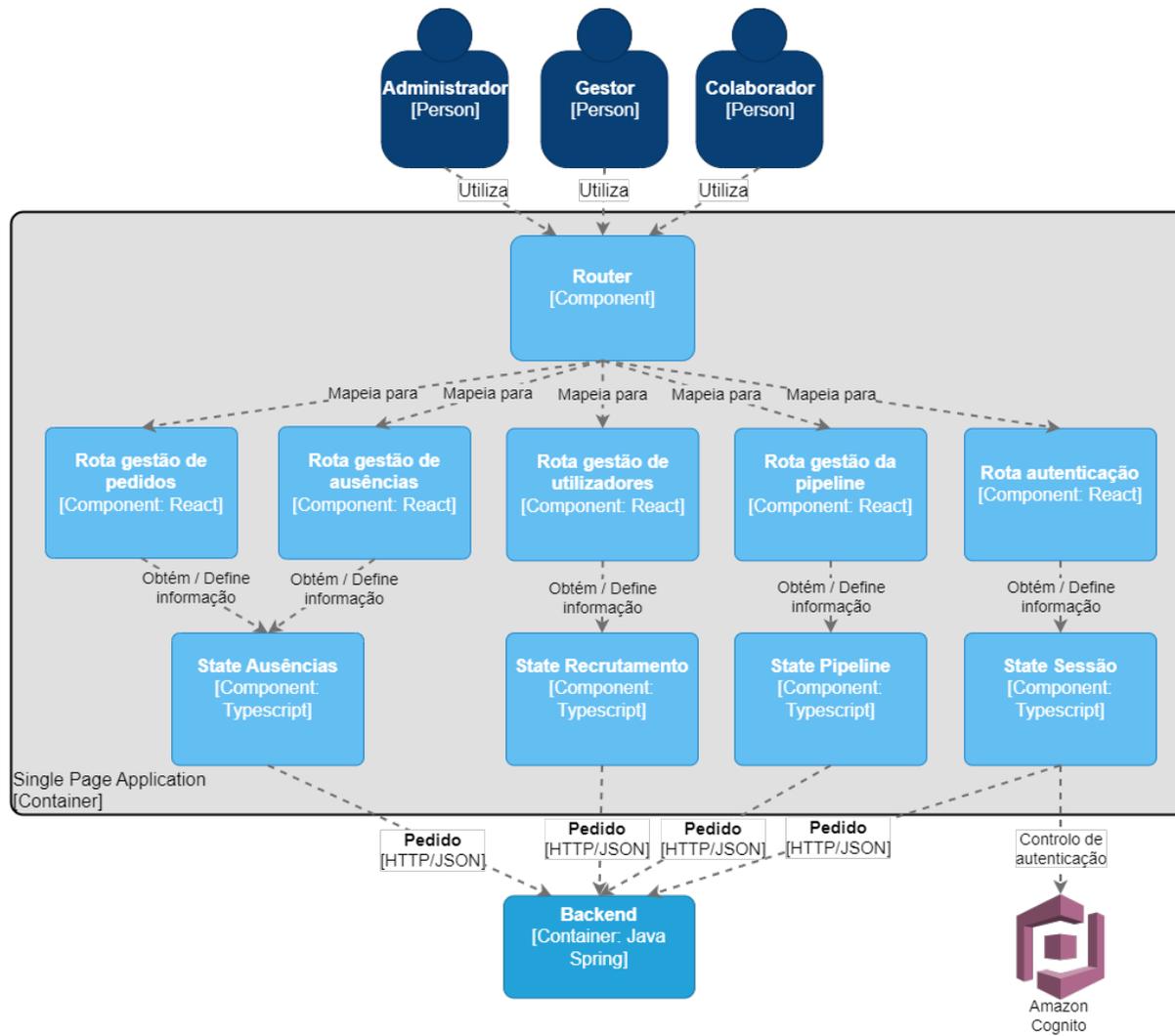


Figura 5.12: Diagrama de componentes relativo ao *frontend* da aplicação.

5.5 Diagrama Entidade-Relacionamento

O Diagrama de Entidade-Relacionamento (ER) permite visualizar qual a relação entre entidades num sistema. Para um melhor controlo das ações efetuadas a cada entrada da base de dados (*audit* [94]) as várias tabelas terão os seguintes campos: “created_by”, “created_at”, “updated_by”, “updated_at”, “deleted_by”, “deleted_at”. Contudo, nos diagramas abaixo, para uma melhor visualização, esses campos não se encontram representados. De notar que, para as operações de remoção, existirá uma propriedade que representará se a entrada foi apagada, isto é, será efetuado um *soft delete* de modo a garantir que a informação não será apagada e será sempre possível de recuperar.

De seguida, serão apresentados os diagramas ER para as bases de dados dos microsserviços representados na secção anterior.

Em relação ao microsserviço de ausências (Figura 5.13), uma ausência (ou pedido de ausência) é representada pela entidade “time_offs” que por sua vez se relaciona com a entidade “time_off_types” onde são guardadas informações sobre os vários tipos de ausências. O balanço dos utilizadores é representado pela entidade “users_balance” que relaciona um utilizador e o seu balanço de ausências.

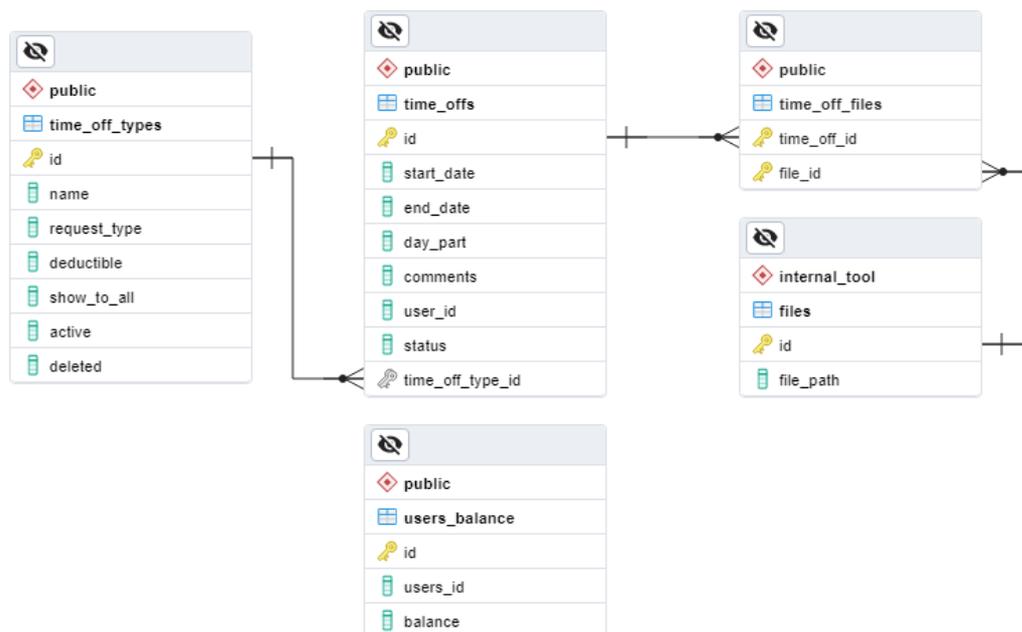


Figura 5.13: Diagrama ER relativo ao módulo de ausências.

No que toca ao processo de recrutamento (Figura 5.14), um candidato é representado pela entidade “candidates” que poderá estar associado a uma coluna (entidade “board_columns”). Já a entidade "interactions" representa as interações (entrevistas) de um ou vários utilizadores com os candidatos.

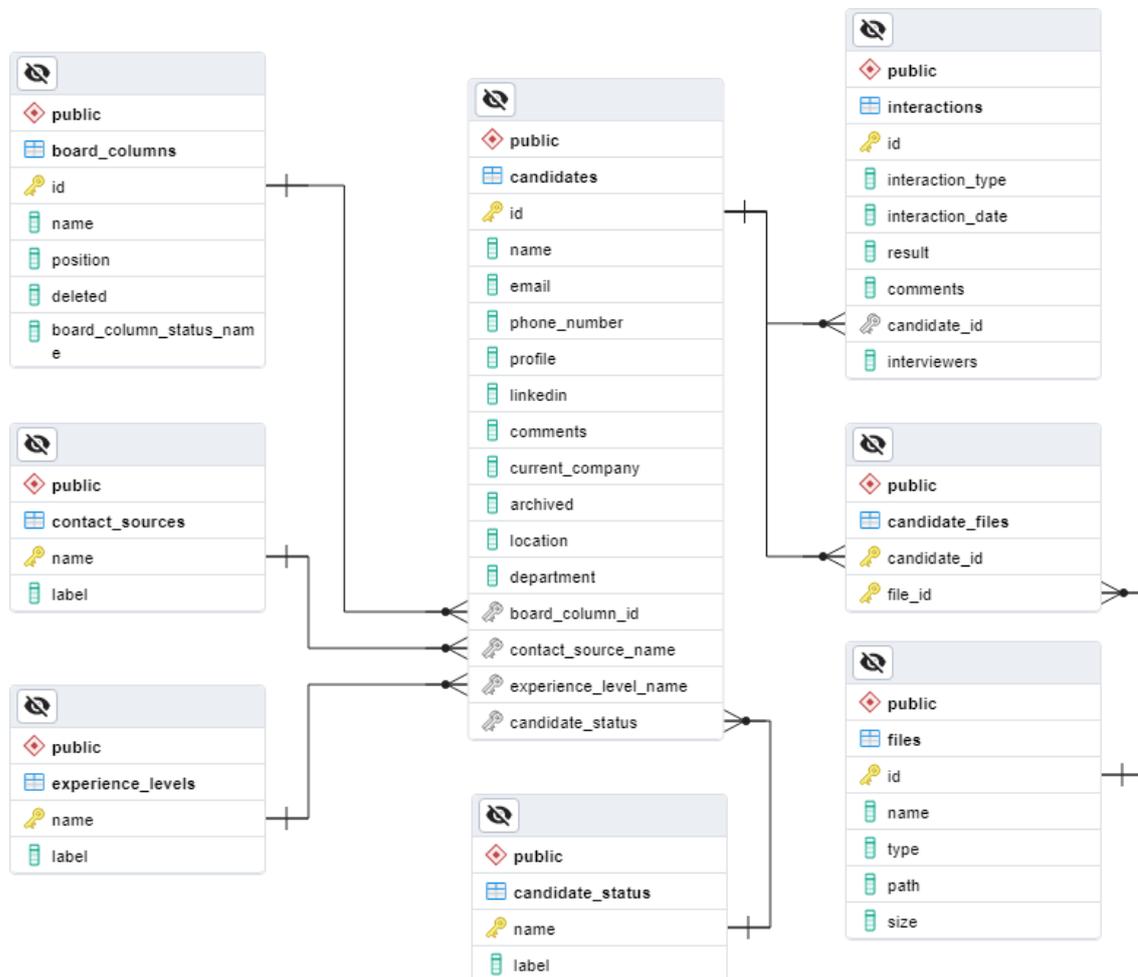


Figura 5.14: Diagrama ER relativo ao módulo de recrutamento.

Capítulo 6

Metodologia e Planeamento

Neste capítulo, será apresentada uma visão abrangente da metodologia utilizada no estágio, assim como o planeamento realizado para o primeiro e segundo semestres. Também serão detalhadamente analisados os riscos associados ao estágio e os seus planos de mitigação.

6.1 Metodologia

O processo de desenvolvimento de um *software* deve ser definido previamente com o objetivo de delinear e estruturar o processo em várias fases. Para tal, existem vários tipos de metodologias, desde as tradicionais como *Waterfall*, às metodologias ágeis como, por exemplo, *Scrum*.

Na década de 1990 existia uma grande necessidade de alterar os métodos tradicionais de desenvolvimento de projetos visto que estes estavam a falhar regularmente. Ou seja, os projetos eram entregues fora do tempo, do orçamento e sem o principal foco, agradar o cliente.

Então, em 2001, surge o **Manifesto Ágil** após um grupo de 17 *developers*, que já utilizavam metodologias ágeis nos seus processos (por exemplo, *Scrum* e *Extreme Programming*), se reunirem pois partilhavam das mesmas frustrações em relação ao desenvolvimento de projetos. Posto isto, uma metodologia ágil assenta em **quatro** valores principais:

- **Foco nas pessoas e nas suas interações** ao invés de nas ferramentas e nos procedimentos;
- **Ter um *software* funcional** é prioritário a uma documentação completa;
- **A colaboração com o cliente** é mais importante que a negociação do contrato;
- **A capacidade de reagir às mudanças** é mais importante do que seguir um plano.

Sendo assim, tal como documentado, enquanto há valor nas ferramentas e nos procedimentos, na documentação, no contrato e no plano, é mais valorizado o **foco**, o **software funcional**, a **colaboração com o cliente** e a **capacidade de reagir às mudanças**. Com estes valores não se pretende desvalorizar a documentação, mas sim definir as prioridades e deixar o processo de desenvolvimento mais flexível [95].

No decorrer do estágio, foi utilizada uma metodologia de trabalho ágil, baseada em *Scrum*. A escolha foi sugerida pela empresa visto que é a abordagem utilizada nos seus projetos.

Scrum

O *Scrum* é uma metodologia ágil de desenvolvimento de projetos dividida em *sprints* (Figura 6.1). Um *sprint* é o período de tempo, normalmente de duas a quatro semanas, em que a equipa desenvolve um conjunto de tarefas para atingir um produto pronto para ser entregue. Após um *sprint* terminar é iniciado outro, ajustando as tarefas com base no *feedback* do cliente e dos *stakeholders* [96]. Durante o segundo semestre cada *sprint* teve a duração de duas semanas.

Utilizar *Scrum* como metodologia de trabalho traz vantagens como [97]:

- As equipas conseguem entregar as suas tarefas de forma mais rápida utilizando o tempo de forma mais eficiente;
- Os projetos grandes são divididos em *sprints* menores facilitando a gestão e a alteração dos requisitos;
- Os *sprints* permitem visualizar o esforço gasto;
- O cliente e os *stakeholders* conseguem fornecer *feedback* entre *sprints*;
- *Sprints* mais curtos facilitam a introdução do *feedback* recebido;
- O desenvolvimento e os testes são realizados durante a revisão do *sprint*.

Funções

Em *Scrum* existem três funções fundamentais, são estas:

- **Scrum Master:** ajuda o *product owner* e a equipa de desenvolvimento de forma a ultrapassar obstáculos e a conduzir o projeto ao sucesso;
- **Product owner:** responsável por construir, priorizar e gerir o *backlog* consoante o *feedback* do cliente e dos *stakeholders*;
- **Development team:** deve-se auto-organizar para implementar as funcionalidades listadas no *backlog*.

O aluno, no contexto do estágio, integra a equipa de desenvolvimento juntamente com o estagiário Gonçalo São Marcos e o *designer* responsável pela criação dos *mockups* das páginas da plataforma. Como forma de aprendizagem, o aluno participou na redação das *user stories*, embora esta seja uma tarefa do *product owner*. A função de *product owner* é desempenhada pela Susana Lourenço, enquanto que a função de *scrum master* será desempenhada pelo orientador da Grama, Marcos Calvo.

Cerimónias

No decorrer do desenvolvimento de um projeto, existem cinco tipos de cerimónias (eventos), ou seja, reuniões com o objetivo de garantir que o *scrum master*, o *product owner* e a equipa de desenvolvimento estão em sintonia. As cerimónias são as seguintes [96]:

- ***Sprint planning***: efetuado no início do *sprint* e com a presença de toda a equipa. Nesta reunião são definidas quais as tarefas a realizar nesse *sprint*;
- ***Daily stand-up***: reunião diária, idealmente realizada pela manhã, entre os membros da equipa. Cada um deve indicar o que fez no dia anterior, o que espera fazer no dia atual e se precisa de ajuda em alguma tarefa;
- ***Sprint review***: tem por objetivo apresentar o produto no final de um *sprint* bem como obter *feedback* dos *stakeholders* sobre o mesmo;
- ***Sprint Retrospective***: é a última cerimónia onde toda a equipa deve analisar o que foi feito de positivo e negativo o que se pode melhorar nas próximas iterações;
- ***Backlog refinement***: reunião realizada no final de um *sprint* cujo objetivo é rever o *backlog* garantido que este continua atualizado para os *sprints* futuros.

Durante o primeiro semestre foram realizadas *weeklies* de acompanhamento. Ao longo do segundo semestre foram realizadas *dailies*, *sprint planning* e *sprint reviews*. As tarefas foram estimadas utilizando a escala de Fibonacci (utilizaram-se os números 1, 2, 3, 5 e 8), onde uma tarefa cujo valor seja 8 foi repartida em subtarefas.

Artefactos

Em *Scrum* os artefactos são utilizados para uma melhor gestão do trabalho. São eles [96]:

- ***Product backlog***: lista de tarefas (requisitos, novas funcionalidades, correções de *bugs*) necessárias para o desenvolvimento do produto. Está em constante evolução;

- **Sprint backlog:** conjunto de tarefas retiradas do *product backlog* e que serão desenvolvidas no próximo *sprint*;
- **Increment:** produto obtido como resultado da conclusão de um *sprint*.

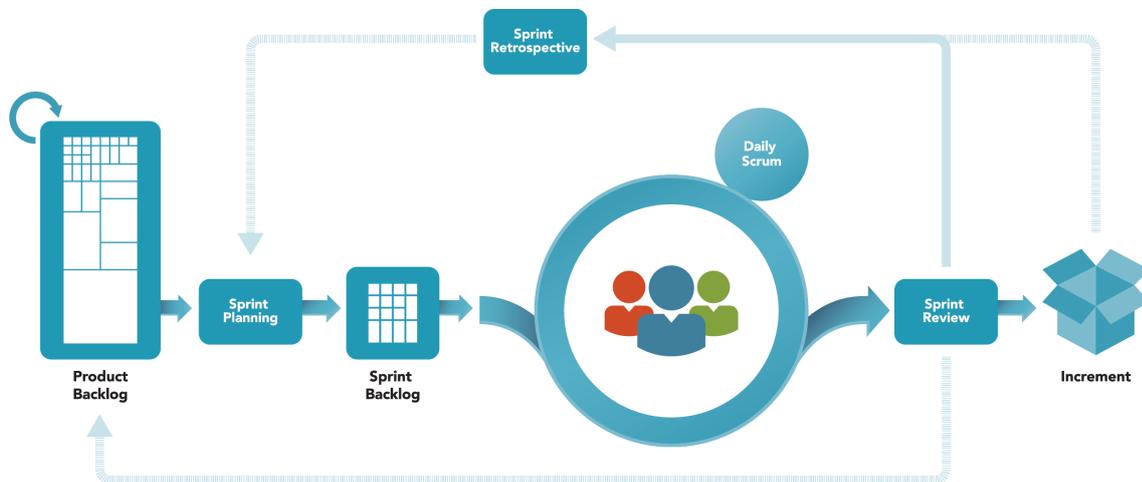


Figura 6.1: Metodologia ágil, *Scrum* [96].

6.2 Planeamento

Nesta secção é apresentado o planeamento temporal do estágio para os dois semestres.

No início de cada semestre, foi estabelecida uma lista de tarefas com base no plano de trabalho proposto no estágio curricular e nas reuniões entre os orientadores e o aluno. De seguida, foi elaborado um diagrama de *Gantt*, tendo em conta as tarefas definidas, com o objetivo de visualizar a ordem cronológica e a duração de cada uma.

No final de cada semestre, foi criado um novo diagrama de *Gantt* que representa a ordem cronológicas de realização das tarefas. A atualização contínua do diagrama auxiliou a acompanhar o progresso em relação ao plano inicial, permitindo identificar eventuais desvios e realizar os ajustes necessários. Por fim, é também possível comparar ambos os diagramas de cada semestre e analisar quais as diferenças entre o planeado e o cumprido.

6.2.1 1º Semestre

Durante o primeiro semestre, o aluno dedicou-se ao estudo do estado da arte, com base nos objetivos propostos para o estágio curricular. O foco foi encontrar concorrentes para a plataforma a ser desenvolvida e também analisar tecnologias e arquiteturas, a fim de tomar uma decisão sólida sobre as quais utilizar no desenvolvimento.

O aluno não só definiu requisitos e *user stories* para o projeto, mas também realizou a arquitetura do mesmo através do modelo C4 e de diagramas entidade-relacionamento.

Em paralelo com as tarefas enumeradas anteriormente, o aluno redigiu a documentação para o relatório intermédio e preparou-se para a defesa intermédia.

Planeamento inicial

Na Figura 6.2, é apresentado o plano inicial para o primeiro semestre. De notar que as tarefas apresentadas foram divididas em subtarefas.

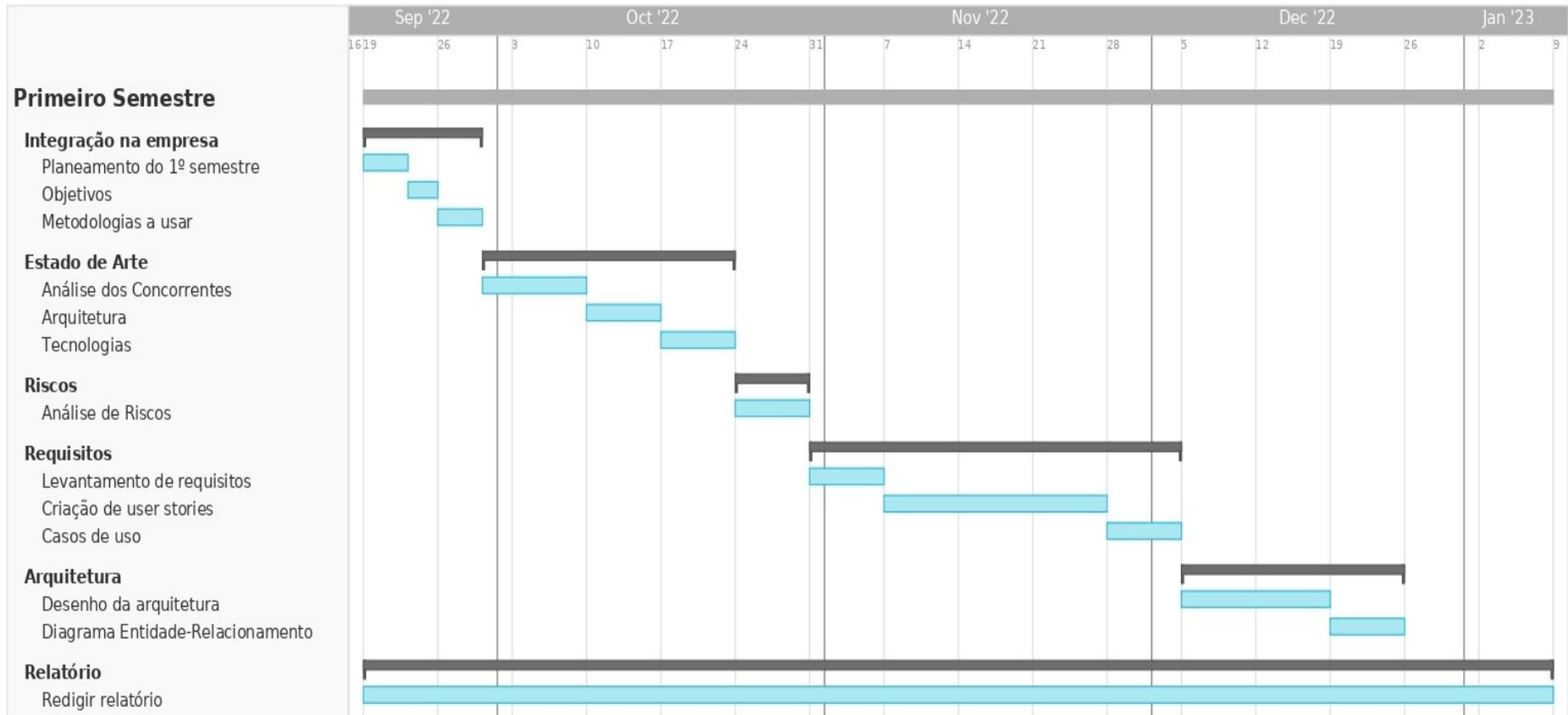


Figura 6.2: Planeamento de temporal do primeiro semestre.

Planeamento executado

A cronologia real do primeiro semestre é representada pela Figura 6.3. Pode-se visualizar que existiram alguns desvios em relação ao planeamento inicial, devido às sugestões dos orientadores sobre os próximos passos após a análise dos concorrentes. Isto porque o aluno deveria conhecer bem os requisitos da plataforma antes de começar a analisar tecnologias e arquitetura.

O maior atraso representado deveu-se à grande quantidade de *user stories*, o que prolongou a sua conclusão e, conseqüentemente, a realização e conclusão do estado da arte.

Pese embora os atrasos nos requisitos, o aluno conseguiu finalizar todas as tarefas planeadas para o primeiro semestre.

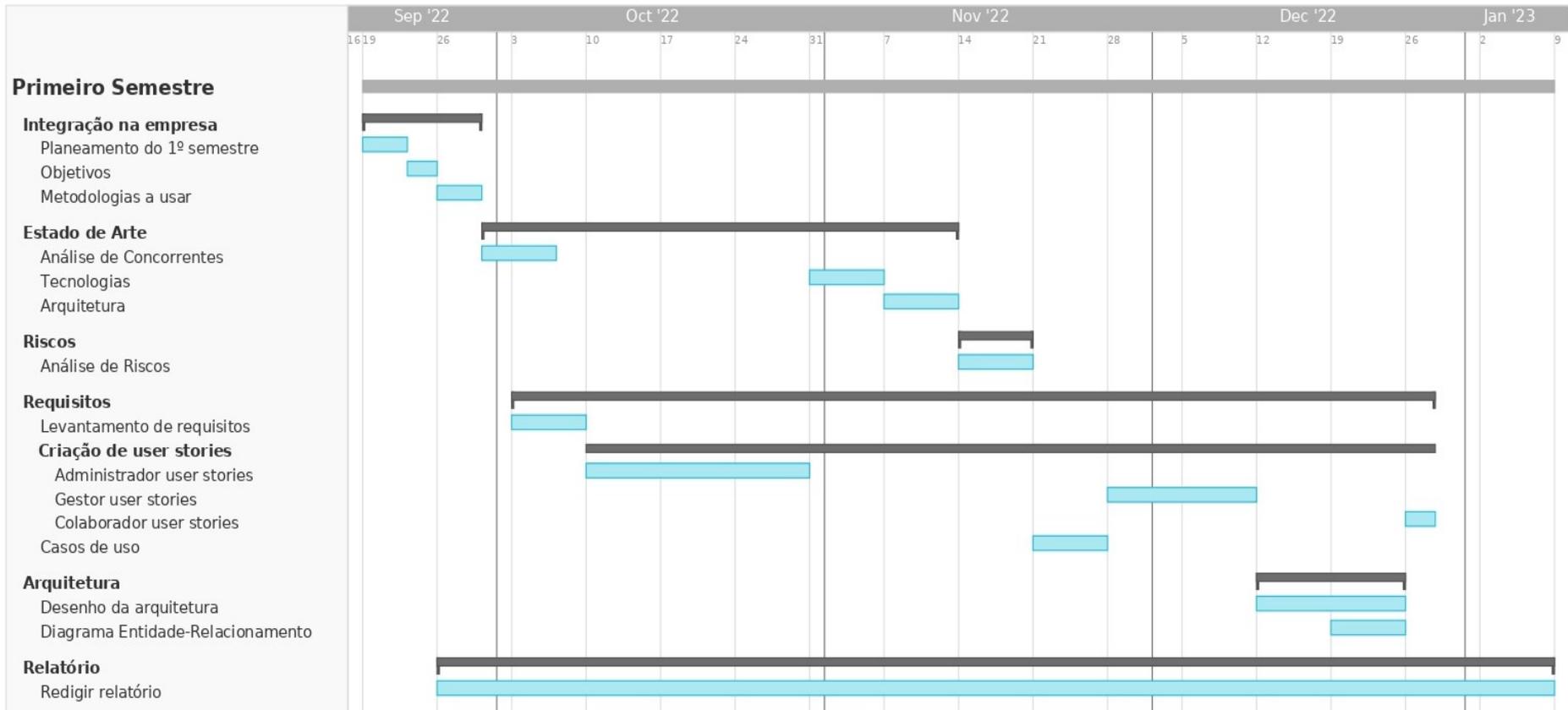


Figura 6.3: Diagrama de *Gantt* das tarefas concluídas ao longo do primeiro semestre.

6.2.2 2º Semestre

O segundo semestre foi maioritariamente focado no desenvolvimento da Plataforma Interna da Grama. Para tal seguiu-se uma metodologia baseada em *Scrum* com *sprints* de duas semanas.

Abaixo serão apresentados os diagramas de *Gantt* com o planeamento real e executado ao longo do segundo semestre. Pretende-se apresentar uma melhor visualização e comparação entre o planeamento efetuado no primeiro semestre para o segundo semestre e o trabalho realizado. De seguida, são descritas as tarefas realizadas ao longo dos *sprints* destinados ao desenvolvimento recorrendo às métricas disponibilizadas pela ferramenta Linear [98].

Planeamento inicial

O planeamento do segundo semestre foi realizado tendo por base a metodologia a utilizar (Figura 6.4). Isto é, como se seguiria uma metodologia baseada em *Scrum*, o semestre foi repartido em dez *sprints* de duas semanas cada. Os *sprints* foram iniciados com um *sprint planning* e concluídos com uma *sprint review* à exceção do último *sprint* (*sprint* 10). A conclusão de uma tarefa implica a realização de alguns testes, contudo a inclusão deste *sprint* tem como objetivo voltar a executar os testes realizados, bem como a revalidação dos requisitos funcionais e não funcionais, sem a adição de novas funcionalidades. Ao longo do semestre o aluno pretende redigir o relatório final, sendo as duas últimas semanas para a sua conclusão e preparação da defesa final.

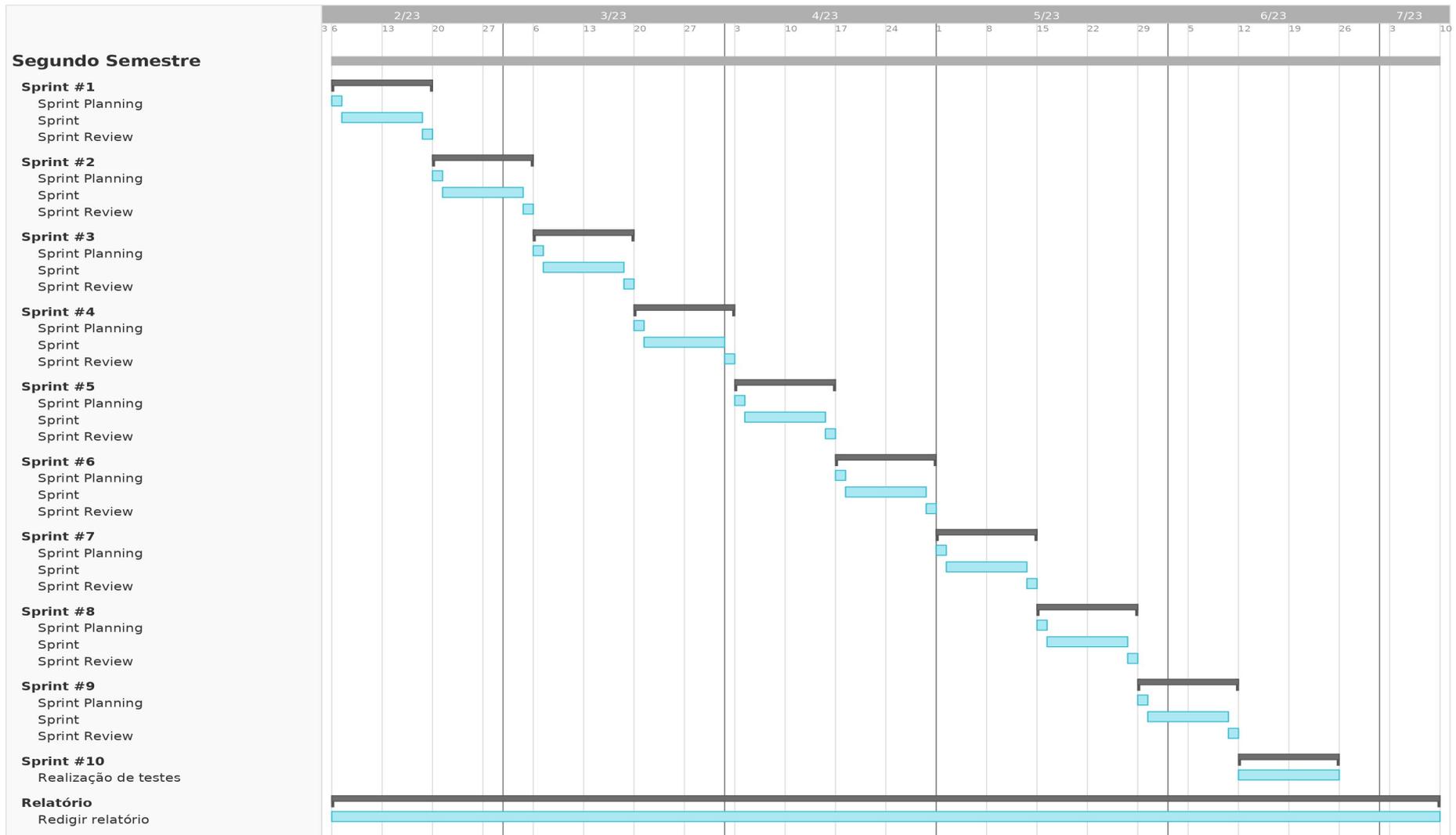


Figura 6.4: Planeamento de temporal do segundo semestre.

Planeamento executado

Com base na Figura 6.5, pode-se concluir que ocorreram alterações em relação ao planeamento inicialmente realizado. Deste modo, foram destinados dez *sprints* para o desenvolvimento, removendo assim o *sprint* destinado aos testes, como previamente apresentado. Isto pois uma tarefa só é considerada concluída quando passa nos respetivos testes, ou seja, ao longo de cada um dos dez *sprints*, foram realizados testes aos requisitos desenvolvidos.

Foi destinada apenas uma semana para os testes, onde foram novamente realizados os testes definidos ao longo do semestre e também validados os requisitos funcionais e não funcionais. Para a escrita do relatório, o aluno utilizou as primeiras duas semanas do semestre para aplicar o *feedback* recebido na defesa intermédia. Por fim, as últimas três semanas do semestre foram dedicadas à redação e melhoramento do relatório final, bem como à preparação para a apresentação e defesa do estágio em paralelo com as restantes tarefas.

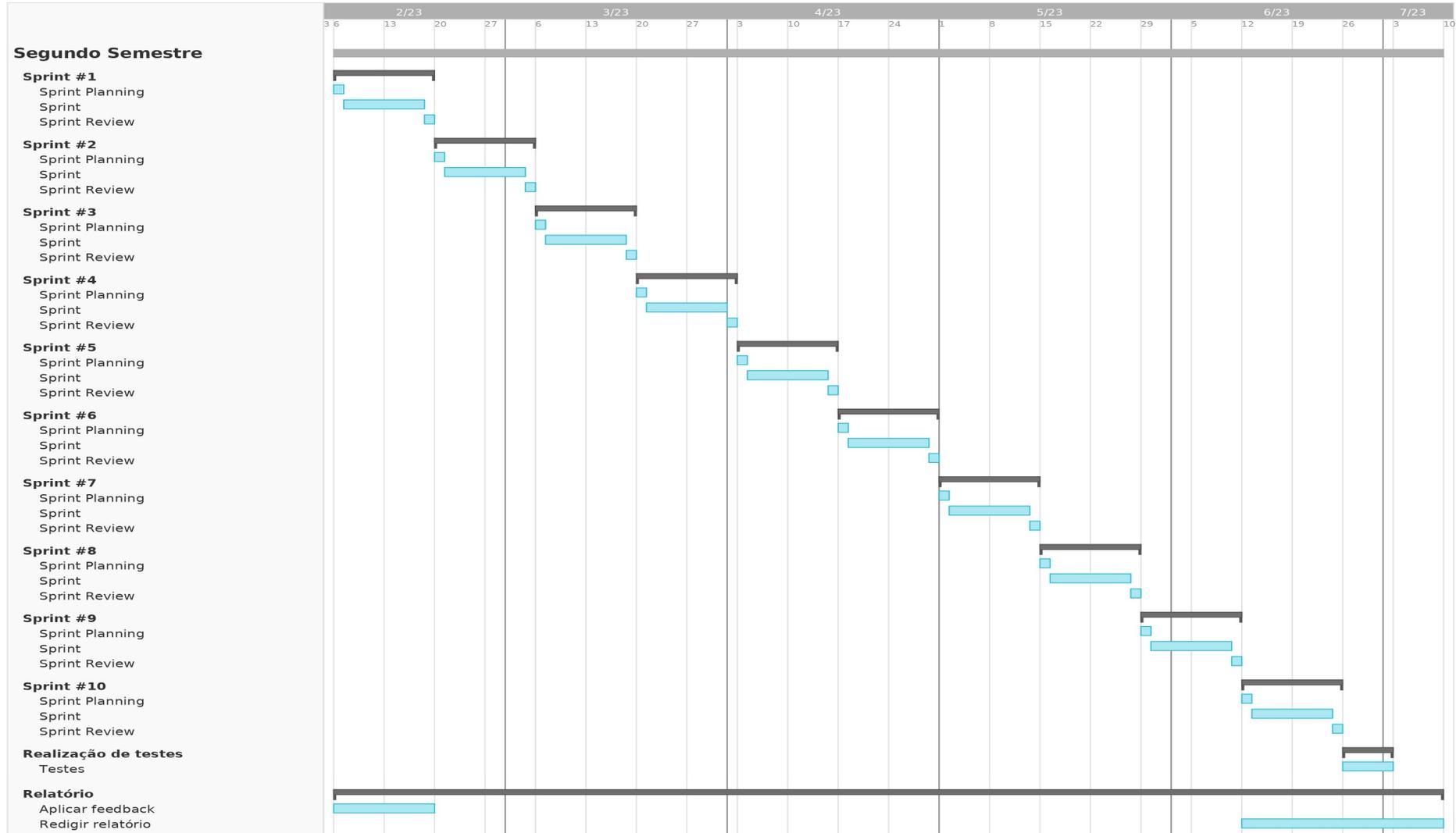


Figura 6.5: Diagrama de Gantt que representa o planejamento real do segundo semestre.

Tarefas realizadas

De seguida, são apresentadas as tarefas cumpridas ao longo dos dez *sprints* de desenvolvimento. Para auxiliar a explicação são apresentadas algumas métricas obtidas através da ferramenta Linear [98]. O aluno tinha dois quadros a ele destinados: o primeiro para realização de tarefas comuns aos vários módulos e o segundo para as tarefas dos módulos de ausências e recrutamento (Figura 6.6). Na ferramenta Linear um *sprint* é representado como um ciclo (*cycle*).

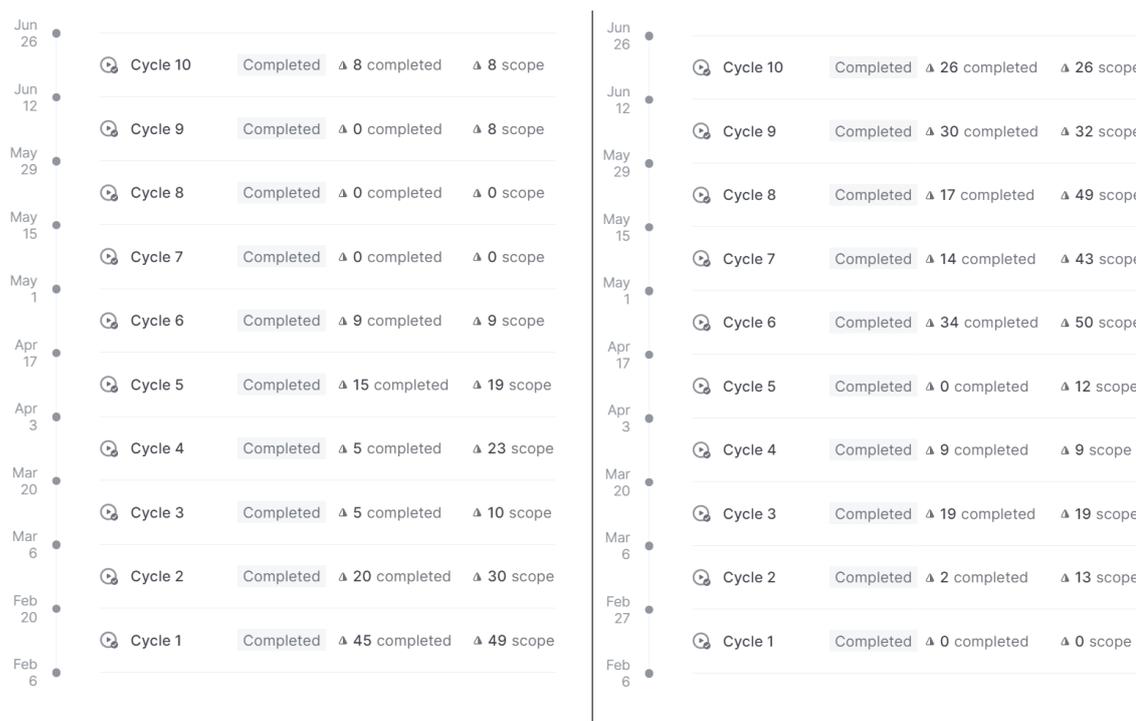


Figura 6.6: Tarefas realizadas durante os *sprints* no quadro de tarefas comuns (esquerda) e no quadro de tarefas dos microsserviços de ausências e recrutamento (direita).

É importante apontar que o aluno foi realizando tarefas em ambos os quadros, o que justifica, por exemplo, a não realização de qualquer das tarefas destinadas aos seus módulos no *sprint* 5. Isto porque o aluno estava focado na implementação da segurança, que era uma tarefa pertencente ao quadro das tarefas comuns entre os dois alunos. De uma forma objetiva, as tarefas realizadas pelo aluno ao longo dos *sprints* podem resumir-se da seguinte forma:

- **Ciclo 1:** Criação de alguns componentes comuns e configuração da AWS;
- **Ciclo 2:** Conclusão de componentes e *endpoint* para CRUD de candidatos;
- **Ciclo 3:** Conclusão do CRUD de candidatos e criação do *endpoint* da *pipeline* de recrutamento;
- **Ciclo 4:** Realização de páginas de autenticação;
- **Ciclo 5:** Implementação de segurança no microsserviço dos candidatos e de páginas de adicionar e listar candidatos;

- **Ciclo 6:** Desenvolvimento do primeiro início de sessão e páginas relacionadas aos detalhes e arquivo dos candidatos;
- **Ciclo 7:** Páginas relacionadas com a *pipeline* de recrutamento;
- **Ciclo 8:** Criação do microsserviço de ausências, *endpoint* de ausências e páginas para pedir ausências;
- **Ciclo 9:** Apresentação das ausências em calendário e em lista com os respectivos detalhes;
- **Ciclo 10:** Integração do serviço de *emails* da AWS, do serviço S3 e desenvolvimento da *API Gateway*.

6.3 Riscos

Antes do início do desenvolvimento de um projeto é importante identificar os potenciais fatores que podem por em causa o sucesso do projeto, ou seja, os riscos. Um risco é composto por um título, uma descrição, consequência, probabilidade de ocorrer, impacto no projeto e um plano de mitigação.

O plano de mitigação será definido tendo por base o impacto do risco no projeto. Enquanto que a probabilidade e o impacto são parametrizados entre baixo, médio e alto [99].

Os níveis de probabilidade podem ser descritos da seguinte forma:

- **Baixo:** improvável mas poderá ocorrer durante operações normais;
- **Médio:** alguma probabilidade de ocorrer durante operações normais;
- **Alto:** existe uma alta probabilidade de ocorrer frequentemente durante operações normais.

Já os níveis de impacto poderão ser definidos do seguinte modo:

- **Baixo:** o risco poderá ser controlado;
- **Médio:** a ocorrência deste risco implicará a aplicação de algumas medidas para o minimizar;
- **Alto:** o risco afetará de uma forma grave sendo necessário medidas imediatas para o mitigar.

De seguida, serão descritos os **cinco** riscos identificados associados ao desenvolvimento do estágio.

Risco #1

Identificador: R-1.

Título: *Design* não estar concluído a tempo.

Descrição: O *design* da plataforma é necessário para o aluno efetuar a implementação da mesma. Caso exista algum atraso ou até mesmo não esteja concluído a tempo do desenvolvimento poderá causar dificuldades na implementação.

Consequência: A implementação não seguirá os padrões desejados pela empresa.

Probabilidade: Baixa.

Impacto: Alto.

Plano de mitigação: Coordenação e comunicação com a equipa de *design* de modo a garantir que as funcionalidades vão de encontro às expectativas da empresa e, assim, não exista atraso nos *designs*.

Risco #2

Identificador: R-2.

Título: Requisitos mal definidos.

Descrição: A definição dos requisitos é muito importante pois estes devem refletir as funcionalidades que a empresa deseja para a plataforma.

Consequência: Os requisitos implementados não correspondem às expectativas da empresa.

Probabilidade: Média.

Impacto: Alto.

Plano de mitigação: No final de cada *sprint* o cliente pode testar a aplicação e fornecer *feedback* de alterações a fazer aos requisitos. O *feedback* do cliente será utilizado pelo aluno para ajustar os requisitos ao longo das iterações seguintes.

Risco #3

Identificador: R-3.

Título: *Scope* do projeto muito ambicioso.

Descrição: O *scope* do projeto muito ambicioso implica que existam muitos requisitos a desenvolver dentro da janela temporal prevista para a realização do estágio.

Consequência: O desenvolvimento do projeto excede o custo temporal devido a

atrasos causados pelas tarefas a desenvolver, comprometendo o produto final.

Probabilidade: Média.

Impacto: Alto.

Plano de mitigação: Avaliação e priorização cuidadosa dos requisitos levantados e, se necessário, efetuar uma revisão do *scope* ao longo do semestre.

Risco #4

Identificador: R-4.

Título: Incapacidade de adaptação às tecnologias por falta de experiência.

Descrição: O desenvolvimento do projeto está dependente das tecnologias escolhidas sendo que estas podem ser desconhecidas pelo aluno.

Consequência: Atraso na implementação das funcionalidades e no desenvolvimento do projeto.

Probabilidade: Média.

Impacto: Médio.

Plano de mitigação: A escolha das tecnologias deve ter por base os conhecimentos do aluno, a adequação ao projeto e a possibilidade de existir suporte quer pelo orientador, quer pela empresa. Sempre que o aluno sentir alguma dificuldade a solucionar um problema este deve comunicar ao orientador.

Risco #5

Identificador: R-5.

Título: Atraso no desenvolvimento do módulo de utilizadores.

Descrição: O módulo de utilizadores é fundamental para a realização de todos os outros módulos, existindo uma dependência dos restantes módulos em relação a esse. O módulo de utilizadores estará ser realizado em paralelo pelo outro estagiário.

Consequência: Atraso na implementação de algumas funcionalidades e no desenvolvimento do projeto.

Probabilidade: Média.

Impacto: Alto.

Plano de mitigação: Trabalho mais síncrono do aluno com o outro estagiário através de reuniões regulares, bem como organização das tarefas de modo ao desenvolvimento ter o mínimo de dependência no início do desenvolvimento.

6.3.1 Síntese dos riscos

Em seguida será apresentada, na Tabela 6.1, uma pequena síntese dos riscos levantados para o projeto.

Tabela 6.1: Tabela sumária dos riscos levantados.

Risco	Título	Consequência	Plano de mitigação
R-1	<i>Design</i> não estar concluído a tempo	A implementação não seguirá os padrões desejados pela empresa	Coordenação com a equipa de <i>design</i> de modo a garantir que as funcionalidades vão de encontro às expectativas da empresa
R-2	Requisitos mal definidos	Os requisitos implementados não correspondem às expectativas da empresa	Utilização do <i>feedback</i> do cliente para efetuar alterações em cada iteração
R-3	<i>Scope</i> do projeto muito ambicioso	O desenvolvimento excede o custo temporal e compromete o produto final	Avaliação e priorização dos requisitos e, se necessário, efetuar uma revisão do <i>scope</i> durante o semestre
R-4	Incapacidade de adaptação às tecnologias por falta de experiência	Atraso na implementação das funcionalidades e no desenvolvimento do projeto	Escolha das tecnologias com base nos conhecimentos do aluno, a adequação ao projeto e a possibilidade de existir suporte pelo orientador
R-5	Atraso no desenvolvimento do módulo de utilizadores	Atraso na implementação de algumas funcionalidades e no desenvolvimento do projeto	Trabalho mais síncrono do aluno com o outro estagiário. Início do desenvolvimento por tarefas que não exijam dependência desse módulo

6.3.2 Matriz de Riscos

A matriz de riscos é uma representação visual da análise dos riscos. Foi elaborada após o levantamento e análise de cada risco, seguido da identificação da probabilidade e impacto associado a cada risco. Na Figura 6.7, podemos visualizar os riscos consoante a sua probabilidade e impacto.

		Impacto		
		Baixo	Médio	Alto
P r o b a b i l i d a d e	Alta			
	Média		R-4	R-2 R-3 R-5
	Baixa			R-1

Figura 6.7: Matriz de riscos.

Capítulo 7

Desenvolvimento

O presente capítulo aborda a fase de desenvolvimento durante o segundo semestre do estágio curricular, destacando o processo adotado pelo aluno e pela equipa de desenvolvimento. Serão apresentadas as funcionalidades desenvolvidas em conformidade com os requisitos funcionais identificados no primeiro semestre (secção 3.3). Serão também abordados os riscos que surgiram durante o desenvolvimento e as medidas tomadas para mitigá-los, terminando com a apresentação do trabalho futuro.

7.1 Processo de desenvolvimento

A fase de desenvolvimento de um *software* envolve várias etapas e recursos essenciais para iniciar o processo de forma adequada. Tal como descrito anteriormente, a equipa de desenvolvimento é composta por um *designer*, dois programadores *full-stack* (os dois alunos), os respetivos orientadores responsáveis pelas *code reviews* e a *product owner*.

O processo de desenvolvimento começou pela **construção dos *mockups*** por parte do *designer*. Com o intuito de estar à frente do processo de desenvolvimento, o *designer* começou a construir os componentes e as diversas páginas antes dos alunos darem início ao desenvolvimento da plataforma. Desta forma, a *product owner* e o *scrum master* conseguiriam ter tempo para rever e aprovar os *mockups* de alta fidelidade. Só após a finalização e aprovação desses *mockups* é que a fase de desenvolvimento por parte dos alunos foi iniciada.

A **fase de desenvolvimento** teve início com a construção dos vários componentes comuns da plataforma. Isto exigiu uma colaboração mais sincronizada entre os dois alunos para definir as tarefas seguintes, tendo em conta as tarefas que cada um estava a desenvolver. Após a conclusão e aprovação dos componentes, o aluno iniciou a implementação dos seus respetivos módulos na plataforma (ausências e recrutamento). Sempre que uma tarefa era concluída, era submetida a *code reviews* realizadas pelo *scrum master* e outro *developer* da Grama, com o objetivo de evitar erros programáticos e alertar para correções relativas a boas práticas de escrita de código. Após essa etapa, as tarefas eram definidas como *ready to*

test.

Por fim, antes de uma tarefa ser marcada como concluída, foram efetuados testes para garantir que a mesma tinha o comportamento esperado e que todos os requisitos que esta englobava estavam presentes. Caso contrário, eram abertos novos *issues* de categoria *Bug* que eram resolvidos pelo aluno nos *sprints* seguintes conforme o *sprint planning*. Para além disso, foram realizados testes de integração com o respetivo microsserviço com o objetivo de garantir que os dados e as permissões dos utilizadores estavam a ser tratadas de forma correta.

Para efetuar o planeamento e estruturar o trabalho a ser realizado em cada *sprint*, a ferramenta utilizada foi o Linear [98]. Visto que a plataforma está a ser realizada de raiz e em paralelo com outro estágio foram criados **três** quadros:

- ***Internal Tool Common*** onde eram definidas as tarefas necessárias em todos os módulos, por exemplo, componentes comuns (*inputs* ou botões) ou configuração de serviços da AWS;
- ***Users and IT Management*** para os módulos de utilizadores e gestão de equipamentos;
- ***Time-off and recruitment*** para os módulos de ausências e recrutamento.

Cada um dos quadros acima indicados contém um *backlog* onde se encontram todas as tarefas a ser desenvolvidas. Além disso, os quadros contêm também uma tabela composta pelas **seis** seguintes colunas (Figura 7.1):

- ***Todo***: tarefas previstas desenvolver durante o presente *sprint*;
- ***In Progress***: tarefas em desenvolvimento;
- ***In Review***: tarefas em *code review*;
- ***Ready to test***: tarefas aprovadas e prontas a serem testadas;
- ***Done***: tarefas que passaram nos testes, dadas como concluídas;
- ***Canceled***: tarefas não implementadas e canceladas com devida justificação.

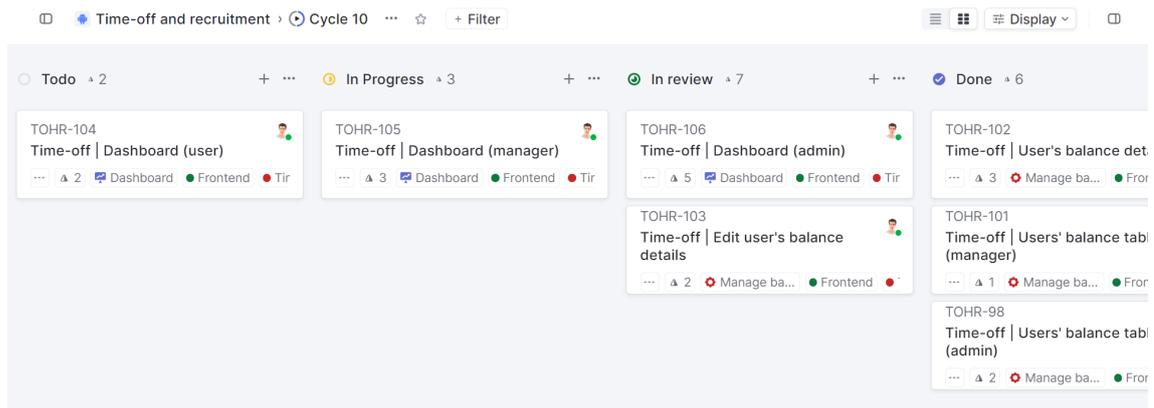


Figura 7.1: Exemplo da organização das tarefas na ferramenta Linear.

7.2 Repriorização dos requisitos

No início do processo de desenvolvimento, o aluno tinha uma compreensão clara dos requisitos com maior e menor prioridade que iria implementar. Isto deve-se à descrição e priorização dos requisitos funcionais realizada durante o primeiro semestre, conforme representado na secção 3.3.

Devido à limitada experiência do aluno em trabalhar em projetos de larga escala e com as tecnologias seleccionadas, inicialmente, o aluno definiu um grande conjunto de requisitos a desenvolver até ao final do estágio. Contudo, por volta de meio do segundo semestre e antes da conclusão do módulo de recrutamento, em conversa com os orientadores, o aluno decidiu que era necessário efetuar uma repriorização dos requisitos. Nesta fase, o aluno já estava mais familiarizado e confortável com as tecnologias utilizadas, o que lhe permitiu identificar cedo esse problema e, consequentemente fazer uso de uma das vantagens de utilizar uma metodologia ágil.

O principal objetivo da repriorização consistiu em definir novamente quais os requisitos mais importantes para o aluno realizar até ao final do período do estágio. Deste modo, sem desvirtuar o objetivo principal do estágio e sem comprometer a arquitetura previamente definida, foram analisados principalmente os requisitos relacionados com o módulo de ausências e definiu-se que a área de "Gerir tipos de ausências" não seria realizada.

O esforço necessário para a realização dessa área em concreto excederia o tempo definido para o estágio, ou seja, o aluno não só ficaria com essa área para concluir como algumas outras funcionalidades definidas como *Must*. Por isso, ficou definido que todos os outros requisitos ir-se-iam manter com a mesma prioridade, enquanto que a área "Gerir tipos de ausência" passaria para *Will not have*.

7.3 Requisitos funcionais desenvolvidos

Com base nos requisitos funcionais descritos na secção 3.3, foram novamente listadas, para cada módulo, as funcionalidades desenvolvidas e não desenvolvidas durante o semestre. Tal como referido anteriormente, existiram algumas alterações ao nível das prioridades de alguns requisitos. Nas tabelas abaixo essa alteração será notória.

É importante salientar que, nestas tabelas, não são incluídos os atores pois cada funcionalidade só é dada como concretizada (*done*) se estiver finalizada para todos os atores envolvidos.

Módulo de Autenticação

Como se pode inferir da Tabela 7.1, para o módulo de autenticação, todos os requisitos foram implementados.

Tabela 7.1: Requisitos funcionais desenvolvidos para o módulo de autenticação.

Área	ID	Requisito	Prioridade	Concluída
Autenticação	RF-1	Enviar convite para a plataforma	M	✓
	RF-2	Efetuar primeiro início de sessão	M	✓
	RF-3	Iniciar sessão	M	✓
	RF-4	Recuperar palavra-passe	M	✓
	RF-5	Terminar sessão	M	✓

Módulo de Ausências

Para o módulo de ausências é possível concluir, pela Tabela 7.2, todos os requisitos *Must have* foram desenvolvidos. No entanto, tal como referido anteriormente, a área de "Gerir tipos de ausências" ficou completamente por realizar após a sua repriorização. Posto isto, os requisitos não concluídos de prioridade *Should have* são bastante semelhantes e relacionados a filtros, contudo, não existiu tempo suficiente para os realizar e, conseqüentemente, os requisitos de prioridade inferior (*Could have*) também não foram realizados.

Tabela 7.2: Requisitos funcionais desenvolvidos para o módulo de ausências.

Área	ID	Requisito	Prioridade	Concluída
Dashboard	RF-6	Visualizar métricas	M	✓
	RF-7	Pesquisar e aplicar filtros à dashboard	S	✗
Gerir calendário	RF-8	Visualizar ausências em calendário - mês	M	✓
	RF-9	Visualizar ausências em calendário - semana	S	✓
	RF-10	Visualizar ausências em calendário - dia	S	✓
	RF-11	Visualizar ausências marcadas em grelha	W	✗
	RF-12	Aplicar filtros ao calendário	S	✗
Gerir pedidos	RF-13	Visualizar pedidos	M	✓
	RF-14	Aprovar pedido	M	✓
	RF-15	Rejeitar pedido	M	✓
	RF-16	Cancelar pedido	M	✓
	RF-17	Editar detalhes do pedido	M	✓
	RF-18	Pesquisar e aplicar filtros à lista de pedidos	S	✗
Gerir ausências	RF-19	Ver detalhes de uma ausência	M	✓
	RF-20	Editar detalhes de uma ausência	M	✓
Requisitar ausência	RF-21	Requisitar ausência - múltiplos dias	M	✓
	RF-22	Requisitar ausência - dia completo	M	✓
	RF-23	Requisitar ausência - dia parcial	M	✓

Notificar	RF-24	Receber <i>email</i> com pedido de aprovação	M	✓
	RF-25	Receber <i>email</i> com o resultado do pedido	M	✓
	RF-26	Enviar <i>email</i> informativo sobre a ausência	M	✓
	RF-27	Receber <i>email</i> informativo sobre a ausência	M	✓
Gerir tipos de ausências	RF-28	Visualizar tipos disponíveis	W	✗
	RF-29	Adicionar tipo	W	✗
	RF-30	Editar tipo	W	✗
	RF-31	Arquivar tipo	W	✗
Gerir balanço	RF-32	Visualizar ausências dos colaboradores ativos	M	✓
	RF-33	Editar balanço	M	✓
	RF-34	Pesquisar e aplicar filtros à lista de colaboradores	S	✗
Gerir eventos especiais	RF-35	Visualizar eventos especiais	C	✗
	RF-36	Adicionar eventos especiais	C	✗

Módulo de Recrutamento

Com base na Tabela 7.3, pode-se verificar que todos os requisitos *Must have* foram concluídos. Contudo, apenas dois requisitos de menor prioridade relacionados com a lista de candidatos não foram realizados, visto que, aquando do término do módulo de recrutamento o aluno ainda teria de desenvolver todo o módulo de ausências. Logo, a decisão tomada foi deixar os dois requisitos em *stand by* para realizar após o desenvolvimento do módulo de ausências, mas, como descrito acima, não houve tempo suficiente para concluir o módulo de ausências e consequentemente estes dois requisitos.

Tabela 7.3: Requisitos funcionais desenvolvidos para o módulo de recrutamento.

Área	ID	Requisito	Prioridade	Concluída
Gerir quadro de recrutamento	RF-37	Ver candidatos ativos	M	✓
	RF-38	Adicionar candidato	M	✓
	RF-39	Adicionar candidato a uma coluna específica	S	✓
	RF-40	Adicionar candidato com informação detalhada	M	✓
	RF-41	Pesquisar e aplicar filtros ao quadro	S	✓
Customizar quadro de recrutamento	RF-42	Adicionar coluna	M	✓
	RF-43	Editar coluna	M	✓
	RF-44	Editar coluna diretamente no quadro	S	✓
	RF-45	Remover coluna	M	✓
Lista de candidatos	RF-46	Editar candidato pela tabela	S	✗
	RF-47	Pré-visualizar informação do candidato	S	✗
	RF-48	Ver candidatos em lista	S	✓
	RF-49	Arquivar candidato	M	✓
	RF-50	Recuperar candidato arquivado	M	✓
	RF-51	Pesquisar e aplicar filtros à lista	S	✓
Analisar candidato	RF-52	Ver detalhes gerais	M	✓
	RF-53	Adicionar interações	M	✓
	RF-54	Editar detalhes via detalhes completos	M	✓
	RF-55	Editar interações	M	✓

7.4 Principais funcionalidades

Nesta secção, serão apresentados exemplos de páginas referentes às principais funcionalidades de acordo com os respetivos módulos. A fim de dar uma visão geral, serão mostrados apenas alguns exemplos, enquanto que os restantes podem ser consultados no Anexo B.

É importante mencionar que muitas das funcionalidades apresentadas nesta secção são do ponto de vista de um administrador, pois são os que têm acesso a mais informação e funcionalidades. A principal diferença no *layout* da plataforma é a barra lateral que, para colaboradores e gestores, não apresenta a separação da área de administração. De notar que, em algumas páginas os administradores terão acesso a botões que os restantes utilizadores estão restritos. No caso de um gestor, este terá acesso a um botão (*toggle*) para alternar entre a vista de gestor - onde vê informações relacionadas ao(s) seu(s) departamento(s) -, e a vista de colaborador - para ver apenas a sua informação.

Módulo de Autenticação

Início e Primeiro Início de Sessão

Um utilizador convidado por um administrador para a plataforma irá receber um *email* customizado enviado através do *Cognito* com uma palavra-passe temporária gerada automaticamente (Figura 7.4). Para efetuar o início de sessão o utilizador deve inserir o seu *email* e a palavra-passe recebida (Figura 7.2) e, de seguida, será redirecionado para um ecrã onde deverá definir a sua palavra-passe definitiva (Figura 7.3).

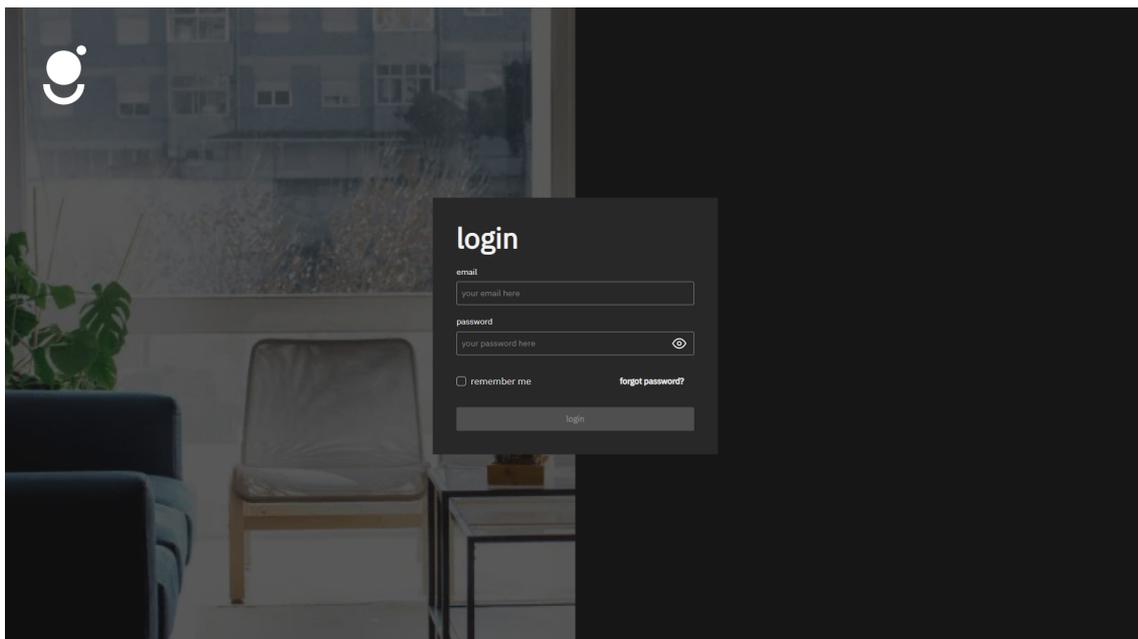


Figura 7.2: Página de início de sessão.

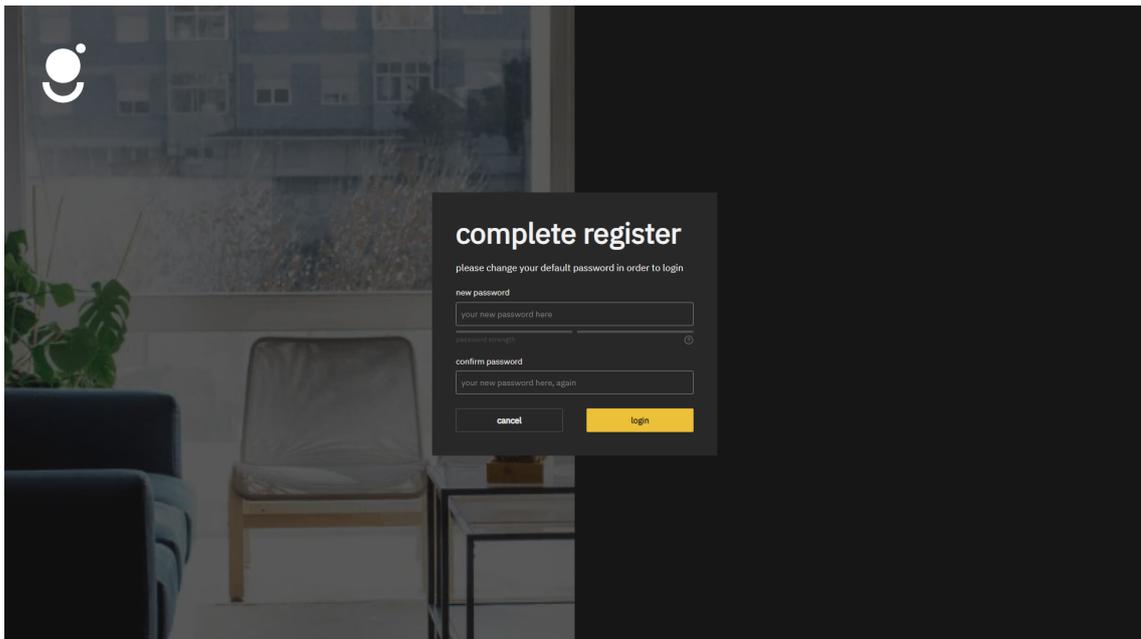
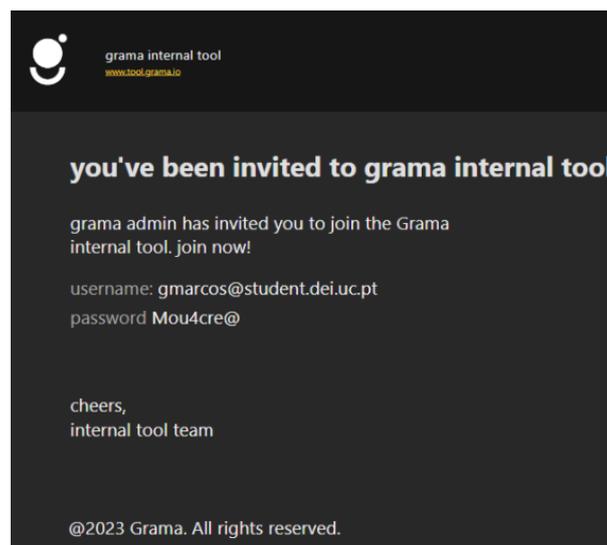


Figura 7.3: Página de primeiro início de sessão.

Figura 7.4: Exemplo de um *email* de convite para a plataforma.

Módulo de Ausências

Visualização em calendário - mês

Os utilizadores podem visualizar, em formato de calendário, todos os colaboradores ausentes ao longo de um mês. É apresentado o nome do colaborador ausente nos respetivos dias, juntamente com a cor associada ao departamento ao qual este pertence. Além disso, é exibido um ícone a indicar se o pedido está pendente, e as expressões "am" e "pm" são utilizadas para identificar se a ausência ocorre de manhã ou à tarde (respetivamente).

Capítulo 7

Na Figura 7.5, é apresentada uma vista de administrador, logo, estes conseguem ver no calendário todos os pedidos ainda não aprovados de todos os colaboradores sem exceção. À direita são apresentados os pedidos de ausência de todos os colaboradores da empresa para aprovação ou cancelamento.

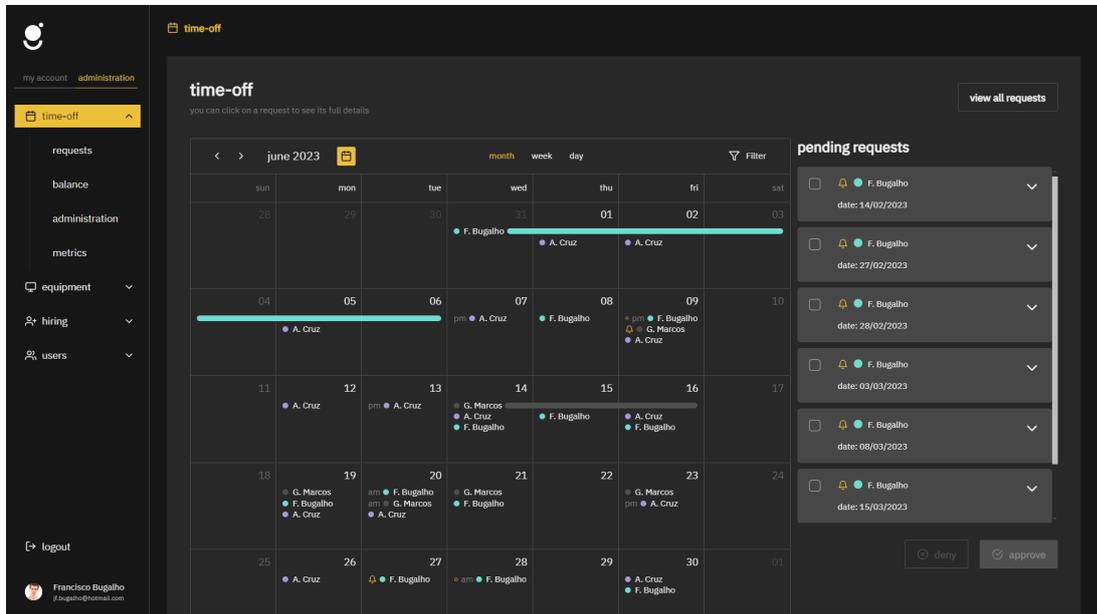


Figura 7.5: Página de vista de um mês em calendário (para administradores).

Os gestores visualizam no calendário todos os colaboradores cujas ausências se encontram aprovadas, para além dos pedidos de ausência dos colaboradores pertencentes aos seus departamentos. Esta restrição também se aplica à lista dos "pending requests", onde estes podem aprovar ou cancelar apenas os pedidos de ausência dos colaboradores dos departamentos que gerem (Figura 7.6).

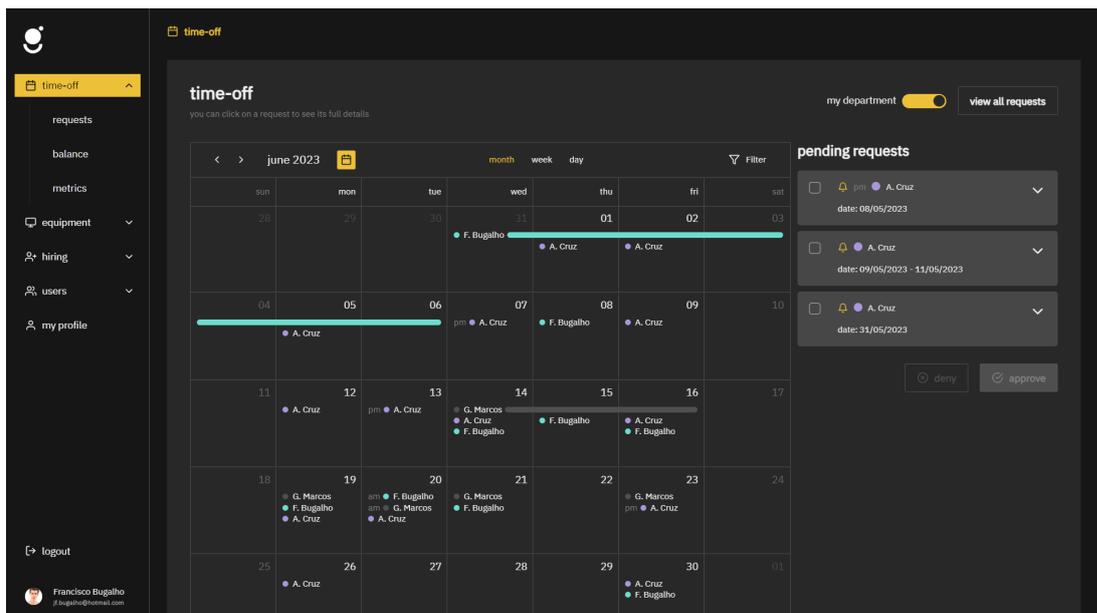


Figura 7.6: Página de vista de um mês em calendário (para gestores).

Já no calendário dos colaboradores são representadas todas as ausências aprovadas para qualquer colaborador e apenas os seus pedidos pendentes. Na lista apenas são apresentados os seus pedidos de ausência pendentes sem qualquer ação possível à exceção. Tal como para os cargos anteriores, é possível expandir o pedido e analisar alguns detalhes adicionais (Figura 7.7).

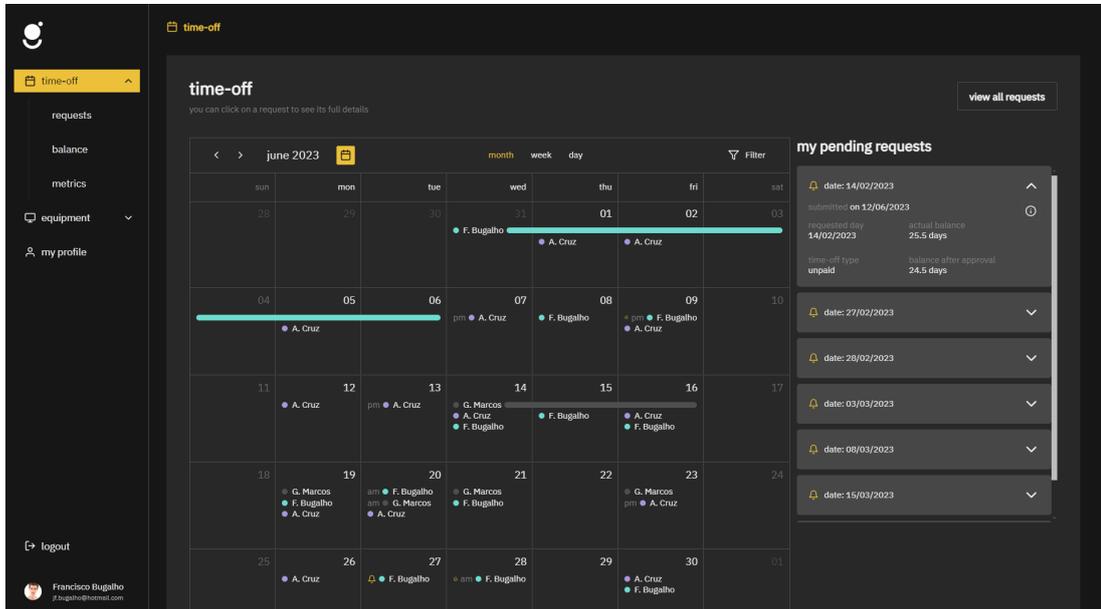


Figura 7.7: Página de vista de um mês em calendário (para colaboradores).

Requisição de ausência

Para um utilizador requisitar ausências, deverá indicar o tipo do pedido (dia completo, parcial ou múltiplos dias), definir a data de início e, por fim, indicar qual o tipo da ausência (por exemplo, férias), Figura 7.8. Caso o tipo de pedido seja múltiplos dias o utilizador deve inserir a data de término da ausência. Posto isto é enviado um email aos administradores e gestores a informar do pedido.

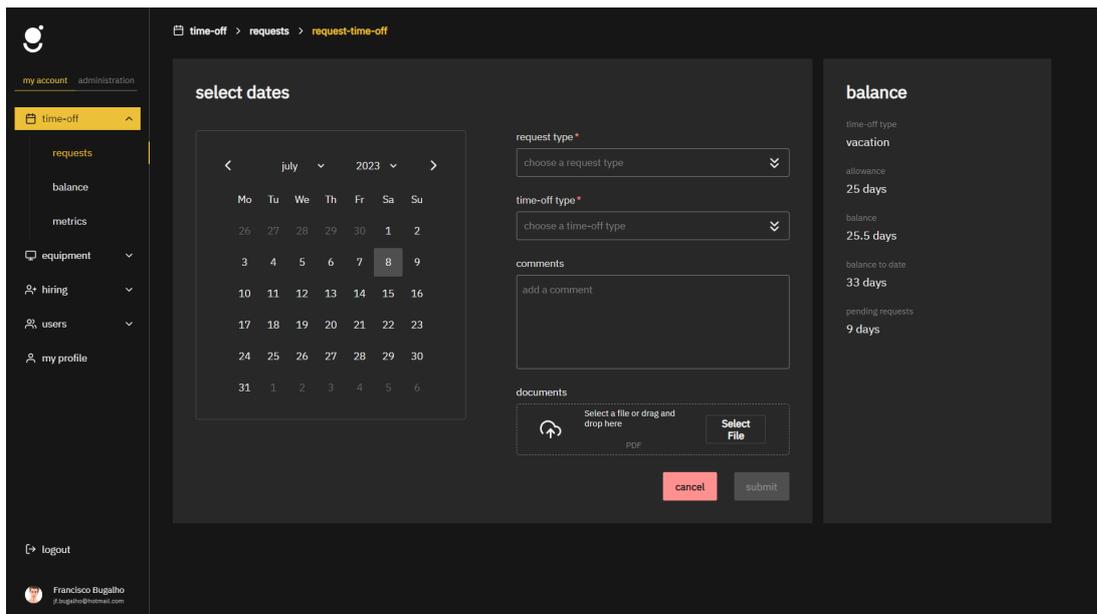


Figura 7.8: Página de requisição de uma ausência.

No caso de ser um administrador, este pode requisitar ausências em nome de outros colaboradores (Figura 7.9). Para tal deve preencher o campo extra "request on behalf of". Ao contrário da funcionalidade anterior, este pedido é automaticamente aprovado notificando não só o colaborador, mas também os administradores e gestores desse colaborador.

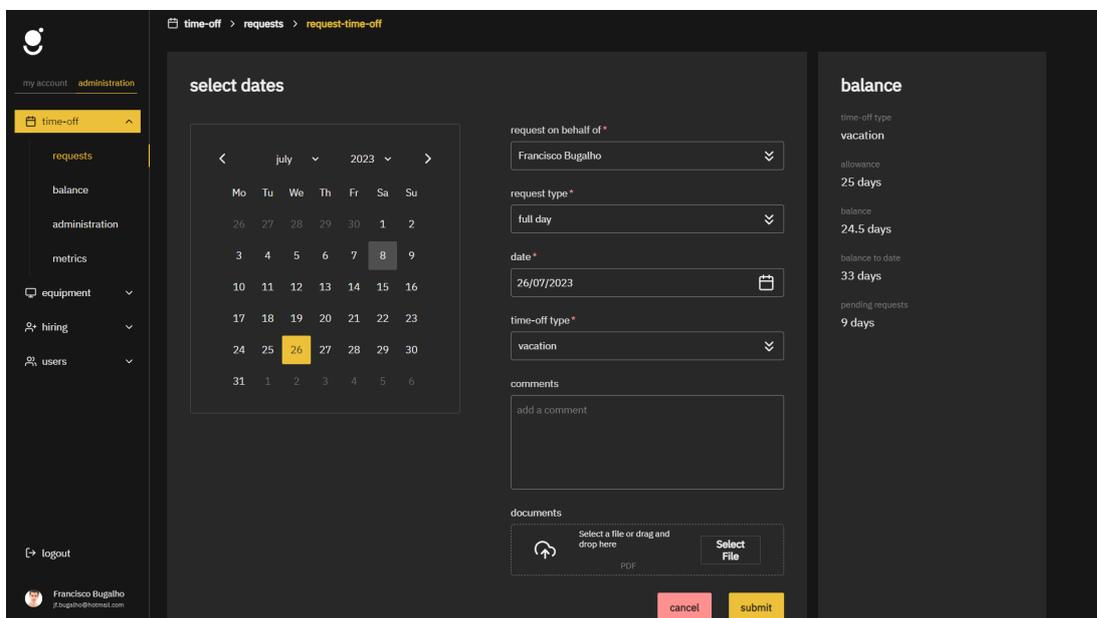


Figura 7.9: Página de requisição de uma ausência (para administradores).

Email de pedido de ausência

Após um utilizador efetuar um pedido de ausência é enviado um *email* ao(s) seu(s) gestor(es) e aos administradores da plataforma com as informações desse pedido (Figura 7.10). Este é apenas um dos exemplos de *emails* enviados pois tendo em conta a ação serão enviados *emails* distintos. Por exemplo, após um cancelamento ou uma edição de um pedido os gestores e administradores, bem como do colaborador a quem se refere o pedido são notificados via *email*.

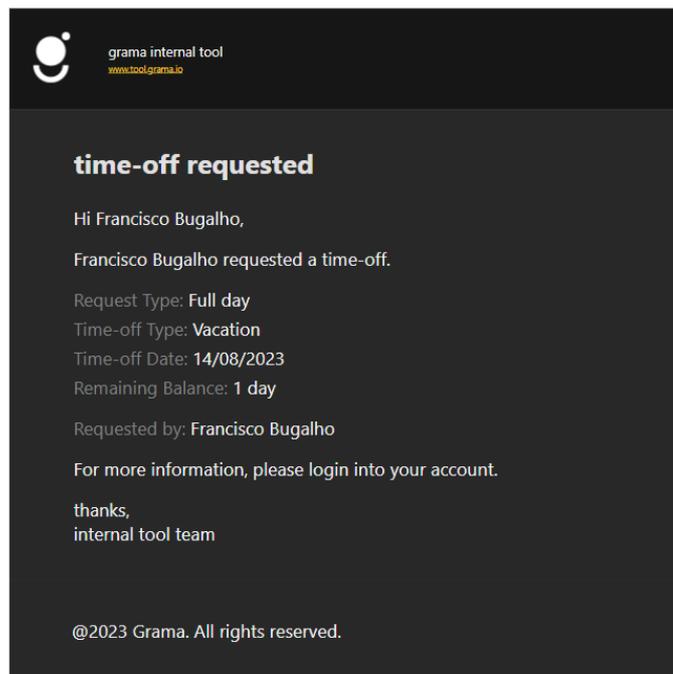


Figura 7.10: Exemplo de um *email* de um pedido de ausência.

Visualizar métricas

Os utilizadores têm um painel de métricas com várias informações referentes às ausências, por exemplo, no caso de um administrador: métricas das ausências por departamento ou número de colaboradores ausentes nos próximos 7 dias (Figura 7.11).

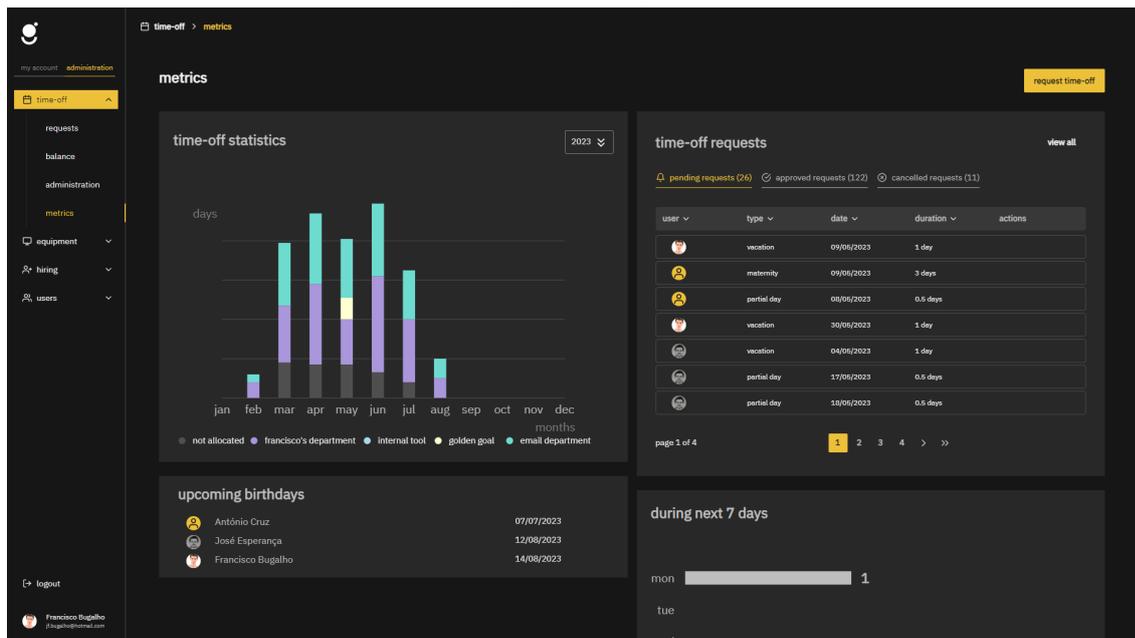


Figura 7.11: Página de métricas sobre as ausências (para administradores).

Módulo de Recrutamento

Pipeline de Recrutamento

No módulo de recrutamento os administradores podem customizar a *pipeline* de recrutamento de acordo com os processos de recrutamento da empresa. Podem não só adicionar novas colunas como removê-las, adaptando a *pipeline* ao atual processo de recrutamento da empresa. Com esse processo bem definido na *pipeline* de recrutamento é possível distribuir os vários candidatos pelas várias fases do processo de recrutamento. Após um candidato avançar ou recuar no processo este pode ser movido para a respetiva coluna através de *drag and drop* (Figura 7.12).

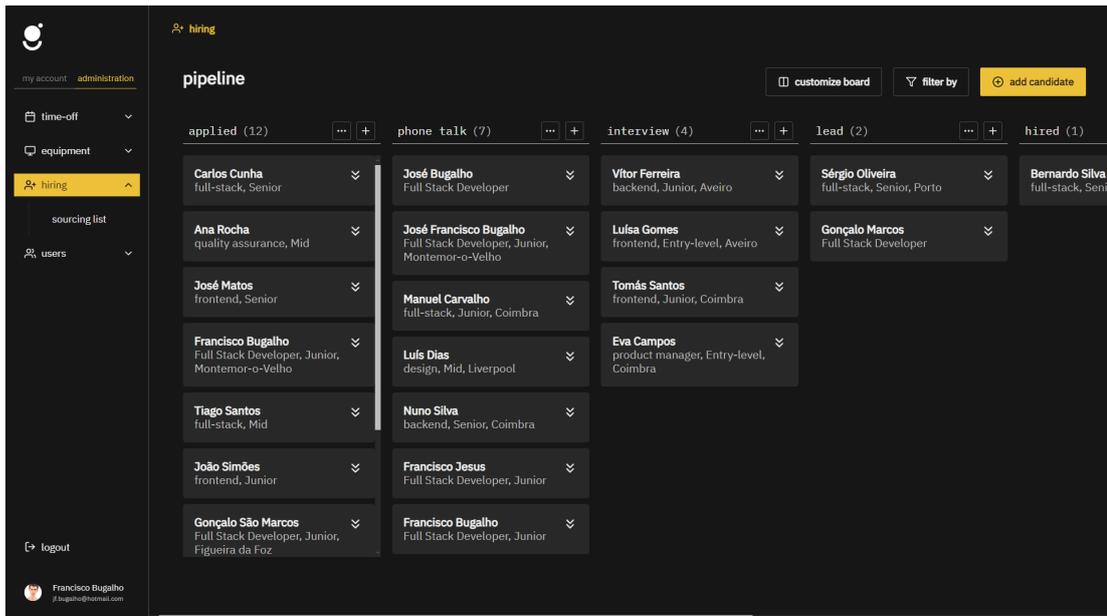


Figura 7.12: Página da *pipeline* de recrutamento.

Adicionar um candidato a uma coluna

Para colocar os candidatos numa coluna da *pipeline* de recrutamento estes devem ser adicionados à plataforma com o estado "ativo". A título de exemplo, é apresentado, na Figura 7.13, a modal para a adição de um candidato diretamente a uma coluna da *pipeline*. Ao clicar no botão "add more details" é possível adicionar outro tipo de detalhes pertinentes sobre cada candidato.

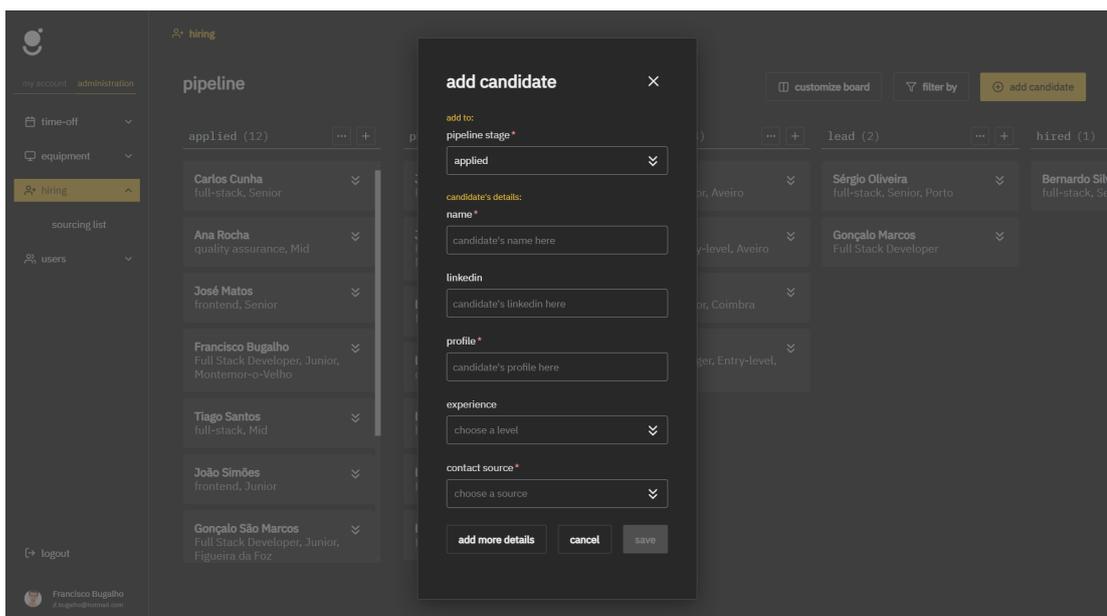


Figura 7.13: Modal de adição de um candidato a uma coluna da *pipeline*.

Lista de candidatos

Caso um candidato seja adicionado à plataforma mas os administradores o definam com um estado diferente de "ativo" este não aparecerá na *pipeline* de recrutamento. Deste modo, serão apresentados numa lista todos os candidatos independentemente do seu estado (Figura 7.14). Para além disso é possível ver duas abas: "sourcing list" onde são apresentados os candidatos em que a empresa tem interesse, e a aba "archived candidates" que representa os candidatos descartados pela empresa.

<input type="checkbox"/>	name	status	profile	experience	contact source	current company	last interaction
<input type="checkbox"/>	Andrela Marques	On-hold	Full Stack Developer	n/a	Referral	n/a	never
<input type="checkbox"/>	Tomás Santos	Active	frontend	Junior	Referral	n/a	never
<input type="checkbox"/>	Inês Rodrigues	Active	designer	Junior	LinkedIn	n/a	never
<input type="checkbox"/>	Bernardo Silva	Active	full-stack	Senior	Events	n/a	never
<input type="checkbox"/>	Maria Silva	Active	product manager	Senior	Referral	n/a	never
<input type="checkbox"/>	Isabel Fernandes	Active	backend	Entry-level	Referral	n/a	24/05/2023
<input type="checkbox"/>	Francisco Bugalho	Active	Full Stack Developer	Junior	Referral	n/a	never
<input type="checkbox"/>	Eva Campos	Active	product manager	Entry-level	Events	n/a	never
<input type="checkbox"/>	Francisco Bugalho	Lead	Full Stack Developer	n/a	LinkedIn	n/a	19/04/2023
<input type="checkbox"/>	João Pedro	Active	None	n/a	Events	n/a	never

Figura 7.14: Lista com todos os candidatos não arquivados.

Detalhes dos candidatos

De forma aos gestores e administradores controlarem as suas interações (entrevistas) com os candidatos, quando se visualiza os detalhes de um candidato é possível não só consultar (ou editar) as informações pessoais (Figura 7.15), mas também consultar (ou adicionar ou editar) as várias interações com cada candidato. Deste modo, os gestores e administradores, após cada entrevista, conseguem manter um registo associado a cada candidato possível de consultar sempre que necessário com informações relevantes sobre a mesma, bem como um resultado obtido (Figura 7.16).

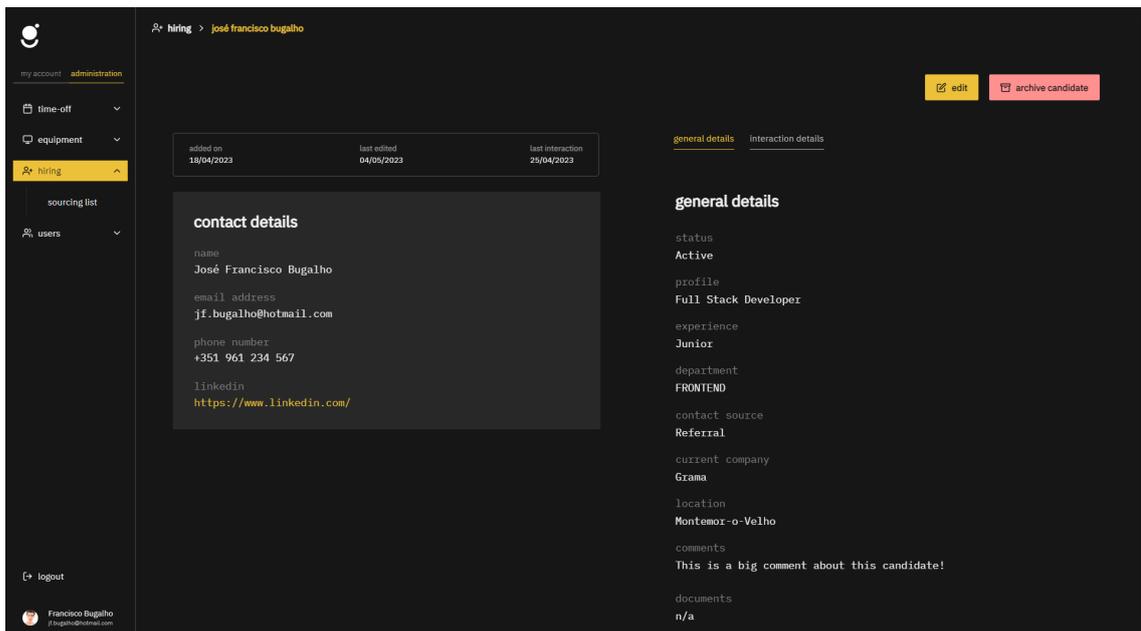


Figura 7.15: Detalhes gerais de um candidato.

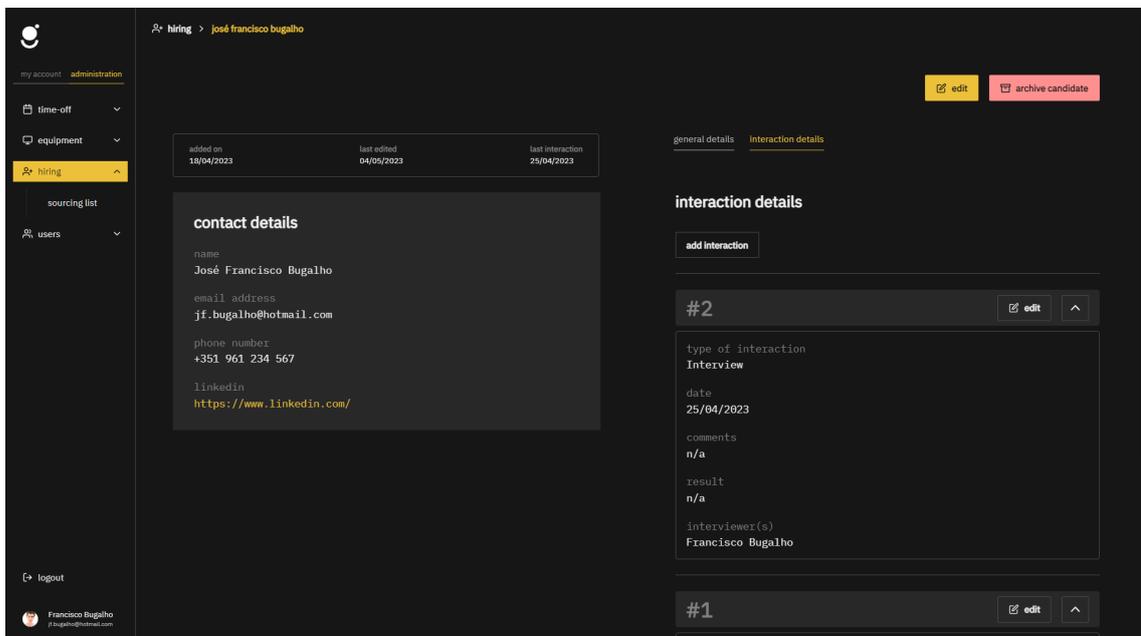


Figura 7.16: Interações de um candidato.

7.5 Análises dos riscos levantados

Ao longo da secção 6.3, constam os possíveis riscos associados ao desenvolvimento do projeto. Com o final do semestre, foi possível realizar uma análise aos riscos levantados. De seguida, serão descritos os riscos ocorridos durante o desenvolvimento e o seu plano de mitigação:

- **R-3: Scope muito ambicioso** - Inicialmente, o aluno definiu e priorizou um conjunto de requisitos que, posteriormente, foram validados pelo cliente. Contudo, com o aproximar de meio do semestre, o aluno em conjunto com o orientador da Grama efetuaram um balanço do progresso até à data e do trabalho a realizar até ao final da fase de desenvolvimento. Em conjunto decidiram que o objetivo principal seria cumprir todas as tarefas priorizadas como *Must Have* à exceção da área de "Gerir tipos de ausências", visto que esta implicaria muito esforço extra que ultrapassava consideravelmente o período temporal do estágio. Deste modo, o aluno conseguiria obter um produto consistente mesmo com os requisitos em falta;
- **R-4: Adaptação às tecnologias** - Embora o aluno já tenha tido experiência com algumas das tecnologias a utilizar, outras eram completamente desconhecidas, o que implicou muita pesquisa causando algum atraso ao desenvolvimento numa fase inicial. É importante realçar toda a disponibilidade, quer do orientador da empresa, quer de outros *developers* que forneceram apoio sempre que necessário;
- **R-5: Atraso no módulo de utilizadores** - Visto que para o desenvolvimento de algumas funcionalidades o aluno necessitava de funcionalidades do módulo de utilizadores, módulo esse a ser desenvolvido em paralelo pelo outro estagiário. O plano passou pela realização do módulo de recrutamento, pois tinha menos dependência do módulo de utilizadores. Deste modo, aquando da necessidade de funcionalidades do módulo de utilizadores estas já se encontravam concluídas.

7.6 Trabalho Futuro

A título de trabalho futuro, inicialmente, concluir os requisitos correspondentes à área "Gerir tipos de ausências" que trará outro tipo de funcionalidades ao método de pedir uma ausência. Através dessas funcionalidades a empresa poderá personalizar os tipos de ausências a serem pedidos, tornando a sua política de ausências mais flexível. Ainda em relação ao módulo de ausências, e analisando os requisitos levantados, adicionar eventos especiais (como, feriados e aniversários) também adicionaria algum valor ao produto, tornando-o ainda mais customizável.

Para além dos restantes requisitos levantados e não concluídos, a plataforma está preparada para qualquer atualização/incremento aos módulos já existentes, bem como incluir outros módulos. Por exemplo, um módulo de partilha de conheci-

mentos onde, os colaboradores poderão requisitar formações e também partilhar informação relevante sobre as mesmas.

Por fim, existem alguns melhoramentos a realizar ao nível de responsividade da plataforma, visto que não era um dos pontos principais do estágio, sendo em algumas tarefas omitida devido ao tempo para o desenvolvimento.

Capítulo 8

Testagem

Neste capítulo serão apresentados e descritos os testes efetuados na Plataforma Interna da Grama. Inicialmente, serão apresentados os testes às funcionalidades da plataforma e, de seguida, os testes aos requisitos não funcionais (secção 3.5).

8.1 Testes funcionais

No decorrer do desenvolvimento, o aluno, antes de terminar uma tarefa, efetuou um conjunto de testes unitários, cujo objetivo era cobrir alguns dos casos possíveis e garantir que cada *endpoint* tinha o funcionamento correto. A ferramenta utilizada para a realização destes testes foi JUnit [100].

De seguida, é apresentado um exemplo de um teste unitário para o *endpoint* dos candidatos. Ao realizar um pedido *GET* com o identificador do candidato, a resposta a esse pedido será um objeto JSON com a informação do respetivo candidato. Por outro lado, caso não exista candidato para o identificador enviado será retornado um objeto com um código de erro e respetiva mensagem.

Teste do *endpoint* *get* candidato por identificador

```
@Test
@DisplayName("Get Candidate By Id")
public void testGetCandidateById(@Autowired
    WebClient webTestClient) {
    Candidate candidate = // Creating a candidate

    CandidateResponse response = webTestClient.get()
        .uri("/candidates/{id}", candidate.getId())
        .header("Authorization", getToken())
        .exchange()
        .expectStatus().isOk()
        .expectBody(/* Response Type */)
        .returnResult()
        .getResponseBody();
}
```

```
assertNotNull(response);
assertEquals(response.getId(), candidate.getId());
// Other asserts...
}
```

Foi ainda desenvolvido, para cada um dos microserviços de ausências e recrutamento, um *endpoint* no *backend* para a realização de testes manuais utilizando a plataforma *Swagger* [101]. Este, para além do objetivo de realizar testes, permite documentar os *endpoints* e respetivos parâmetros.

Na Figura 8.1, é apresentada uma visão geral dos controladores disponíveis e respetivos *endpoints*, para o microserviço de ausências. Adicionalmente, ao expandir cada método (*GET*, por exemplo) são descritos os parâmetros e apresentados exemplos dos mesmos.

Internal Tool Time-Off API 0.0.1-SNAPSHOT OAS3
/api/v1/internal-tool-time-off/v3/api-docs
The Time Off service is responsible for managing all time offs in Grama Internal Tool.
Contact Francisco Bugalho

Servers
/api/v1/internal-tool-time-off

Time Off Controller Endpoints to CRUD time offs.
GET /time-offs List time offs
POST /time-offs Create time off
GET /time-offs/{id} Get a time off
DELETE /time-offs/{id} Delete a time off
PATCH /time-offs/{id} Update a time off
GET /time-offs/summary Get the time offs summary

Time Off Metrics Controller Endpoints get metrics time-off related.

Time Off Type Controller Endpoints to CRUD time off types.

Time-Off File Controller Endpoints to CRUD time-off files.

Time-off User Balances Controller Endpoints to CRUD time-off users balances.

Figura 8.1: Exemplo do *endpoint* do *Swagger* para o microserviço de ausências.

Após o término do desenvolvimento, foram realizados testes de integração entre o *frontend* e o *backend* da plataforma. Deste modo foi possível testar a plataforma e garantir que todos os requisitos desenvolvidos tinham o comportamento desejado.

A Tabela 8.1, é um exemplo de uma das tabelas de testes que constam do Anexo C. Esta representa o plano de testes para a área "Gerir Quadro de Recrutamento" do módulo de recrutamento. É composta pelas seguintes colunas: US referente ao teste a realizar, ID do teste, cenário, passos executados, resultado esperado, resultado obtido e, por fim, uma conclusão que indica se o teste passou ou falhou.

Note-se que as US foram agrupadas nos casos em que as funcionalidades eram semelhantes. A diferença mais significativa reside nas funcionalidades de colaborador, visto que estes são os atores com maiores restrições à informação que podem aceder. Logo, sempre que necessário foram realizados testes adicionais tendo em conta essas funcionalidades.

Tabela 8.1: Plano de testes para a área Gerir Quadro de Recrutamento.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
62 e 81	39	Ver os candidatos ativos	1. Aceder à página "hiring"	Quadro com os candidatos ativos distribuídos pela respetiva fase do processo de recrutamento	Quadro com os candidatos ativos distribuídos pela respetiva fase do processo de recrutamento	Passou
63 e 82	40	Pesquisar e aplicar filtros ao quadro de recrutamento	1. Aceder à página "hiring" 2. Clicar no botão "filter by" 3. Escolher os filtros na modal 4. Confirmar a ação	Quadro com os candidatos filtrados	Quadro com os candidatos filtrados	Passou
64	41	Adicionar um candidato	1. Aceder à página "hiring" 2. Clicar no botão "add candidate" 3. Na página de adicionar preencher os campos obrigatórios: - Nome - Estado - Perfil - Fonte de contacto	Candidato adicionado com sucesso	Candidato adicionado com sucesso	Passou
65	42	Adicionar um candidato diretamente a uma coluna	1. Aceder à página "hiring" 2. Clicar no símbolo '+' numa coluna 3. Preencher os campos obrigatórios: - Coluna - Nome - Perfil - Fonte de contacto	Candidato associado à coluna escolhida	Candidato associado à coluna escolhida	Passou
66	42	Adicionar um candidato com informação detalhada	1. Aceder à página "hiring" 2. Clicar no símbolo '+' numa coluna 3. Na modal clicar no botão "add more details" 4. Na página de adicionar preencher os campos obrigatórios: - Nome - Estado - Perfil - Fonte de contacto	Candidato associado à coluna escolhida	Candidato associado à coluna escolhida	Passou

8.2 Testes não funcionais

Nesta secção, são apresentados os testes realizados para cada requisito não funcional. Deste modo é possível garantir que os requisitos não funcionais definidos anteriormente na secção 3.5 foram devidamente cumpridos.

Robustez

Tal como referido anteriormente (secção 3.5.1), a robustez de um sistema representa a capacidade deste manter o correto funcionamento na presença de falhas. Os testes de robustez foram acompanhando o desenvolvimento, isto é, aquando da realização de *endpoints* quer no microserviço de ausências, quer no de recrutamento, foram executados alguns testes unitários cujos cenários correspondiam à inserção de valores inválidos para avaliar o comportamento do sistema. Por exemplo, durante a criação de um candidato, o utilizador ao inserir um número de telefone inválido espera-se que o sistema responda com uma mensagem de erro.

Posto isto, aquando a integração do *frontend* com o *backend*, foram testadas algumas possibilidades, por exemplo, inserir um número de caracteres superior ao suportado pela base de dados. Neste caso, existiu alguma limitação na quantidade de testes realizados devido à validação de valores realizada no *frontend* antes da submissão de um formulário. Por fim, no final do semestre ambos os testes foram novamente executados para garantir a validade dos resultados anteriormente obtidos.

De seguida, é apresentado um exemplo de uma tabela (Tabela 8.2) com alguns testes de aceitação realizados. Nesta estão descritos os dados de entrada e é comparado o resultado esperado com o resultado obtido, concluindo se o teste passou ou não com sucesso.

Note-se que, para simplificar a análise e sintetizar a informação, os testes estão agrupados por US. Os restantes resultados poderão ser consultados no Anexo D.

Tabela 8.2: Testes de aceitação para a adição e edição de colunas do quadro de recrutamento.

US	ID	Cenário	Dados de entrada	Resultado Esperado	Resultado Obtido	Conclusões
67, 68 e 69	43	Nome superior a 128 caracteres	Qualquer <i>string</i> superior a 128 caracteres	Campo 'column name' transita para a cor vermelha com a mensagem: "must be at most 128 characters"	Campo 'column name' transita para a cor vermelha com a mensagem: "must be at most 128 characters"	Passou
	44	Nome igual a 128 caracteres	Qualquer <i>string</i> igual a 128 caracteres	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Passou
	45	Nome inferior a 128 caracteres	Qualquer <i>string</i> inferior a 128 caracteres	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Passou
	46	Nome inferior a 128 caracteres e valores numéricos	Coluna 2	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Passou
	47	Nome inferior a 128 caracteres e caracteres especiais	#Primeira Coluna	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Passou

Com base na análise das tabelas, é possível concluir que a plataforma demonstrou ser robusta, mantendo sempre o comportamento adequado para os diferentes tipos de dados de entrada testados. Este comportamento estende-se aos componentes que foram projetados para aceitar apenas alguns tipos de dados, reduzindo possíveis erros numa fase inicial.

Performance

Para a realização dos testes de *performance* foi necessário efetuar o *deploy* da plataforma na AWS. Esta fornece recursos dedicados que permitem obter conclusões mais precisas em relação ao ambiente real, em contraste com a realização dos testes na máquina onde o desenvolvimento foi feito, uma vez que essa abordagem pode ser afetada pela execução de outros serviços, comprometendo a confiabilidade dos resultados. No entanto, o *deploy* foi um processo que demorou alguns dias, atrasando a realização destes testes.

Este requisito foi priorizado como baixo pois, numa fase inicial, a plataforma não tem uma utilização recorrente. De qualquer modo, deverá garantir que os tempos de carregamento de informação não são morosos e não prejudicam a experiência do utilizador. Daí, os testes realizados procuraram garantir que a métrica definida na secção 3.5.2 é cumprida, ou seja, se as principais páginas da plataforma são apresentadas ao utilizador em menos de **2,5 segundos**.

Para a realização dos testes, utilizou-se a ferramenta de desenvolvimento do navegador Google Chrome. Anotou-se 20 vezes o tempo apresentado na coluna "finish" para cada uma das páginas indicadas na Tabela 8.3, pois este representa o carregamento total da página com todos os pedidos realizados e a receção da respetiva resposta. De seguida, foi calculada a média dos tempos de carregamento na várias páginas dos dois módulos e, pode-se concluir que apenas em **dois** casos o tempo de carregamento foi inferior a 2,5 segundos.

Considerando o referido na secção 3.5.2 e mesmo não existindo páginas com tempos de carregamento superiores a 4 segundos, é importante rever as páginas para analisar se é possível melhorar o tempo de carregamento. Embora este não seja um requisito de alta prioridade para o estágio, será algo a abordar no futuro para garantir aos utilizadores a uma boa experiência na utilização da plataforma.

Tabela 8.3: Tempos médios de carregamento das principais páginas nos módulos de ausências e recrutamento.

Módulos	Página	Média (em segundos)	Conclusões
Ausências	Ausências em calendário - mês	2.832s	Requer Melhoramento
	Ausências em calendário - semana	1.295s	Passou
	Ausências em calendário - dia	1.013s	Passou
	Pedidos de ausência	2.693	Requer Melhoramento
	Detalhes de uma ausência	2.633s	Requer Melhoramento
	Métricas	3.16s	Requer Melhoramento
Recrutamento	Pipeline de recrutamento	3.594s	Requer Melhoramento
	Candidatos em lista	2.726s	Requer Melhoramento
	Detalhes gerais de um candidato	3.356s	Requer Melhoramento

Segurança

O requisito não funcional de segurança foi descrito em dois cenários distintos: não autenticação e não autorização. Tal como para a robustez, alguns dos testes de segurança foram realizados ao longo do desenvolvimento. Começando pela autenticação, os alunos focaram-se em garantir que apenas os utilizadores autenticados têm acesso aos conteúdos da plataforma através da integração do *Amazon Cognito*. Posto isto, depois da realização do primeiro *endpoint*, a tarefa seguinte consistia na adição de segurança ao respetivo microsserviço. Para tal, era importante fazer a validação dos *tokens* do *Cognito*, bem como das permissões dos utilizadores aquando cada pedido aos vários *endpoints*. Por exemplo, no módulo de recrutamento, apenas os administradores podem criar candidatos, o que significa que os gestores e colaboradores não devem ter acesso a essa funcionalidade.

Começando pela **não autenticação**, foram efetuados testes de forma a garantir que um utilizador não autenticado não tem permissões para aceder a qualquer página da plataforma sejam elas páginas destinadas apenas a utilizadores ou páginas de administração. Na Tabela 8.4 encontram-se os testes realizados para o cenário de não autenticação.

Tabela 8.4: Testes ao requisito não funcional de segurança, não autenticação.

ID	Cenário	Resultado Esperado	Resultado Obtido	Conclusões
1	Qualquer utilizador sem registo tenta aceder a uma página da plataforma	Utilizador é redirecionado para a página de início de sessão	Utilizador é redirecionado para a página de início de sessão	Passou
2	Qualquer utilizador sem registo tenta aceder a uma página de administração da plataforma	Utilizador é redirecionado para a página de início de sessão	Utilizador é redirecionado para a página de início de sessão	Passou
3	Pedido a um <i>endpoint</i> sem <i>token JWT</i>	Pedido rejeitado e resposta com código de erro 401	Pedido rejeitado e resposta com código de erro 401	Passou
4	Pedido a um <i>endpoint</i> com um <i>token JWT</i> inválido	Pedido rejeitado e resposta com código de erro 401	Pedido rejeitado e resposta com código de erro 401	Passou
5	Pedido a um <i>endpoint</i> com um <i>token JWT</i> válido expirado	Pedido rejeitado e resposta com código de erro 401	Pedido rejeitado e resposta com código de erro 401	Passou
6	Pedido a um <i>endpoint</i> com um <i>token JWT</i> válido	Pedido aceite e resposta com código 200	Pedido aceite e resposta com código 200	Passou

Para o cenário de **não autorização** os testes realizados têm como objetivo analisar se os vários cargos do utilizador estão corretamente implementados. Ou seja, se de facto os colaboradores apenas têm acesso às funcionalidades de colaborador, os gestores às funcionalidades de gestor e colaborador e os administradores às funcionalidades de todos os cargos. Na Tabela 8.5, pode-se concluir que os testes realizados foram cumpridos com sucesso e cada utilizador apenas consegue aceder às funcionalidades destinadas aos seus cargos.

Tabela 8.5: Testes ao requisito não funcional de segurança, não autorização.

ID	Cenário	Resultado Esperado	Resultado Obtido	Conclusões
7	Colaborador tenta aceder a uma página de administração	É apresentada uma página de erro	É apresentada uma página de erro	Passou
8	Colaborador tenta aceder a uma página da área pessoal	Redirecionado para a página e conteúdo apresentado	Redirecionado para a página e conteúdo apresentado	Passou
9	Gestor tenta aceder a uma página de administração	É apresentada uma página de erro	É apresentada uma página de erro	Passou
10	Gestor tenta aceder a uma página da área pessoal	Redirecionado para a página e conteúdo apresentado	Redirecionado para a página e conteúdo apresentado	Passou
11	Administrador tenta aceder à uma página de administração	Redirecionado para a página e conteúdo apresentado	Redirecionado para a página e conteúdo apresentado	Passou
12	Administrador tenta aceder à uma página da área pessoal	Redirecionado para a página e conteúdo apresentado	Redirecionado para a página e conteúdo apresentado	Passou
13	Qualquer utilizador faz um pedido de uma origem não autorizada	O pedido é bloqueado pela política de CORS na <i>API Gateway</i>	O pedido é bloqueado pela política de CORS na <i>API Gateway</i>	Passou
14	Qualquer utilizador faz um pedido direto ao microsserviço de ausências	O pedido é bloqueado pelas <i>inbound rules</i>	O pedido é bloqueado pelas <i>inbound rules</i>	Passou
15	Qualquer utilizador faz um pedido direto ao microsserviço de recrutamento	O pedido é bloqueado pelas <i>inbound rules</i>	O pedido é bloqueado pelas <i>inbound rules</i>	Passou
16	Qualquer utilizador tenta aceder diretamente à base de dados RDS	O pedido é bloqueado pelas <i>inbound rules</i>	O pedido é bloqueado pelas <i>inbound rules</i>	Passou
17	Qualquer utilizador tenta aceder diretamente ao <i>bucket S3</i>	O pedido é bloqueado pelas permissões e políticas do <i>bucket</i>	O pedido é bloqueado pelas permissões e políticas do <i>bucket</i>	Passou

Note-se que, para além de testar a não autorização aos cargos dos utilizadores, realizaram-se alguns testes aos serviços da AWS após a plataforma estar *deployed* para garantir a proteção dos serviços do tráfego externo. Para tal, na *API Gateway* foi implementada uma política de *Cross-Origin Resource Sharing* (CORS) de

forma a bloquear os pedidos efetuados de fora da plataforma. Além disso, os microsserviços de ausências e recrutamento apenas podem ser acedidos via API Gateway, deste modo todos os pedidos externos serão bloqueados. Já as bases de dados e os *buckets* S3 serão acedidos apenas através dos respetivos microsserviços, enquanto que o *Cloudfront*, relativo ao *frontend* da plataforma, tem acesso exclusivo ao seu *bucket*.

Após a realização dos testes acima, foi utilizada a ferramenta OWASP Zed Attack Proxy (ZAP) [102] para identificar algumas vulnerabilidades e/ou problemas de segurança relacionados à plataforma. As vulnerabilidades encontradas são representadas na ferramenta com uma bandeira colorida, Figura 8.2.



Figura 8.2: Exemplo dos alertas no OWASP ZAP [103].

Deste modo, foi efetuado um *scan* automático sobre o endereço da plataforma no qual foram identificados **seis** vulnerabilidades (Figura 8.3). Não foram encontrados riscos altos, apenas dois de risco médio, dois de risco baixo e dois informativos.

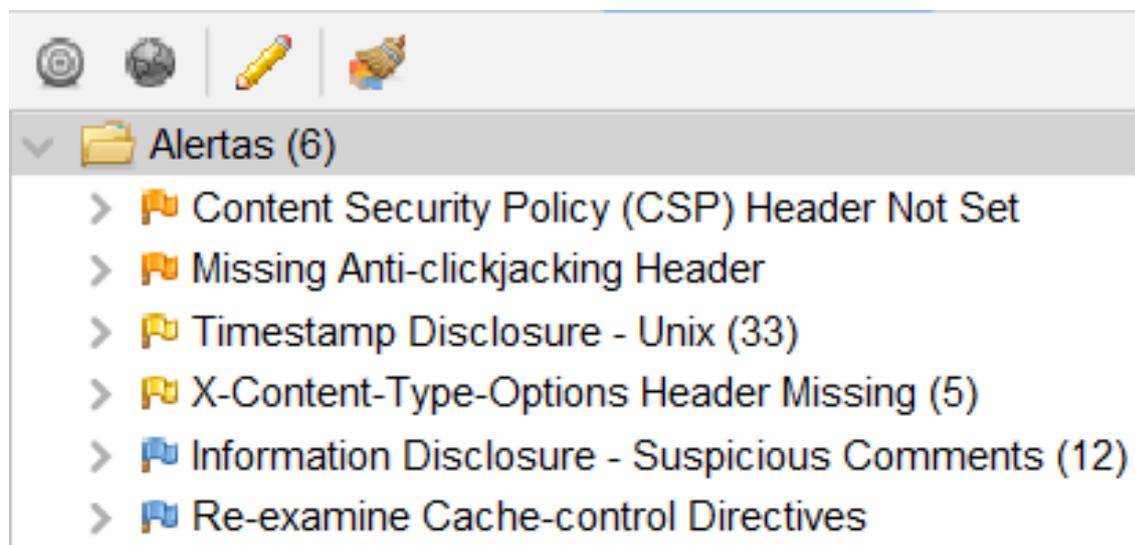


Figura 8.3: Vulnerabilidades encontradas após o teste com a ferramenta OWASP ZAP.

Devido ao curto prazo dedicado aos testes durante o último *sprint*, não houve tempo suficiente para corrigir os alertas identificados. Contudo, foi efetuada uma pequena pesquisa sobre cada vulnerabilidade e definida uma possível solução para cada uma. É importante referir que corrigir os alertas encontrados é uma das prioridades futuras, de forma a tornar a plataforma mais segura e menos suscetível a ataques.

Content Security Policy (CSP)

Content Security Policy é uma medida de segurança utilizada para proteger as aplicações *web* contra ataques como *cross-site scripting* (XSS) [104]. Este tipo de ataques é usado não só com o objetivo de roubar de dados sensíveis, mas também para realizar injeções de *malware*.

Ao definir corretamente o *header Content-Security-Policy*, é possível restringir a execução de *scripts* provenientes de fontes não confiáveis e prevenir a execução de código malicioso. Deste modo, uma possível solução seria garantir que os vários servidores *web* retornam o *header Content-Security-Policy* devidamente configurado, pois este permite à aplicação *web* saber quais os recursos permitidos carregar [105].

Missing Anti-clickjacking Header

Os ataques de *clickjacking* estão relacionados à alteração de objetos na página *web* por parte do atacante levando o utilizador a efetuar uma ação sem o seu conhecimento ou consentimento. O principal objetivo destes ataques é direcionar os cliques do utilizador na página para a realização de ações maliciosas, por exemplo, a instalação de um *malware* [106].

Para solucionar este problema deve-se garantir que os servidores *web* retornem ou o *header Content-Security-Policy* com as diretivas *frame-ancestors* ou *header X-Frame-Options* [107].

Timestamp Disclosure Unix

Este alerta refere-se à possibilidade de exposição da hora do servidor. Com esta informação um atacante pode obter informação sensível, por exemplo, um *token* de autenticação [108].

Neste caso, após analisar os *timestamps* com algum cuidado, estes alertas aparentam ser falsos positivos pois não passam de números *nothing up my sleeve* cujo propósito é inicializar o algoritmo de encriptação SHA-256. Estes números são cuidadosamente escolhidos para demonstrar transparência e evitar qualquer suspeita de padrões ocultos que possam ser explorados por atacantes, garantindo a segurança do SHA-256. Visto que estes são representados no formato "unsigned int 32", o *scan* considerou-os como potenciais *timestamps* que divulgavam informação sensível do servidor [109].

X-Content-Type-Options Header Missing

O alerta indica que o *header X-Content-Type-Options* está em falta, este tem como objetivo indicar ao *browser* como este deve tratar o tipo do conteúdo recebido. Ao não definir este *header* os atacantes podem explorar essa debilidade forçando o *browser* a interpretar o conteúdo de uma maneira não intencional através de ataques *MIME type sniffing* [110].

Para mitigar este problema as respostas devem incluir o *header X-Content-Type-Options* definido para "nosniff". Assim, com este *header* devidamente definido o *browser* interpretará o tipo do conteúdo como o definido na resposta [110].

Information Disclosure - Suspicious Comments

Este alerta informa que algumas respostas contêm comentários suspeitos, isto é, comentários que podem, de alguma forma, auxiliar o atacante, pois podem revelar informações sensíveis ou até detalhes sobre o código [111].

Após uma pequena análise aos comentários indicados pela ferramenta, pode-se concluir que nenhum destes apresenta informação sensível. Estes ocorrem em ficheiros específicos de licenças do React, React-Router, e AWS Amplify, logo, não representam um risco de segurança. Contudo, caso existisse esse risco dever-se-ia rever o código e remover os respetivos comentários.

Re-examine Cache - control Directives

As diretivas de controlo de *cache* indicam ao *browser* como este deve lidar com o armazenamento dos recursos em *cache*. Deste modo, esta vulnerabilidade indica que as diretivas de controlo de *cache* não estão definidas, e existe o risco de alguma informação sensível ser guardada em *cache*. Embora este comportamento possa ser desejado para conteúdos estáticos como ficheiros JavaScript ou CSS é importante garantir que nenhuma informação sensível fica guardada [112].

Por isso, para corrigir esta vulnerabilidade deve-se definir corretamente o *header Cache-Control* com os valores "no-cache, no-store, must-revalidate" para respostas com conteúdo sensível [112].

Capítulo 9

Conclusão

Com o finalizar do presente estágio curricular, o aluno adquiriu valiosas competências no âmbito do funcionamento de uma empresa com processos reais e também uma compreensão das metodologias de trabalho utilizadas no processo de desenvolvimento de um *software*. Ao longo do processo de desenvolvimento, o aluno aplicou os conhecimentos obtidos durante o Mestrado em Engenharia Informática, bem como aprimorou conceitos devido ao trabalho próximo com colaboradores experientes da Grama.

Sendo o principal objetivo o desenvolvimento da Plataforma Interna da Grama, em paralelo com outro estagiário. Os alunos, sempre que necessário, trabalharam em conjunto de modo a obterem uma plataforma funcional e de qualidade, cumprindo todos os requisitos da empresa. Para tal, ao longo do primeiro semestre o aluno planeou o seu ano curricular, priorizando as tarefas de modo a gerir o tempo de forma eficaz. Definiu ainda a metodologia a usar e um plano de riscos.

O estudo do estado da arte permitiu ao aluno conhecer os concorrentes diretos e indiretos no mercado e concretizar a ideia da necessidade da plataforma para a empresa. De seguida, o aluno definiu os requisitos funcionais através de *user stories* e os requisitos não funcionais da plataforma que o auxiliaram, posteriormente, na escolha das tecnologias a utilizar no desenvolvimento.

Posto isto, o aluno teve de efetuar estudos sobre arquiteturas fomentando o seu conhecimento e, por vezes, envolvendo-se em debates com o colega estagiário e respetivos orientadores sobre o estudo realizado. Concluiu o semestre com o desenho da arquitetura da plataforma através do modelo C4, seguindo então uma abordagem de microsserviços e utilizando os serviços da AWS.

Findado o primeiro semestre, o aluno partiu para o segundo semestre com um planeamento cuidado e requisitos bem definidos para o desenvolvimento. Este iniciou o desenvolvimento em conjunto com o outro estagiário mantendo-se ambos em sincronia, dado que nos primeiros *sprints* foram realizadas tarefas comuns como a criação de componentes ou a integração do serviço *Cognito* na plataforma. Com a maioria dos componentes realizados, o aluno começou a desenvolver os seus microsserviços (ausências e recrutamento) cujo *backend* foi realizado em

Spring Boot e o *frontend* em React.

Perto de meio da fase de desenvolvimento foi necessário ajustar o plano pois previa-se falta de tempo para concluir todas as tarefas propostas. Feito esse ajuste o aluno conseguiu realizar os requisitos de maior prioridade a que se propôs, aprendendo bastante não só nas tecnologias em que desenvolveu a plataforma, mas também com as *code reviews* feitas por *developers* experientes da empresa que, sempre que necessário, corrigiam boas práticas de programação.

Após o término da fase de desenvolvimento os requisitos funcionais desenvolvidos foram novamente testados. Assim como foram efetuados testes aos requisitos não funcionais para garantir as métricas previamente definidas.

Pese embora as dificuldades encontradas ao nível das tecnologias, o principal obstáculo foi, sem dúvida, a falta de tempo para concluir todas as tarefas com qualidade, tendo sido necessário uma repriorização dos requisitos. No entanto, as expectativas do aluno foram superadas pois não só adquiriu valiosas competências técnicas mas também ao nível pessoal e de trabalho em equipa através da exposição a problemas do mundo real e com a colaboração com *developers* experientes.

Em suma, embora seja ainda necessária uma revisão ao nível da *performance* e responsividade das páginas, bem como o desenvolvimento de algumas funcionalidades que complementam os módulos implementados, o aluno considerara que os principais objetivos do estágio foram cumpridos. Todos os requisitos de alta prioridade foram concluídos, e a plataforma encontra-se preparada para futuramente receber mais módulos e funcionalidades. Deste modo, a Plataforma Interna de Grama é modular e customizável às necessidades da empresa, sendo possível produtizá-la para outras empresas com realidades semelhantes às da Grama.

Referências

- [1] PurelyHR. <https://www.purelyhr.com/product/overview/>. Acedido a 30 de setembro de 2022.
- [2] Microsoft Excel. <https://www.microsoft.com/pt-pt/microsoft-365/excel>. Acedido a 23 de junho de 2023.
- [3] *PurelyHR Reviews & Product Details*. <https://www.g2.com/products/purelyhr/reviews>. Acedido a 26 de junho de 2023.
- [4] Absence.io. <https://www.absence.io/features/vacation-tracker/>. Acedido a 30 de setembro de 2022.
- [5] BambooHR. <https://www.bamboohr.com/>. Acedido a 29 de setembro de 2022.
- [6] BambooHR. *Applicant Tracking System*. <https://www.bamboohr.com/applicant-tracking/>. Acedido a 29 de setembro de 2022.
- [7] BambooHR. *Time-Off Management Software*. <https://www.bamboohr.com/paid-time-off/>. Acedido a 29 de setembro de 2022.
- [8] Calamari. *Leave management system*. <https://calamari.io/leave-management>. Acedido a 30 de setembro de 2022.
- [9] ClanHR. <http://www.clanhr.com/>. Acedido a 30 de setembro de 2022.
- [10] Greenhouse. <https://www.greenhouse.io/uk/recruiting>. Acedido a 30 de setembro de 2022.
- [11] JazzHR. <https://www.jazzhr.com/>. Acedido a 29 de setembro de 2022.
- [12] Teamtailor. <https://www.teamtailor.com/>. Acedido a 30 de setembro de 2022.
- [13] Workable. <https://www.workable.com/>. Acedido a 30 de setembro de 2022.
- [14] Zoho. <https://www.zoho.com/>. Acedido a 28 de setembro de 2022.
- [15] Zoho Analytics. <https://www.zoho.com/analytics/>. Acedido a 24 de junho de 2023.
- [16] Zoho Payroll. <https://www.zoho.com/payroll/>. Acedido a 24 de junho de 2023.

- [17] Zoho Sign. <https://www.zoho.com/sign/>. Acedido a 24 de junho de 2023.
- [18] Zoho Sign. <https://www.zoho.com/workerly/>. Acedido a 24 de junho de 2023.
- [19] Zoho. Zoho people, *Attendance management that empowers your workforce*. <https://www.zoho.com/people/attendance-tracker.html?src=ft>. Acedido a 28 de setembro de 2022.
- [20] Zoho. Zoho recruit. <https://www.zoho.com/recruit/?src=zoho-home&ireft=ohome>. Acedido a 28 de setembro de 2022.
- [21] Zoho. Zoho shifts. <https://www.zoho.com/shifts/?src=zoho-home&ireft=ohome>. Acedido a 28 de setembro de 2022.
- [22] BambooHR. *Pricing*. <https://www.bamboohr.com/pricing/>. Acedido a 30 de setembro de 2022.
- [23] Microsoft. <https://www.microsoft.com>. Acedido a 23 de junho de 2023.
- [24] Google Sheets. <https://www.google.com/sheets/about/>. Acedido a 23 de junho de 2023.
- [25] Zoho Sheet. <https://www.zoho.com/ar/sheet/>. Acedido a 23 de junho de 2023.
- [26] LibreOffice Calc. <https://www.libreoffice.org/discover/calc/>. Acedido a 23 de junho de 2023.
- [27] Microsoft. *Employee absence schedule*. <https://templates.office.com/en-us/employee-absence-schedule-tm03987167>. Acedido a 5 de outubro de 2022.
- [28] Simple Sheets. *Recruitment Management Excel Template*. <https://www.simplesheets.co/recruitment-management>. Acedido a 5 de outubro de 2022.
- [29] Ivo Anjos. Mapa de férias em excel: 5 razões para não o fazer, 2019. <http://www.clanhr.com/blog/mapa-ferias-excel-razoes-para-nao-fazer>. Acedido a 5 de outubro de 2022.
- [30] Max Rehkopf. *Agile epics: definition, examples, and templates*. <https://www.atlassian.com/agile/project-management/epics>. Acedido a 9 de outubro de 2022.
- [31] Max Rehkopf. *User stories with examples and a template*. <https://www.atlassian.com/agile/project-management/user-stories>. Acedido a 9 de outubro de 2022.
- [32] MoSCoW *Prioritisation*. <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html>. Acedido a 9 de outubro de 2022.
- [33] Andrei Gridnev. *Non-functional requirements, quality attributes and better software architecture*. <https://andreigridnev.medium.com/non-functional-requirements-quality-attributes-and-better-software-architecture-855425310e60>. Acedido a 3 de outubro de 2022.

-
- [34] RickKazman Len Bass, Paul Clements. *Software Architecture in Practice*. Universidade de São Paulo, 3 edition, 2013. Consultado a 3 de outubro de 2022.
- [35] *How To Measure And Optimize Page Load Time*. <https://www.debugbear.com/docs/page-load-time#what-is-a-good-page-load-time>. Acedido a 23 de junho de 2023.
- [36] A proteção de dados ao abrigo do RGPD. https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/index_pt.htm. Acedido a 6 de outubro de 2022.
- [37] *Stack Overflow Trends*. <https://insights.stackoverflow.com/trends?tags=reactjs%2Cangular%2Cvue.js%2Csvelte%2Cember.js%2Cbackbone.js>. Acedido a 3 de novembro de 2022.
- [38] *What is Angular?* <https://angular.io/guide/what-is-angular>. Acedido a 3 de novembro de 2022.
- [39] *Two-way binding*. <https://angular.io/guide/two-way-binding>. Acedido a 3 de novembro de 2022.
- [40] Jamal Osman. *Top 10 Popular Frontend Frameworks to Use in 2022*. <https://appmaster.io/blog/popular-frontend-frameworks>. Acedido a 2 de novembro de 2022.
- [41] React. <https://reactjs.org/>. Acedido a 3 de novembro de 2022.
- [42] Svelte. <https://svelte.dev/>. Acedido a 3 de novembro de 2022.
- [43] *What is Vue?* <https://vuejs.org/guide/introduction.html#what-is-vue>. Acedido a 3 de novembro de 2022.
- [44] Angular Github. <https://github.com/angular/angular>. Acedido a 3 de novembro de 2022.
- [45] React Github. <https://github.com/facebook/react>. Acedido a 3 de novembro de 2022.
- [46] Svelte Github. <https://github.com/sveltejs/svelte>. Acedido a 3 de novembro de 2022.
- [47] Vue.js Github. <https://github.com/vuejs/vue>. Acedido a 3 de novembro de 2022.
- [48] Carlos Garcia. *Top 10 Best Web Backend Frameworks in 2022 for Web Development*. <https://appmaster.io/blog/10-best-web-backend-frameworks>. Acedido a 2 de novembro de 2022.
- [49] *Why Django?* <https://www.djangoproject.com/start/overview/>. Acedido a 3 de novembro de 2022.
- [50] *Pros and Cons of Django as Web Framework*. <https://techvidvan.com/tutorials/pros-and-cons-of-django/>. Acedido a 3 de novembro de 2022.

- [51] Spring Boot *Overview*. <https://spring.io/projects/spring-boot>. Acedido a 3 de novembro de 2022.
- [52] *What is Quarkus?* <https://quarkus.io/about/>. Acedido a 3 de novembro de 2022.
- [53] Iryna Deremuk. *Quarkus vs Spring Boot – What’s the Difference Between Popular Techs?* <https://litslink.com/blog/slug-quarkus-vs-spring-boot-the-difference>. Acedido a 2 de novembro de 2022.
- [54] Express. <https://expressjs.com/>. Acedido a 3 de novembro de 2022.
- [55] Tejas Kaneriya. *Advantages & Disadvantages of Node.js : Why to Use Node.js?* <https://expressjs.com/>. Acedido a 3 de novembro de 2022.
- [56] Django Github. <https://github.com/django/django>. Acedido a 3 de novembro de 2022.
- [57] Spring Boot Github. <https://github.com/spring-projects/spring-boot>. Acedido a 3 de novembro de 2022.
- [58] Quarkus Github. <https://github.com/quarkusio/quarkus>. Acedido a 3 de novembro de 2022.
- [59] Express.js Github. <https://github.com/expressjs/express>. Acedido a 3 de novembro de 2022.
- [60] Chandler Harris. *Microservices vs. monolithic architecture*. <https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith>. Acedido a 7 de novembro de 2022.
- [61] Chris Richardson. *Pattern: Microservice Architecture*. <https://microservices.io/patterns/microservices.html>. Acedido a 7 de novembro de 2022.
- [62] Chris Richardson. *What are microservices?* <https://microservices.io/>. Acedido a 6 de novembro de 2022.
- [63] Dick Gabriel. *What are Patterns*. <https://hillside.net/patterns/50-patterns-library/patterns/222-design-pattern-definition>. Acedido a 6 de novembro de 2022.
- [64] Chris Richardson. *A pattern language for microservices*. <https://microservices.io/patterns/>. Acedido a 6 de novembro de 2022.
- [65] Database per-service pattern. <https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-data-persistence/database-per-service.html>. Acedido a 9 de novembro de 2022.
- [66] Chris Richardson. *Pattern: Database per service*. <https://microservices.io/patterns/data/database-per-service.html>. Acedido a 9 de novembro de 2022.
- [67] Chris Richardson. *Pattern: Saga*. <https://microservices.io/patterns/data/saga.html>. Acedido a 9 de novembro de 2022.

-
- [68] Chris Richardson. *Pattern: API Composition*. <https://microservices.io/patterns/data/api-composition.html>. Acedido a 9 de novembro de 2022.
- [69] *CQRS pattern*. <https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-data-persistence/cqrs-pattern.html>. Acedido a 9 de novembro de 2022.
- [70] Chris Richardson. *Pattern: Shared database*. <https://microservices.io/patterns/data/shared-database.html>. Acedido a 9 de novembro de 2022.
- [71] *Shared-database-per-service pattern*. <https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-data-persistence/shared-database.html>. Acedido a 9 de novembro de 2022.
- [72] Chris Richardson. *Pattern: API Gateway / Backends for Frontends*. <https://microservices.io/patterns/apigateway.html>. Acedido a 9 de novembro de 2022.
- [73] Json Web Token. <https://jwt.io/>. Acedido a 23 de junho de 2023.
- [74] Chris Richardson. *Pattern: Access token*. <https://microservices.io/patterns/security/access-token.html>. Acedido a 9 de novembro de 2022.
- [75] *What is cloud computing?* <https://www.oracle.com/pt/cloud/what-is-cloud-computing/>. Acedido a 10 de novembro de 2022.
- [76] *What are public, private, and hybrid clouds?* <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-private-public-hybrid-clouds/#overview>. Acedido a 10 de novembro de 2022.
- [77] *Types of Cloud Computing*. https://aws.amazon.com/types-of-cloud-computing/?WICC-N=tile&tile=types_of_cloud. Acedido a 10 de novembro de 2022.
- [78] *Cloud computing with AWS*. https://aws.amazon.com/what-is-aws/?nc1=h_ls. Acedido a 10 de novembro de 2022.
- [79] *AWS Global Infrastructure*. https://aws.amazon.com/about-aws/global-infrastructure/?nc1=h_ls. Acedido a 10 de novembro de 2022.
- [80] *What's the Difference Between AWS vs. Azure vs. Google Cloud?* <https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud>. Acedido a 10 de novembro de 2022.
- [81] *What is Azure?* <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/>. Acedido a 10 de novembro de 2022.
- [82] *Google Cloud overview*. <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/>. Acedido a 10 de novembro de 2022.

- [83] José Vidal. *Google Cloud Storage: Pros/Cons and how to use it with JavaScript*. <https://medium.com/dailyjs/google-cloud-storage-pros-cons-and-how-to-use-it-with-javascript-ea9ce60a94c0>. Acedido a 10 de novembro de 2022.
- [84] *Compare AWS and Azure services to Google Cloud*. <https://cloud.google.com/free/docs/aws-azure-gcp-service-comparison>. Acedido a 10 de novembro de 2022.
- [85] *Cloud infrastructure services vendor market share worldwide from 4th quarter 2017 to 1st quarter 2022*. <https://www.statista.com/statistics/967365/worldwide-cloud-infrastructure-services-market-share-vendor/>. Acedido a 11 de novembro de 2022.
- [86] Amazon CloudFront. <https://aws.amazon.com/cloudfront/>. Acedido a 24 de junho de 2023.
- [87] Amazon Cognito. <https://aws.amazon.com/cognito/>. Acedido a 24 de junho de 2023.
- [88] Amazon EC2. <https://aws.amazon.com/ec2/>. Acedido a 24 de junho de 2023.
- [89] Amazon RDS. <https://aws.amazon.com/rds/>. Acedido a 24 de junho de 2023.
- [90] Amazon SES. <https://aws.amazon.com/ses/>. Acedido a 24 de junho de 2023.
- [91] Amazon S3. <https://aws.amazon.com/s3/>. Acedido a 24 de junho de 2023.
- [92] Amazon ELB. <https://aws.amazon.com/elasticloadbalancing/>. Acedido a 24 de junho de 2023.
- [93] *The C4 model for visualising software architecture*. <https://c4model.com/>. Acedido a 12 de dezembro de 2022.
- [94] *Audit database*. <https://www.ibm.com/docs/en/psfa/7.2.1?topic=history-audit-database>. Acedido a 19 de dezembro de 2022.
- [95] *Manifesto for Agile Software Development*. <https://agilemanifesto.org/>. Acedido a 26 de outubro de 2022.
- [96] *What is Scrum?* <https://www.scrum.org/resources/what-is-scrum>. Acedido a 26 de outubro de 2022.
- [97] Chandana. *Scrum Project Management: Advantages and Disadvantages*. https://www.simplilearn.com/scrum-project-management-article#advantages_and_disadvantages_of_scrum_project_management. Acedido a 26 de outubro de 2022.
- [98] Linear. <https://linear.app/>. Acedido a 23 de junho de 2023.

-
- [99] *Levels of a Risk Matrix*. <https://www.vectorsolutions.com/resources/blogs/levels-of-a-risk-matrix/>. Acedido a 15 de novembro de 2022.
- [100] JUNIT. <https://junit.org/junit5/>. Acedido a 26 de junho de 2023.
- [101] Swagger. <https://swagger.io/>. Acedido a 26 de junho de 2023.
- [102] OWASP Zed Attack Proxy. <https://www.zaproxy.org/>. Acedido a 26 de junho de 2023.
- [103] OWASP ZAP Alerts. <https://www.zaproxy.org/docs/desktop/start/features/alerts/>. Acedido a 26 de junho de 2023.
- [104] *Content Security Policy (CSP)*. <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>. Acedido a 26 de junho de 2023.
- [105] *Content Security Policy*. <https://content-security-policy.com/>. Acedido a 26 de junho de 2023.
- [106] *Clickjacking*. <https://owasp.org/www-community/attacks/Clickjacking>. Acedido a 26 de junho de 2023.
- [107] *Missing Anti-clickjacking Header*. <https://www.zaproxy.org/docs/alerts/10020-1/>. Acedido a 26 de junho de 2023.
- [108] *Timestamp Disclosure - Unix*. <https://scanrepeat.com/web-security-knowledge-base/timestamp-disclosure---unix>. Acedido a 26 de junho de 2023.
- [109] *Nothing up my sleeve numbers*. <https://www.cryptologie.net/article/221/nothing-up-my-sleeve-numbers/>. Acedido a 26 de junho de 2023.
- [110] *X-Content-Type-Options Header Missing*. <https://www.zaproxy.org/docs/alerts/10021/>. Acedido a 26 de junho de 2023.
- [111] *Information Disclosure - Suspicious Comments*. <https://www.zaproxy.org/docs/alerts/10027/>. Acedido a 26 de junho de 2023.
- [112] *Re-examine Cache-control Directives*. <https://www.zaproxy.org/docs/alerts/10015/>. Acedido a 26 de junho de 2023.

Apêndices

Apêndice A

Epic e User stories

Módulo de Autenticação

Módulo referente à autenticação de todos os utilizadores (administrador, gestor e colaborador) na Plataforma Interna da Grama.

Autenticação

ES-1: Como administrador autenticado **quero** enviar um convite **de modo a** adicionar um utilizador à plataforma.

US-1: Enviar convite para a plataforma

- (a) Como administrador autenticado **quero** enviar um convite para a plataforma **de modo a** que o utilizador possa efetuar o seu primeiro registo.
- (b) Como administrador autenticado **quero** poder reenviar um convite para a plataforma **de modo a** que o utilizador possa efetuar o seu primeiro registo.
- (c) Como administrador autenticado **quero** poder reenviar um convite para a plataforma **de modo a** que o utilizador possa recuperar o seu acesso.

ES-2: Como administrador/gestor/colaborador que nunca acedeu à plataforma **quero** efetuar o meu registo através de um convite **de modo a** obter acesso à plataforma.

US-2: Efetuar primeiro início de sessão

- (a) Como utilizador não autenticado **quero** efetuar o primeiro registo na plataforma **de modo a** aceder ao conteúdo da plataforma.

ES-3: Como administrador/gestor/colaborador não autenticado **quero** iniciar sessão **de modo a** conseguir aceder à plataforma.

US-3: Iniciar sessão

(a) **Como** administrador/gestor/colaborador **quero** iniciar sessão na plataforma **de modo a** aceder ao conteúdo da plataforma. Quero inserir:

- *Email**;
- Palavra-passe*.

(* campo obrigatório)

ES-4: **Como** administrador/gestor/colaborador não autenticado **quero** poder recuperar a minha palavra-passe **de modo a** recuperar o acesso à minha conta.

US-4: Recuperar palavra-passe

(a) **Como** administrador/gestor/colaborador **quero** recuperar a minha palavra-passe **de modo a** conseguir recuperar o acesso à minha conta. Quero inserir o meu *email* e, de seguida, receber um *email* com um código de recuperação. É obrigatório definir uma nova palavra-passe para poder aceder à conta.

ES-5: **Como** administrador/gestor/colaborador autenticado **quero** terminar a sessão **de modo a** sair da plataforma em segurança.

US-5: Terminar sessão

(a) **Como** administrador/gestor/colaborador **quero** terminar a sessão **de modo a** poder sair da plataforma.

Módulo Ausências

Este módulo é relativo à gestão de ausências dos colaboradores da Grama. Este módulo poderá ser acedido pelos três tipos de atores e, como cada ator terá diferentes funcionalidades, as seguintes secções representam separadamente as *epic/user stories* para cada um destes.

Administrador

Na seguinte secção são descritas as *epic* e *user stories* relativas ao administrador da plataforma no módulo de ausências.

Dashboard

ES-6: **Como** administrador autenticado **quero** um painel onde me sejam apresentadas algumas métricas referentes às ausências dos colaboradores **de modo a** analisar e gerir os novos pedidos de ausência.

US-6: Visualizar métricas

(a) **Como** administrador autenticado **quero** ver métricas referentes às ausências dos colaboradores **de modo a** simplificar a sua gestão. Quero ver:

- Sumário de ausências:
 - Número colaboradores ausentes por mês por departamento;
 - Selecionar o ano para o qual quero visualizar as ausências.
- Pedidos de ausência:
 - Aprovados;
 - Pendentes;
 - Cancelados.
- Próximos aniversários de colaboradores;
- Gráfico do número de colaboradores ausentes nos próximos 7 dias.

US-7: Pesquisar e aplicar filtros à *dashboard*

(a) **Como** administrador autenticado **quero** poder filtrar a informação da *dashboard* **de modo a** pesquisar e aplicar filtros às métricas da *dashboard*. Quero poder pesquisar e filtrar:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Gerir calendário

ES-7: **Como** administrador autenticado **quero** uma representação esquemática **de modo a** poder ver facilmente que colaboradores estão ausentes e em que dias.

US-8: Visualizar ausências em calendário - mês

(a) **Como** administrador autenticado **quero** ver as ausências dos colaboradores num calendário mensal **de modo a** conseguir visualizar facilmente quem são os colaboradores ausentes e em que dias do mês. Quero ver:

- Dia da semana;
- Dia do mês;
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação caso o colaborador esteja ausente só de manhã ou só de tarde;
 - Indicação se o pedido está aprovado ou pendente.

US-9: Visualizar ausências em calendário - semana

(a) **Como** administrador autenticado **quero** ver as ausências dos colaboradores num calendário semanal **de modo a** poder analisar quem são os colaboradores ausentes e em que dias da semana. Quero ver:

- Dia do mês;
- Dia da semana;
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação caso o colaborador esteja ausente só de manhã ou só de tarde;
 - Indicação se o pedido está aprovado ou pendente.

US-10: Visualizar ausências em calendário - dia

(a) **Como** administrador autenticado **quero** ver as ausências dos colaboradores num dia, **de modo a** saber quem são os colaboradores ausentes e em que períodos desse dia. **Quero** também conseguir de forma fácil selecionar novos dias para serem analisados. Quero ver:

- Dia do mês;
- Dia da semana;
- Período do dia:
 - Dia completo;
 - Manhã;
 - Tarde.
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação se o pedido está aprovado ou pendente.

US-11: Aplicar filtros ao calendário

(a) **Como** administrador autenticado **quero** poder filtrar a informação do calendário **de modo a** apresentar a informação que procuro. Quero filtrar por:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Gerir pedidos

ES-8: **Como** administrador autenticado **quero** gerir os pedidos de ausência dos colaboradores **de modo a** ser possível analisar e avaliar os pedidos de ausência dos colaboradores.

US-12: Visualizar pedidos

(a) **Como** administrador autenticado **quero** visualizar os pedidos de ausência **de modo a** analisar os pedidos pendentes para os aprovar ou rejeitar. Quero ver:

- Colaborador que requisitou o pedido;
- Tipo da ausência;
- Dia(s) do pedido;
- Tipo do pedido de ausência;
- Número de dias pedidos pelo colaborador.

US-13: Aprovar pedido

(a) **Como** administrador autenticado **quero** aprovar um pedido **de modo a** que o período pedido seja concedido ao colaborador, este seja notificado e o seu balanço de ausências seja atualizado.

US-14: Rejeitar pedido

(a) **Como** administrador autenticado **quero** rejeitar um pedido **de modo a** que o período pedido não seja concedido ao colaborador e este seja notificado, atualizando seu balanço de ausências.

US-15: Apagar pedido

(a) **Como** administrador autenticado **quero** apagar um pedido **de modo a** que o pedido não seja efetuado.

US-16: Editar detalhes do pedido

(a) **Como** administrador autenticado **quero** editar os detalhes de um pedido **de modo a** efetuar alterações ao mesmo.

US-17: Pesquisar e aplicar filtros à lista de pedidos

(a) **Como** administrador autenticado **quero** pesquisar e aplicar filtros à lista de pedidos **de modo a** encontrar os pedidos que desejo. Quero filtrar por:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Gerir ausências

ES-9: Como administrador autenticado **quero** gerir as ausências aprovadas e canceladas dos colaboradores **de modo a** ser possível analisar e editar as suas informações.

US-18: Ver detalhes de uma ausência

(a) **Como** administrador autenticado **quero** ver os detalhes de uma ausência **de modo a** obter mais informação sobre a ausência. Quero ver:

- Fotografia do colaborador;
- Nome do colaborador;
- Tipo de ausência;
- Data(s) da ausência;
- Período de ausência:
 - Dia completo;
 - Manhã;
 - Tarde.
- Número de dias pedidos pelo colaborador.

US-19: Editar detalhes de uma ausência

(a) **Como** administrador autenticado **quero** editar os detalhes de uma ausência **de modo a** alterar informação relacionada com a ausência. Quero editar:

- Data(s) da ausência;
- Tipo do pedido;
- Período de ausência:
 - Dia completo;
 - Manhã;
 - Tarde.

Requisitar ausência

ES-10: Como administrador autenticado **quero** efetuar um pedido de ausência **de modo a** este ser posteriormente enviado para aprovação.

US-20: Requisitar ausência - dia completo

(a) **Como** administrador autenticado **quero** pedir um dia completo de ausência **de modo a** obter aprovação. Quero inserir:

- Se o pedido é feito em nome de outro utilizador*;

- Tipo do pedido*;
- Dia da ausência*;
- Tipo de ausência*;
- Adicionar uma nota;
- Anexar ficheiro.

(* campo obrigatório)

(b) **Como** administrador autenticado **quero** poder ver o balanço do colaborador de quem estou a fazer o pedido **de modo a** saber quantos dias o colaborador tem disponíveis. Quero ver:

- Tipo de ausência;
- Número de dias permitidos;
- Balanço total;
- Número de dias usados até à data;
- Número de dias pendentes.

US-21: Requisitar ausência - dia parcial

(a) **Como** administrador autenticado **quero** pedir um dia parcial de ausência **de modo a** que possa enviar o pedido para aprovação. Quero inserir:

- Se o pedido é feito em nome de outro utilizador*;
- Tipo do pedido*;
- Dia da ausência*:
- Período do dia*:
 - Manhã;
 - Tarde.
- Tipo de ausência*;
- Adicionar uma nota;
- Anexar ficheiro.

(* campo obrigatório)

(b) **Como** administrador autenticado **quero** poder ver o balanço do colaborador de quem estou a fazer o pedido **de modo a** saber quantos dias o colaborador tem disponíveis. Quero ver:

- Tipo de ausência;
- Número de dias permitidos;
- Balanço total;
- Número de dias usados até à data;

- Número de dias pendentes.

US-22: Requisitar ausência - múltiplos dias

(a) **Como** administrador autenticado **quero** pedir mais do que um dia de ausência **de modo a** enviar o pedido para aprovação. Quero inserir:

- Se o pedido é feito em nome de outro utilizador*;
- Tipo do pedido*;
- Data de início da ausência*;
- Data de fim da ausência*;
- Tipo de ausência*;
- Adicionar uma nota;
- Anexar ficheiro.

(* campo obrigatório)

(b) **Como** administrador autenticado **quero** poder ver o balanço do colaborador de quem estou a fazer o pedido **de modo a** saber quantos dias o colaborador tem disponíveis. Quero ver:

- Tipo de ausência;
- Número de dias permitidos;
- Balanço total;
- Número de dias usados até à data;
- Número de dias pendentes.

Notificar

ES-11: **Como** administrador autenticado **quero** ser notificado via *email* **de modo a** tomar conhecimento das ações que necessitam de aprovação.

US-23: Receber *email* com pedido de aprovação

(a) **Como** administrador autenticado **quero** receber um *email* com o pedido de aprovação **de modo a** ficar informado com o novo pedido de ausência do colaborador. Quero receber:

- Nome do colaborador que efetuou o pedido;
- Detalhes da ausência:
 - Data e hora em que o pedido foi efetuado;
 - Tipo do pedido;
 - Tipo da ausência;
 - Período da ausência;

- Duração da ausência;
- Notas adicionais.
- *Link* direto para a gestão de pedidos.

US-24: Enviar *email* informativo sobre a ausência

- (a) **Como** administrador autenticado **quero** que seja enviado um *email* após a submissão de uma ausência **de modo a** notificar o colaborador.

Gerir tipos de ausências

ES-12: **Como** administrador autenticado **quero** ter acesso ao tipo de ausências existentes **de modo a** conseguir, de forma simples, aceder e gerir todos os tipos de ausências.

US-25: Visualizar tipos disponíveis

- (a) **Como** administrador autenticado **quero** ver a lista de tipos de ausências disponíveis **de modo a** ver facilmente todos os tipos de ausências existentes. Quero ver:

- Nome do tipo de ausência;
- Detalhes do tipo de ausência.

ES-13: **Como** administrador autenticado **quero** gerir os tipos de ausência **de modo a** estes ficarem disponíveis e atualizados para serem utilizados aquando de um pedido.

US-26: Adicionar tipo

- (a) **Como** administrador autenticado **quero** adicionar novos tipos de ausência **de modo a** que os colaboradores possam efetuar pedidos com esse tipo. Quero inserir:

- Nome*;
- Dedutível*;
- Pago*.

(* campo obrigatório)

US-27: Editar tipo

- (a) **Como** administrador autenticado **quero** editar tipos de ausência existentes **de modo a** estes ficarem atualizados. Quero editar:

- Nome*;

- Dedutível*;
- Pago*.

(* campo obrigatório)

US-28: Arquivar tipo

- (a) **Como** administrador autenticado **quero** arquivar os tipos de ausência existentes **de modo a** estes não poderem voltar a ser utilizados pelos colaboradores.

Gerir balanço

ES-14: Como administrador autenticado **quero** ter acesso ao balanço de dias de ausência de cada colaborador **de modo a** ser possível gerir esse balanço.

US-29: Visualizar balanço dos colaboradores ativos

- (a) **Como** administrador autenticado **quero** ver a lista de colaboradores ativos **de modo a** conseguir analisar o balanço atual e gasto dos colaboradores. Quero ver:

- Nome do colaborador;
- Data da última atualização do balanço;
- Total de dias disponíveis por ano;
- Quantidade de dias de ausência que possui;
- Quantidade de dias de ausência gastos (até à data);
- Data de entrada do colaborador.

US-30: Editar balanço

- (a) **Como** administrador autenticado **quero** editar o balanço de um colaborador **de modo a** adicionar ou remover dias de ausência, atualizando assim o seu balanço. Quero editar:

- Número de dias*;

(* campo obrigatório)

US-31: Aplicar filtros à lista de colaboradores

- (a) **Como** administrador autenticado **quero** aplicar filtros à lista de colaboradores **de modo a** analisar e gerir o balanço atual e gasto dos colaboradores. Quero filtrar por:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Gerir eventos especiais

ES-15: Como administrador autenticado **quero** gerir a existência de eventos especiais **de modo a** que os colaboradores vejam esses eventos nos seus calendários.

US-32: Visualizar eventos especiais

(a) **Como** administrador autenticado **quero** ver a lista de eventos especiais **de modo a** saber em que dias já existem eventos. Quero ver:

- Nome do evento;
- Data de início;
- Data de fim;
- Calendário de eventos.

US-33: Adicionar eventos especiais

(a) **Como** administrador autenticado **quero** adicionar eventos especiais **de modo a** estes serem adicionados ao calendário dos colaboradores. Quero inserir:

- Nome do evento;
- Dias em que o evento ocorre:
 - Dia único;
 - Múltiplos dias.
- Data de início;
- Data de fim.

Gestor

Na seguinte secção são descritas as *epic* e *user stories* relativas ao gestor da plataforma no módulo de ausências.

Dashboard

ES-16: Como gestor autenticado **quero** um painel onde me sejam apresentadas algumas métricas referentes à minha equipa **de modo a** analisar e gerir as ausências.

US-34: Visualizar métricas

(a) **Como** gestor autenticado **quero** ver métricas referentes às ausências da minha equipa **de modo a** simplificar a sua gestão. Quero ver:

- Sumário de ausências:

- Número de ausências, por departamento, por mês;
- Selecionar o ano para o qual quero visualizar as ausências.

- Pedidos de ausência:
 - Aprovados;
 - Pendentes;
 - Rejeitados;
 - Cancelados.

- Próximos aniversários de colaboradores;

- Gráfico do número de colaboradores ausentes nos próximos 7 dias.

US-35: Pesquisar e aplicar filtros à *dashboard*

(a) **Como** gestor autenticado **quero** poder filtrar a informação da *dashboard* **de modo a** pesquisar e aplicar filtros às métricas da *dashboard*. Quero poder pesquisar e filtrar:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Gerir calendário

ES-17: **Como** gestor autenticado **quero** uma representação esquemática **de modo a** poder ver facilmente que colaboradores estão ausentes e em que dias.

US-36: Visualizar ausências em calendário - mês

(a) **Como** gestor autenticado **quero** ver as ausências de todos colaboradores num calendário mensal **de modo a** conseguir visualizar facilmente quem são os colaboradores ausentes e em que dias do mês. Quero ver:

- Dia da semana;
- Dia do mês;
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação caso o colaborador esteja ausente só de manhã ou só de tarde;
 - Indicação se o pedido está aprovado ou pendente para os colaboradores do seu departamento.

US-37: Visualizar ausências em calendário - semana

(a) **Como** gestor autenticado **quero** ver as ausências dos colaboradores num calendário semanal **de modo a** poder analisar quem são os colaboradores ausentes e em que dias da semana. Quero ver:

- Dia do mês;
- Dia da semana;
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação caso o colaborador esteja ausente só de manhã ou só de tarde;
 - Indicação se o pedido está aprovado ou pendente para os colaboradores do seu departamento.

US-38: Visualizar ausências em calendário - dia

(a) **Como** gestor autenticado **quero** ver as ausências dos colaboradores num dia, **de modo a** saber ver quem são os colaboradores ausentes e em que períodos desse dia. **Quero** também conseguir de forma fácil selecionar novos dias para serem analisados. Quero ver:

- Dia do mês;
- Dia da semana;
- Período do dia:
 - Dia completo;
 - Manhã;
 - Tarde.
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação se o pedido está aprovado ou pendente para os colaboradores do seu departamento.

US-39: Aplicar filtros ao calendário

(a) **Como** gestor autenticado **quero** poder filtrar a informação do calendário **de modo a** apresentar a informação que procuro. Quero filtrar por:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Gerir pedidos

ES-18: Como gestor autenticado **quero** ver os pedidos de ausência dos colaboradores da minha equipa **de modo a** ser possível analisar e avaliar os pedidos de ausência.

US-40: Visualizar pedidos

(a) Como gestor autenticado **quero** visualizar os pedidos de ausência da minha equipa **de modo a** tomar conhecimento quais os colaboradores fora e em que dias. Quero ver:

- Colaborador que requisitou o pedido;
- Tipo da ausência;
- Dia(s) do o pedido;
- Tipo do pedido de ausência;
- Número de dias pedidos pelo colaborador;

US-41: Aprovar pedido

(a) Como gestor autenticado **quero** aprovar um pedido de um colaborador da minha equipa **de modo a** que o período pedido seja concedido ao colaborador, este seja notificado e o seu balanço de ausências seja atualizado.

US-42: Rejeitar pedido

(a) Como gestor autenticado **quero** rejeitar um pedido de um colaborador da minha equipa **de modo a** que o período pedido não seja concedido ao colaborador e este seja notificado, atualizando o seu balanço.

US-43: Ver detalhes de uma ausência

(a) Como gestor autenticado **quero** ver os detalhes de uma ausência **de modo a** obter mais informação sobre a ausência. Quero ver:

- Fotografia do colaborador;
- Nome do colaborador;
- Tipo de ausência;
- Data(s) da ausência;
- Período de ausência:
 - Dia completo;
 - Manhã;
 - Tarde.
- Número de dias pedidos pelo colaborador.

US-44: Pesquisar e aplicar filtros à lista de pedidos

(a) **Como** gestor autenticado **quero** pesquisar e aplicar filtros à lista de pedidos **de modo a** encontrar os pedidos que desejo. Quero filtrar por:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Notificar

ES-19: **Como** gestor autenticado **quero** ser notificado via *email* **de modo a** tomar conhecimento das ações da minha equipa que necessitam de aprovação.

US-45: Receber *email* com pedido de aprovação

(a) **Como** gestor autenticado **quero** receber um *email* com o pedido de aprovação **de modo a** ficar informado com o novo pedido de ausência do colaborador da minha equipa. Quero receber:

- Nome do colaborador que efetuou o pedido;
- Detalhes da ausência:
 - Data e hora em que o pedido foi efetuado;
 - Tipo do pedido;
 - Tipo da ausência;
 - Período da ausência;
 - Duração da ausência;
 - Notas adicionais.
- *Link* para a aba de gestão de pedidos.

Gerir balanço

ES-20: **Como** gestor autenticado **quero** ter acesso ao balanço de dias de ausência de cada colaborador da minha equipa **de modo a** ser possível gerir esse balanço.

US-46: Visualizar balanço dos colaboradores ativos

(a) **Como** gestor autenticado **quero** ver a lista de colaboradores ativos pertencentes à minha equipa **de modo a** conseguir analisar o balanço atual e gasto dos colaboradores da minha equipa. Quero ver:

- Nome do colaborador;
- Data da última atualização do balanço;

- Total de dias disponíveis por ano;
- Quantidade de dias de ausência que possui;
- Quantidade de dias de ausência gastos (até à data);
- Data de entrada do colaborador.

US-47: Aplicar filtros à lista de colaboradores

(a) **Como** gestor autenticado **quero** aplicar filtros à lista de colaboradores **de modo a** analisar e gerir o balanço atual e gasto dos colaboradores. Quero filtrar por:

- Nome do colaborador;
- Departamento;
- Tipo de ausência;
- Tipo do pedido.

Colaborador

Na seguinte secção são descritas as *epic* e *user stories* relativas ao colaborador da plataforma no módulo de ausências. Os administradores e também os gestores herdam as permissões de colaborador, por isso, as seguintes *user stories* também são associadas a esses tipos de utilizadores.

Dashboard

ES-21: **Como** colaborador autenticado **quero** um painel onde me sejam apresentadas algumas métricas referentes às minhas ausências **de modo a** poder analisar e gerir as minhas ausências.

US-48: Visualizar métricas

(a) **Como** colaborador autenticado **quero** ver métricas referentes às minhas ausências **de modo a** conseguir gerir as minhas ausências. Quero ver:

- Sumário de ausências:
 - Número dias de ausência por mês;
 - Selecionar o ano para o qual quero visualizar as ausências;
 - Alternar entre tipo de ausência.
- Pedidos de ausência:
 - Aprovados;
 - Pendentes;
 - Rejeitados;
 - Cancelados.

- O meu sumário de ausências;

US-49: Pesquisar e aplicar filtros à *dashboard*

(a) **Como** colaborador autenticado **quero** poder filtrar a informação da *dashboard* **de modo a** pesquisar e aplicar filtros às métricas da *dashboard*. Quero poder pesquisar e filtrar:

- Tipo de ausência;
- Tipo do pedido.

Gerir calendário

ES-22: **Como** colaborador autenticado **quero** uma representação esquemática **de modo a** poder ver facilmente que colaboradores estão ausentes e em que dias.

US-50: Visualizar ausências em calendário - mês

(a) **Como** colaborador autenticado **quero** ver as ausências aprovadas de todos os colaboradores num calendário mensal **de modo a** conseguir visualizar facilmente quem são os colaboradores ausentes e em que dias do mês. Quero ver:

- Dia da semana;
- Dia do mês;
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação caso o colaborador esteja ausente só de manhã ou só de tarde.

US-51: Visualizar ausências em calendário - semana

(a) **Como** colaborador autenticado **quero** ver as ausências aprovadas de todos os colaboradores num calendário semanal **de modo a** poder analisar quem são os colaboradores ausentes e em que dias da semana. Quero ver:

- Dia do mês;
- Dia da semana;
- Informação do colaborador associada aos dias em que está ausente:
 - Nome;
 - Indicação caso o colaborador esteja ausente só de manhã ou só de tarde.

US-52: Visualizar ausências em calendário - dia

(a) **Como** colaborador autenticado **quero** ver as ausências aprovadas de todos os colaboradores num dia, **de modo a** saber ver quem são os colaboradores ausentes e em que períodos desse dia. **Quero** também conseguir de forma fácil selecionar novos dias para serem analisados. Quero ver:

- Dia do mês;
- Dia da semana;
- Período do dia:
 - Dia completo;
 - Manhã;
 - Tarde.
- Nome do colaborador associado aos dias em que está ausente.

US-53: Aplicar filtros ao calendário

(a) **Como** colaborador autenticado **quero** poder filtrar a informação do calendário **de modo a** apresentar a informação que procuro. Quero filtrar por:

- Nome do colaborador;
- Departamento.

Gerir pedidos

ES-23: **Como** colaborador autenticado **quero** ver os meus pedidos de ausência **de modo a** ser possível controlar os meus pedidos de ausência.

US-54: Visualizar pedidos

(a) **Como** colaborador autenticado **quero** visualizar os meus pedidos de ausência **de modo a** tomar conhecimento de quais os pedidos que efetuei. Quero ver:

- Tipo da ausência;
- Dia(s) do pedido;
- Tipo do pedido de ausência;
- Número de dias pedidos.

US-55: Cancelar pedido

(a) **Como** colaborador autenticado **quero** cancelar um pedido **de modo a** que o pedido não seja efetuado e o meu balanço atualizado.

US-56: Editar detalhes do pedido

(a) **Como** administrador autenticado **quero** editar os detalhes de um pedido por mim efetuado **de modo a** efetuar alterações ao mesmo.

Requisitar ausência

ES-24: Como colaborador autenticado **quero** efetuar um pedido de ausência **de modo a** este ser posteriormente enviado para aprovação.

US-57: Requisitar ausência - dia completo

(a) **Como** colaborador autenticado **quero** pedir um dia completo de ausência **de modo a** obter aprovação. Quero inserir:

- Tipo do pedido*;
- Dia da ausência*;
- Tipo de ausência*;
- Adicionar uma nota;
- Anexar ficheiro.

(* campo obrigatório)

(b) **Como** colaborador autenticado **quero** poder ver o meu balanço de ausências **de modo a** saber quantos dias tenho disponíveis. Quero ver:

- Tipo de ausência;
- Número de dias permitidos;
- Balanço total;
- Número de dias usados até à data;
- Número de dias pendentes.

US-58: Requisitar ausência - dia parcial

(a) **Como** colaborador autenticado **quero** pedir um dia parcial de ausência **de modo a** que possa enviar o pedido para aprovação. Quero inserir:

- Tipo do pedido*;
- Período do dia*:
 - Manhã;
 - Tarde.
- Tipo de ausência*;
- Adicionar uma nota;
- Anexar ficheiro.

(* campo obrigatório)

(b) **Como** colaborador autenticado **quero** poder ver o meu balanço de ausências **de modo a** saber quantos dias tenho disponíveis. Quero ver:

- Tipo de ausência;
- Número de dias permitidos;
- Balanço total;
- Número de dias usados até à data;
- Número de dias pendentes.

US-59: Requisitar ausência - múltiplos dias

(a) **Como** colaborador autenticado **quero** pedir mais do que um dia de ausência **de modo a** enviar o pedido para aprovação. Quero inserir:

- Tipo do pedido*;
- Data de início da ausência*;
- Data de fim da ausência*;
- Tipo de ausência*;
- Adicionar uma nota;
- Anexar ficheiro.

(* campo obrigatório)

(b) **Como** colaborador autenticado **quero** poder ver o meu balanço de ausências **de modo a** saber quantos dias tenho disponíveis. Quero ver:

- Tipo de ausência
- Número de dias permitidos;
- Balanço total;
- Número de dias usados até à data;
- Número de dias pendentes.

Notificar

ES-25: **Como** colaborador autenticado **quero** ser notificado via *email* **de modo a** tomar conhecimento dos resultados dos meus pedidos.

US-60: Receber *email* com o resultado do pedido

(a) **Como** colaborador autenticado **quero** receber um *email* com o resultado do pedido **de modo a** receber informação sobre a situação do meu pedido de ausência.

US-61: Receber *email* informativo sobre ausência

(a) **Como** colaborador autenticado **quero** que receber um *email* após a submissão de uma ausência em meu nome **de modo a** ser notificado dessa ausência

Módulo Recrutamento

Este capítulo é relativo ao módulo de recrutamento de colaboradores para a Grama. Este módulo será acedido pelos administradores e gestores, logo, as seguintes secções representam separadamente as *epic / user stories* para cada um destes.

Administrador

Na seguinte secção são descritas as *epic e user stories* relativas ao administrador da plataforma no módulo de recrutamento.

Gerir quadro de recrutamento

ES-26: Como administrador autenticado **quero** ter acesso à *pipeline* de recrutamento **de modo a**, de forma simples, conseguir gerir o processo de recrutamento.

US-62: Ver candidatos ativos

(a) **Como** administrador autenticado **quero** ver quais os candidatos ativos **de modo a** ser facilmente identificável a sua fase no recrutamento. Quero ver:

- Nome do candidato;
- Vaga à qual se candidata;
- Experiência profissional;
- Fase no processo de recrutamento.

US-63: Pesquisar e aplicar filtros ao quadro

(a) **Como** administrador autenticado **quero** pesquisar e aplicar filtros ao quadro **de modo a** encontrar os candidatos que pretendo. Quero pesquisar/filtrar:

- Entrevistador;
- Perfil;
- Experiência;
- Fonte de contacto.

ES-27: Como administrador autenticado **quero** adicionar candidatos **de modo a** estes ficarem associados à *pipeline* de recrutamento.

US-64: Adicionar candidato

(a) **Como** administrador autenticado **quero** adicionar um novo candidato **de modo a** este ser adicionado diretamente no quadro. Quero inserir:

- Coluna*;
- Nome*;
- LinkedIn;
- Entrevistador(es);
- Experiência*;
- Fonte de contacto*;
- Perfil*;
- Comentários.

(* campo obrigatório)

US-65: Adicionar candidato a uma coluna específica

(a) **Como** administrador autenticado **quero** adicionar um novo candidato a uma coluna específica **de modo a** este ficar associado à sua fase de recrutamento. Quero inserir:

- Nome*;
- LinkedIn;
- Entrevistador;
- Experiência*;
- Fonte de contacto*;
- Perfil*;
- Comentários.

(* campo obrigatório)

US-66: Adicionar candidato com informação detalhada

(a) **Como** administrador autenticado **quero** adicionar informação detalhada a um novo candidato **de modo a** que o seu perfil fique mais completo. Quero inserir:

- Nome*;
- Estado*;
- Perfil*;
- Experiência;
- Departamento;
- LinkedIn;
- *Email*;
- Número de telefone;
- Entrevistador(es)*;

- Fonte de contacto*;
- Empresa atual;
- Localização;
- Comentários;
- Anexar ficheiros.

(* campo obrigatório)

Customizar quadro de recrutamento

ES-28: Como administrador autenticado **quero** customizar o quadro de recrutamento **de modo a** adicionar, editar e remover as suas colunas.

US-67: Adicionar coluna

(a) **Como** administrador autenticado **quero** ter a possibilidade de adicionar uma coluna à *pipeline* **de modo a** controlar as fases de recrutamento de forma eficiente. Quero inserir:

- Posição*;
- Nome*;
- Estado*.

(* campo obrigatório)

US-68: Editar coluna

(a) **Como** administrador autenticado **quero** conseguir editar a informação de uma coluna **de modo a** atualizar a *pipeline* de recrutamento. Quero editar:

- Posição;
- Nome;
- Estado*.

US-69: Editar coluna diretamente no quadro

(a) **Como** administrador autenticado **quero** conseguir editar o nome de uma coluna diretamente no quadro **de modo a** atualizar essa coluna. **Quero** ainda poder editar a posição de uma coluna **de modo a** alterar a sua posição no quadro.

US-70: Remover coluna

(a) **Como** administrador autenticado **quero** remover uma coluna **de modo a** que a *pipeline* de recrutamento se mantenha atualizada.

Lista de candidatos

ES-29: Como administrador autenticado **quero** visualizar a lista de candidatos **de modo a** poder de forma simples controlar e gerir os candidatos.

US-71: Editar candidato pela tabela

(a) **Como** administrador autenticado **quero** editar a informação de um candidato através da tabela **de modo a** que os dados do candidato fiquem atualizados. Quero editar:

- Estado;
- Perfil;
- Experiência;
- Fonte de contacto.

US-72: Pré-visualizar informação do candidato

(a) **Como** administrador autenticado **quero** pré-visualizar a informação de um candidato **de modo a** obter mais informação sobre o candidato. Quero ver:

- Nome;
- Perfil;
- Experiência;
- LinkedIn;
- *Email*;
- Data da última edição;
- Comentários;
- Possibilidade de ver os detalhes completos.

US-73: Ver candidatos em lista

(a) **Como** administrador autenticado **quero** ver os candidatos através de uma lista **de modo a** identificar rapidamente a sua fase no recrutamento entre outras informações. Quero ver:

- Nome do candidato;
- Estado do candidato;
- Vaga à qual se candidata;
- Experiência profissional;
- Contacto;
- Empresa atual;

- Data da última interação com o candidato.

US-74: Arquivar candidato

- (a) **Como** administrador autenticado **quero** arquivar um candidato **de modo a** removê-lo da lista de recrutamento.

US-75: Recuperar candidato arquivado

- (a) **Como** administrador autenticado **quero** recuperar um candidato previamente arquivado **de modo a** recolocá-lo na lista de recrutamento.

US-76: Pesquisar e aplicar filtros à lista

- (a) **Como** administrador autenticado **quero** pesquisar e aplicar filtros à lista **de modo a** encontrar os candidatos na lista. Quero pesquisar/filtrar:

- Estado;
- Perfil;
- Experiência;
- Fonte de contacto.

Analisar candidatos

ES-30: **Como** administrador autenticado **quero** visualizar e gerir os vários candidatos **de modo a** poder ver e editar as suas informações.

US-77: Ver detalhes gerais

- (a) **Como** administrador autenticado **quero** visualizar todos os detalhes de um candidato **de modo a** aceder à sua informação. Quero ver:

- Data da adição;
- Data da última edição;
- Data da última interação;
- Detalhes de contacto:
 - Nome;
 - *Email*;
 - Número de telefone;
 - LinkedIn.
- Detalhes gerais:
 - Estado;
 - Perfil;

- Experiência;
- Departamento;
- Fonte de contacto;
- Empresa atual;
- Localização;
- Comentários;
- Ficheiros relacionados.
- Detalhes de cada interação:
 - Tipo de interação;
 - Data;
 - Comentários;
 - Resultado;
 - Entrevistador(es).

US-78: Adicionar interações

(a) **Como** administrador autenticado **quero** adicionar interações com o candidato **de modo a** estas ficarem registadas na plataforma. Quero inserir:

- Tipo de interação*;
- Data*;
- Comentários;
- Resultado;
- Entrevistador(es)*.

(* campo obrigatório)

US-79: Editar candidato via detalhes completos

(a) **Como** administrador autenticado **quero** editar a informação de um candidato através da aba dos detalhes completos **de modo a** que os dados do candidato fiquem atualizados. Quero editar:

- Detalhes de contacto:
 - Nome;
 - *Email*;
 - Número de telefone;
 - LinkedIn.
- Detalhes gerais:
 - Estado*;
 - Perfil*;
 - Experiência;
 - Departamento;

- Fonte de contacto*;
- Empresa atual;
- Localização;
- Comentários;
- Ficheiros relacionados.

(* campo obrigatório)

US-80: Editar interações

(a) **Como** administrador autenticado **quero** editar interações com o candidato **de modo a** que estas fiquem atualizadas. Quero editar:

- Tipo de interação*;
- Data*;
- Comentários;
- Resultado;
- Entrevistador(es).

(* campo obrigatório)

Gestor

Na seguinte secção são descritas as *epic* e *user stories* relativas ao gestor da plataforma no módulo de recrutamento.

Gerir quadro de recrutamento

ES-31: Como gestor autenticado **quero** ter acesso à *pipeline* de recrutamento **de modo a**, de forma simples, conseguir gerir processos de recrutamento.

US-81: Ver candidatos ativos

(a) **Como** gestor autenticado **quero** ver quais os candidatos ativos **de modo a** ser facilmente identificável a sua fase no recrutamento. Quero ver:

- Nome do candidato;
- Vaga à qual se candidata;
- Experiência profissional;
- Fase no processo de recrutamento.

US-82: Pesquisar e aplicar filtros ao quadro

(a) **Como** gestor autenticado **quero** pesquisar e aplicar filtros **de modo a** apresentar os candidatos que desejo no quadro. Quero pesquisar/filtrar:

- Entrevistador;
- Perfil;
- Experiência;
- Fonte de contacto.

Lista de candidatos

ES-32: Como gestor autenticado **quero** visualizar a lista de candidatos **de modo a** poder de forma simples controlar e gerir os candidatos.

US-83: Pré-visualizar informação do candidato

(a) **Como** gestor autenticado **quero** pré-visualizar a informação de um candidato **de modo a** obter mais informação sobre o candidato. Quero ver:

- Nome;
- Perfil;
- Experiência;
- LinkedIn;
- *Email*;
- Data da última edição;
- Comentários;
- Possibilidade de ver os detalhes completos.

US-84: Ver candidatos em lista

(a) **Como** gestor autenticado **quero** ver os candidatos através de uma lista **de modo a** identificar rapidamente a sua fase no recrutamento entre outras informações. Quero ver:

- Nome do candidato;
- Estado do candidato;
- Vaga à qual se candidata;
- Experiência profissional;
- Contacto;
- Empresa atual;
- Data da última interação com o candidato.

US-85: Pesquisar e aplicar filtros à lista

(a) **Como** gestor autenticado **quero** pesquisar e aplicar filtros à lista **de modo a** encontrar os candidatos na lista. Quero pesquisar/filtrar:

- Estado;
- Perfil;
- Experiência;
- Fonte de contacto.

Analisar candidatos

ES-33: Como gestor autenticado quero visualizar e gerir os vários candidatos **de modo a** poder ver e editar as suas informações.

US-86: Ver detalhes gerais

(a) **Como gestor autenticado quero** visualizar todos os detalhes de um candidato **de modo a** aceder à sua informação. Quero ver:

- Data da adição;
- Data da última edição;
- Data da última interação;
- Detalhes de contacto:
 - Nome;
 - *Email*;
 - Número de telefone;
 - LinkedIn.
- Detalhes gerais:
 - Estado;
 - Perfil;
 - Experiência;
 - Departamento;
 - Fonte de contacto;
 - Empresa atual;
 - Localização;
 - Comentários;
 - Ficheiros relacionados.
- Detalhes de cada interação:
 - Tipo de interação;
 - Data;
 - Comentários;
 - Resultado;
 - Entrevistador(es).

US-87: Adicionar interações

(a) **Como** gestor autenticado **quero** adicionar interações com o candidato **de modo a** estas ficarem registadas na plataforma. Quero inserir:

- Tipo de interação*;
- Data*;
- Comentários;
- Resultado;
- Entrevistador(es)*.

(* campo obrigatório)

Apêndice B

Funcionalidades Desenvolvidas

Neste apêndice, para além das funcionalidades de autenticação, estarão representadas as funcionalidades desenvolvidas para os administradores da plataforma, visto que a maior diferença é os dados a que cada ator tem acesso.

Módulo de Autenticação

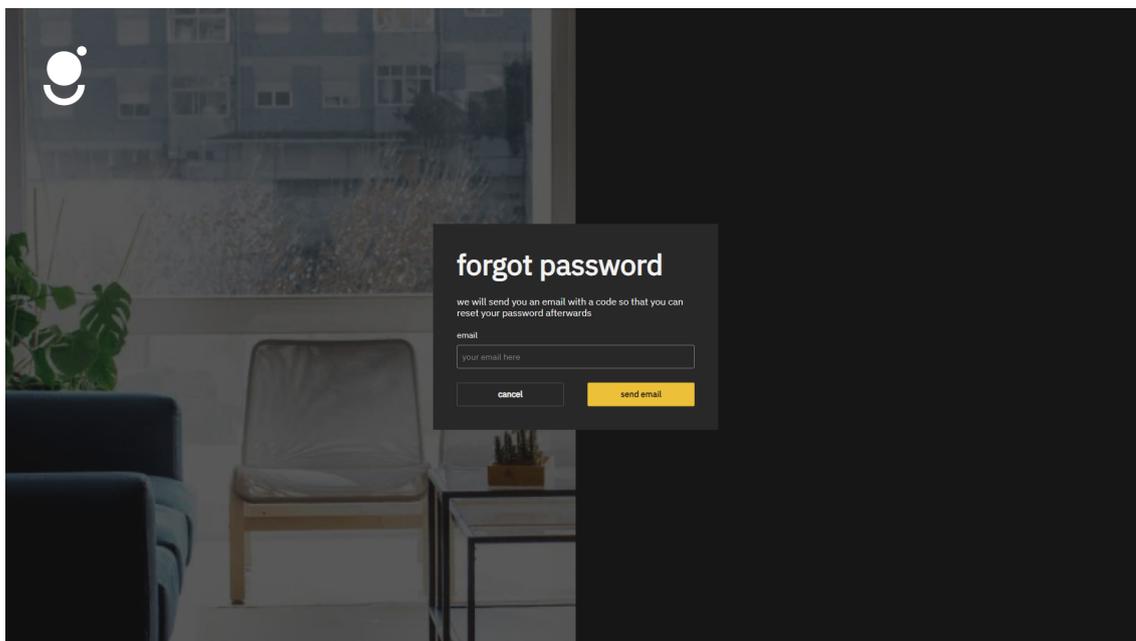


Figura B.1: Ecrã de recuperação da palavra-passe.

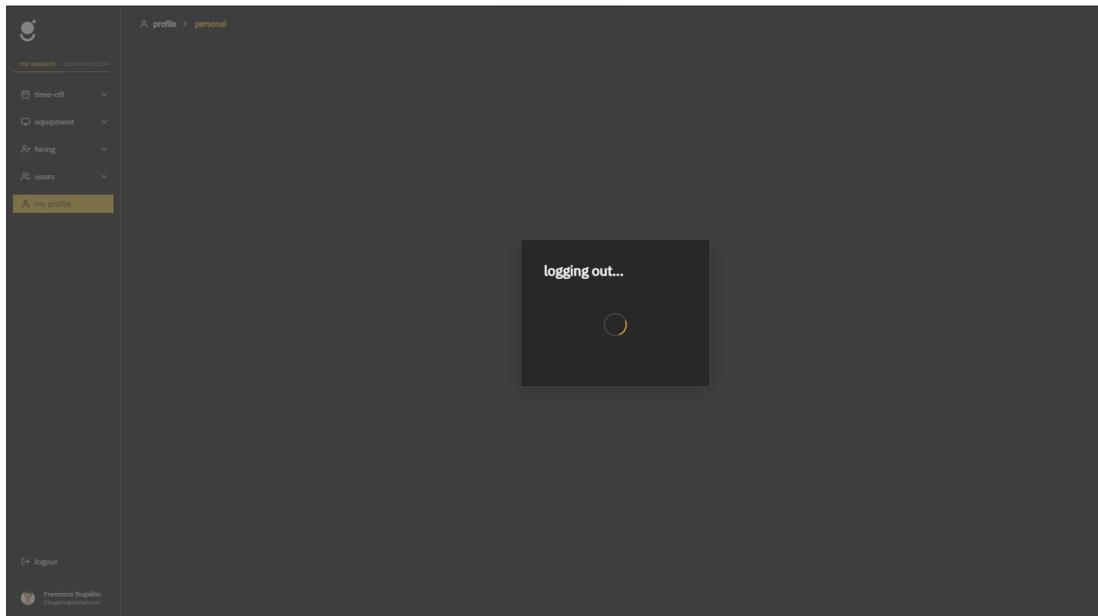


Figura B.2: Modal de término de sessão.

Módulo de Ausências

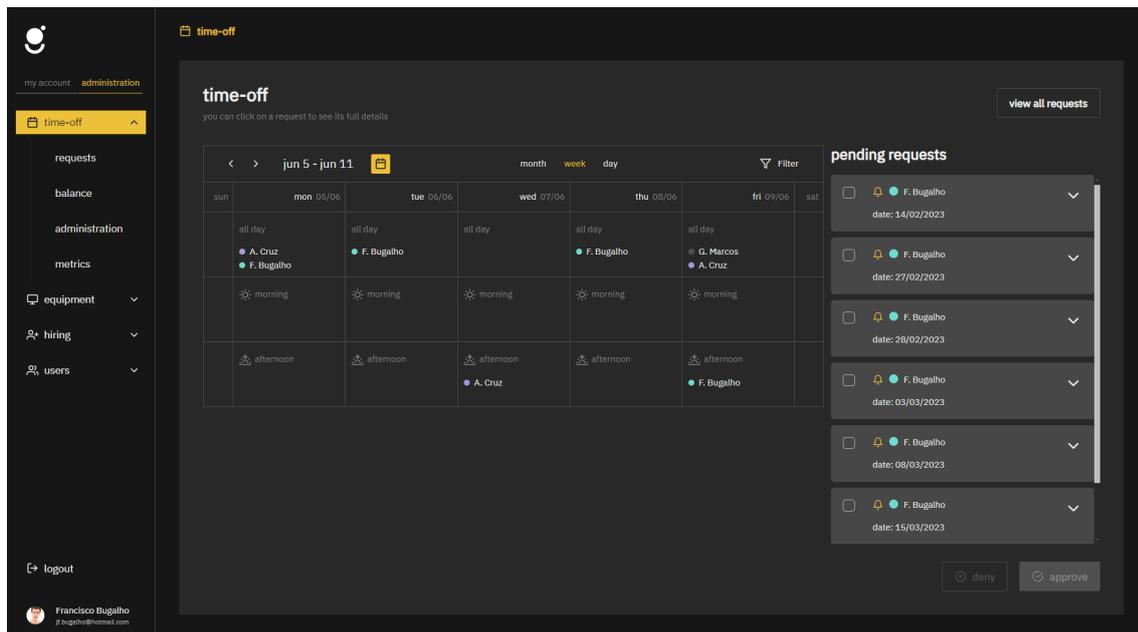


Figura B.3: Vista de uma semana em calendário.

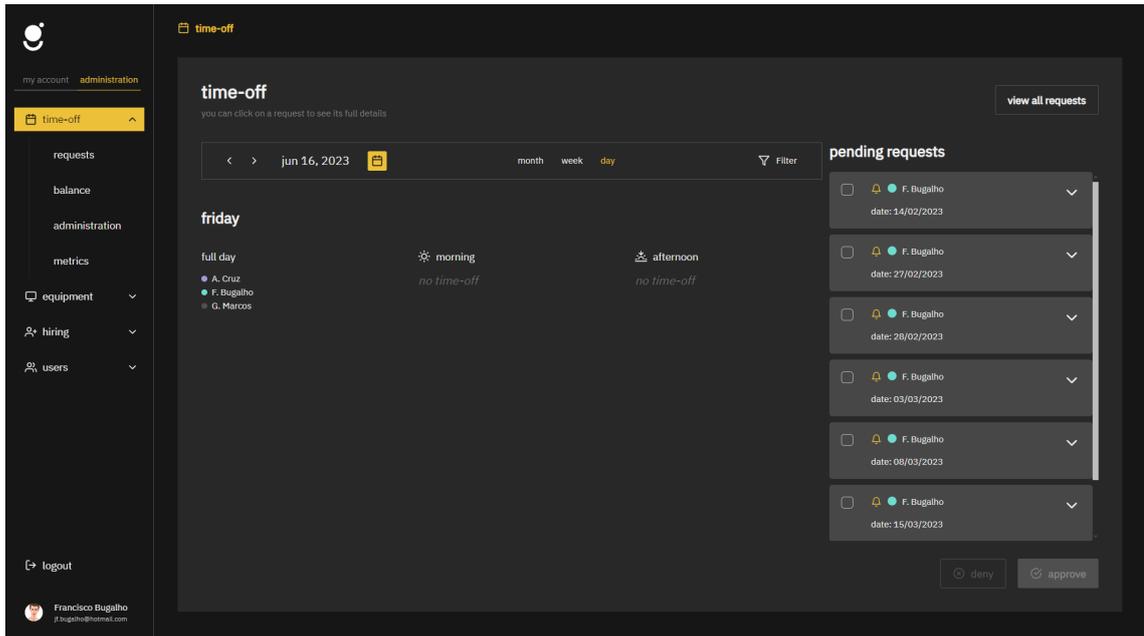


Figura B.4: Vista de um dia em calendário.

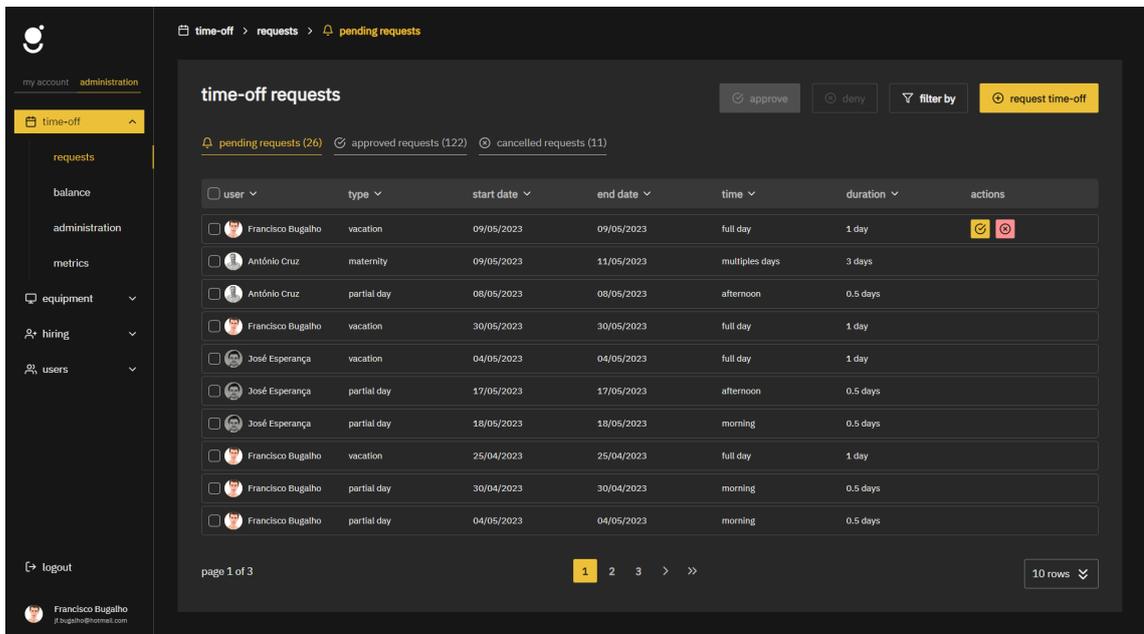


Figura B.5: Lista dos pedidos de ausência.

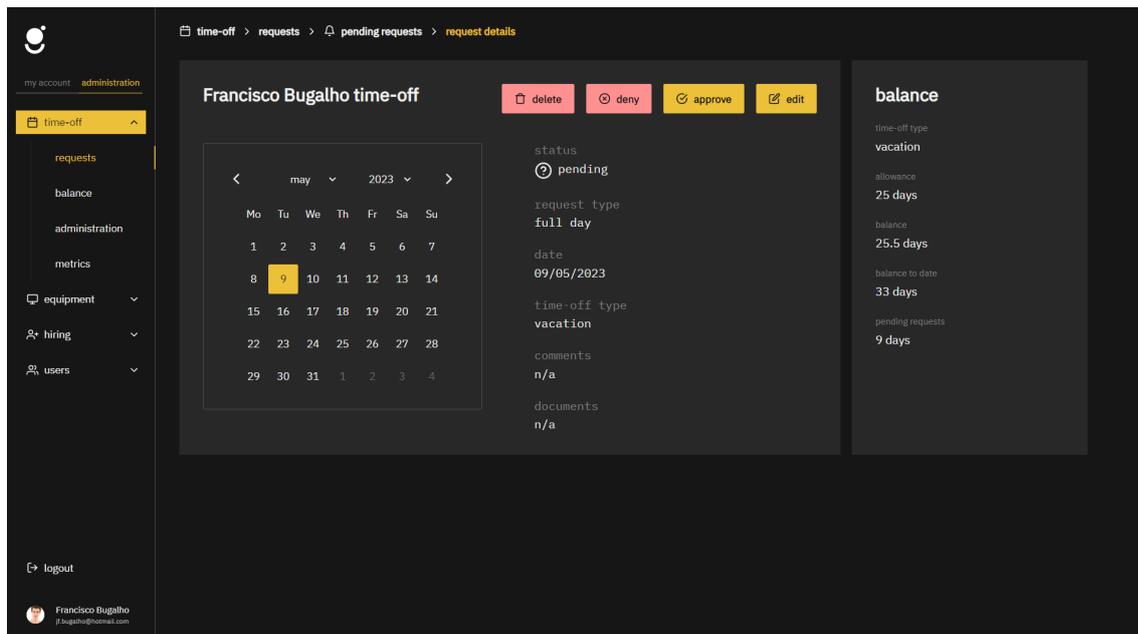


Figura B.6: Detalhes de um pedido de ausência.

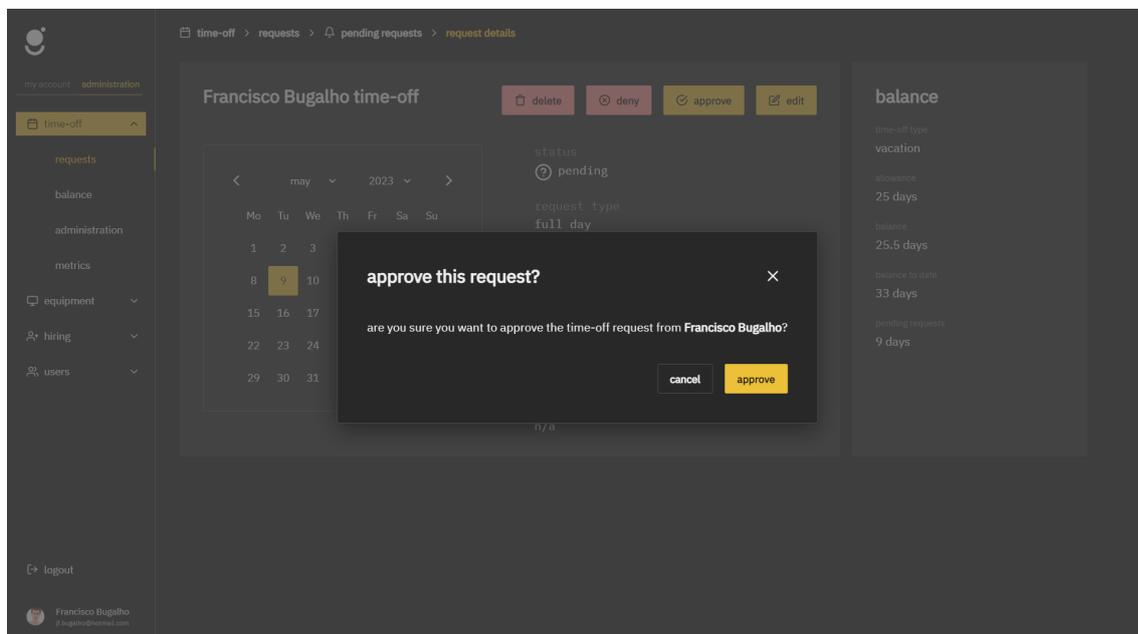


Figura B.7: Modal para aprovar um pedido de ausência (igual para cancelar e apagar).

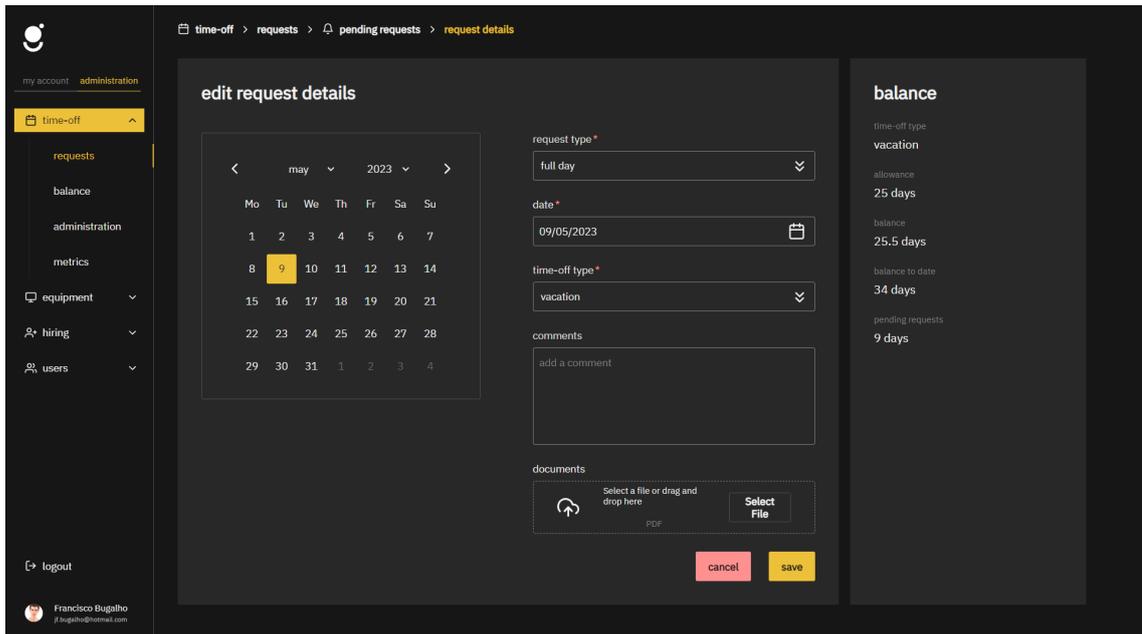


Figura B.8: Modo de edição de um pedido de ausência.

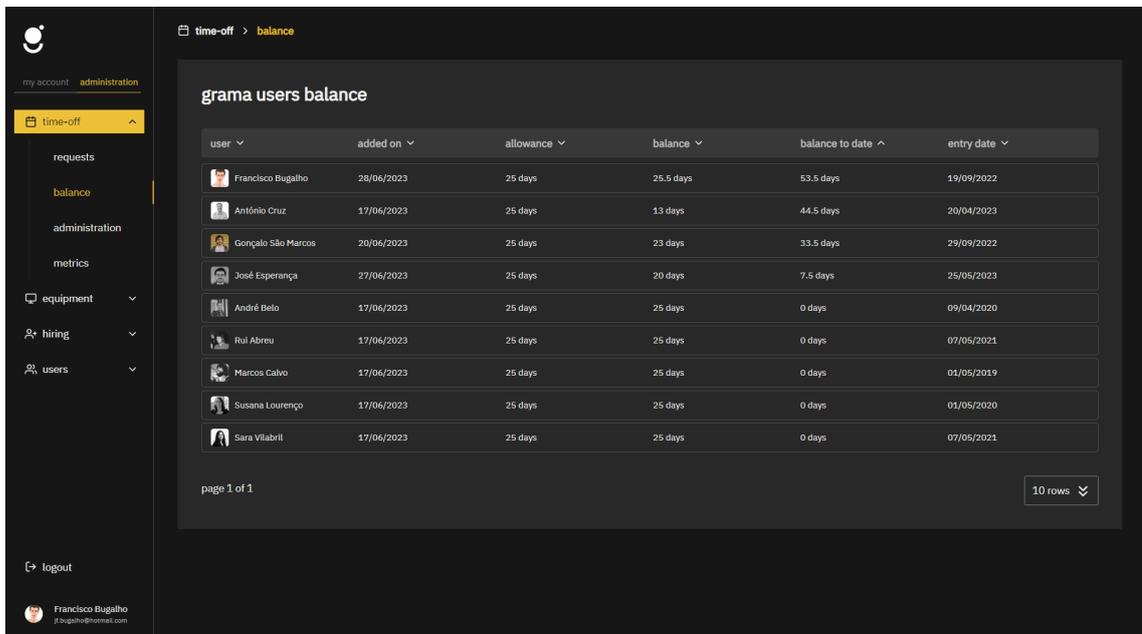


Figura B.9: Lista do balanço dos utilizadores.

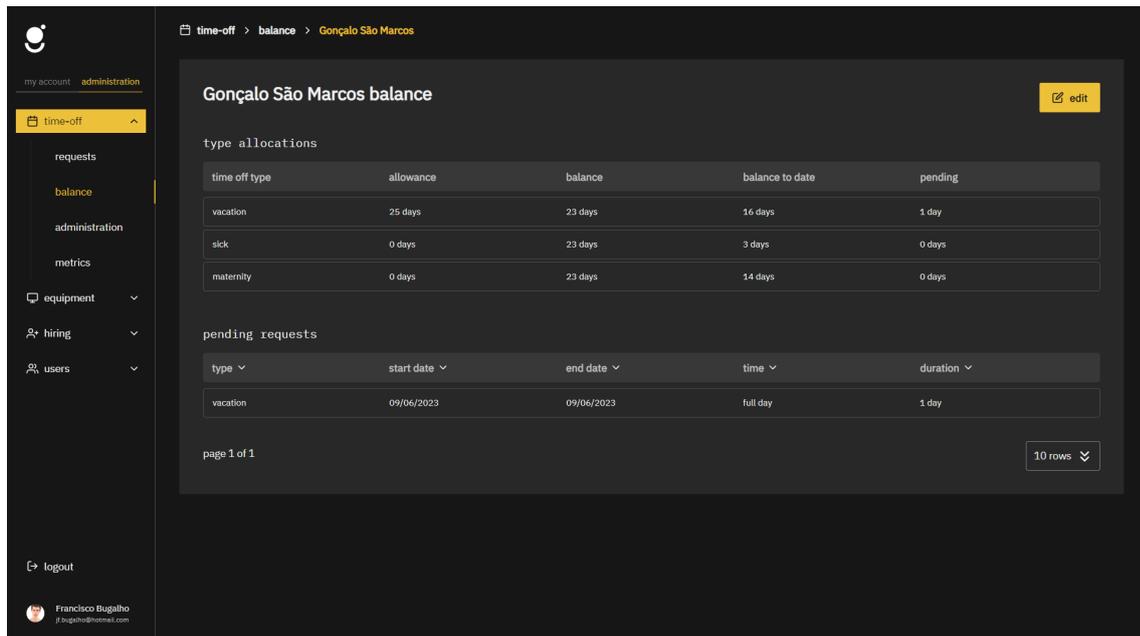


Figura B.10: Detalhes do balanço de um utilizador.

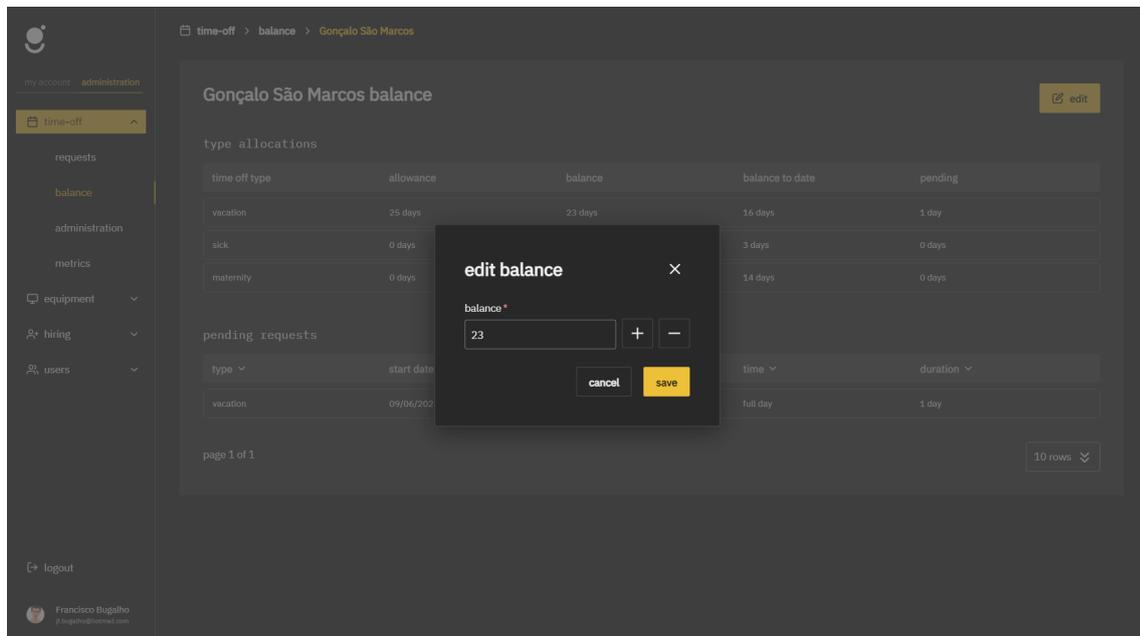


Figura B.11: Modo de edição do balanço de um utilizador.

Módulo de Recrutamento

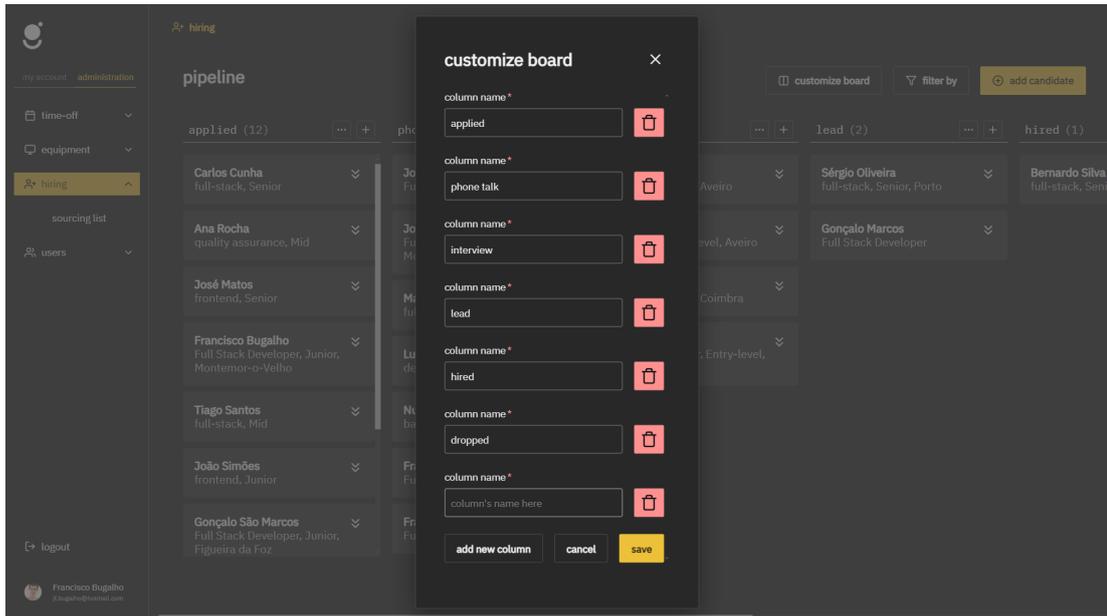


Figura B.12: Modo de edição da *pipeline* de recrutamento.

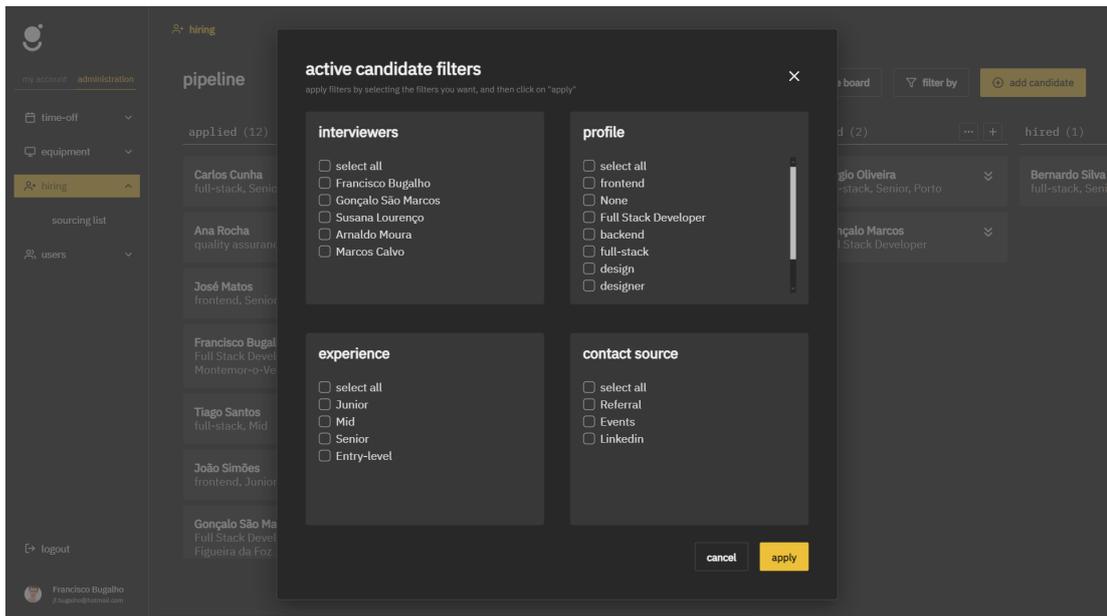


Figura B.13: Modal de filtros a aplicar na *pipeline* de recrutamento.

Figura B.14: Adição de um candidato com todos os detalhes.

<input type="checkbox"/>	name	status	profile	experience	contact source	current company	last interaction
<input type="checkbox"/>	J. Francisco Bugalho	Not Interested	Full Stack Developer	n/a	Events	n/a	24/04/2023
<input type="checkbox"/>	G. São Marcos	Not Interested	Full Stack Developer	n/a	Referral	n/a	never
<input type="checkbox"/>	Francisco J. Bugalho	Not Interested	Full Stack Developer	n/a	Referral	n/a	never

Figura B.15: Lista os candidatos arquivados.

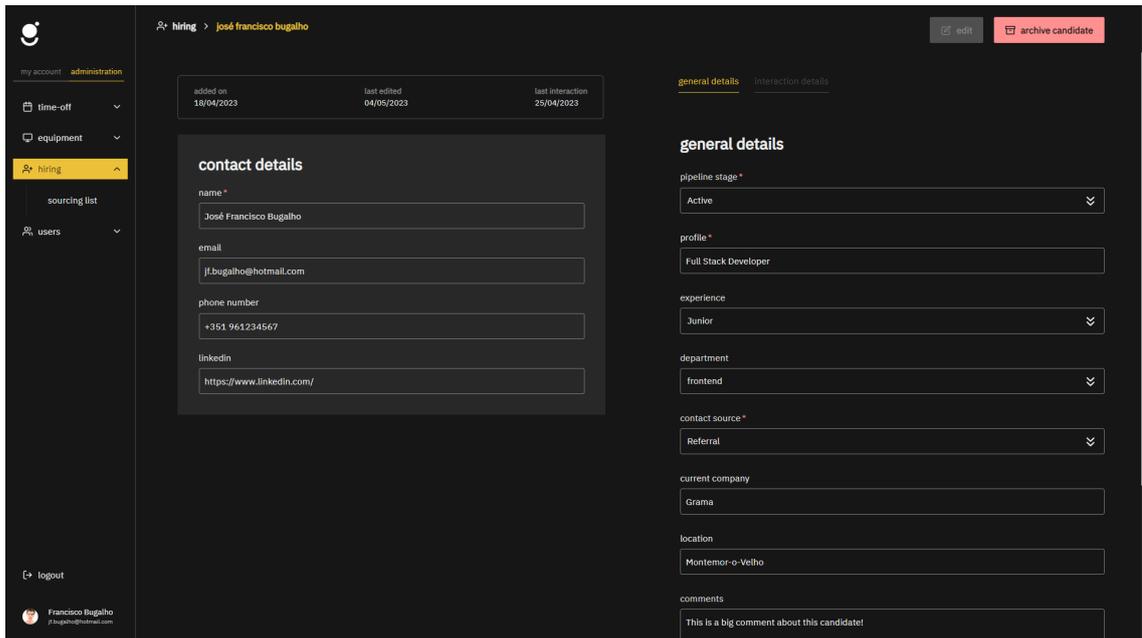


Figura B.16: Edição dos detalhes gerais de um candidato.

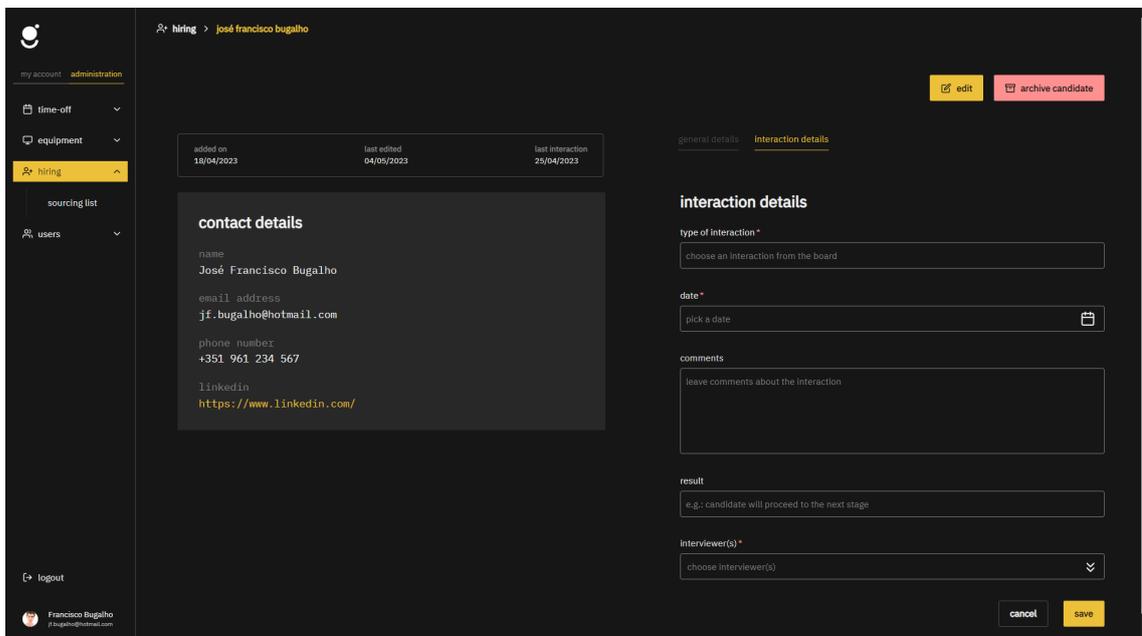


Figura B.17: Edição das interações de um candidato.

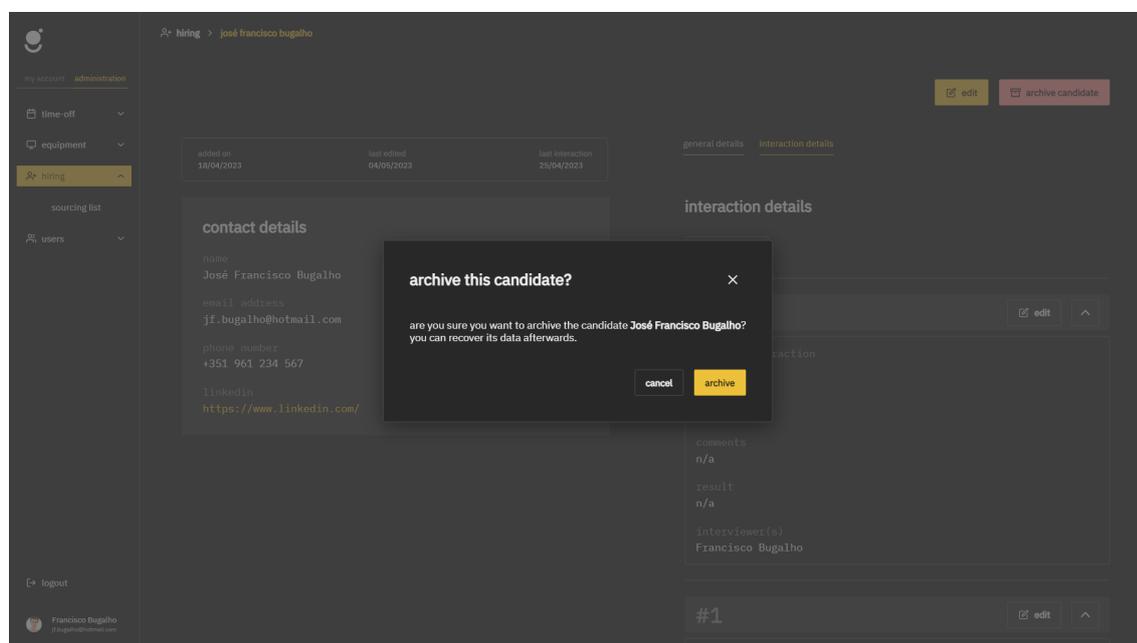


Figura B.18: Modal para arquivar um candidato (igual para recuperá-lo).

Apêndice C

Plano de Testes de Integração

Módulo de Autenticação

Tabela C.1: Plano de testes para o módulo de autenticação.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
1 (a)	1	Enviar convite para a plataforma	1. Criar um utilizador 2. Clicar no botão "save and invite"	O utilizador recebe um <i>email</i> com uma palavra-passe temporária	O utilizador recebe um <i>email</i> com uma palavra-passe temporária	Passou
1 (b) (c)	2	Reenviar convite para a plataforma	1. Aceder à vista detalhada de um utilizador 2. Clicar no botão "re-invite user"	O utilizador recebe um <i>email</i> com uma palavra-passe temporária	O utilizador recebe um <i>email</i> com uma palavra-passe temporária	Passou
2	3	Utilizador efetua o seu primeiro início de sessão	1. Aceder ao <i>site</i> da plataforma 2. Inserir <i>email</i> e palavra-passe temporária 3. Redefinir palavra-passe	O utilizador entra na plataforma	O utilizador entra na plataforma	Passou
3	4	Utilizador inicia sessão	1. Aceder ao <i>site</i> da plataforma 2. Inserir <i>email</i> e palavra-passe	O utilizador entra na plataforma	O utilizador entra na plataforma	Passou
4	5	Utilizador pretende recuperar a sua palavra-passe	1. Aceder ao <i>site</i> da plataforma 2. Selecionar o <i>link</i> "forgot password" 3. Colocar o <i>email</i> 4. Receber um <i>email</i> com um código 5. Preencher os campos: - Código - Palavra-passe - Confirmação da palavra-passe	A palavra-passe é alterada com sucesso e o utilizador entra na plataforma	A palavra-passe é alterada com sucesso e o utilizador entra na plataforma	Passou
5	6	Utilizador termina sessão	1. Utilizador com sessão iniciada clica no botão "logout"	É redirecionado para a página de início de sessão	É redirecionado para a página de início de sessão	Passou

Módulo de Ausências

Para a realização dos testes abaixo, as US foram agrupadas de forma a sintetizar a informação, sendo a diferença mais significativa as funcionalidades de colaborador pois este é o cargo com maiores restrições ao nível das funcionalidades e informação que pode aceder. Deste modo, para cada secção abaixo estarão representados os testes agrupados por US, contudo, em algumas secções existirá uma subsecção relativa aos testes de colaborador.

Dashboard

Tabela C.2: Plano de testes para a área *Dashboard*.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
6 e 34	7	Visualizar métricas	1. Aceder ao painel de métricas	Um gráfico com o sumário das ausências, pedidos de ausência, próximos aniversários e um gráfico com o número de colaboradores ausentes nos próximos 7 dias	Um gráfico com o sumário das ausências, pedidos de ausência, próximos aniversários e um gráfico com o número de colaboradores ausentes nos próximos 7 dias	Passou
7, 35 e 49	-	Não desenvolvido				

Colaborador

Tabela C.3: Plano de testes para a área *Dashboard* para colaboradores.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
48	31	Visualizar métricas	1. Aceder ao painel de métricas	Um gráfico com as suas ausências, pedidos de ausência e o seu sumário das ausências	Um gráfico com as suas ausências, pedidos de ausência e o seu sumário das ausências	Passou

Gerir Calendário

Tabela C.4: Plano de testes para a área Gerir Calendário.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
8, 36 e 50	8	Visualizar ausências em calendário mensal	1. Aceder à página "time-off"	É apresentado um calendário mensal com os colaboradores ausentes nos respetivos dias do mês	É apresentado um calendário mensal com os colaboradores ausentes nos respetivos dias do mês	Passou
9, 37 e 51	9	Visualizar ausências em calendário semanal	1. Aceder à página "time-off" 2. Clicar no <i>link</i> "week" no calendário	É apresentado um calendário semanal com os colaboradores ausentes nos respetivos dias da semana	É apresentado um calendário semanal com os colaboradores ausentes nos respetivos dias da semana	Passou
10, 38 e 53	10	Visualizar ausências por dia	1. Aceder à página "time-off" 2. Clicar no <i>link</i> "day" no calendário ou selecionar um dia no calendário mensal/semanal	É apresentado o respetivo dia com os colaboradores ausentes todo o dia, só de manhã ou só de tarde	É apresentado o respetivo dia com os colaboradores ausentes todo o dia, só de manhã ou só de tarde	Passou
11, 39 e 53	-	Não desenvolvido				

Gerir Pedidos

Tabela C.5: Plano de testes para a área Gerir Pedidos.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões	
12, 40 e 54	11	Visualizar lista dos pedidos de ausência	1. Aceder à página "requests"	Listagem dos pedidos de ausência	Listagem dos pedidos de ausência	Passou	
13 e 41	12	Aprovar um pedido pela tabela	1. Aceder à página "requests" 2. Na aba "pending requests" 3. Colocar o cursor no colaborador 4. Clicar no botão de aprovar	Pedido de ausência aprovado e colaborador notificado	Pedido de ausência aprovado e colaborador notificado	Passou	
13 e 41	13	Aprovar um pedido através de checkboxes	1. Aceder à página "requests" 2. Na aba "pending requests" 3. Clicar na checkbox do colaborador 4. Clicar no botão de aprovar ("approve") 5. Confirmar a ação na modal	Pedido de ausência aprovado e colaborador notificado	Pedido de ausência aprovado e colaborador notificado	Passou	
13 e 41	14	Aprovar um pedido na página de detalhes	1. Clicar na linha da tabela da página "requests" 2. Na página de detalhes clicar no botão de aprovar ("approve") 3. Confirmar a ação na modal	Pedido de ausência aprovado e colaborador notificado	Pedido de ausência aprovado e colaborador notificado	Passou	
13 e 41	15	Aprovar um pedido na página "time-off"	1. Aceder à página "time-off" 2. Selecionar o pedido clicando na checkbox 3. Clicar no botão de aprovar ("approve") 4. Confirmar a ação na modal	Pedido de ausência aprovado e colaborador notificado	Pedido de ausência aprovado e colaborador notificado	Passou	
14 e 42	16	Rejeitar um pedido pela tabela	1. Aceder à página "requests" 2. Na aba "pending requests" 3. Colocar o cursor no colaborador 4. Clicar no botão de rejeitar	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Passou	
14 e 42	17	Rejeitar um pedido através de checkboxes	1. Aceder à página "requests" 2. Na aba "pending requests" 3. Clicar na checkbox do colaborador 4. Clicar no botão de rejeitar ("deny") 5. Confirmar a ação na modal	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Passou	
14 e 42	18	Rejeitar um pedido na página de detalhes	1. Clicar na linha da tabela da página "requests" 2. Na página de detalhes clicar no botão de rejeitar ("deny") 3. Confirmar a ação na modal	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Passou	
14 e 42	19	Rejeitar um pedido na página "time-off"	1. Aceder à página "time-off" 2. Selecionar o pedido clicando na checkbox 3. Clicar no botão de rejeitar ("deny") 4. Confirmar a ação na modal	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Pedido de ausência rejeitado, balanço adicionado ao colaborador e colaborador notificado	Passou	
15	20	Apagar um pedido	1. Clicar na linha da tabela da página "requests" 2. Na página de detalhes clicar no botão de apagar ("delete") 3. Confirmar a ação na modal	Pedido de ausência apagado e balanço adicionado ao colaborador	Pedido de ausência apagado e balanço adicionado ao colaborador	Passou	
16	21	Editar detalhes de um pedido	1. Clicar na linha da tabela da página "requests" 2. Na página de detalhes clicar no botão de editar ("edit") 3. Efetuar as ações necessárias preenchendo sempre os campos obrigatórios	Pedido de ausência editado, balanço atualizado e colaborador notificado	Pedido de ausência editado, balanço atualizado e colaborador notificado	Passou	
17 e 44	-	Não desenvolvido					

Colaborador

Tabela C.6: Plano de testes para a área Gerir Pedidos para colaboradores.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
55	32	Cancelar um pedido	<ol style="list-style-type: none"> 1. Aceder à página "requests" 2. Clicar na linha da coluna que deseja 3. Na página de detalhes clicar no botão de cancelar ("cancel") 4. Confirmar a ação na modal 	Pedido cancelado e balanço atualizado	Pedido cancelado e balanço atualizado	Passou
56	33	Editar detalhes de um pedido	<ol style="list-style-type: none"> 1. Clicar na linha da tabela da página "requests" 2. Na página de detalhes clicar no botão de editar ("edit") - Efetuar as ações necessárias preenchendo sempre os campos obrigatórios 	Pedido de ausência editado, balanço atualizado e colaborador notificado	Pedido de ausência editado, balanço atualizado e colaborador notificado	Passou

Gerir Ausências

Tabela C.7: Plano de testes para a área Gerir Ausências.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
18 e 43	22	Visualizar lista dos pedidos de ausência aprovados	<ol style="list-style-type: none"> 1. Aceder à página "requests" 2. Selecionar a aba "approved requests" 	Listagem dos pedidos de ausência aprovados	Listagem dos pedidos de ausência aprovados	Passou
19 e 43	23	Visualizar lista dos pedidos de ausência cancelados	<ol style="list-style-type: none"> 1. Aceder à página "requests" 2. Selecionar a aba "cancelled requests" 	Listagem dos pedidos de ausência cancelados	Listagem dos pedidos de ausência cancelados	Passou

Requisitar Ausências

Tabela C.8: Plano de testes para a área Requisitar Ausências.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
20	24	Requisitar um dia completo de ausência	<ol style="list-style-type: none"> 1. Aceder à página "requests" 2. Clicar no botão "request time-off" 3. Preencher os campos obrigatórios: <ul style="list-style-type: none"> - Nome do utilizador para quem é o pedido - Tipo do pedido - Dia da ausência - Tipo da ausência 	Pedido é criado automaticamente aprovado e atualiza o balanço do colaborador	Pedido é criado automaticamente aprovado e atualiza o balanço do colaborador	Passou
21	25	Requisitar um dia parcial de ausência	<ol style="list-style-type: none"> 1. Aceder à página "requests" 2. Clicar no botão "request time-off" 3. Preencher os campos obrigatórios: <ul style="list-style-type: none"> - Nome do utilizador para quem é o pedido - Tipo do pedido - Dia da ausência - Período do dia - Tipo da ausência 	Pedido é criado automaticamente aprovado e atualiza o balanço do colaborador	Pedido é criado automaticamente aprovado e atualiza o balanço do colaborador	Passou
22	26	Requisitar múltiplos dias de ausência	<ol style="list-style-type: none"> 1. Aceder à página "requests" 2. Clicar no botão "request time-off" 3. Preencher os campos obrigatórios: <ul style="list-style-type: none"> - Nome do utilizador para quem é o pedido - Tipo do pedido - Dia de início da ausência - Dia de fim da ausência - Tipo da ausência 	Pedido é criado automaticamente aprovado e atualiza o balanço do colaborador	Pedido é criado automaticamente aprovado e atualiza o balanço do colaborador	Passou

Colaborador

Tabela C.9: Plano de testes para a área Requisitar Ausências para colaboradores.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
57	34	Requisitar um dia completo de ausência	1. Aceder à página "requests" 2. Clicar no botão "request time-off" 3. Preencher os campos obrigatórios: - Tipo do pedido - Dia da ausência - Tipo da ausência	Pedido é criado e atualiza o balanço do colaborador	Pedido é criado e atualiza o balanço do colaborador	Passou
58	35	Requisitar um dia parcial de ausência	1. Aceder à página "requests" 2. Clicar no botão "request time-off" 3. Preencher os campos obrigatórios: - Tipo do pedido - Dia da ausência - Período do dia - Tipo da ausência	Pedido é criado e atualiza o balanço do colaborador	Pedido é criado e atualiza o balanço do colaborador	Passou
59	36	Requisitar múltiplos dias de ausência	1. Aceder à página "requests" 2. Clicar no botão "request time-off" 3. Preencher os campos obrigatórios: - Tipo do pedido - Dia de início da ausência - Dia de fim da ausência - Tipo da ausência	Pedido é criado e atualiza o balanço do colaborador	Pedido é criado e atualiza o balanço do colaborador	Passou

Notificar

Tabela C.10: Plano de testes para a área Notificar.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
23 e 45	27	Receber <i>email</i> com pedido de aprovação	Confirmar a existência de um <i>email</i> de aprovação de uma ausência após um pedido de ausência	Receção de um <i>email</i> com pedido de aprovação	Receção de um <i>email</i> com pedido de aprovação	Passou
24	28	Enviar <i>email</i> informativo sobre uma ausência	Após a submissão de uma ausência verificar o envio de um <i>email</i> informativo sobre a mesma	Receção de um <i>email</i> com a informação da ausência	Receção de um <i>email</i> com a informação da ausência	Passou

Colaborador

Tabela C.11: Plano de testes para a área Notificar para colaboradores.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
60	37	Receber <i>email</i> com resultado do pedido	Confirmar a existência de um <i>email</i> a notificar a atualização do estado de um pedido de ausência	Receção de um <i>email</i> com a alteração do estado do pedido	Receção de um <i>email</i> com a alteração do estado do pedido	Passou
61	38	Receber <i>email</i> informativo sobre a ausência	Confirmar a existência de um <i>email</i> após a submissão de um pedido de ausência	Receção de um <i>email</i> com informação sobre a ausência pedida	Receção de um <i>email</i> com informação sobre a ausência pedida	Passou

Gerir Tipos de Ausência

Tabela C.12: Plano de testes para a área Gerir Tipos de Ausências.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
25	-	Não desenvolvido				
26	-	Não desenvolvido				
27	-	Não desenvolvido				
28	-	Não desenvolvido				

Gerir Balanço

Tabela C.13: Plano de testes para a área Gerir Balanço.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
29 e 46	29	Visualizar balanço dos colaboradores ativos	1. Aceder à página "balance"	Listagem do balanço dos colaboradores ativos	Listagem do balanço dos colaboradores ativos	Passou
30	30	Editar balanço	1. Aceder à página "balance" 2. Selecionar um colaborador 3. Clicar no botão de editar ("edit") 4. Alterar o valor do balanço 5. Confirmar a ação	Balanço do colaborador atualizado	Balanço do colaborador atualizado	Passou
31 e 47	-	Não desenvolvido				

Gerir Eventos Especiais

Tabela C.14: Plano de testes para a área Gerir Eventos Especiais.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
32	-			Não desenvolvido		
33	-			Não desenvolvido		

Módulo de Recrutamento

Tal como para o módulo anterior as US foram agrupadas na realização dos testes. No entanto, apenas administradores e gestores têm acesso a este módulo.

Gerir Quadro de Recrutamento

Tabela C.15: Plano de testes para a área Gerir Quadro de Recrutamento.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
62 e 81	39	Ver os candidatos ativos	1. Aceder à página "hiring"	Quadro com os candidatos ativos distribuídos pela respetiva fase do processo de recrutamento	Quadro com os candidatos ativos distribuídos pela respetiva fase do processo de recrutamento	Passou
63 e 82	40	Pesquisar e aplicar filtros ao quadro de recrutamento	1. Aceder à página "hiring" 2. Clicar no botão "filter by" 3. Escolher os filtros na modal 4. Confirmar a ação	Quadro com os candidatos filtrados	Quadro com os candidatos filtrados	Passou
64	41	Adicionar um candidato	1. Aceder à página "hiring" 2. Clicar no botão "add candidate" 3. Na página de adicionar preencher os campos obrigatórios: - Nome - Estado - Perfil - Fonte de contacto	Candidato adicionado com sucesso	Candidato adicionado com sucesso	Passou
65	42	Adicionar um candidato diretamente a uma coluna	1. Aceder à página "hiring" 2. Clicar no símbolo '+' numa coluna 3. Preencher os campos obrigatórios: - Coluna - Nome - Perfil - Fonte de contacto	Candidato associado à coluna escolhida	Candidato associado à coluna escolhida	Passou
66	42	Adicionar um candidato com informação detalhada	1. Aceder à página "hiring" 2. Clicar no símbolo '+' numa coluna 3. Na modal clicar no botão "add more details" 4. Na página de adicionar preencher os campos obrigatórios: - Nome - Estado - Perfil - Fonte de contacto	Candidato associado à coluna escolhida	Candidato associado à coluna escolhida	Passou

Customizar Quadro de Recrutamento

Tabela C.16: Plano de testes para a área Customizar Quadro de Recrutamento.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
67	43	Adicionar uma coluna ao quadro	<ol style="list-style-type: none"> 1. Aceder à página "hiring" 2. Clicar botão "customize board" 3. Na modal clicar no botão "add new column" 4. Preencher o campo com o nome da coluna 5. Confirmar a ação 	Nova coluna adicionada ao quadro de recrutamento	Nova coluna adicionada ao quadro de recrutamento	Passou
68	44	Editar coluna	<ol style="list-style-type: none"> 1. Aceder à página "hiring" 2. Clicar botão "customize board" 3. Na modal editar os nomes das colunas 4. Confirmar a ação 	Nome da coluna editado com sucesso	Nome da coluna editado com sucesso	Passou
69	45	Editar coluna diretamente no quadro	<ol style="list-style-type: none"> 1. Aceder à página "hiring" 2. Duplo clique no nome de uma coluna 3. Efetuar a alteração do nome 4. Clicar na tecla ENTER para confirmar a ação 	Nome da coluna editado com sucesso	Nome da coluna editado com sucesso	Passou
70	46	Remover coluna	<ol style="list-style-type: none"> 1. Aceder à página "hiring" 2. Clicar botão "customize board" 3. Clicar no botão de apagar 	Coluna apagada com sucesso	Coluna apagada com sucesso	Passou

Lista de Candidatos

Tabela C.17: Plano de testes para a área Lista de Candidatos.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
71 e 83	-	Não desenvolvido				
72	-	Não desenvolvido				
73 e 84	47	Visualizar candidatos em lista	1. Aceder à página "sourcing list"	Listagem dos candidatos ativos	Listagem dos candidatos ativos	Passou
74	48	Arquivar um candidato via lista	1. Aceder à página "sourcing list" 2. Selecionar os candidatos marcando-os na <i>checkbox</i> 3. Clicar no botão de arquivar "archive" 4. Confirmar a ação na modal	Candidato arquivado com sucesso	Candidato arquivado com sucesso	Passou
74	49	Arquivar um candidato via detalhes	1. Aceder à vista detalhada de um candidato 2. Clicar no botão de arquivar ("archive") 3. Confirmar a ação na modal	Candidato arquivado com sucesso	Candidato arquivado com sucesso	Passou
75	50	Recuperar um candidato via lista	1. Aceder à página "sourcing list" 2. Selecionar os candidatos marcando-os na <i>checkbox</i> 3. Clicar no botão de recuperar ("recover") 4. Confirmar a ação na modal	Candidato recuperado com sucesso	Candidato recuperado com sucesso	Passou
75	51	Recuperar um candidato via detalhes	1. Aceder à vista detalhada de um candidato 2. Clicar no botão de recuperar ("recover") 3. Confirmar a ação na modal	Candidato recuperado com sucesso	Candidato recuperado com sucesso	Passou
76 e 85	52	Pesquisar e aplicar filtros à lista	1. Aceder à página "sourcing list" 2. Clicar no botão "filter by" 3. Escolher os filtros na modal 4. Confirmar a ação	Lista com os candidatos filtrados	Lista com os candidatos filtrados	Passou

Analisar Candidatos

Tabela C.18: Plano de testes para a área Analisar Candidatos.

US	ID	Cenário	Passos	Resultado Esperado	Resultado Obtido	Conclusões
77 e 86	53	Ver detalhes de um candidato pelo quadro	<ol style="list-style-type: none"> 1. Aceder à página "hiring" 2. Clicar no candidato 	Página com os detalhes do candidato escolhido	Página com os detalhes do candidato escolhido	Passou
77 e 86	54	Ver detalhes de um candidato pela lista	<ol style="list-style-type: none"> 1. Aceder à página "sourcing list" 2. Clicar no candidato 	Página com os detalhes do candidato escolhido	Página com os detalhes do candidato escolhido	Passou
78 e 87	55	Adicionar interações a um candidato	<ol style="list-style-type: none"> 1. Aceder aos detalhes de um candidato 2. Clicar na aba "interactions details" 3. Clicar no botão de "add interaction" 4. Preencher os campos obrigatórios: <ul style="list-style-type: none"> - Tipo de interação - Data da entrevista - Entrevistador(es) 	Interação adicionada com sucesso	Interação adicionada com sucesso	Passou
79	56	Editar candidato via detalhes completos	<ol style="list-style-type: none"> 1. Aceder aos detalhes de um candidato 2. Clicar no botão "edit" 3. Preencher os campos obrigatórios: <ul style="list-style-type: none"> - Nome - Estado - Perfil - Fonte de contacto 	Candidato editado com sucesso	Candidato editado com sucesso	Passou
80	57	Editar interações	<ol style="list-style-type: none"> 1. Aceder aos detalhes de um candidato 2. Selecionar a aba "interaction details" 3. Preencher os campos obrigatórios: <ul style="list-style-type: none"> - Tipo de interação - Data - Entrevistador(es) 	Interação editada com sucesso	Interação editada com sucesso	Passou

Apêndice D

Plano de Testes de Aceitação

Módulo de Ausências

Tabela D.1: Testes de aceitação para adição e edição de ausências.

US	ID	Cenário	Dados de entrada	Resultado Esperado	Resultado Obtido	Conclusões
16, 19, 20, 21, 22, 56, 57, 58 e 59	1	Identificador do utilizador inválido	null	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Passou
	2	Identificador do utilizador inválido	Identificador que não existe	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Passou
	3	Identificador do utilizador válido	Qualquer identificador presente na base de dados dos <i>users</i>	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Passou
	4	Data de início inválida	null	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Passou
	5	Data de início válida	Qualquer data válida onde o utilizador não tenha uma ausência	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Passou
	6	Data de fim inválida	null	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Passou
	7	Data de fim válida	Qualquer data válida onde o utilizador não tenha uma ausência	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Passou
	8	Estado do pedido inválido	null	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Passou
	9	Estado do pedido válido	Qualquer valor que seja: PENDING, APPROVED, CANCELLED	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Passou
	10	Data de outra ausência do utilizador	Uma data na qual o utilizador já tenha um pedido/ausência	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Resposta "Bad Request" e <i>popup</i> "sorry, the time-off was not requested, please try again"	Passou
64, 66 e 79	11	Ficheiro sem a extensão .pdf	test_file.jpg	Campo 'related files' transita para a cor vermelha com a mensagem: "The file (test_file.jpg) don't have the correct extension"	Campo 'related files' transita para a cor vermelha com a mensagem: "The file (test_file.jpg) don't have the correct extension"	Passou
	12	Ficheiro com a extensão .pdf	test_file.pdf	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Ausência adicionada/editada com sucesso e <i>popup</i> com título "time-off requested" / "time-off request edited"	Passou

Módulo de Recrutamento

Tabela D.2: Testes de aceitação para a adição e edição de candidatos.

US	ID	Cenário	Dados de entrada	Resultado Esperado	Resultado Obtido	Conclusões
64, 65, 66 e 79	13	Nome superior a 128 caracteres	Qualquer <i>string</i> superior a 128 caracteres	Campo 'name' transita para a cor vermelha com a mensagem: "name must be at most 128 characters"	Campo 'name' transita para a cor vermelha com a mensagem: "name must be at most 128 characters"	Passou
64, 65, 66 e 79	14	Nome igual a 128 caracteres	Qualquer <i>string</i> igual a 128 caracteres	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Passou
64, 65, 66 e 79	15	Nome inferior a 128 caracteres	Qualquer <i>string</i> inferior a 128 caracteres	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Passou
64, 65, 66 e 79	16	Nome inferior a 128 caracteres e caracteres especiais	J. Francisco	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Passou
64, 65, 66 e 79	17	Nome inferior a 128 caracteres e caracteres especiais	José-Francisco	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Passou
64, 65, 66 e 79	18	Nome inferior a 128 caracteres e caracteres especiais	José #Francisco	Campo 'name' transita para a cor vermelha com a mensagem: "invalid name"	Campo 'name' transita para a cor vermelha com a mensagem: "invalid name"	Passou
64, 65, 66 e 79	19	Nome inferior a 128 caracteres e caracteres especiais	José Francisco 1	Campo 'name' transita para a cor vermelha com a mensagem: "invalid name"	Campo 'name' transita para a cor vermelha com a mensagem: "invalid name"	Passou
64, 66 e 79	20	Email inválido	thisIs.MyEmail	Campo 'email' transita para a cor vermelha com a mensagem: "the email is invalid"	Campo 'email' transita para a cor vermelha com a mensagem: "the email is invalid"	Passou
64, 66 e 79	21	Email válido	francisco@email.com	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Passou
64, 66 e 79	22	Número de telefone inválido para o país correto	País: Portugal Número de telefone: 123574392	Campo 'phone number' transita para a cor vermelha com a mensagem: "invalid phone number"	Campo 'phone number' transita para a cor vermelha com a mensagem: "invalid phone number"	Passou
64, 66 e 79	23	Número de telefone válido para o país correto	País: Portugal Número de telefone: 915681422	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Candidato adicionado/editado com sucesso e <i>popup</i> com título "added/updated candidate"	Passou

Tabela D.3: Testes de aceitação para a adição e edição de candidatos (cont.).

US	ID	Cenário	Dados de entrada	Resultado Esperado	Resultado Obtido	Conclusões
64, 65, 66 e 79	24	Perfil superior a 64 caracteres	Qualquer string superior a 64 caracteres	Campo 'profile' transita para a cor vermelha com a mensagem "profile must be at most 64 characters"	Campo 'profile' transita para a cor vermelha com a mensagem "profile must be at most 64 characters"	Passou
64, 65, 66 e 79	25	Perfil igual a 64 caracteres	Qualquer string igual a 64 caracteres	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Passou
64, 65, 66 e 79	26	Perfil igual a 64 caracteres	Qualquer string inferior a 64 caracteres	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Passou
64, 66 e 79	27	Localização superior a 128 caracteres	Qualquer string superior a 128 caracteres	Campo 'location' transita para a cor vermelha com a mensagem: "location must be at most 128 characters"	Campo 'location' transita para a cor vermelha com a mensagem: "location must be at most 128 characters"	Passou
64, 66 e 79	28	Localização igual a 128 caracteres	Qualquer string igual a 128 caracteres	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Passou
64, 66 e 79	29	Localização inferior a 128 caracteres	Qualquer string inferior a 128 caracteres	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Passou
64, 66 e 79	30	Ficheiro sem a extensão .pdf	test_file.jpg	Campo 'related files' transita para a cor vermelha com a mensagem: "The file (test_file.jpg) don't have the correct extension"	Campo 'related files' transita para a cor vermelha com a mensagem: "The file (test_file.jpg) don't have the correct extension"	Passou
64, 66 e 79	31	Ficheiro com a extensão .pdf	test_file.pdf	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Passou
64, 66 e 79	32	Ficheiro com tamanho superior a 10MB	Ficheiro test_file.pdf Tamanho superior a 10MB	Campo 'related files' transita para a cor vermelha com a mensagem: "The file (test_file.pdf) is bigger than expected!"	Campo 'related files' transita para a cor vermelha com a mensagem: "The file (test_file.pdf) is bigger than expected!"	Passou
64, 66 e 79	33	Ficheiro com tamanho igual a 10MB	Ficheiro test_file.pdf Tamanho igual a 10MB	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Passou
64, 66 e 79	34	Ficheiro com tamanho inferior a 10MB	Ficheiro test_file.pdf Tamanho inferior a 10MB	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Candidato adicionado/editado com sucesso e popup com título "added/updated candidate"	Passou

Tabela D.4: Testes de aceitação para a adição e edição de interações.

US	ID	Cenário	Dados de entrada	Resultado Esperado	Resultado Obtido	Conclusões
78, 80 e 87	35	Tipo de interação superior a 128 caracteres	Qualquer <i>string</i> superior a 128 caracteres	Campo 'type of interaction' transita para a cor vermelha com a mensagem: "type of interaction must be at most 128 characters"	Campo 'type of interaction' transita para a cor vermelha com a mensagem: "type of interaction must be at most 128 characters"	Passou
	36	Tipo de interação igual a 128 caracteres	Qualquer <i>string</i> igual a 128 caracteres	Interação adicionada/editada com sucesso e <i>popup</i> com título "saved interaction"	Interação adicionada/editada com sucesso e <i>popup</i> com título "saved interaction"	Passou
	37	Tipo de interação inferior a 128 caracteres	Qualquer <i>string</i> inferior a 128 caracteres	Interação adicionada/editada com sucesso e <i>popup</i> com título "saved interaction"	Interação adicionada/editada com sucesso e <i>popup</i> com título "saved interaction"	Passou
	38	Tipo de interação inferior a 128 caracteres e números	Interação 1	Campo 'type of interaction' transita para a cor vermelha com a mensagem: "invalid type of interaction"	Campo 'type of interaction' transita para a cor vermelha com a mensagem: "invalid type of interaction"	Passou
	39	Tipo de interação inferior a 128 caracteres e caracteres especiais	Interacao&new	Campo 'type of interaction' transita para a cor vermelha com a mensagem: "invalid type of interaction"	Campo 'type of interaction' transita para a cor vermelha com a mensagem: "invalid type of interaction"	Passou
	40	Data inválida	Carater espaço ('')	Campo 'date' transita para a cor vermelha com a mensagem: "interaction date is required"	Campo 'date' transita para a cor vermelha com a mensagem: "interaction date is required"	Passou
	41	Data inválida	05/2024/07	Campo 'date' transita para a cor vermelha com a mensagem: "interaction date is required"	Campo 'date' transita para a cor vermelha com a mensagem: "interaction date is required"	Passou
42	Data válida	11/01/2023	Interação adicionada/editada com sucesso e <i>popup</i> com título "saved interaction"	Interação adicionada/editada com sucesso e <i>popup</i> com título "saved interaction"	Passou	

Tabela D.5: Testes de aceitação para a adição e edição de colunas do quadro de recrutamento.

US	ID	Cenário	Dados de entrada	Resultado Esperado	Resultado Obtido	Conclusões
67, 68 e 69	43	Nome superior a 128 caracteres	Qualquer <i>string</i> superior a 128 caracteres	Campo 'column name' transita para a cor vermelha com a mensagem: "must be at most 128 characters"	Campo 'column name' transita para a cor vermelha com a mensagem: "must be at most 128 characters"	Passou
	44	Nome igual a 128 caracteres	Qualquer <i>string</i> igual a 128 caracteres	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Passou
	45	Nome inferior a 128 caracteres	Qualquer <i>string</i> inferior a 128 caracteres	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Coluna adicionada/editada com sucesso e <i>popup</i> com título "added/updated column"	Passou
	46	Nome inferior a 128 caracteres e valores numéricos	Coluna 2	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Passou
	47	Nome inferior a 128 caracteres e caracteres especiais	#Primeira Coluna	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Campo 'column name' transita para a cor vermelha com a mensagem: "invalid column name"	Passou

