



**UNIVERSIDADE DE AVEIRO**  
**Escola Superior de Tecnologia e Gestão de Águeda**  
Licenciatura em Tecnologias da Informação

## **TeamFoundry**

Projeto Temático em Desenvolvimento Web

Nome dos autores

Gabriel Oliveira – 120364

João Nunes – 117589

Luís Assis – 112763

Luís Nantes – 120401

Pedro Sampaio - 119213

Rodrigo Ferreira - 120099

Docentes responsáveis da UC:

**Dr. Professor Fábio Marques**

**Dr. Professor Ivan Pires**

Águeda | 07 de novembro de 2025

# Agradecimentos

Gostaríamos de expressar a nossa gratidão aos nossos professores e em especial ao orientador do projeto, pelo suporte, pelo conhecimento partilhado e pelas orientações fundamentais ao longo do desenvolvimento. Um especial agradecimento às nossas famílias e amigos pelo incentivo constante e compreensão nos momentos desafiantes. Também reconhecemos o empenho e colaboração de cada membro do grupo, cujo esforço conjunto permitiu a concretização deste trabalho. Por fim, agradecemos a todos os que, direta ou indiretamente, contribuíram para o sucesso deste projeto.

# Resumo

O projeto **TeamFoundry** consiste no desenvolvimento de uma aplicação web voltada para a gestão de recrutamento no setor industrial. O principal objetivo é digitalizar e centralizar os processos de contratação, aproximando **candidatos, empresas e administradores** num único sistema, promovendo transparência, eficiência e rastreabilidade em todo o ciclo de recrutamento.

A plataforma permite que **empresas publiquem ofertas de trabalho, candidatos gerem o seu perfil profissional e administradores supervisionem o funcionamento global do sistema**. A solução responde à carência de ferramentas específicas para este setor, onde as contratações ainda são frequentemente feitas de forma informal.

O projeto seguiu o **modelo de desenvolvimento em cascata**, abrangendo fases de planeamento, modelação, prototipagem, implementação e testes. Foram utilizadas tecnologias modernas como **React, TailwindCSS e DaisyUI** no frontend, e **Spring Boot com PostgreSQL e Docker** no backend, garantindo robustez, segurança e escalabilidade.

Os testes de usabilidade confirmaram uma **interface intuitiva, acessível e consistente**, adequada aos diferentes perfis de utilizador. O sistema permite a gestão integrada de perfis, equipas, certificações e histórico de colaborações, oferecendo uma solução prática e inovadora para a gestão de recursos humanos industriais.

Em suma, o **TeamFoundry** representa um contributo relevante para a digitalização da gestão de pessoal na indústria, alinhando-se com os princípios da **Indústria 4.0** e promovendo uma cultura de recrutamento mais moderna e estruturada.

# Abstract (Tradução do Resumo em inglês)

The **TeamFoundry** project consists of developing a web application designed for **industrial recruitment management**. Its main goal is to **digitalize and centralize hiring processes**, connecting **candidates, companies, and administrators** in a unified platform that promotes transparency, efficiency, and traceability throughout the recruitment cycle.

The platform enables **companies to publish job offers**, **candidates to manage professional profiles**, and **administrators to oversee global operations**. It addresses the current lack of specialized digital tools in the industrial sector, where hiring often occurs through informal and unstructured methods.

The project followed the **waterfall development model**, comprising stages of planning, modelling, prototyping, implementation, and testing. Modern technologies were used, including **React, TailwindCSS, and DaisyUI** for the frontend, and **Spring Boot with PostgreSQL and Docker** for the backend, ensuring scalability, reliability, and security.

Usability tests confirmed an **intuitive and accessible user interface**, well adapted to different user roles. The system integrates profile management, team coordination, certification validation, and work history tracking, providing a comprehensive and efficient human resources solution for the industrial context.

In conclusion, **TeamFoundry** represents a significant contribution to the **digital transformation of industrial workforce management**, aligning with **Industry 4.0** principles and fostering a more structured and transparent recruitment culture.

**Keywords:** Industrial recruitment, Web application, Profile management, Spring Boot, React.

# Listas de siglas e abreviaturas

**Lista ordenada alfabeticamente (ordem ascendente) das abreviaturas e siglas usadas no texto, seguida das palavras ou expressões correspondentes escritas por extenso.**

**Exemplo**

**SI - Sistema de Informação**

**UML - Unified Modelling Language**

**WBS - Work Breakdown Structure**

# Índice Geral

1. Introdução.....	11
1.1 Identificação do Tema do Projeto .....	11
1.2 Enquadramento Geral do Projeto.....	11
1.3 Justificação da Importância.....	12
1.4 Objetivos do Projeto .....	12
1.5 Metodologia de Trabalho.....	12
1.6 Organização Interna do Relatório .....	13
2. Análise e Planeamento .....	15
2.1 Atividades e Tarefas Planeadas.....	15
2.1.1 Calendarização – Gantt.....	17
2.1.3 Quadro-Resumo das Atividades Planeadas.....	18
2.2 Atividades e Tarefas Executadas .....	19
2.2.1 1º Fase de Desenvolvimento .....	19
2.2.2 2º Fase de Desenvolvimento .....	20
2.3 Utilizadores e Requisitos de Utilização .....	20
2.4.2 Público-Alvo e Necessidades.....	20
2.4.3 Personas .....	21
2.5 Levantamento do Estado da Arte .....	23
2.6 Levantamento de Requisitos .....	24
2.6.1 Requisitos Funcionais .....	24
2.6.2 Requisitos Não Funcionais .....	25
3. Modelação e Design .....	27
3.1 Modelo de Casos de Utilização.....	27
3.1.1 Visão Geral .....	27
3.1.2 Atores do Sistema .....	28
3.1.3 Descrição de Casos de Utilização .....	29
3.1.4 Diagramas de Sequência .....	49
3.2 Prototipagem do Sistema .....	62
3.2.1 Protótipo de Baixa-Fidelidade .....	62
3.2.2 Protótipo de Alta-Fidelidade.....	69

3.3 Identificação das Tecnologias para Implementação.....	79
3.4 Modelo de Dados .....	79
3.4.1 Diagrama de Conceitos .....	80
3.4.2 Diagrama de Entidades e Relações .....	80
3.4.3 Esquema Físico .....	81
3.5 Distribuição e Estrutura do Site .....	81
3.5.1 Distribuição do Website.....	82
3.5.2 Estrutura Implementada (Padrão MVC) .....	83
3.5.3 Mapa de Navegação.....	84
4. Implementação da Aplicação.....	85
4.1 Camada de Apresentação .....	85
4.2 Camada de Lógica.....	85
4.3 Arquitetura e Organização da Aplicação .....	85
5. Testes e Validação .....	86
5.1 Estratégia de Testes e Validações .....	86
5.2 Testes Unitários .....	87
5.3 Testes Estruturais .....	87
5.4 Validação da Interface e Usabilidade.....	87
5.5 Verificação do Cumprimento dos Requisitos.....	88
6. Análise dos Resultados.....	89
6.1 Verificação do Cumprimento dos Requisitos Funcionais .....	89
6.2 Análise Crítica dos Resultados .....	89
7. Reflexão e Autocrítica.....	90
7.1 Atividades Desenvolvidas.....	90
7.2 Estratégias de Trabalho Adotadas.....	90
7.3 Tecnologias Utilizadas.....	90
7.4 Planeamento vs Execução.....	91
7.5 Sugestões de Melhoria .....	91
7.6 Síntese das Experiências .....	92
7.7 Perspetivas Futuras .....	92
8. Conclusão .....	93

Referências Bibliográficas.....	94
---------------------------------	----

Figura 1 - Diagrama de Gantt.....	17
Figura 2-Diagrama de Casos de Utilização .....	27
Figura 3-Diagrama de Sequência-aceder área pessoal/empresarial [Invalido] .....	50
Figura 4-Diagrama de Sequência-aceder área pessoal/empresarial [Valido] .....	50
Figura 5-Diagrama de Sequência-Administrador monta equipa para requisição .....	51
Figura 6-Diagrama de Sequência-Administrador monta equipa para requisição [Erro ao consultar a base de dados] .....	52
Figura 7-Diagrama de Sequência-Administrador monta equipa para requisição [Falha ao atualizar requisição] .....	53
Figura 8-Diagrama de Sequência-Administrador monta equipa para requisição [Funcionário indisponível].	54
Figura 9-Diagrama de Sequência-Administrador monta equipa para requisição [Nenhuma requisição disponível].....	55
Figura 10-Diagrama de Sequência-Administrador monta equipa para requisição [Nenhum funcionário corresponde aos filtros] .....	55
Figura 11-Diagrama de Sequência-Analisar empregado [opcional].....	56
Figura 12-Diagrama de Sequência-Avaliar empregador [opcional] .....	56
Figura 13-Diagrama de Sequência-Editar perfil.....	57
Figura 14-Diagrama de Sequência-Gerir requisição [Inválido].....	57
Figura 15-Diagrama de Sequência-Gerir requisições [Valido] .....	58
Figura 16-Diagrama de Sequência-Login [Invalido].....	58
Figura 17-Diagrama de Sequência-Login [Valido] .....	59
Figura 18-Diagrama de Sequência-Registrar [Invalido] .....	59
Figura 19-Diagrama de Sequência-Registrar [Válido] .....	60
Figura 20-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador .....	60
Figura 21-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador [Nenhum admin disponível].....	61
Figura 22-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador [Nenhuma requisição pendente].....	61
Figura 23-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador [Requisição já delegada a outro administrador] .....	62
Figura 24-Wireframe Homepage.....	63
Figura 25-Wireframe página de login .....	64
Figura 26-Wireframe Dados pessoais.....	64
Figura 27-Wireframe trabalhos ativos .....	65
Figura 28-Wireframe Admin escolher equipas.....	65
Figura 29-Wireframe Gestão de credenciais do Super-Admin.....	66
Figura 30-Wireframe Metricas do Super-Admin .....	67
Figura 31-Protótipo de alta-fidelidade-página de login .....	70
Figura 32-Protótipo de alta-fidelidade.....	71
Figura 33-Protótipo de alta-fidelidade-Página de dados pessoais .....	72

Figura 34-Protótipo de alta-fidelidade-Gestão de credenciais Super-Admin .....	73
Figura 35-Paleta de cores/Tipografia/Espaçamentos e bases/Grelhas .....	74
Figura 36-Componentes reutilizáveis .....	75
Figura 37-Diagrama de Conceitos .....	80
Figura 38-Diagrama de Entidades e Relações .....	81
Figura 39-Mapa de Navegação.....	84

# Índice de Tabelas

Tabela 1 - Quadro-Resumo das Atividades Planeadas .....	18
Tabela 2 - Persona Carlos Ferreira .....	21
Tabela 3 - Persona Marta Oliveira.....	22
Tabela 4 - Persona João Santos .....	22
Tabela 5 - Requisitos Não Funcionais .....	25
Tabela 6 - Atores do Sistema.....	28
Tabela 7 - Caso de Utilização #1 - Login/Validação de Credenciais .....	29
Tabela 8 - Caso de Utilização #2 - Editar Perfil.....	30
Tabela 9 - Caso de Utilização #3 - Gerir Requisições.....	31
Tabela 10 - Caso de Utilização #4 - Avaliar Empregador .....	32
Tabela 11 - Caso de Utilização #5 - Criar Administradores.....	33
Tabela 12 - Caso de Utilização #6 - Editar Funcionários Escolhidos.....	34
Tabela 13 - Caso de Utilização #7 - Visualizar Emails .....	35
Tabela 14 - Caso de Utilização #8 - Delegar Requisições.....	36
Tabela 15 - Caso de Utilização #9 - Exportar/Importar Dados de uma Requisição .....	36
Tabela 16 - Caso de Utilização #10 - Ver Estatísticas da Aplicação.....	37
Tabela 17 - Caso de Utilização #11 - Gerir Requisições .....	38
Tabela 18 - Caso de Utilização #12 - Editar Perfil .....	39
Tabela 19 - Caso de Utilização #13 - Aceder Dashboard Administrativo.....	40
Tabela 20 - Caso de Utilização #14 - Consultar Histórico de Trabalhos.....	41
Tabela 21 - Caso de Utilização #15 - Avaliar Empregados.....	42
Tabela 22 - Caso de Utilização #16 - Aceder Área Pessoal/Empresarial (Perfil).....	43
Tabela 23 - Caso de Utilização #17 - Consultar Histórico de Colaborações.....	43
Tabela 24 - Caso de Utilização #18 - Gerar Email .....	44
Tabela 25 - Caso de Utilização #19 - Ver Detalhes de uma Requisição .....	45
Tabela 26 - Caso de Utilização #20 - Listar/Filtrar Pedidos.....	46
Tabela 27 - Caso de Utilização #21 - Pesquisar/Filtrar Funcionários.....	47
Tabela 28 - Caso de Utilização #22 - Adicionar Funcionário .....	48
Tabela 29 - Caso de Utilização #23 - Gerir Administradores.....	48

# 1. Introdução

O presente relatório descreve o desenvolvimento do projeto TeamFoundry, realizado no âmbito da unidade curricular Projeto Temático em Desenvolvimento Web do curso de Licenciatura em Tecnologias da Informação da Escola Superior de Tecnologia e Gestão de Águeda (Universidade de Aveiro).

O projeto tem como objetivo criar uma aplicação web que centralize e digitalize os processos de recrutamento e gestão de trabalhadores no setor industrial, promovendo uma abordagem mais estruturada, transparente e eficiente à contratação de profissionais especializados.

Esta introdução apresenta o tema, o enquadramento e a relevância do projeto, os seus objetivos principais e a metodologia adotada pela equipa de desenvolvimento.

## 1.1 Identificação do Tema do Projeto

O projeto TeamFoundry consiste no desenvolvimento de uma plataforma web de gestão de recrutamento industrial, que conecta candidatos, empresas e administradores num ambiente unificado.

A aplicação permite que empresas publiquem ofertas de trabalho e avaliem os trabalhadores contratados, enquanto os candidatos podem gerir o seu perfil profissional, enviar candidaturas e acompanhar o seu histórico de experiências. Os administradores são responsáveis pela gestão global da plataforma, montagem de equipas e análise de métricas relacionadas ao desempenho e ao recrutamento.

O projeto insere-se na área de desenvolvimento web voltado à gestão de recursos humanos, com enfoque no contexto industrial, onde há carência de soluções tecnológicas especializadas para este tipo de processo.

## 1.2 Enquadramento Geral do Projeto

O setor industrial enfrenta atualmente grandes desafios na formalização e organização dos seus processos de recrutamento. Em muitos casos, as contratações são feitas de forma informal, recorrendo a mensagens em aplicações como o WhatsApp, o que resulta em falta de registo histórico, repetição de processos de avaliação e ausência de centralização das informações dos candidatos.

Este cenário leva a ineficiências tanto para as empresas, que perdem tempo e recursos a repetir processos, como para os trabalhadores, que frequentemente têm de refazer testes e apresentar novamente certificações já comprovadas.

O **TeamFoundry** surge como resposta a esta necessidade, propondo uma plataforma que centraliza todas as informações relevantes a currículos, certificações, histórico de trabalhos e avaliações, permitindo às empresas gerir as suas ofertas e selecionar trabalhadores com base em dados reais.

Durante a fase de conceção, a equipa estudou soluções já existentes no mercado, destacando-se a Eurofirms, que serviu como principal referência funcional, ainda que voltada a um público mais generalista. Foram também analisados elementos do LinkedIn, nomeadamente a gestão de perfis e competências, que inspiraram o desenho da área pessoal dos utilizadores.

Desta forma, o TeamFoundry combina conceitos de rede profissional e de gestão de recrutamento, adaptados à realidade industrial, promovendo eficiência, rastreabilidade e profissionalização nos processos de contratação.

## 1.3 Justificação da Importância

A relevância do projeto está diretamente ligada à necessidade de modernização **do processo de recrutamento industrial**, um setor que, apesar de tecnologicamente avançado nas áreas produtivas, mantém práticas administrativas ainda muito informais.

A falta de uma solução digital específica para este segmento dificulta a gestão de candidatos, o acompanhamento de competências e a formação de equipas qualificadas. O TeamFoundry pretende colmatar essa lacuna ao oferecer um sistema centralizado e validado, que elimina redundâncias e garante maior transparência entre empresas e trabalhadores.

O impacto esperado é triplo:

- Para as empresas, redução de tempo e custos de contratação;
- Para os trabalhadores, criação de uma identidade profissional digital permanente;
- Para os administradores, acesso a dados fiáveis que facilitam a tomada de decisão.

Além disso, o projeto alinha-se com os princípios da **Indústria 4.0**, ao promover a digitalização e automatização de processos de gestão de pessoas, reforçando o papel das tecnologias web como motor de eficiência e inovação no ambiente industrial.

## 1.4 Objetivos do Projeto

O objetivo geral do projeto é desenvolver uma aplicação web funcional e acessível que digitalize e centralize os processos de recrutamento no setor industrial.

Os objetivos específicos incluem:

- Levantar e analisar os requisitos do sistema com base nas necessidades identificadas;
- Adaptar boas práticas de plataformas de referência, como Eurofirms e LinkedIn;
- Desenvolver protótipos de baixa e alta-fidelidade para validação das interfaces e fluxos de navegação;
- Implementar as funcionalidades principais da aplicação, incluindo gestão de perfis, ofertas e equipas;
- Realizar testes de usabilidade e validação com utilizadores internos e externos;
- Documentar o processo de desenvolvimento, refletindo criticamente sobre os resultados obtidos.

A concretização destes objetivos permitirá criar uma plataforma sólida, intuitiva e tecnicamente viável, com potencial de aplicação prática no contexto industrial.

## 1.5 Metodologia de Trabalho

O desenvolvimento da TeamFoundry seguiu o **modelo em cascata**, caracterizado pela execução sequencial das suas fases: modelação, prototipagem, implementação e testes. Este método foi escolhido por oferecer uma estrutura clara, ideal para projetos académicos com prazos definidos e entregas planeadas.

A equipa foi dividida em duplas responsáveis por diferentes componentes do projeto, mantendo uma comunicação constante através de reuniões semanais e ferramentas colaborativas. Foram utilizadas:

- Notion e OneDrive para organização de ideias e ficheiros;
- Discord e WhatsApp para comunicação diária entre os membros;
- Microsoft Teams para reuniões e partilha de progresso com o orientador.

O planeamento seguiu o diagrama de Gantt, que definiu as dependências entre tarefas e a sequência temporal de execução.

A validação da aplicação será realizada em duas etapas: inicialmente por testes conduzidos pelos próprios desenvolvedores, e posteriormente por utilizadores externos ao grupo, incluindo o professor orientador, que acompanhará o progresso e fornecerá feedback técnico.

Esta abordagem garante um desenvolvimento organizado, com documentação completa e progressão linear das etapas, assegurando coerência com o modelo em cascata adotado.

## 1.6 Organização Interna do Relatório

O presente relatório encontra-se estruturado de forma lógica e sequencial, permitindo uma leitura clara e uma compreensão gradual de todas as etapas que compuseram o desenvolvimento do projeto **TeamFoundry**. A organização do documento segue uma abordagem que acompanha o ciclo de vida do sistema — desde a sua conceção inicial até à implementação e avaliação final.

O **Capítulo 1** introduz o contexto geral do projeto, apresentando a motivação, os objetivos principais e específicos, o enquadramento teórico e tecnológico, bem como a metodologia de desenvolvimento adotada. Este capítulo inclui ainda a descrição da organização do relatório, proporcionando ao leitor uma visão global da estrutura do documento.

O **Capítulo 2** descreve em detalhe a **análise do problema e os requisitos do sistema**, abrangendo os casos de uso, os tipos de utilizadores, as funcionalidades pretendidas e as restrições identificadas. São também apresentadas as necessidades do mercado e as justificações técnicas e funcionais para as decisões tomadas.

O **Capítulo 3** é dedicado ao **desenho e modelação da solução**, incluindo diagramas UML, arquitetura geral do sistema e estrutura da base de dados. Este capítulo evidencia as ligações entre os diferentes componentes do sistema e o modo como interagem para garantir a coerência e escalabilidade da aplicação.

O **Capítulo 4** aborda a **implementação da aplicação**, descrevendo as tecnologias utilizadas, a organização do código, os módulos desenvolvidos e a integração entre frontend e backend. São explicados os principais desafios técnicos enfrentados e as soluções adotadas para garantir a eficiência e a segurança do sistema.

O **Capítulo 5** apresenta a **fase de testes e validação**, onde são descritos os procedimentos de verificação, os resultados obtidos e as melhorias aplicadas após a análise dos testes funcionais e de usabilidade.

O **Capítulo 6** expõe as **conclusões e o trabalho futuro**, avaliando os objetivos atingidos, o impacto do projeto e as possíveis evoluções da aplicação em contextos reais de utilização.

Por fim, o relatório é complementado por um conjunto de **anexos**, que incluem documentação técnica adicional, excertos de código relevantes, diagramas detalhados e outros elementos de apoio à compreensão global do sistema.

## 2. Análise e Planeamento

O planeamento do projeto foi estruturado em fases sequenciais, de acordo com o guião da unidade curricular disponibilizada pelos docentes, de forma a garantir uma progressão lógica desde a análise inicial até à implementação e entrega final da aplicação.

### 2.1 Atividades e Tarefas Planeadas

As atividades planeadas foram organizadas em seis fases principais, esta divisão permite acompanhar de forma clara a evolução do trabalho, bem como identificar os pontos críticos e as dependências entre tarefas. Além da calendarização, importa destacar a importância do planeamento no projeto, pois permite controlar o tempo, alocar corretamente os recursos e gerir dependências entre atividades, e desta forma assegurar que o grupo avança de forma estruturada e eficiente.

O mapa de *Gantt* foi utilizado como ferramenta de apoio, o que nos permitiu representar graficamente a calendarização, as dependências e as metas intermédias (*milestones*).

- Fase 1: **Especificação e Planeamento:** Inclui a preparação e realização da reunião com o cliente, levantamento do estado da arte, definição de requisitos funcionais e não funcionais, bem como a identificação e seleção dos casos de utilização iniciais. Esta fase é fundamental para alinhar expectativas com o cliente e assegurar que o projeto responde às necessidades reais dos utilizadores.
- Fase 2: **Prototipagem de baixa fidelidade e análise de tecnologias:** Desenvolvimento de *wireframes*, avaliação inicial com utilizadores externos e identificação das tecnologias a considerar. Esta fase tem como objetivo validar precocemente a experiência de utilização e recolher feedback antes do investimento em implementações complexas.
- Fase 3: **Prototipagem de alta-fidelidade não funcional e desenho do sistema:** Construção do protótipo visual detalhado, guia de estilos, definição da arquitetura (MVC) e modelo de dados, incluindo testes preliminares de acessibilidade e compatibilidade. O propósito é consolidar a visão da aplicação, garantindo consistência visual e técnica, antes de avançar para a implementação.
- Fase 4: **Primeira versão funcional da aplicação:** Implementação inicial das funcionalidades selecionadas, testes unitários e melhorias resultantes da primeira iteração. O foco está em obter uma versão funcional mínima que permita validar o funcionamento do sistema.
- Fase 5: **Testes finais:** Testes de usabilidade, acessibilidade, compatibilidade e desempenho junto de utilizadores com características próximas do público-alvo. Esta fase assegura a qualidade global da aplicação e a sua adequação às necessidades identificadas.
- Fase 6: **Produção e entrega final:** Ajustes finais, documentação técnica, manual de instalação/manutenção e entrega da versão final da aplicação. Representa a conclusão formal do projeto e a disponibilização do produto ao cliente.

Em termos de dependências, as mesmas foram consideradas de forma rigorosa:

- A definição de requisitos precedeu a prototipagem.

- A avaliação do protótipo de baixa fidelidade foi concluída antes do desenvolvimento do protótipo de alta-fidelidade.
- A definição da arquitetura e modelo de dados foi realizada antes da implementação funcional.
- A versão funcional foi validada com testes antes da produção final.

Este encadeamento assegurou uma evolução progressiva e validada do projeto, posteriormente, através dos diagramas de *gantt* fica mais explícito as dependências entre as diferentes tarefas.

A distribuição temporal atribuída a cada fase foi estabelecida de acordo com a sua complexidade e esforço esperado. Posto isto, o planeamento seguido foi definido de forma a equilibrar estes pontos.

- A ordem escolhida garante que as fases de análise e prototipagem antecedem a implementação, reduzindo o risco de erros futuros.
- A duração atribuída a cada fase foi definida com base no seu grau de complexidade e esforço esperado.

Por exemplo:

- O levantamento de requisitos exigiu menos tempo por ser uma fase inicial, mas crítica.
- A fase de desenvolvimento teve maior duração, dado o esforço de implementação e testes.
- As fases de testes finais e entrega foram planeadas para consolidar e validar o produto.

É através desta abordagem que se mostra a preocupação da equipa em equilibrar o esforço disponível com os prazos estabelecidos.

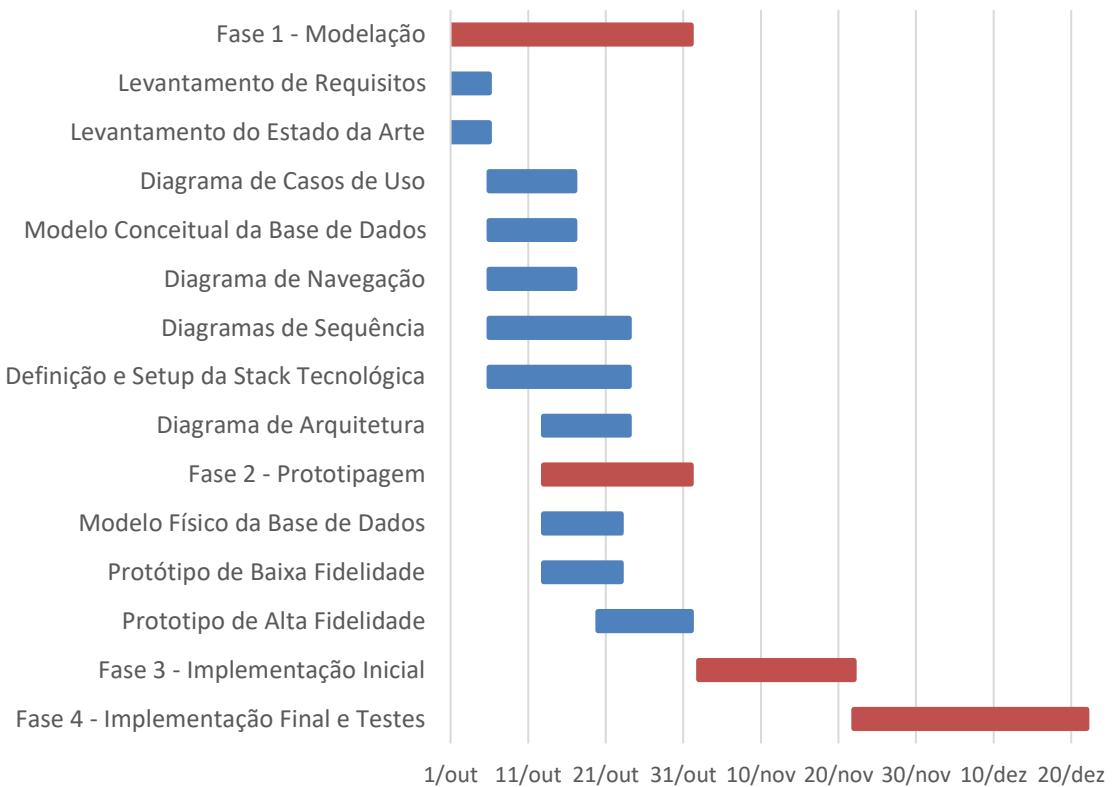
Relativamente aos recursos, a equipa foi organizada em papéis distintos para análise, design, desenvolvimento e testes, assegurando uma divisão eficiente do trabalho. Por isso, podemos dividir os recursos em 3 tipos:

- Recursos humanos:
  - Distribuição de tarefas entre membros da equipa (análise, design, desenvolvimento, testes).
  - Definição de papéis para garantir eficiência (ex.: responsável pelo front-end, responsável pela base de dados, responsável pela documentação, etc.).
- Recursos técnicos:
  - Ferramentas de design (Figma, Excalidraw).
  - IDEs de desenvolvimento e linguagens de programação.
  - Sistemas de controlo de versões (Git/GitHub).
  - Plataformas de colaboração (Teams, Notion).

- Riscos e restrições:
  - Gestão do tempo, dado o calendário académico apertado.
  - Conciliar este projeto com outras unidades curriculares.
  - Possíveis limitações técnicas na escolha de ferramentas ou na curva de aprendizagem de tecnologias.

Esta reflexão reforça que o planeamento não se limitou a datas e tarefas, mas considerou também a viabilidade prática e os recursos reais disponíveis.

### 2.1.1 Calendarização – *Gantt*



*Figura 1 - Diagrama de Gantt*

### 2.1.3 Quadro-Resumo das Atividades Planeadas

Para complementar os diagramas de *Gantt*, apresenta-se na Tabela 1 - Quadro-Resumo das Atividades Planeadas um resumo das tarefas planeadas, indicando a sua duração prevista, os recursos envolvidos, as dependências identificadas e os principais marcos (*milestones*) associados a cada fase.

Tabela 1 - Quadro-Resumo das Atividades Planeadas

Fase	Tarefa Macro	Duração Prevista	Recursos Envolvidos	Dependências	Entrega/Milestone
1	Especificação e Planeamento	26/09 – 10/10	Equipa completa, Início do projeto reuniões, documentos de requisitos		Requisitos definidos
2	Prototipagem de baixa fidelidade	11/10 – 24/10	Designer, equipa Fase 1 de análise, utilizadores externos		Protótipo validado
3	Prototipagem de alta-fidelidade e desenho do sistema	25/10 – 02/11	Equipa técnica, Fase 2 ferramentas de prototipagem		Arquitetura + Modelo de dados definidos
	Entrega 1	02/11	Equipa completa Fase 1-3		Relatório intermédio
4	Primeira versão funcional	03/11 – 27/11	Programadores, Fase 3 IDE, GitHub		Protótipo funcional
5	Testes finais	28/11 – 12/12	Utilizadores externos, equipa de testes	Fase 4	Testes concluídos
6	Produção e entrega final	13/12 – 22/12	Equipa completa, Fase 5 documentação		Entrega final

## 2.2 Atividades e Tarefas Executadas

Nesta secção devem ser descritas as **atividades que realmente foram realizadas** durante o projeto, comparando com o planeamento inicial definido no subcapítulo anterior. Aqui é importante destacar:

- Quais tarefas foram concluídas conforme planeado.
- Quais sofreram alterações (adiamentos, antecipações, ajustes).
- Eventuais dificuldades ou desvios em relação ao cronograma inicial.
- Os resultados obtidos em cada tarefa.

O objetivo é mostrar a **execução prática** do planeamento, evidenciando o progresso e as decisões tomadas durante o desenvolvimento.

### 2.2.1 1º Fase de Desenvolvimento

Durante esta primeira fase de desenvolvimento foram definidas várias atividades para serem desenvolvidas ao longo do período definido para a primeira entrega. As fases 1, 2 e 3 consistiam em fazer pesquisa e especificação, seguido de uma prototipagem de baixa-fidelidade mais análise de tecnologias e por fim seleção de tecnologias, fazer uma prototipagem de alta-fidelidade não funcionais e um desenho do sistema.

As atividades que foram realizadas dessas fases foram as seguintes-

- Definição dos Requisitos da Aplicação
- Definição da Stack de Tecnológica
- Benchmarking
- Protótipo da Interface
- Diagrama Entidade e Relações da SBD
- Diagrama Conceitual SBD
- SiteMapping
- Diagramas de Sequência
- Criação de Componentes no Figma
- Protótipo de Alta-Fidelidade no Figma
- SetUp da Stack Tecnológica
- E Realização dos 3 primeiros pontos do relatório

Os protótipos que fomos desenvolvendo ao longo desta primeira fase estão expostos nos pontos 3.2 do relatório. Ao longo deste período foram também realizadas 5 reuniões síncronas com o Professor orientador onde nos foram dadas indicações e dicas de como melhorar e progredir com o projeto. Toda a documentação produzida está também devidamente colocada no relatório.

Como resultado desta primeira fase, ficaram definidos todos os requisitos da aplicação, concluído o protótipo de alta-fidelidade, escolhida e configurada a stack tecnológica e elaborada a documentação correspondente às três primeiras fases do projeto.

## 2.2.2 2º Fase de Desenvolvimento

Nesta subsecção devem ser descritas as tarefas **da segunda fase** (até à entrega final).

Deve incluir:

- As atividades das fases 4, 5 e 6.
- Implementação da aplicação, testes finais e ajustes realizados.
- Preparação da versão final, documentação técnica e entrega ao cliente.
- Principais resultados alcançados nesta fase.

O objetivo é mostrar como o projeto evoluiu da primeira versão funcional até à **versão final entregue na segunda entrega**.

## 2.3 Utilizadores e Requisitos de Utilização

### 2.3.1 Público-Alvo e Necessidades

O público-alvo da aplicação é composto por profissionais e empresas do setor industrial, abrangendo tanto os trabalhadores que procuram oportunidades de emprego como as empresas que necessitam de recrutar pessoal qualificado para projetos temporários ou permanentes.

#### Funcionários (Candidatos)

Os funcionários representam os utilizadores que procuram oportunidades de trabalho no setor industrial. As suas principais necessidades incluem:

- Criar e manter um perfil profissional com informações pessoais, competências, certificações e histórico de experiências.
- Candidatar-se a ofertas de trabalho publicadas pelas empresas.
- Consultar o estado das suas candidaturas e os projetos em que participaram.
- Receber avaliações por parte das empresas após a conclusão de trabalhos, permitindo construir um histórico de desempenho.

Estes utilizadores valorizam a simplicidade de utilização e a clareza da informação apresentada, sendo essencial que a aplicação ofereça um ambiente intuitivo e acessível.

## Empresas

As empresas utilizadoras da plataforma têm como principal objetivo recrutar e gerir trabalhadores qualificados para diferentes funções e períodos. As suas necessidades incluem:

- Publicar ofertas de emprego com especificações detalhadas (função, requisitos, duração, local, remuneração, etc.).
- Consultar e selecionar candidatos com base nas suas competências e histórico de avaliações.
- Avaliar o desempenho dos trabalhadores após cada colaboração.
- Gerir as equipas ativas e manter um registo das contratações realizadas.

Para as empresas, a plataforma deve fornecer uma interface de gestão eficiente, que simplifique o processo de recrutamento e elimine a informalidade comum no setor.

---

## Administradores

Os administradores são responsáveis pela gestão global da aplicação. As suas necessidades incluem:

- Gerir e validar regtos de utilizadores (funcionários e empresas).
- Supervisionar as ofertas de emprego e candidaturas ativas.
- Formar equipas com base nas competências e disponibilidade dos trabalhadores.
- Acompanhar estatísticas e relatórios de utilização da plataforma.
- Garantir a integridade e segurança dos dados armazenados.

Este perfil exige uma interface de controlo clara e funcionalidades avançadas de administração, uma vez que é responsável por manter o bom funcionamento e a credibilidade da aplicação.

### 2.3.2 Personas

*Tabela 2 - Persona Carlos Ferreira*

Nome:	Carlos Ferreira
Idade:	32 anos
Função:	Funcionário
Profissão:	Eletricista industrial
Contexto:	Trabalha por conta própria e procura projetos temporários em fábricas e empresas de manutenção industrial.
Motivações:	Pretende encontrar oportunidades de trabalho de curta duração sem depender de contactos informais.

Dificuldades:	Tem pouca familiaridade com plataformas digitais complexas e perde tempo a repetir testes ou enviar documentos para cada nova empresa.
Necessidades	Ter um perfil único que centralize as suas informações e possa ser reutilizado em futuras candidaturas.

*Tabela 3 - Persona Marta Oliveira*

Nome:	Marta Oliveira
Idade:	41 anos
Função:	Representante de Empresa
Cargo:	Gestora de Recursos Humanos numa empresa de manutenção industrial
Contexto:	É responsável pela contratação de técnicos especializados para diferentes projetos.
Motivações:	Precisa encontrar rapidamente trabalhadores qualificados, com certificações válidas e boas avaliações anteriores.
Dificuldades:	Lida com grande volume de contactos informais, o que torna difícil gerir e validar candidatos.
Necessidades	Utilizar uma plataforma que centralize os perfis de candidatos e simplifique o processo de recrutamento.

*Tabela 4 - Persona João Santos*

Nome:	João Santos
Idade:	28 anos
Função:	Administrador
Cargo:	Administrador da plataforma TeamFoundry
Contexto:	Supervisiona as interações entre empresas e candidatos, assegurando a integridade dos dados e o bom funcionamento do sistema.
Motivações:	Garantir que a aplicação opera de forma eficiente, segura e transparente.

Dificuldades:	Precisa monitorizar simultaneamente múltiplas operações e validar registos de novos utilizadores.
Necessidades	Aceder a um painel de administração que permita gerir utilizadores, ofertas e equipas de forma prática e organizada.

## 2.4 Levantamento do Estado da Arte

### Eurofirms

A Eurofirms é uma plataforma de emprego com foco na ligação entre candidatos e empresas em diversos setores de atividade. Atua como uma agência de trabalho temporário digital, permitindo às empresas publicar ofertas e aos candidatos submeter candidaturas e gerir o seu perfil profissional.

Uma das suas principais vantagens é a organização clara do processo de recrutamento, oferecendo áreas distintas para candidatos e empresas, e funcionalidades como o histórico de candidaturas, alertas de novas ofertas e filtros avançados de pesquisa.

No entanto, a Eurofirms apresenta uma abordagem genérica, não adaptada às especificidades do meio industrial. A falta de funcionalidades voltadas para a validação de certificações técnicas, a gestão de equipas industriais e o acompanhamento do desempenho dos trabalhadores limita a sua aplicabilidade neste contexto.

Apesar disso, a plataforma foi uma das principais referências funcionais para o desenvolvimento da TeamFoundry, servindo como base para a estrutura de comunicação entre candidatos e empresas e para o modelo de publicação e gestão de ofertas.

---

### LinkedIn

O LinkedIn é atualmente a maior rede profissional digital do mundo, reunindo milhões de utilizadores e empresas de diferentes setores. A sua principal força está na construção de perfis pessoais detalhados, que funcionam como currículos dinâmicos, permitindo aos utilizadores exibir competências, formações, certificações e experiências anteriores.

O sistema de recomendações e validações de competências oferece credibilidade e reforça o valor profissional dos utilizadores. Além disso, o LinkedIn integra uma área de ofertas de emprego e de comunicação direta entre candidatos e recrutadores, criando um ecossistema profissional completo.

Contudo, a sua abordagem é muito ampla e genérica, não contemplando as necessidades específicas do recrutamento industrial, como a verificação de certificações obrigatórias, a comprovação de experiência em ambientes industriais e a criação automatizada de equipas técnicas.

Ainda assim, o LinkedIn inspirou a estrutura da área pessoal do utilizador na TeamFoundry, especialmente no que diz respeito à organização do perfil, histórico de experiências e exibição de competências.

## 2.5 Levantamento de Requisitos

O levantamento de requisitos tem como objetivo identificar e descrever as funcionalidades e características essenciais da aplicação TeamFoundry, com base nas necessidades dos utilizadores previamente definidas. Esta etapa é fundamental para garantir que a solução proposta responde de forma coerente às expectativas dos diferentes perfis, Funcionário, Empresa e Administrador, e que a sua implementação é viável dentro do tempo e recursos disponíveis.

Os requisitos foram definidos a partir da análise dos utilizadores, dos objetivos do projeto e das referências estudadas no levantamento do estado da arte. Foram organizados em duas categorias principais: requisitos funcionais, que definem o que o sistema deve fazer, e requisitos não funcionais, que descrevem as condições e qualidades que o sistema deve cumprir.

### 2.5.1 Requisitos Funcionais

#### **Requisitos Funcionais do Funcionário**

1. O funcionário deve poder criar uma conta e autenticar-se na plataforma.
2. O funcionário deve poder editar o seu perfil, incluindo dados pessoais, competências, certificações e experiência profissional.
3. O funcionário deve poder consultar ofertas de trabalho disponíveis.
4. O funcionário deve poder candidatar-se a uma ou mais ofertas publicadas pelas empresas.
5. O funcionário deve poder consultar o estado das suas candidaturas (pendente, aceite, recusada).
6. O funcionário deve poder visualizar o histórico de trabalhos realizados e as avaliações associadas.
7. O funcionário deve poder anexar ficheiros ou certificados relevantes ao seu perfil (ex.: curso técnico, formação em segurança, etc.).
8. O funcionário deve poder receber notificações sobre candidaturas e avaliações.

---

#### **Requisitos Funcionais da Empresa**

1. A empresa deve poder registar-se e autenticar-se na aplicação.
2. A empresa deve poder criar, editar e remover ofertas de trabalho, especificando cargo, descrição, competências exigidas e duração.

4. A empresa deve poder avaliar candidatos após o término de um trabalho.
  5. A empresa deve poder gerir equipas ativas, adicionando ou removendo trabalhadores conforme necessário.
  6. A empresa deve poder receber notificações sobre candidaturas e avaliações pendentes.
- 

## Requisitos Funcionais do Administrador

1. O administrador deve poder gerir contas de utilizadores, aprovando ou removendo funcionários e empresas.
2. O administrador deve poder consultar e moderar ofertas de trabalho publicadas pelas empresas.
3. O administrador deve poder aceder a relatórios e estatísticas sobre a utilização da plataforma (número de candidaturas, empresas registadas, avaliações, etc.).  
  
O administrador deve poder consultar candidatos disponíveis e os seus perfis completos.  
  
O administrador deve poder consultar o histórico de contratações e as avaliações anteriores.
4. O administrador deve poder criar equipas de trabalho com base nas competências e disponibilidade dos funcionários.
5. O administrador deve poder gerir notificações e comunicações internas do sistema.
6. O administrador deve poder editar configurações gerais da plataforma, como categorias de trabalho, permissões e definições de segurança.

### 2.5.2 Requisitos Não Funcionais

*Tabela 5 - Requisitos Não Funcionais*

#### Usabilidade

- A interface deve ser intuitiva e de fácil navegação, mesmo para utilizadores com pouca experiência digital.
- O design deve ser consistente e responsivo, adaptando-se a diferentes tamanhos de ecrã (computador e tablet).
- Devem ser utilizados rótulos claros e linguagem acessível em todos os botões, menus e formulários.

#### Desempenho

- A aplicação deve garantir um tempo de resposta rápido nas principais operações (inferior a 3 segundos em condições normais de rede).
- O carregamento das páginas e submissão de formulários deve ser fluido e sem bloqueios visíveis.
- O sistema deve suportar simultaneamente pelo menos 10 utilizadores ativos sem comprometer o desempenho (escala adequada para ambiente de testes académico).

## Segurança

- As palavras-passe dos utilizadores devem ser armazenadas de forma encriptada.
- O acesso às funcionalidades deve ser controlado por autenticação e gestão de permissões.
- Os dados pessoais dos utilizadores devem ser protegidos contra acesso não autorizado.
- A aplicação deve incluir validação de formulários para evitar injeções de dados incorretos.

## Manutenibilidade

- O código-fonte deve ser organizado segundo boas práticas de programação, facilitando a manutenção futura.
- Deve ser utilizada uma estrutura modular (ex.: padrão MVC) que permita alterações sem comprometer outras partes do sistema.
- A documentação técnica deve ser atualizada ao longo do desenvolvimento.

## Fiabilidade e Testabilidade

- O sistema deve ser testado antes da entrega, incluindo testes unitários e funcionais básicos.
- A aplicação deve comportar-se de forma previsível, evitando falhas durante operações comuns.
- O sistema deve ser capaz de recuperar de pequenos erros de entrada (ex.: campos em branco, formatos incorretos).

## Compatibilidade

- A aplicação deve ser compatível com os principais navegadores (Google Chrome, Microsoft Edge e Mozilla Firefox).
- Deve ser possível aceder à aplicação a partir de diferentes sistemas operativos (Windows, macOS, Linux).

## 3. Modelação e Design

### 3.1 Modelo de Casos de Utilização

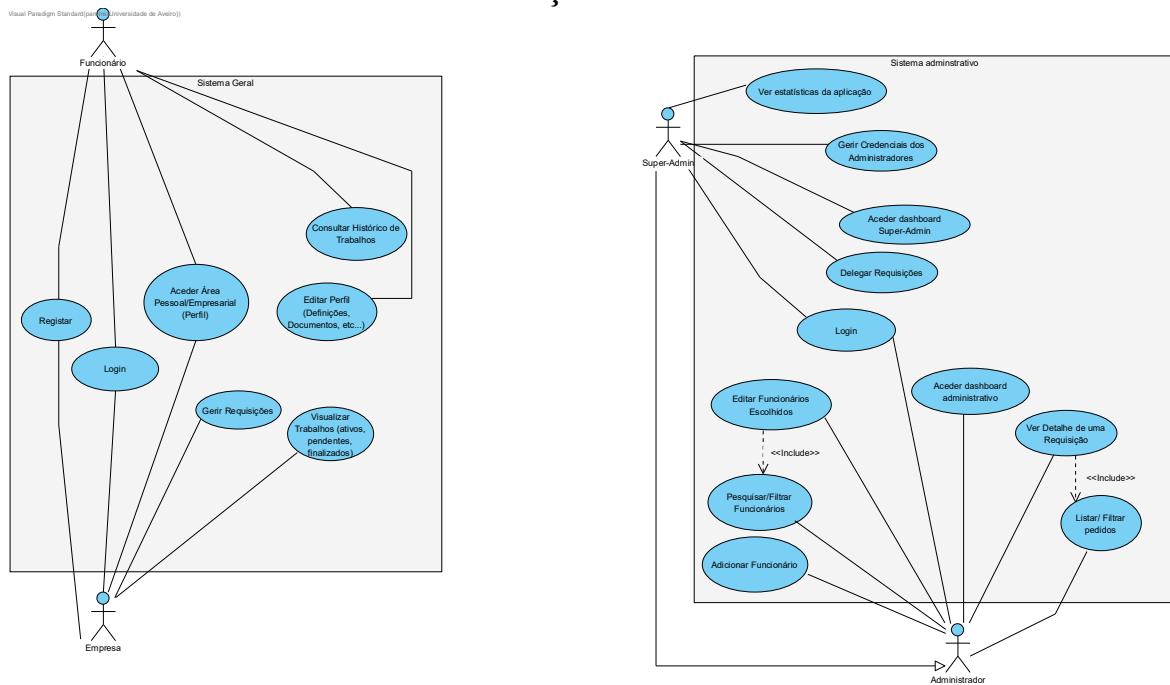


Figura 2-Diagrama de Casos de Utilização

#### 3.1.1 Visão Geral

O modelo de casos de Utilização do sistema TeamFoundry tem como objetivo mostrar, de forma clara e resumida, como os diferentes utilizadores interagem com as principais funcionalidades da aplicação. Esta representação ajuda a compreender rapidamente o papel de cada tipo de utilizador e como o sistema foi pensado para responder às suas necessidades. Além disso, serve de base para o desenvolvimento técnico e para o design das várias partes da aplicação.

O sistema envolve três perfis principais de utilizadores, que são o funcionário, a empresa e o administrador. Além destes, existe ainda o super-administrador e, de forma mais pontual, o ator RH, que lida com o sistema através da importação e exportação de dados.

O funcionário usa a aplicação para acompanhar o seu percurso profissional. É na aplicação que pode atualizar o perfil, candidatar-se a trabalhos e ver o que já fez no passado. Também consegue gerir as suas requisições e deixar a sua opinião sobre as empresas com quem trabalhou. No fundo, esta parte da plataforma serve como um espaço pessoal onde o funcionário tem tudo reunido e pode organizar o seu percurso de forma simples e prática.

A empresa usa a plataforma para tratar das suas necessidades de recrutamento e acompanhar os trabalhos que tem em curso. Pode criar novas requisições quando precisa de contratar alguém e consultar os projetos que já realizou. Depois de cada colaboração, também pode deixar uma avaliação sobre os trabalhadores. Portanto, é

uma forma simples de gerir tudo o que envolve contratações e manter o histórico de colaborações sempre à mão.

O administrador tem um papel mais abrangente, sendo responsável pela gestão da aplicação. É quem supervisiona as atividades dos funcionários e das empresas, pode adicionar e editar funcionários, pesquisar utilizadores, gerir pedidos e aceder ao dashboard administrativo, que oferece uma visão global do sistema. Além disso, o administrador pode delegar tarefas e assegurar que a informação se mantém coerente e atualizada.

Acima do administrador encontra-se o super-administrador, que possui permissões mais avançadas. Este utilizador pode criar e gerir administradores, consultar estatísticas gerais da aplicação, aceder ao dashboard principal e realizar operações de controlo mais complexas, como a gestão de permissões e requisições. A sua função assegura uma hierarquia clara e uma administração mais segura do sistema.

Por fim, o RH interage com o sistema apenas para importar ou exportar dados de requisições, isto garante a integração com outros sistemas externos de gestão de recursos humanos.

De forma transversal, a aplicação inclui também um módulo de comunicação por email, que facilita a troca de mensagens entre empresas, funcionários e administradores, promovendo uma comunicação mais rápida e eficiente.

O diagrama de casos de utilização que acompanha esta secção representa graficamente estas interações, mostra a estrutura geral do sistema e a ligação entre os atores e as suas funcionalidades. Esta visão de conjunto serve de ponto de partida para as fases seguintes de modelação, design e implementação da aplicação.

### 3.1.2 Atores do Sistema

Os atores do sistema representam os diferentes tipos de utilizadores e entidades que interagem com a aplicação TeamFoundry, desempenham papéis consoante as suas responsabilidades e objetivos dentro da plataforma.

*Tabela 6 - Atores do Sistema*

Ator	Descrição
Funcionário	O funcionário usa a aplicação para gerir o seu perfil e procurar trabalho na área industrial. Pode criar uma conta, fazer login, atualizar os seus dados e adicionar certificações. Também pode ver os trabalhos que já fez e deixar a sua opinião sobre as empresas com quem colaborou. Esta plataforma serve para o funcionário ter tudo o que precisa num só sítio e acompanhar o seu percurso profissional de forma prática.
Empresa	A empresa usa a aplicação para publicar ofertas e gerir contratações. Pode criar requisições, ver perfis de

	candidatos e avaliar os funcionários depois de cada projeto. Tem uma área própria onde acompanha colaborações e guarda o histórico das contratações. Assim, o processo de recrutamento fica mais simples, rápido e transparente para todos.
Administrador	O administrador é o utilizador encarregado de supervisionar o funcionamento geral da aplicação. As suas funções incluem gerir contas de utilizadores, validar registo, moderar requisições e candidaturas, e aceder ao dashboard administrativo. Este ator assegura a integridade dos dados e o cumprimento das regras de utilização do sistema
Super-Administrador	O super-administrador possui privilégios avançados e é responsável pela gestão hierárquica da aplicação. Pode criar e gerir contas de administradores, aceder a estatísticas globais, delegar tarefas e supervisionar a utilização da plataforma a um nível organizacional. Este papel garante a manutenção e controlo centralizado da infraestrutura administrativa do sistema
RH	O ator RH representa entidades externas ou sistemas que interagem com o sistema administrativo para importar e exportar dados de requisições. A sua função é assegurar a integração e compatibilidade entre o TeamFoundry e outros sistemas de gestão de recursos humanos

### 3.1.3 Descrição de Casos de Utilização

#### 3.1.3.1 Caso de Utilização #1 – Login/Validação de Credenciais

*Tabela 7 - Caso de Utilização #1 - Login/Validação de Credenciais*

Nome:	Login/Validação de Credenciais
Atores:	Funcionário, Empresa, Administrador, Super-administrador

Prioridade (1/3):	1
Finalidade:	Garantir que só quem tem conta consiga entrar na aplicação e manter os dados protegidos.
Requisitos Funcionais:	O sistema tem de confirmar se o nome e a palavra-passe estão corretos. Se estiverem, o utilizador entra, se não, aparece uma mensagem a dizer que algo não está bem.
Pré-condições:	O utilizador tem de ter uma conta criada e o sistema tem de estar ligado à base de dados.
Sumário:	O utilizador entra na página de login, mete o nome e a palavra-passe e tenta aceder. Se os dados estiverem corretos, entra sem problema, se estiverem incorretos, aparece uma mensagem a dizer que algo está errado.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
O utilizador acede à página de login.	O sistema apresenta o formulário de autenticação.
O utilizador insere nome de utilizador e palavra-passe.	O sistema verifica as credenciais na base de dados.
O utilizador clica em “Entrar”.	O sistema concede acesso ou exibe mensagem de erro.
<b>Sequências alternativas</b>	
<b>Se as credenciais forem incorretas:</b>	O sistema exibe uma mensagem de erro.

### 3.1.3.2 Caso de Utilização #2 - Editar Perfil

*Tabela 8 - Caso de Utilização #2 - Editar Perfil*

Nome:	Editar Perfil
Atores:	Funcionário, Empresa
Prioridade (1/3):	2
Finalidade:	Deixar o utilizador atualizar as suas informações pessoais ou da empresa, para que os dados fiquem sempre certos e atuais.

Requisitos Funcionais:	O sistema deve deixar o utilizador mudar campos como o nome, o contacto, as competências e as certificações (caso seja funcionário), ou os dados da empresa (caso seja empresa). Depois das alterações, o sistema verifica se está tudo bem e guarda as novas informações na base de dados.
Pré-condições:	O utilizador tem de estar com a conta ligada e ter permissão para editar o seu perfil.
Sumário:	O utilizador entra na área do perfil, faz as alterações que quer e confirma. O sistema verifica os dados e guarda as mudanças.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
O utilizador acede à área “Perfil”.	O sistema apresenta as informações atuais.
O utilizador altera os dados pretendidos.	O sistema valida os novos dados introduzidos.
O utilizador confirma as alterações.	O sistema atualiza o perfil na base de dados e confirma a operação.
<b>Sequências alternativas</b>	
<b>Se algum campo obrigatório estiver vazio:</b>	O sistema exibe a mensagem de erro e impede a atualização.
<b>Se ocorrer falha de ligação à base de dados:</b>	O sistema exibe uma notificação de erro técnico.

### 3.1.3.3 Caso de Utilização #3 - Gerir Requisições

*Tabela 9 - Caso de Utilização #3 - Gerir Requisições*

<b>Nome:</b>	<b>Gerir Requisições</b>
Atores:	Empresa, Administrador
Prioridade (1/3):	1
Finalidade:	Permitir que as empresas e administradores criem, consultem, editem e removam requisições de trabalho no sistema.
Requisitos Funcionais:	O sistema deve disponibilizar formulários para criação e edição de requisições, associar informações como função, duração, requisitos e estado. Deve também permitir filtrar e listar requisições.
Pré-condições:	O ator deve estar autenticado com o perfil de empresa ou administrador.

Sumário:	O utilizador acede à secção de requisições, onde pode criar novas ofertas ou gerir as existentes. O sistema armazena as alterações e disponibiliza as requisições aos candidatos.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
O utilizador seleciona “Gerir Requisições”.	O sistema apresenta a lista de requisições existentes.
O utilizador cria ou edita uma requisição.	O sistema valida os dados introduzidos.
O utilizador confirma a operação.	O sistema atualiza a base de dados e apresenta a confirmação.
<b>Sequências alternativas</b>	
<b>Se os dados forem inválidos:</b>	O sistema informa o utilizador e solicita correção.
<b>Se o utilizador tentar eliminar uma requisição ativa:</b>	O sistema solicita confirmação antes de remoção.

### 3.1.3.4 Caso de Utilização #4 - Avaliar Empregador

*Tabela 10 - Caso de Utilização #4 - Avaliar Empregador*

<b>Nome:</b>	<b>Avaliar Empregador</b>
Atores:	Funcionário
Prioridade (1/3):	2
Finalidade:	Permitir que o funcionário avalie a empresa depois de terminar um trabalho, para que ajuda a manter um registo justo e transparente sobre o desempenho das entidades empregadoras.
Requisitos Funcionais:	O sistema deve deixar o funcionário dar uma classificação à empresa onde trabalhou e, se quiser, escrever um comentário. A avaliação fica guardada no histórico do utilizador e pode ser consultada pelos administradores.
Pré-condições:	O funcionário tem de estar com a conta ligada e ter concluído uma colaboração.
Sumário:	Depois de acabar um trabalho, o funcionário vai à secção “Histórico de Trabalhos” e escolhe a opção “Avaliar Empregador”. Preenche os campos e envia a avaliação, em que fica registada no sistema.
<b>Sequência típica dos eventos</b>	

Ações dos atores	Respostas do sistema
O funcionário acede ao seu histórico de trabalhos.	O sistema mostra a lista de colaborações anteriores.
O funcionário seleciona uma empresa e escolhe avaliar.	O sistema apresenta o formulário de avaliação.
O funcionário preenche a classificação e comentário e confirma.	O sistema regista a avaliação e confirma o envio.
<b>Sequências alternativas</b>	
<b>Se o funcionário tentar avaliar uma colaboração ainda ativa:</b>	O sistema mostra uma mensagem a informar que a avaliação só é possível após o fim do trabalho.
<b>Se faltar algum campo obrigatório:</b>	O sistema avisa o utilizador e pede que complete a informação antes de submeter.

### 3.1.3.5 Caso de Utilização #5 - Criar Administradores

*Tabela 11 - Caso de Utilização #5 - Criar Administradores*

Nome:	Criar Administradores
Atores:	Super-Administrador
Prioridade (1/3):	1
Finalidade:	Permitir que o super-administrador crie novas contas de administradores para apoiar a gestão da plataforma.
Requisitos Funcionais:	O sistema deve permitir introduzir dados como nome, email, e permissões do novo administrador. Após a criação, o novo utilizador recebe acesso à área administrativa.
Pré-condições:	O super-administrador tem de estar autenticado no sistema.
Sumário:	O super-administrador acede ao painel principal, entra na área “Gerir Administradores” e cria uma nova conta, onde define as permissões e informações de acesso.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
O super-administrador entra na área “Gerir Administradores”.	O sistema apresenta a lista de administradores existentes e a opção de criar novo.
O super-administrador preenche o formulário de criação.	O sistema valida os dados inseridos.

O super-administrador confirma a operação.	O sistema cria a conta e confirma a adição do novo administrador.
<b>Sequências alternativas</b>	
<b>Se o email já existir:</b>	O sistema mostra uma mensagem de erro e não cria o utilizador.
<b>Se faltar informação obrigatória:</b>	O sistema alerta o super-administrador e impede o registo até os campos estarem completos.

### 3.1.3.6 Caso de Utilização #6 - Editar Funcionários Escolhidos

*Tabela 12 - Caso de Utilização #6 - Editar Funcionários Escolhidos*

<b>Nome:</b>	<b>Editar Funcionários Escolhidos</b>
Atores:	Administrador
Prioridade (1/3):	2
Finalidade:	Permitir ao administrador editar dados dos funcionários registados, para garantir que as informações estão corretas e atualizadas.
Requisitos Funcionais:	O sistema deve permitir alterar dados pessoais, competências ou estado de disponibilidade do funcionário. Todas as alterações devem ficar registadas no histórico do sistema.
Pré-condições:	O administrador tem de estar autenticado e possuir permissões de gestão de utilizadores.
Sumário:	O administrador acede à lista de funcionários, seleciona um e faz as alterações necessárias no perfil. No fim, guarda as modificações.
<b>Sequência típica dos eventos</b>	
<b>Ações dos atores</b>	
O administrador entra na secção “Funcionários”.	O sistema apresenta a lista de funcionários.
O administrador escolhe um funcionário e clica em “Editar”.	O sistema mostra o formulário com os dados atuais.
O administrador altera os dados e confirma.	O sistema atualiza as informações na base de dados e confirma a operação.
<b>Sequências alternativas</b>	
<b>Se o funcionário não existir mais:</b>	O sistema informa que o registo foi removido.
<b>Se houver erro de validação:</b>	O sistema mostra uma mensagem de erro e não grava as alterações.

### 3.1.3.7 Caso de Utilização #7 - Visualizar Emails

*Tabela 13 - Caso de Utilização #7 - Visualizar Emails*

<b>Nome:</b>		<b>Visualizar Emails</b>
Atores:		Funcionário, Empresa, Administrador
Prioridade (1/3):		2
Finalidade:		Permitir que os utilizadores accedam às mensagens trocadas dentro da plataforma, facilita a comunicação entre funcionários, empresas e administradores.
Requisitos Funcionais:		O sistema deve apresentar uma lista de emails recebidos e enviados, permite a leitura do conteúdo e a pesquisa por remetente, assunto ou data.
Pré-condições:		O utilizador deve estar autenticado e ter mensagens associadas à sua conta.
Sumário:		O utilizador entra na secção “Emails”, onde pode ver as mensagens trocadas, abrir uma conversa e ler o seu conteúdo.
<b>Sequência típica dos eventos</b>		
<b>Ações dos atores</b>		<b>Respostas do sistema</b>
O utilizador acede à área de emails.		O sistema mostra a lista de mensagens organizadas cronologicamente.
O utilizador seleciona uma mensagem.		O sistema exibe o conteúdo completo da mensagem.
O utilizador pode arquivar ou eliminar a mensagem.		O sistema atualiza o estado da mensagem conforme a ação.
<b>Sequências alternativas</b>		
<b>Se o utilizador não tiver emails:</b>	O sistema mostra uma mensagem informativa.	
<b>Se a ligação ao servidor falhar:</b>	O sistema avisa que não foi possível mostrar os emails.	

### 3.1.3.8 Caso de Utilização #8 - Delegar Requisições

Tabela 14 - Caso de Utilização #8 - Delegar Requisições

<b>Nome:</b>		<b>Delegar Requisições</b>
Atores:		Super-administrador
Prioridade (1/3):		1
Finalidade:		Permitir que o super-administrador atribua requisições específicas a administradores, e distribuir responsabilidades dentro da equipa de gestão.
Requisitos Funcionais:		O sistema deve permitir selecionar uma requisição e associá-la a um administrador disponível, e guarda essa atribuição na base de dados.
Pré-condições:		O super-administrador deve estar autenticado e devem existirem administradores registados no sistema.
Sumário:		O super-administrador entra na secção “Requisições”, escolhe uma e designa um administrador responsável pela sua gestão.
<b>Sequência típica dos eventos</b>		
<b>Ações dos atores</b>		<b>Respostas do sistema</b>
O super-administrador acede à lista de requisições		O sistema apresenta todas as requisições pendentes.
O super-administrador seleciona uma requisição e um administrador.		O sistema valida a seleção.
O super-administrador confirma a atribuição.		O sistema atualiza a base de dados e confirma a delegação.
<b>Sequências alternativas</b>		
<b>Se não houver administradores disponíveis:</b>		O sistema mostra uma mensagem a informar que a requisição não pode ser delegada.
<b>Se o super-administrador cancelar a operação:</b>		O sistema ignora a ação e volta à lista principal.

### 3.1.3.9 Caso de Utilização #9 - Exportar/Importar Dados de uma Requisição

Tabela 15 - Caso de Utilização #9 - Exportar/Importar Dados de uma Requisição

<b>Nome:</b>		<b>Exportar/Importar Dados de uma Requisição</b>
Atores:		Administrador, RH
Prioridade (1/3):		3

Finalidade:	Permitir a troca de informação entre o TeamFoundry e sistemas externos de gestão de recursos humanos, através da exportação ou importação de dados de requisições.
Requisitos Funcionais:	O sistema deve permitir exportar dados de requisições em formatos padrão e importar novas requisições provenientes de ficheiros externos. A operação deve ser feita apenas pelo administrador, enquanto o RH atua como fonte ou destino dos dados.
Pré-condições:	O administrador tem de estar autenticado no sistema. O RH atua como entidade externa, sem necessidade de autenticação.
Sumário:	O administrador entra na área “Gestão de Dados”, escolhe a opção de importar ou exportar informações de requisições e o sistema realiza a operação, permitindo a comunicação com o sistema de RH.
<b>Sequência típica dos eventos</b>	
<b>Ações dos atores</b>	
O administrador seleciona “Importar / Exportar Dados”.	O sistema apresenta as opções disponíveis (importar ou exportar).
O administrador escolhe o tipo de operação e o formato do ficheiro.	O sistema valida o formato e inicia o processo.
O sistema comunica com o sistema de RH externo.	Os dados são trocados (importados ou exportados) e o sistema confirma o sucesso da operação.
<b>Sequências alternativas</b>	
<b>Se o ficheiro fornecido pelo RH estiver inválido:</b>	O sistema mostra um aviso e cancela a importação.
<b>Se ocorrer erro na exportação:</b>	O sistema informa o administrador e sugere repetir a operação.

### 3.1.3.10 Caso de Utilização #10 - Ver Estatísticas da Aplicação

*Tabela 16 - Caso de Utilização #10 - Ver Estatísticas da Aplicação*

<b>Nome:</b>	<b>Ver Estatísticas da Aplicação</b>
Atores:	Super Administrador
Prioridade (1/3):	3

Finalidade:	Permitir que o super administrador visualize dados e métricas gerais sobre a utilização da plataforma, como número de utilizadores registados, requisições criadas e avaliações submetidas.
Requisitos Funcionais:	O sistema deve apresentar um painel com estatísticas atualizadas em tempo real, utilizando dados da base de dados da aplicação. Deve também permitir filtrar informação por período ou tipo de utilizador.
Pré-condições:	O super administrador tem de estar autenticado no sistema.
Sumário:	O super administrador entra na área “Estatísticas”, onde pode consultar métricas e relatórios sobre a utilização da aplicação.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
O super administrador acede ao menu “Estatísticas”.	O sistema apresenta o painel de estatísticas.
O super administrador seleciona filtros.	O sistema atualiza os dados no gráfico.
O super administrador analisa as métricas e sai da secção.	O sistema mantém as estatísticas atualizadas e regista a consulta.
<b>Sequências alternativas</b>	
<b>Se a base de dados estiver temporariamente inacessível:</b>	O sistema informa que não é possível carregar as estatísticas.

### 3.1.3.11 Caso de Utilização #11 - Gerir Requisições

*Tabela 17 - Caso de Utilização #11 - Gerir Requisições*

<b>Nome:</b>	<b>Gerir Requisições</b>
Atores:	Empresa, Administrador
Prioridade (1/3):	1
Finalidade:	Permitir criar, editar e eliminar requisições de trabalho, de modo a que as empresas possam recrutar profissionais e os administradores possam acompanhar o processo.
Requisitos Funcionais:	O sistema deve permitir preencher um formulário de criação de requisições com informação como cargo, duração, local e requisitos. Deve também permitir a edição e remoção das mesmas.

Pré-condições:	O utilizador tem de estar autenticado com perfil de empresa ou administrador.
Sumário:	O utilizador entra na área “Requisições”, onde pode criar novas ofertas, atualizar as existentes ou removê-las.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
O utilizador seleciona “Gerir Requisições”.	O sistema mostra a lista de requisições existentes.
O utilizador escolhe criar, editar ou eliminar uma requisição.	O sistema apresenta o formulário correspondente.
O utilizador confirma ação.	O sistema guarda as alterações na base de dados e confirma a operação.
<b>Sequências alternativas</b>	
<b>Se o utilizador tentar eliminar uma requisição ativa:</b>	O sistema pede confirmação antes de remover.
<b>Se faltar informação obrigatória:</b>	O sistema bloqueia o envio e mostra uma mensagem de erro.

### 3.1.3.12 Caso de Utilização #12 - Editar Perfil

*Tabela 18 - Caso de Utilização #12 - Editar Perfil*

Nome:	Editar Perfil
Atores:	Funcionário, Empresa
Prioridade (1/3):	2
Finalidade:	Permitir que o utilizador atualize os seus dados pessoais ou empresariais, garantindo que o perfil se mantém correto e completo.
Requisitos Funcionais:	O sistema deve permitir alterar dados como nome, email, contacto, competências e certificações. As alterações devem ser guardadas automaticamente após validação.
Pré-condições:	O utilizador tem de estar autenticado e ter um perfil existente no sistema.
Sumário:	O utilizador entra na área “Editar perfil”, faz as alterações desejadas e guarda as mudanças.

Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador acede ao perfil.	O sistema mostra os dados atuais.
O utilizador altera as informações.	O sistema valida as alterações.
O utilizador confirma a edição.	O sistema atualiza os dados e mostra mensagem de sucesso.
Sequências alternativas	
<b>Se algum campo estiver vazio ou incorreto:</b>	O sistema mostra um aviso e não permite guardar.
<b>Se ocorrer um erro de ligação:</b>	O sistema avisa o utilizador e pede para tentar mais tarde.

### 3.1.3.13 Caso de Utilização #13 - Aceder Dashboard Administrativo

*Tabela 19 - Caso de Utilização #13 - Aceder Dashboard Administrativo*

<b>Nome:</b>	Aceder Dashboard Administrativo
Atores:	Administrador, Super Administrador
Prioridade (1/3):	1
Finalidade:	Permitir que os administradores visualizem uma visão geral do sistema, com informações sobre requisições, funcionários e empresas ativas.
Requisitos Funcionais:	O sistema deve apresentar um painel com estatísticas e atalhos para funcionalidades principais.
Pré-condições:	O utilizador tem de estar autenticado com perfil administrativo.
Sumário:	O utilizador acede ao painel administrativo, onde encontra um resumo das atividades e atalhos para gestão.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador faz login e entra na área administrativa.	O sistema carrega o painel principal.
O utilizador consulta as informações e métricas apresentadas.	O sistema atualiza os dados automaticamente.
O utilizador escolhe navegar para uma funcionalidade específica.	O sistema redireciona para a secção selecionada.

Sequências alternativas	
<b>Se o utilizador não tiver permissões:</b>	O sistema mostra uma mensagem de acesso negado.
<b>Se os dados não estiverem disponíveis:</b>	O sistema mostra uma mensagem informativa e sugere atualizar a página.

### 3.1.3.14 Caso de Utilização #14 - Consultar Histórico de Trabalhos

*Tabela 20 - Caso de Utilização #14 - Consultar Histórico de Trabalhos*

<b>Nome:</b> Consultar Histórico de Trabalhos	
Atores:	Funcionário
Prioridade (1/3):	2
Finalidade:	Permitir que o funcionário visualize os trabalhos realizados anteriormente e as avaliações recebidas, para que possa acompanhar o seu histórico profissional.
Requisitos Funcionais:	O sistema deve apresentar uma lista de trabalhos concluídos, com detalhes como empresa, função, duração e classificação. Deve ainda permitir aceder à avaliação associada a cada trabalho.
Pré-condições:	O funcionário tem de estar autenticado no sistema e ter colaborações finalizadas.
Sumário:	O funcionário entra na área “Histórico de Trabalhos”, onde pode consultar as experiências anteriores e visualizar as avaliações atribuídas pelas empresas.
Sequência típica dos eventos	
Ações dos atores	
O funcionário acede à secção “Histórico de Trabalhos”.	O sistema mostra a lista de colaborações concluídas.
O funcionário seleciona um trabalho da lista.	O sistema exibe os detalhes da colaboração e respetiva avaliação.
Sequências alternativas	
<b>Se o funcionário ainda não tiver colaborações:</b>	O sistema mostra a mensagem “Ainda não existem trabalhos concluídos”.

### 3.1.3.15 Caso de Utilização #15 - Avaliar Empregados

*Tabela 21 - Caso de Utilização #15 - Avaliar Empregados*

<b>Nome:</b>		<b>Avaliar Empregados</b>
Atores:		Empresa
Prioridade (1/3):		2
Finalidade:		Permitir que a empresa avalie os funcionários após o término de uma colaboração, criar um registo transparente do desempenho dos trabalhadores.
Requisitos Funcionais:		O sistema deve permitir à empresa atribuir uma classificação e um comentário opcional a cada funcionário que tenha colaborado num projeto. As avaliações devem ficar associadas ao perfil do funcionário.
Pré-condições:		A empresa tem de estar autenticada e a colaboração com o funcionário deve estar concluída.
Sumário:		Após a conclusão de um trabalho, a empresa acede ao histórico de colaborações e preenche uma avaliação para cada funcionário.
<b>Sequência típica dos eventos</b>		
<b>Ações dos atores</b>		<b>Respostas do sistema</b>
A empresa acede ao histórico de colaborações.		O sistema mostra a lista de funcionários que participaram em trabalhos anteriores.
A empresa seleciona um funcionário e escolhe “Avaliar”.		O sistema apresenta o formulário de avaliação.
A empresa insere a avaliação e confirma.		O sistema regista a avaliação e atualiza o perfil do funcionário.
<b>Sequências alternativas</b>		
<b>Se a colaboração ainda estiver ativa:</b>	O sistema mostra uma mensagem a indicar que a avaliação só é possível após a conclusão do trabalho.	
<b>Se faltar informação obrigatória:</b>	O sistema avisa e impede o envio até a avaliação estar completa.	

### 3.1.3.16 Caso de Utilização #16 - Aceder Área Pessoal/Empresarial (Perfil)

Tabela 22 - Caso de Utilização #16 - Aceder Área Pessoal/Empresarial (Perfil)

<b>Nome:</b>		<b>Aceder Área Pessoal/Empresarial (Perfil)</b>
Atores:		Funcionário, Empresa
Prioridade (1/3):		1
Finalidade:		Permitir que o utilizador, seja funcionário ou empresa, entre na sua área privada para gerir dados, ver atividades e fazer as ações que precisa.
Requisitos Funcionais:		O sistema deve reconhecer o tipo de utilizador e levá-lo automaticamente para a área certa. Pessoal se for funcionário, ou empresarial, se for empresa.
Pré-condições:		O utilizador tem de ter conta criada e estar com a sessão iniciada.
Sumário:		Depois de fazer login, o utilizador é levado para a sua área pessoal ou empresarial, onde pode gerir as informações e aceder às opções disponíveis.
<b>Sequência típica dos eventos</b>		
<b>Ações dos atores</b>		<b>Respostas do sistema</b>
O utilizador faz login.		O sistema identifica o tipo de utilizador.
O sistema direciona para a área correspondente.		O sistema carrega as informações personalizadas.
<b>Sequências alternativas</b>		
<b>Se o tipo de utilizador não for reconhecido:</b>	O sistema mostra uma mensagem de erro.	

### 3.1.3.17 Caso de Utilização #17 - Consultar Histórico de Colaborações

Tabela 23 - Caso de Utilização #17 - Consultar Histórico de Colaborações

<b>Nome:</b>		<b>Consultar Histórico de Colaborações</b>
Atores:		Empresa
Prioridade (1/3):		2

Finalidade:	Permitir que a empresa visualize as colaborações anteriores realizadas com funcionários, consulta informações sobre os projetos e o desempenho de cada trabalhador.
Requisitos Funcionais:	O sistema deve apresentar a lista de colaborações anteriores com detalhes como nome do funcionário, data, função e avaliação.
Pré-condições:	A empresa tem de estar autenticada e possuir histórico de colaborações registadas.
Sumário:	A empresa entra na área “Histórico de Colaborações” e consulta as informações dos funcionários que já integraram equipas anteriores.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
A empresa entra na secção “Histórico de Colaborações”.	O sistema apresenta a lista de trabalhos realizados.
A empresa seleciona uma colaboração.	O sistema mostra os detalhes da colaboração e o desempenho do funcionário.
<b>Sequências alternativas</b>	
<b>Se não existirem colaborações anteriores:</b>	O sistema mostra uma mensagem como “Sem colaborações registadas”.
<b>Se o histórico não estiver disponível:</b>	O sistema informa o utilizador que houve um erro de carregamento.

### 3.1.3.18 Caso de Utilização #18 - Gerar Email

*Tabela 24 - Caso de Utilização #18 - Gerar Email*

Nome:	Gerar Email
Atores:	Funcionário, Empresa, Administrador
Prioridade (1/3):	2
Finalidade:	Permitir que os utilizadores falem entre si dentro da plataforma, para que a comunicação entre funcionários, empresas e administradores seja mais fácil e rápida.

Requisitos Funcionais:	O sistema deve deixar o utilizador escrever uma mensagem, escolher para quem quer enviar e depois mandá-la. As mensagens ficam guardadas e aparecem na caixa de saída de quem as enviou.
Pré-condições:	O utilizador tem de ter a conta ligada e poder enviar mensagens.
Sumário:	O utilizador vai à secção “Emails”, escreve uma nova mensagem e envia para o outro utilizador da plataforma.
<b>Sequência típica dos eventos</b>	
Ações dos atores	Respostas do sistema
O utilizador escolhe “Novo email”.	O sistema abre o formulário de envio.
O utilizador escreve o assunto, a mensagem e seleciona o destinatário.	O sistema valida os campos e prepara o envio.
O utilizador clica em “Enviar”.	O sistema guarda o email na base de dados e confirma o envio.
<b>Sequências alternativas</b>	
<b>Se faltar o destinatário:</b>	O sistema avisa o utilizador e bloqueia o envio.

### 3.1.3.19 Caso de Utilização #19 - Ver Detalhes de uma Requisição

*Tabela 25 - Caso de Utilização #19 - Ver Detalhes de uma Requisição*

<b>Nome:</b>	<b>Ver Detalhes de uma Requisição</b>
Atores:	Administrador
Prioridade (1/3):	2
Finalidade:	Permitir que o administrador visualize toda a informação detalhada sobre uma requisição específica, isto inclui descrição, requisitos, duração, estado e candidatos associados.
Requisitos Funcionais:	O sistema deve apresentar os dados completos da requisição selecionada, permite aceder também às candidaturas associadas e às ações de gestão, como editar ou remover.
Pré-condições:	O administrador deve estar autenticado e ter acesso à área de requisições.
Sumário:	O administrador entra na lista de requisições e escolhe visualizar uma em detalhe para consultar as informações associadas.

<b>Sequência típica dos eventos</b>	
<b>Ações dos atores</b>	<b>Respostas do sistema</b>
O administrador acede à secção “Requisições”.	O sistema apresenta a lista de requisições registadas.
O administrador seleciona uma requisição.	O sistema exibe todos os detalhes da requisição escolhida.
O administrador consulta as informações e, se necessário, regressa à lista.	O sistema mantém o histórico de navegação.
<b>Sequências alternativas</b>	
<b>Se a requisição tiver sido removida:</b>	O sistema mostra mensagem de aviso e atualiza a lista.

### 3.1.3.20 Caso de Utilização #20 - Listar / Filtrar Pedidos

*Tabela 26 - Caso de Utilização #20 - Listar/Filtrar Pedidos*

<b>Nome:</b>	<b>Listar / Filtrar Pedidos</b>
<b>Atores:</b>	Administrador
<b>Prioridade (1/3):</b>	2
<b>Finalidade:</b>	Permitir ao administrador visualizar e filtrar os pedidos submetidos pelas empresas, para facilitar o acompanhamento e gestão das requisições.
<b>Requisitos Funcionais:</b>	O sistema deve permitir aplicar filtros como estado, empresa, ou data de emissão. Deve ainda possibilitar ordenar a lista por critérios específicos.
<b>Pré-condições:</b>	O administrador deve estar autenticado no painel administrativo.
<b>Sumário:</b>	O administrador entra na área de pedidos e utiliza as opções de filtro para encontrar rapidamente as requisições pretendidas.
<b>Sequência típica dos eventos</b>	
<b>Ações dos atores</b>	<b>Respostas do sistema</b>
O administrador acede à secção “Pedidos”.	O sistema mostra todos os pedidos existentes.
O administrador aplica filtros.	O sistema atualiza a lista conforme os filtros aplicados.
O administrador seleciona um pedido para análise.	O sistema exibe as informações detalhadas desse pedido.
<b>Sequências alternativas</b>	

<b>Se não existirem resultados para os filtros escolhidos:</b>	O sistema mostra uma mensagem como “Nenhum pedido encontrado”.
<b>Se o administrador limpar os filtros:</b>	O sistema volta a mostrar a lista completa.

### 3.1.3.21 Caso de Utilização #21 - Pesquisar / Filtrar Funcionários

*Tabela 27 - Caso de Utilização #21 - Pesquisar/Filtrar Funcionários*

<b>Pesquisar / Filtrar Funcionários</b>	
Atores:	Administrador
Prioridade (1/3):	1
Finalidade:	Permitir que o administrador encontre rapidamente funcionários com base em critérios específicos, como nome, função, disponibilidade ou certificações..
Requisitos Funcionais:	O sistema deve incluir um campo de pesquisa e filtros avançados, onde devolve resultados em tempo real e permitindo aceder diretamente ao perfil do funcionário.
Pré-condições:	O administrador tem de estar autenticado e ter acesso à lista de funcionários.
Sumário:	O administrador acede à secção “Funcionários” e utiliza a pesquisa para localizar perfis específicos.

#### Sequência típica dos eventos

<b>Ações dos atores</b>	<b>Respostas do sistema</b>
O administrador entra na lista de funcionários.	O sistema mostra todos os perfis registados.
O administrador usa o campo de pesquisa ou aplica filtros.	O sistema atualiza a lista com os resultados correspondentes.
O administrador seleciona um funcionário para ver detalhes.	O sistema abre o perfil do funcionário escolhido.

#### Sequências alternativas

<b>Se nenhum funcionário corresponder à pesquisa:</b>	O sistema mostra algo como “Nenhum resultado encontrado”.
<b>Se o administrador remover os filtros:</b>	A lista volta a apresentar todos os funcionários.

### 3.1.3.22 Caso de Utilização #22 - Adicionar Funcionário

*Tabela 28 - Caso de Utilização #22 - Adicionar Funcionário*

<b>Nome:</b>	<b>Adicionar Funcionário</b>
Atores:	Administrador
Prioridade (1/3):	1
Finalidade:	Permitir que o administrador registe novos funcionários no sistema, garantindo que a base de dados de utilizadores se mantém atualizada.
Requisitos Funcionais:	O sistema deve permitir inserir informações básicas do funcionário e validar automaticamente os campos obrigatórios.
Pré-condições:	O administrador tem de estar autenticado e ter permissões de gestão de utilizadores.
Sumário:	O administrador entra na área “Funcionários” e escolhe a opção “Adicionar”, e preenche o formulário com os dados necessários.
<b>Sequência típica dos eventos</b>	
<b>Ações dos atores</b>	
O administrador seleciona “Adicionar Funcionário”.	O sistema mostra o formulário de registo.
O administrador preenche os campos e confirma.	O sistema valida os dados introduzidos.
O administrador conclui o registo.	O sistema adiciona o novo funcionário à base de dados.
<b>Sequências alternativas</b>	
<b>Se algum campo obrigatório estiver em falta:</b>	O sistema pede para completar antes de continuar.
<b>Se o email já existir:</b>	O sistema informa que o utilizador já está registado.

### 3.1.3.23 Caso de Utilização #23 - Gerir Administradores

*Tabela 29 - Caso de Utilização #23 - Gerir Administradores*

<b>Nome:</b>	<b>Gerir Administradores</b>
Atores:	Super Administrador
Prioridade (1/3):	1

Finalidade:	Permitir ao super administrador visualizar, editar ou remover administradores existentes na plataforma, para assegurar a gestão centralizada de permissões.
Requisitos Funcionais:	O sistema deve listar todos os administradores registados e permitir ações como editar dados, redefinir passwords e eliminar contas.
Pré-condições:	O super administrador tem de estar autenticado com permissões totais.
Sumário:	O super administrador entra na área “Gestão de Administradores”, onde pode ver a lista de contas e efetuar alterações conforme necessário.
<b>Sequência típica dos eventos</b>	
<b>Ações dos atores</b>	<b>Respostas do sistema</b>
O super administrador entra na área de gestão.	O sistema mostra a lista de administradores atuais.
O super administrador seleciona um administrador e escolhe editar ou eliminar.	O sistema apresenta o formulário correspondente.
O super administrador confirma a ação.	O sistema atualiza ou remove o registo da base de dados.
<b>Sequências alternativas</b>	
<b>Se o super administrador tentar eliminar um administrador:</b>	O sistema mostra um aviso para confirmar a sua decisão.

### 3.1.4 Diagramas de Sequência

Os diagramas de sequência representam a interação temporal entre os diferentes atores e os componentes internos do sistema, mostrando passo a passo como as funcionalidades são executadas. Servem para completar os casos de utilização, ajudando a perceber a dinâmica real da comunicação entre os utilizadores, interfaces e serviços da aplicação. Estes diagramas mostram o fluxo de mensagens ao longo do tempo, e evidenciam como cada caso de utilização é concretizado dentro do sistema.

Cada diagrama identifica os atores envolvidos, as entidades internas e a ordem das mensagens trocadas.

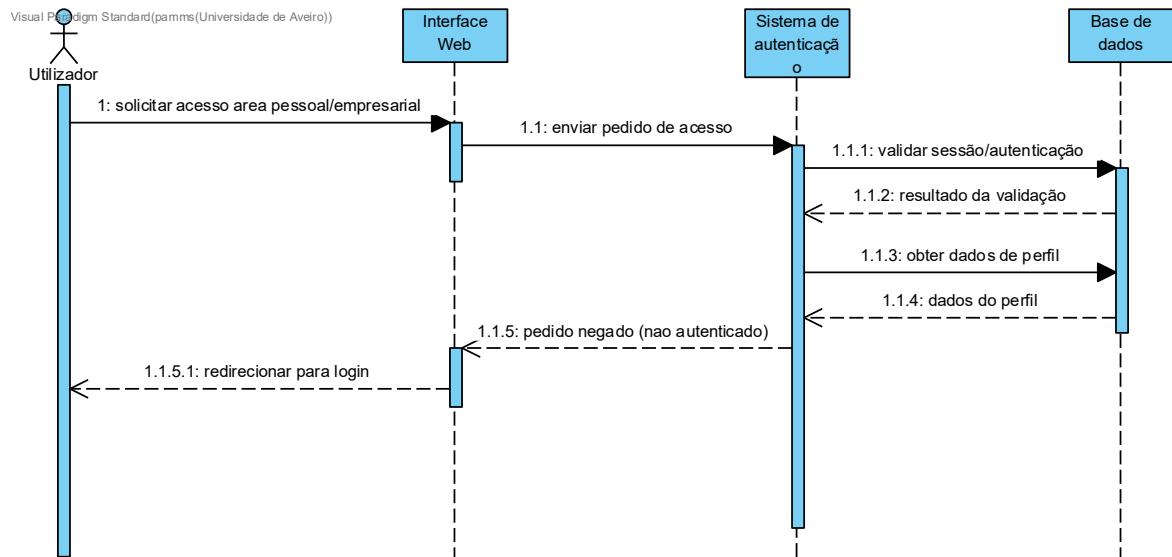


Figura 3-Diagrama de Sequência-aceder área pessoal/empresarial [Invalido]

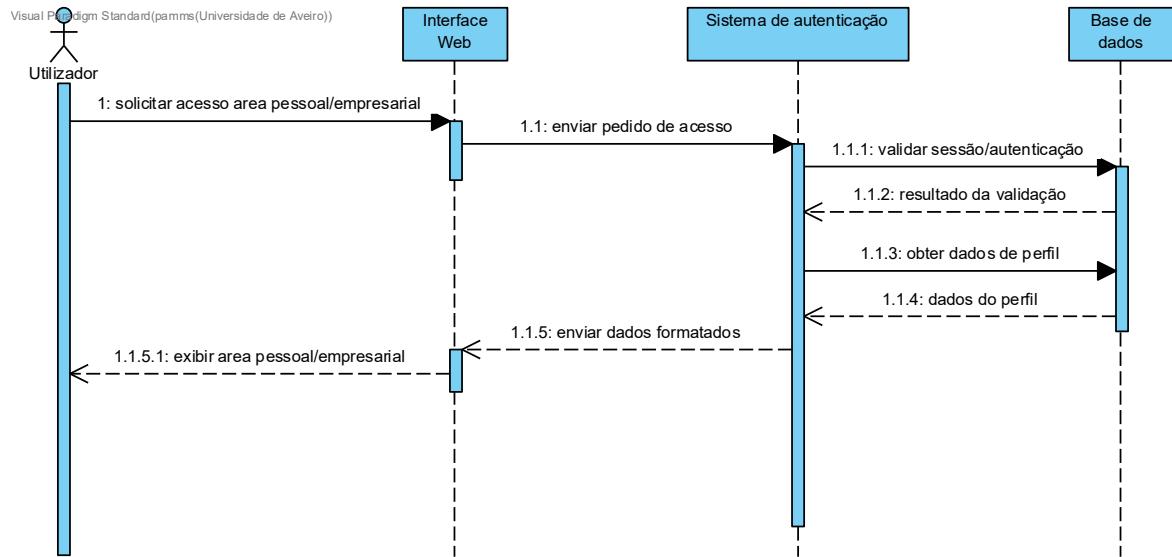


Figura 4-Diagrama de Sequência-aceder área pessoal/empresarial [Valido]

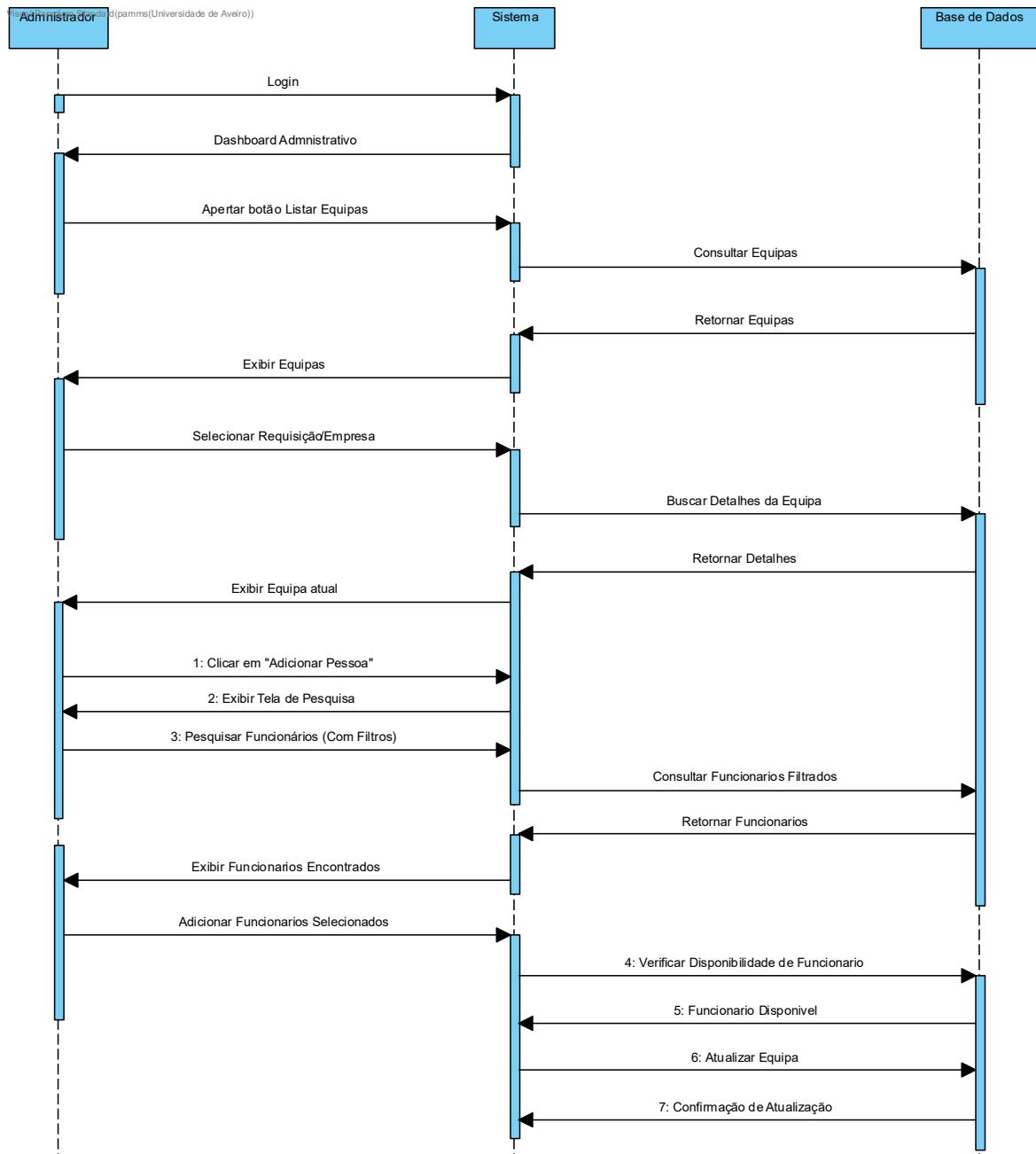


Figura 5-Diagrama de Sequência-Administrador monta equipa para requisição

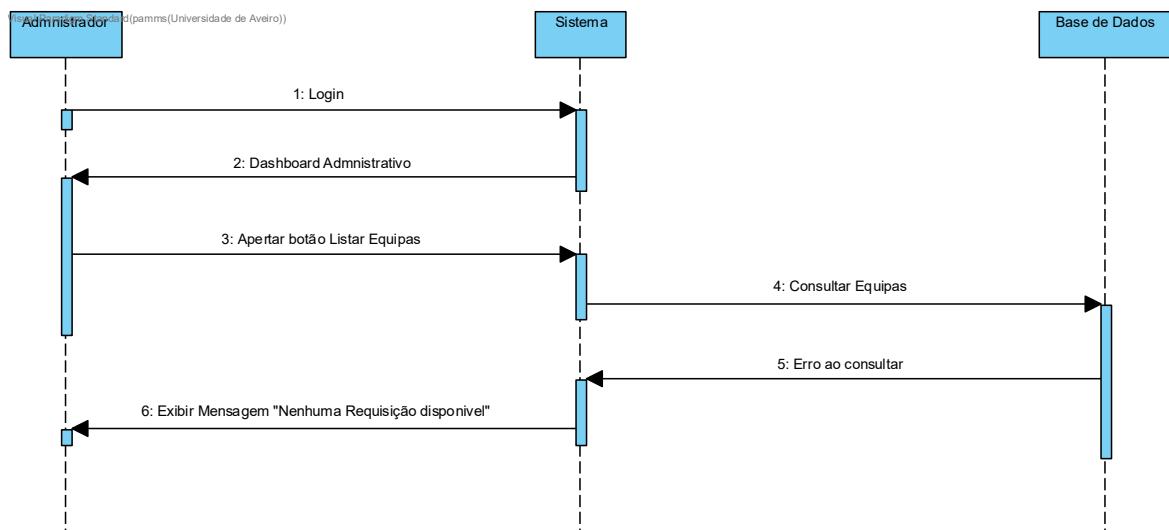


Figura 6-Diagrama de Sequência-Administrador monta equipa para requisição [Erro ao consultar a base de dados]

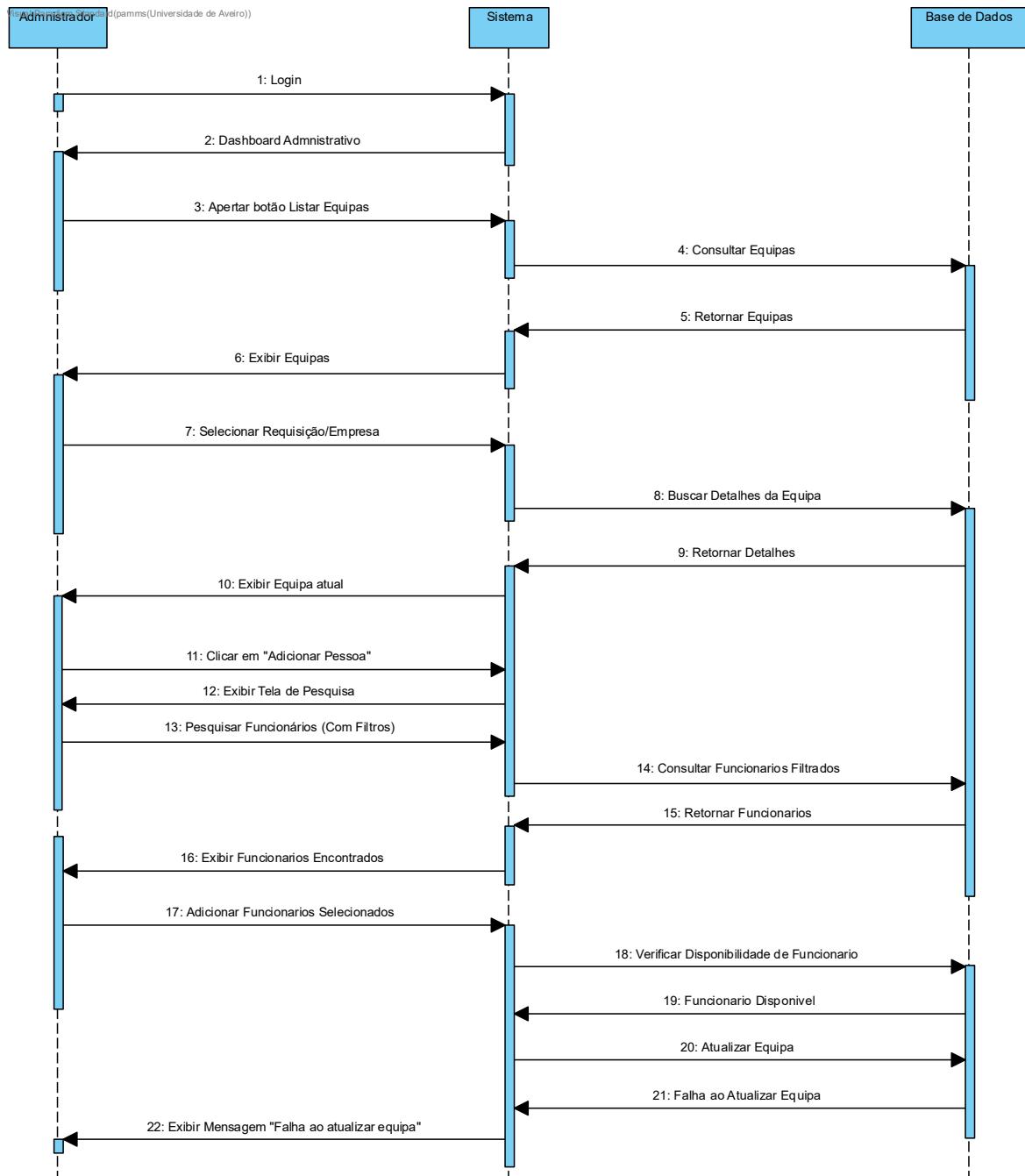


Figura 7-Diagrama de Sequência-Administrador monta equipa para requisição [Falha ao atualizar requisição]

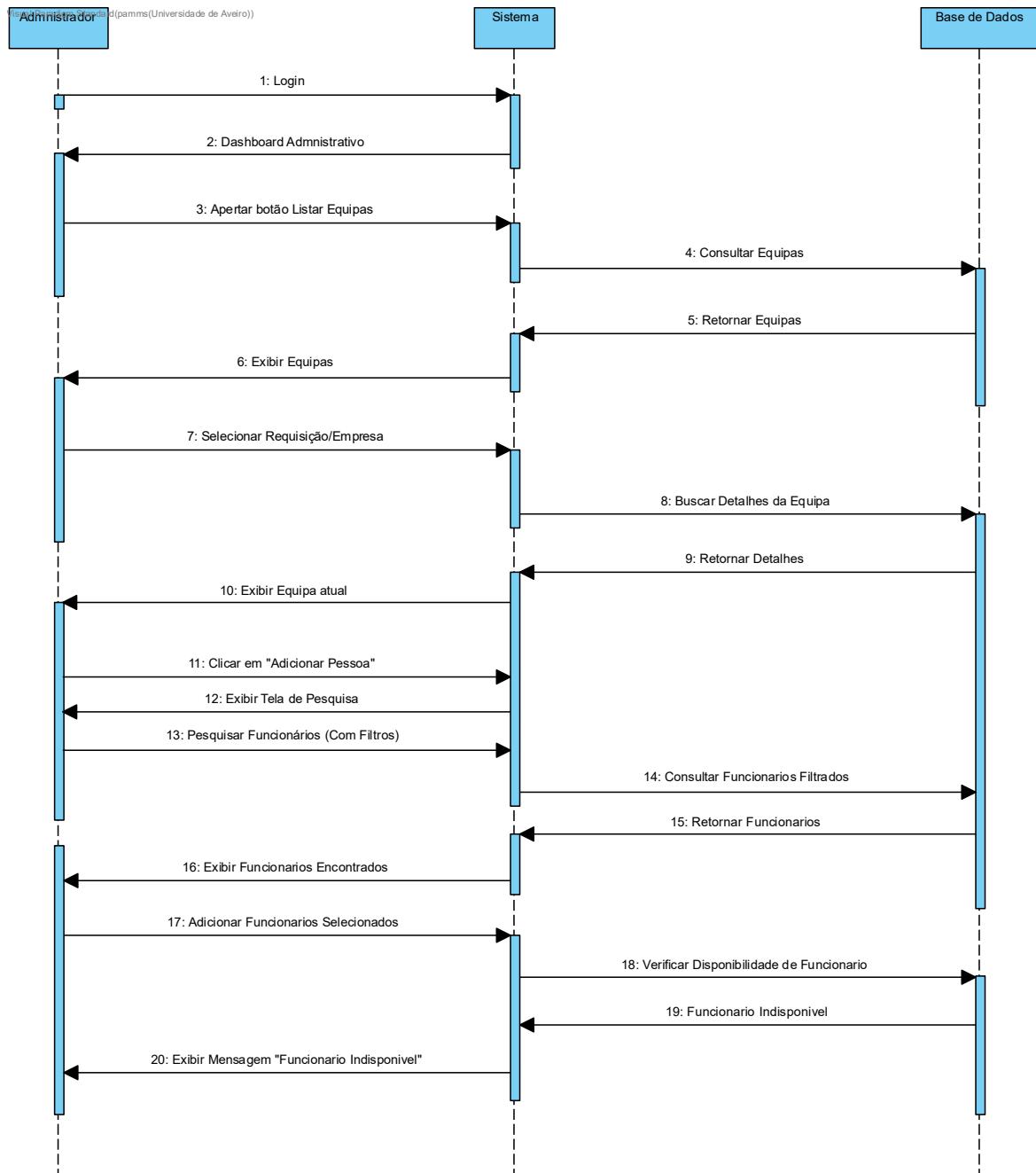


Figura 8-Diagrama de Sequência-Administrador monta equipa para requisição [Funcionário indisponível]

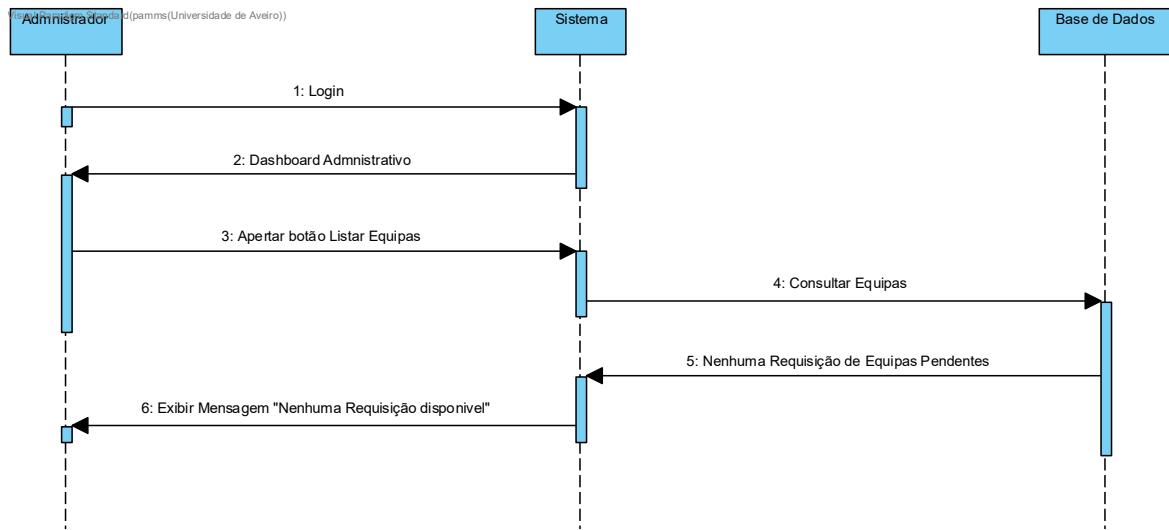


Figura 9-Diagrama de Sequência-Administrador monta equipa para requisição [Nenhuma requisição disponível]

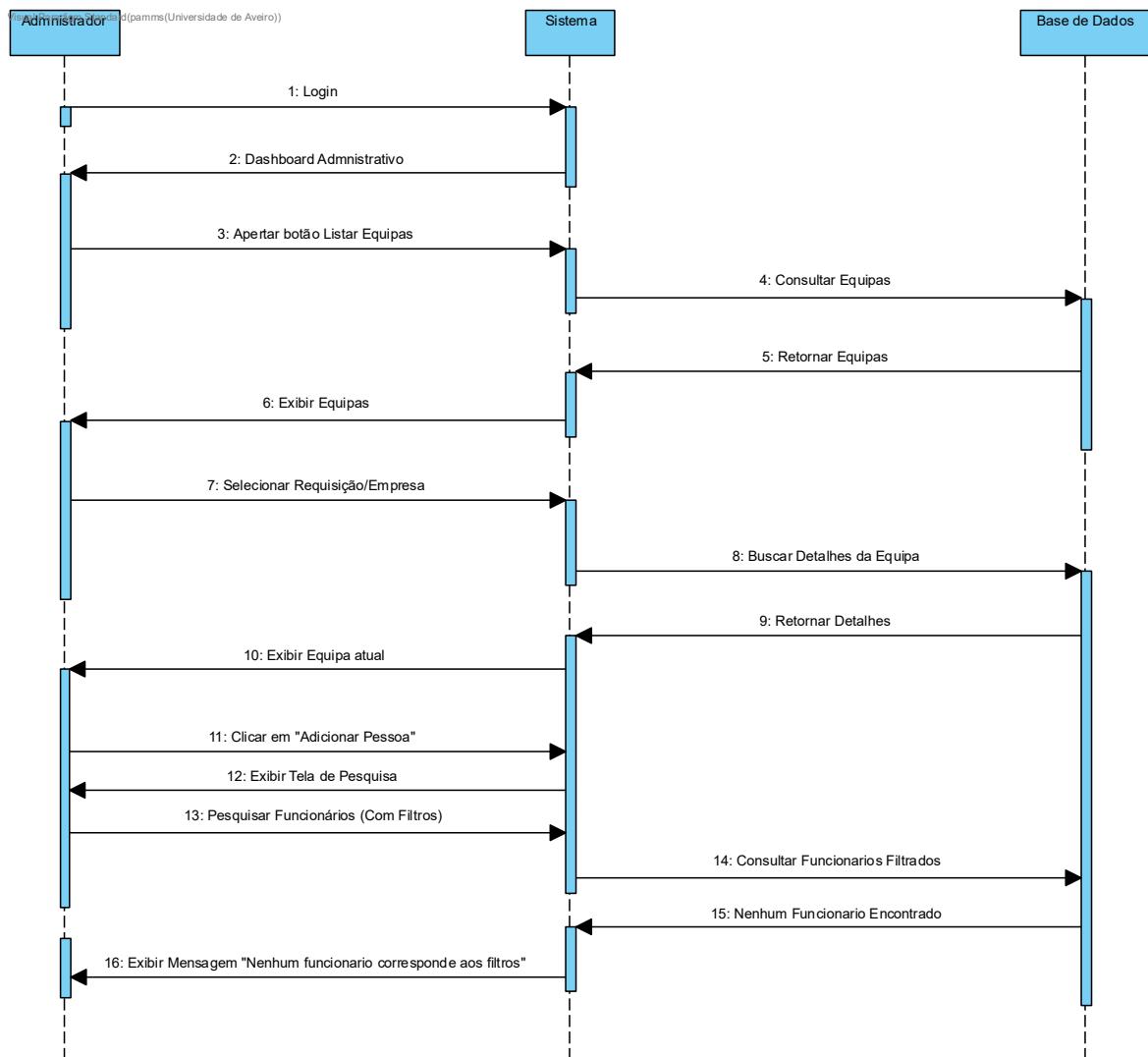


Figura 10-Diagrama de Sequência-Administrador monta equipa para requisição [Nenhum funcionário corresponde aos filtros]

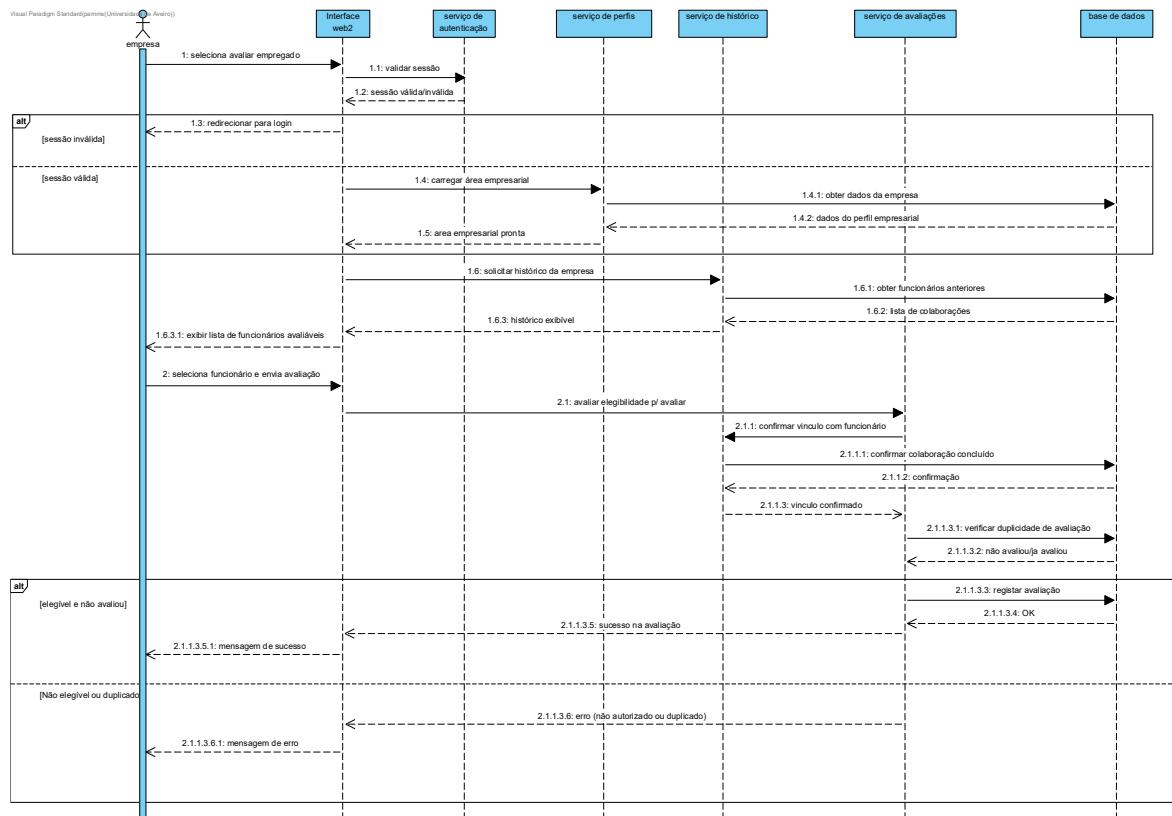


Figura 11-Diagrama de Sequência-Analisar empregado [opcional]

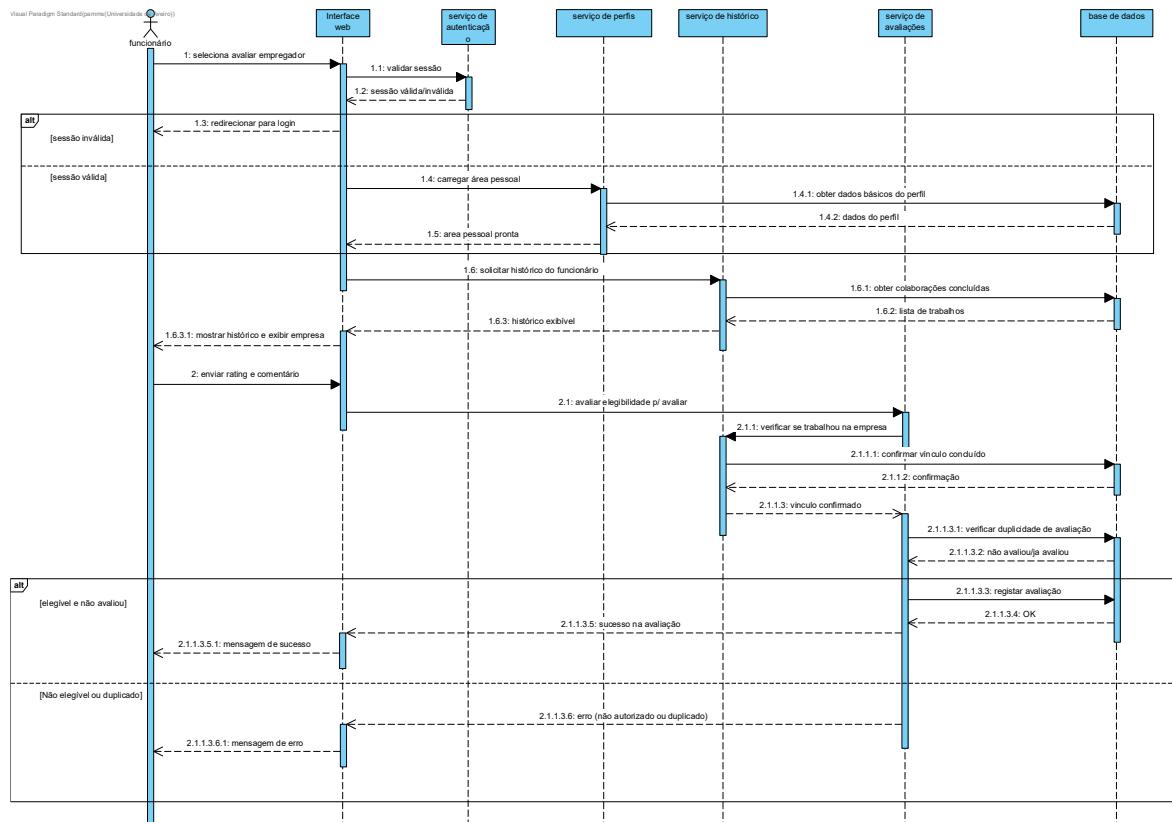


Figura 12-Diagrama de Sequência-Avaliar empregador [opcional]

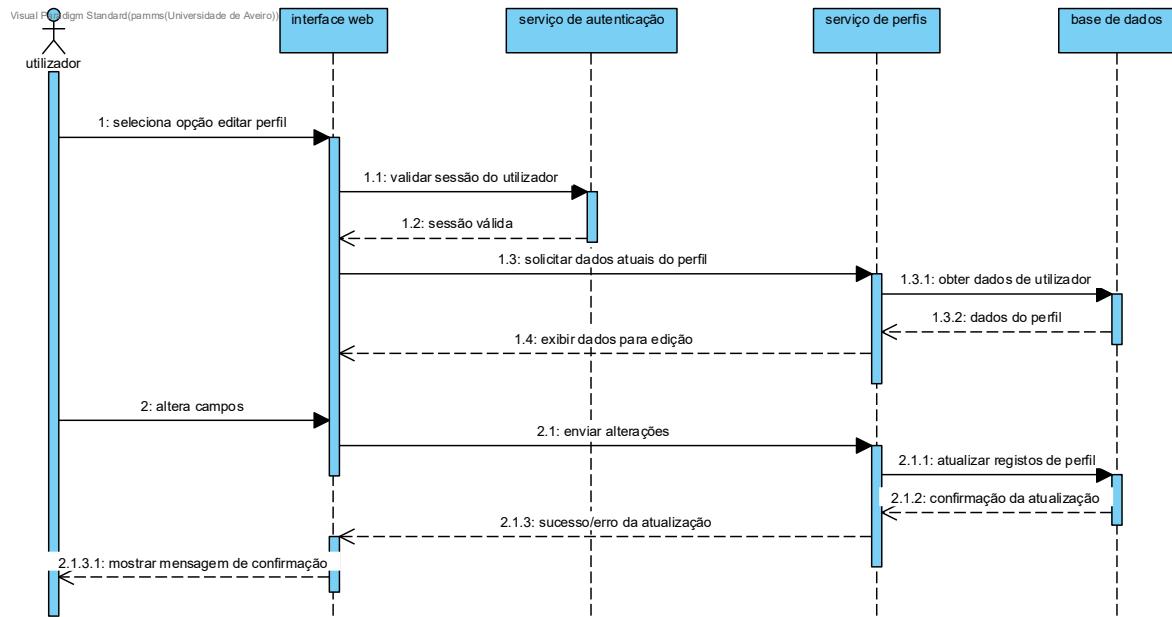


Figura 13-Diagrama de Sequência-Editar perfil

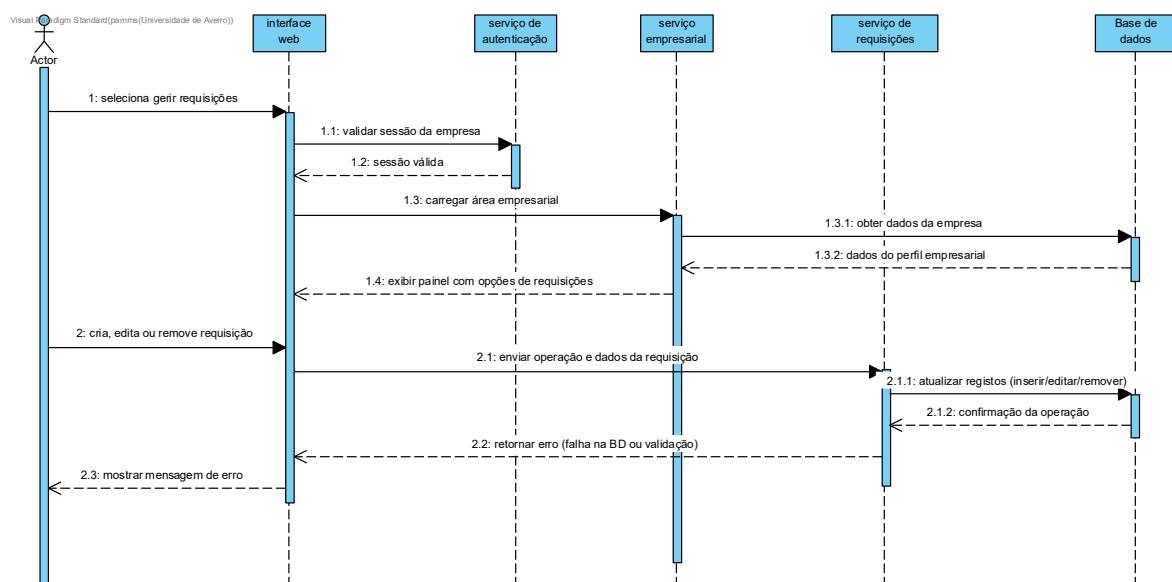


Figura 14-Diagrama de Sequência-Gerir requisição [Inválido]

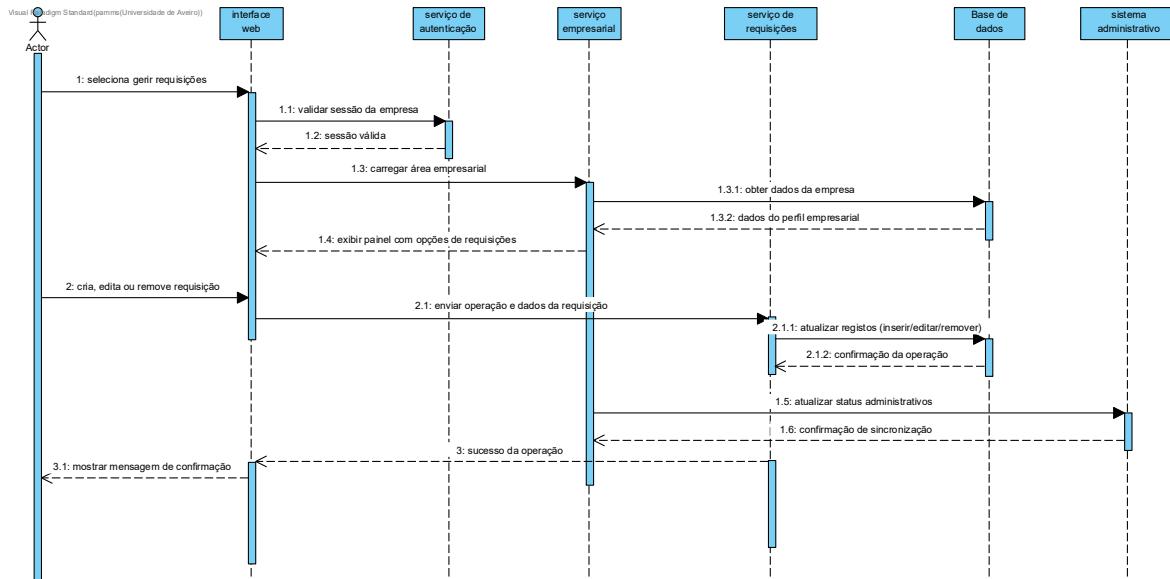


Figura 15-Diagrama de Sequência-Gerir requisições [Valido]

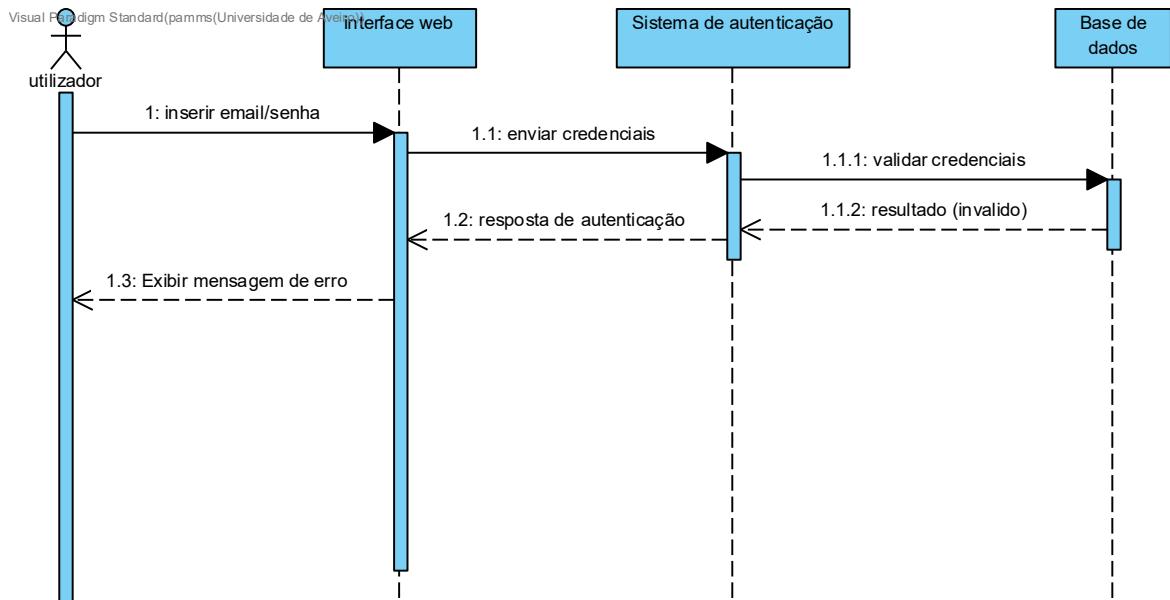


Figura 16-Diagrama de Sequência-Login [Invalido]

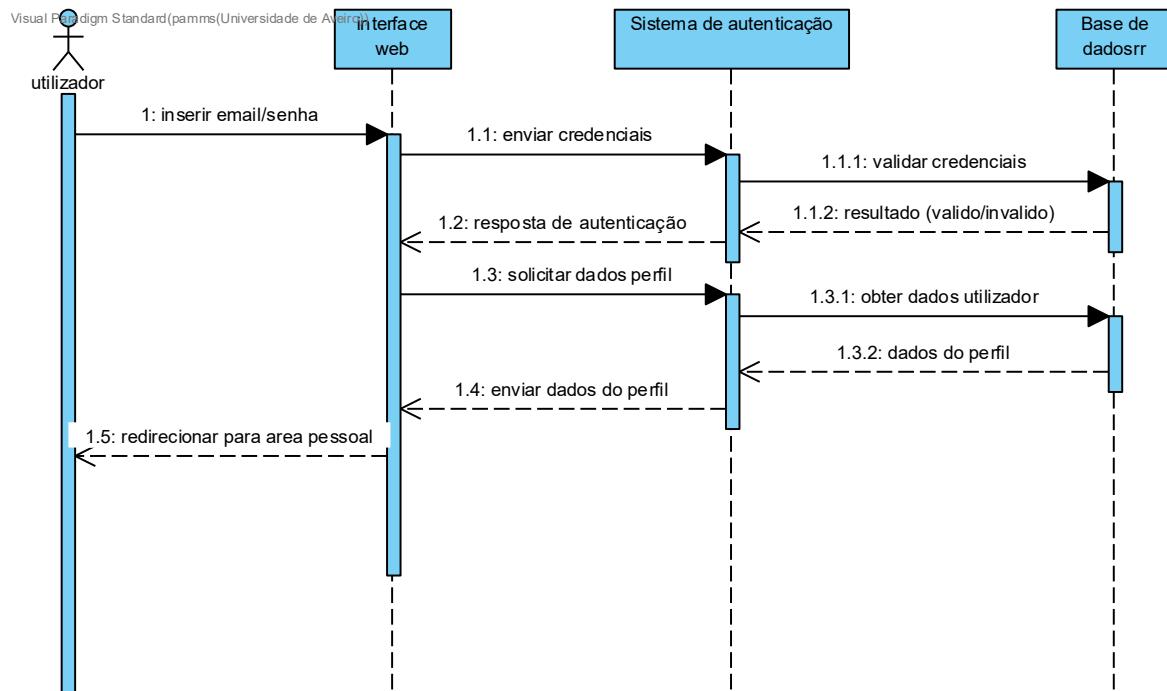


Figura 17-Diagrama de Sequência-Login [Valido]

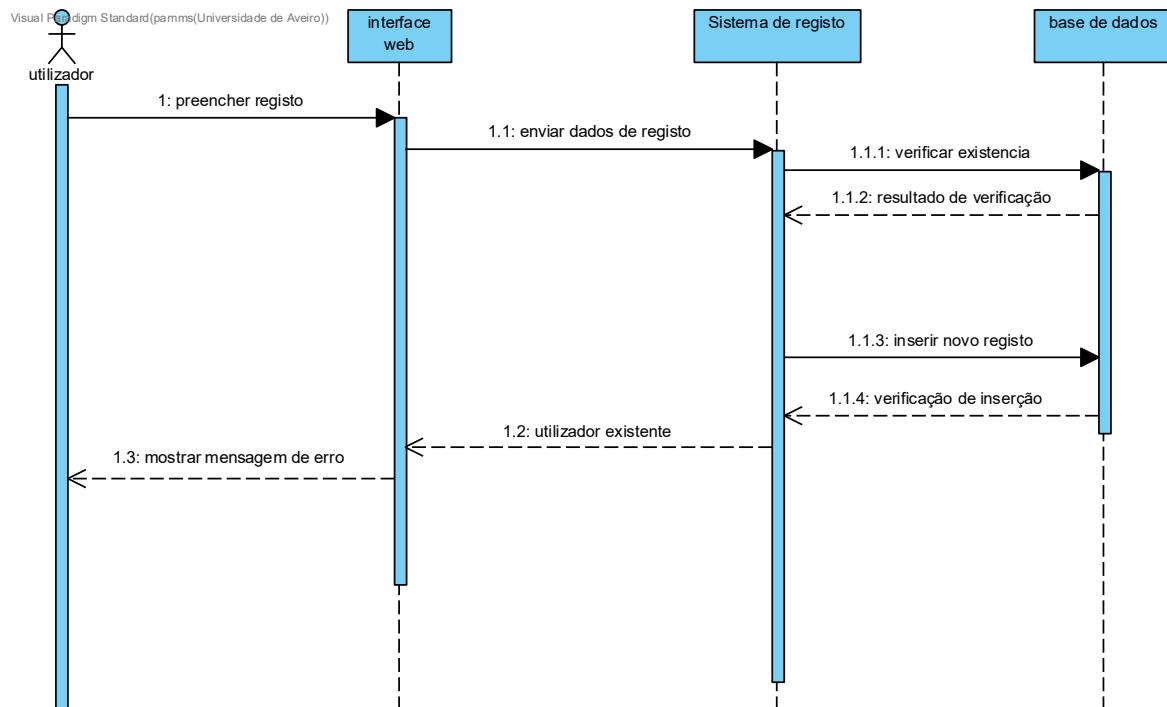


Figura 18-Diagrama de Sequência-Registrar [Invalido]

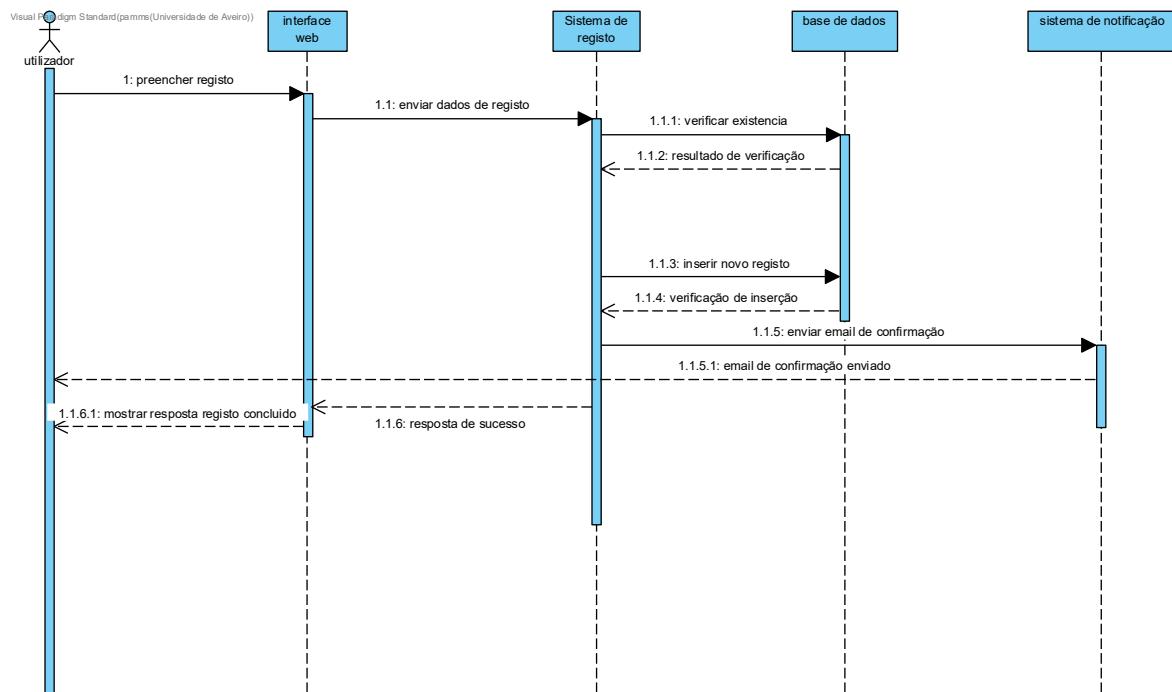


Figura 19-Diagrama de Sequência-Registar [Válido]

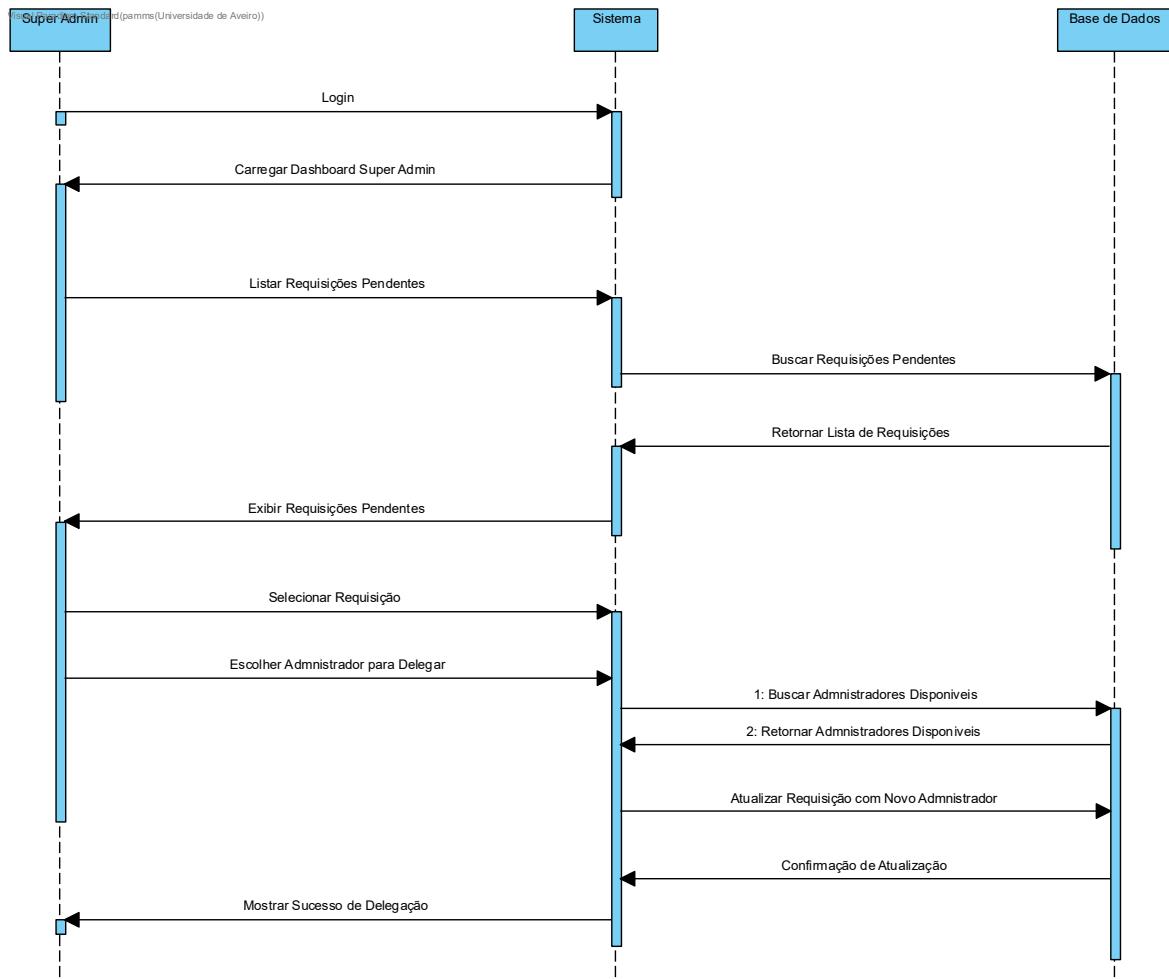


Figura 20-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador

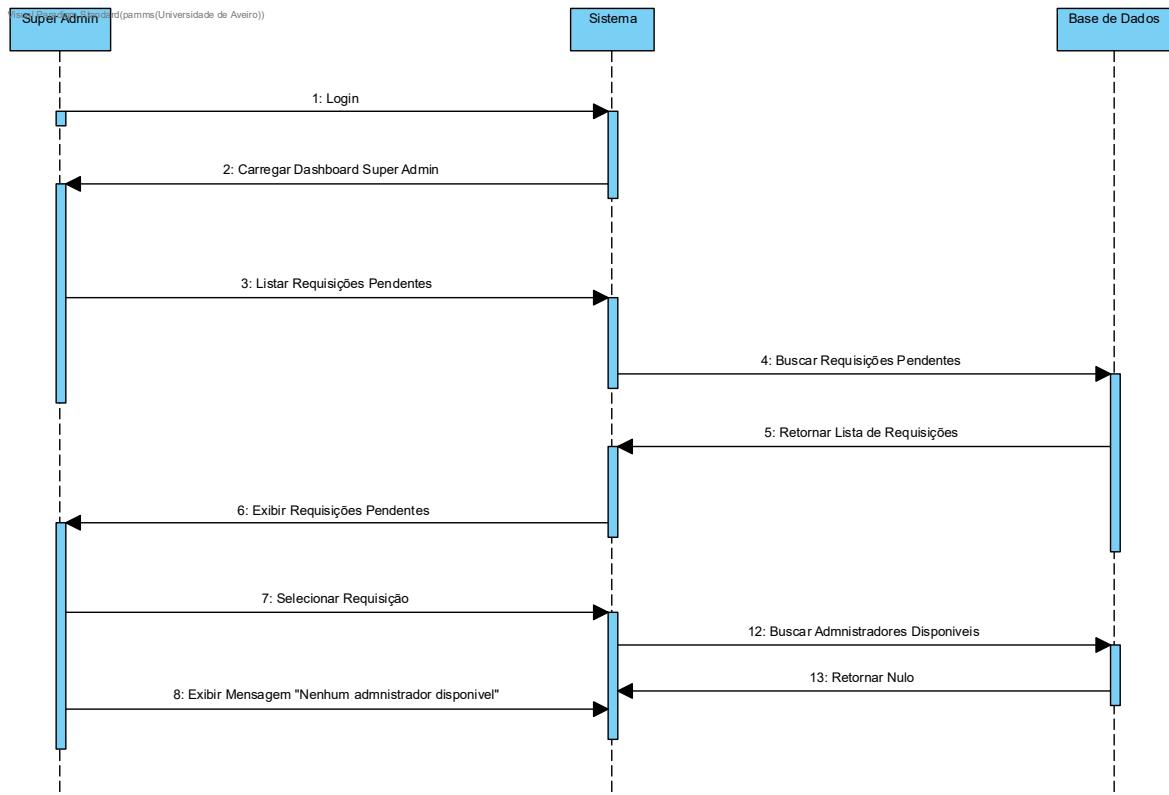


Figura 21-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador [Nenhum admin disponível]

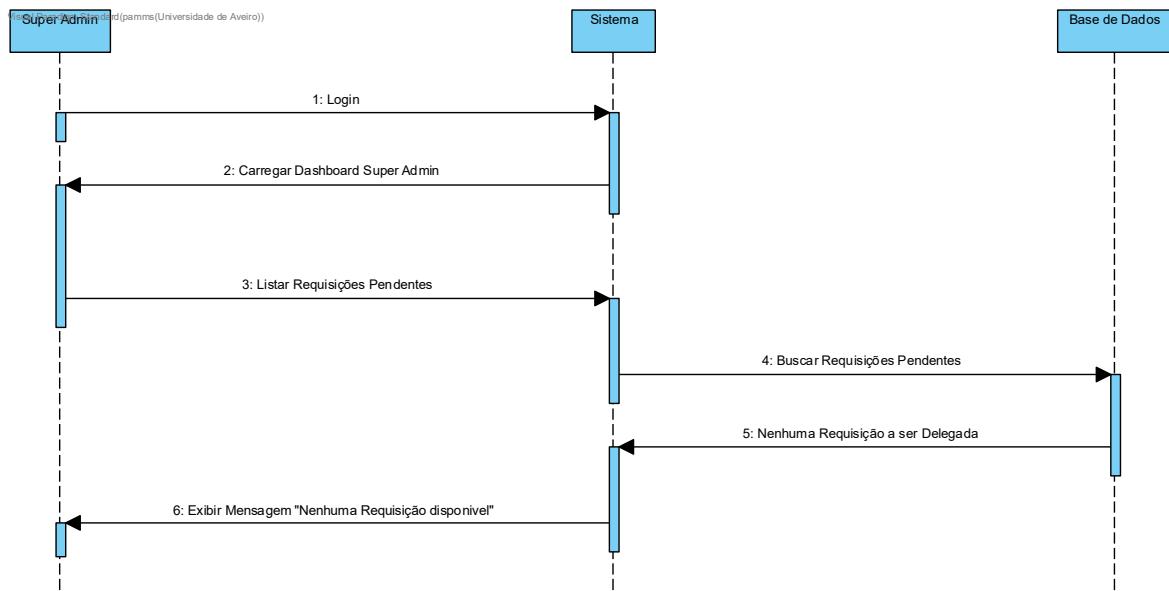


Figura 22-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador [Nenhuma requisição pendente]

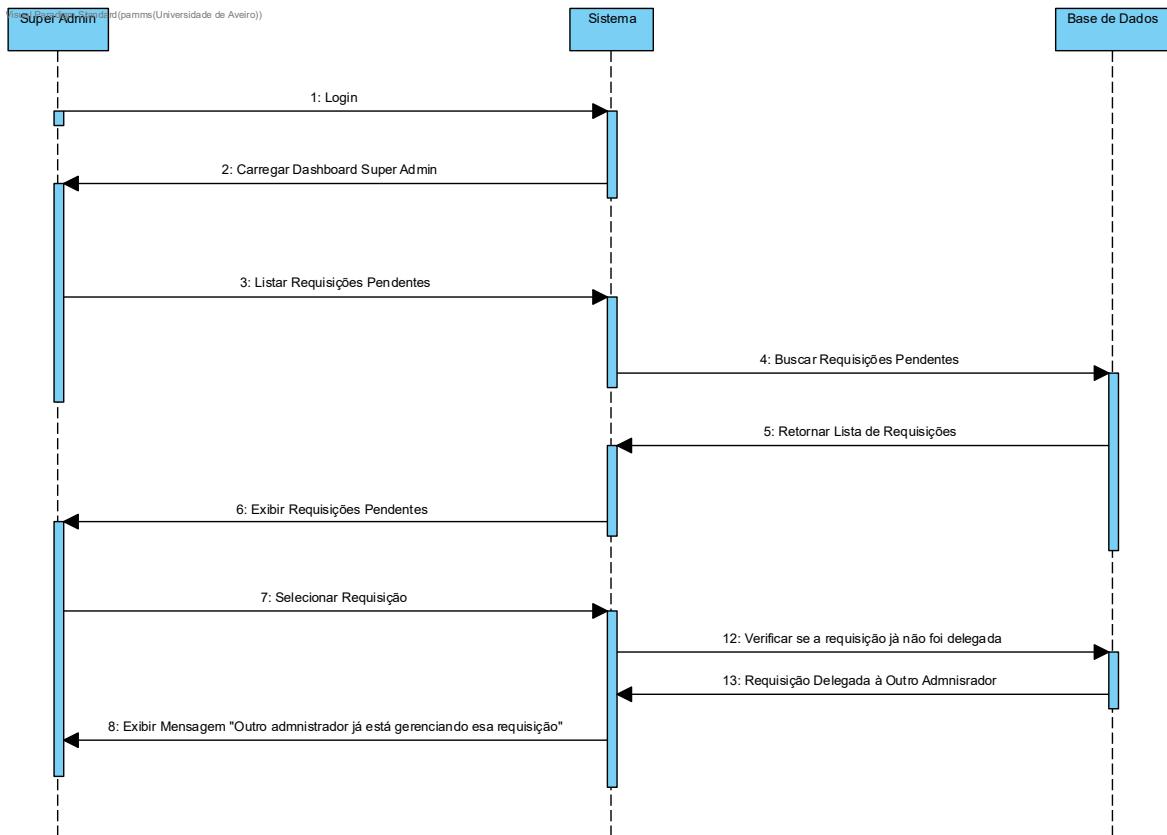


Figura 23-Diagrama de Sequência-Super-Admin delega uma requisição a Administrador [Requisição já delegada a outro administrador]

## 3.2 Prototipagem do Sistema

### 3.2.1 Protótipo de Baixa-Fidelidade

Os wireframes apresentados nesta secção representam a estrutura inicial da aplicação, mostram de forma simples e sem detalhes visuais como cada ecrã foi planeado. O objetivo principal desta fase foi validar a organização da interface, o posicionamento dos elementos e a navegação entre ecrãs, para garantir que a aplicação cumpra os requisitos funcionais definidos anteriormente.

Estes wireframes correspondem aos principais ecrãs identificados nos casos de utilização, nomeadamente:

- Homepage da aplicação, que será a primeira impressão do utilizador.
- Página de login, onde o utilizador autentica-se para aceder á sua área.
- Área pessoal do funcionário, que permite consultar perfil, histórico e candidaturas.
- Área da empresa, destinada à gestão de requisições e colaborações.
- Painel administrativo, que reúne as principais funcionalidades de controlo do sistema.

Os wireframes foram feitos para representar de forma simples e clara os requisitos definidos nas partes anteriores, como a gestão de utilizadores, a criação e visualização de requisições e a troca de mensagens entre os diferentes perfis. Esta etapa foi importante para criar uma base sólida antes de passar para o protótipo de alta-fidelidade, que é apresentada na secção seguinte.

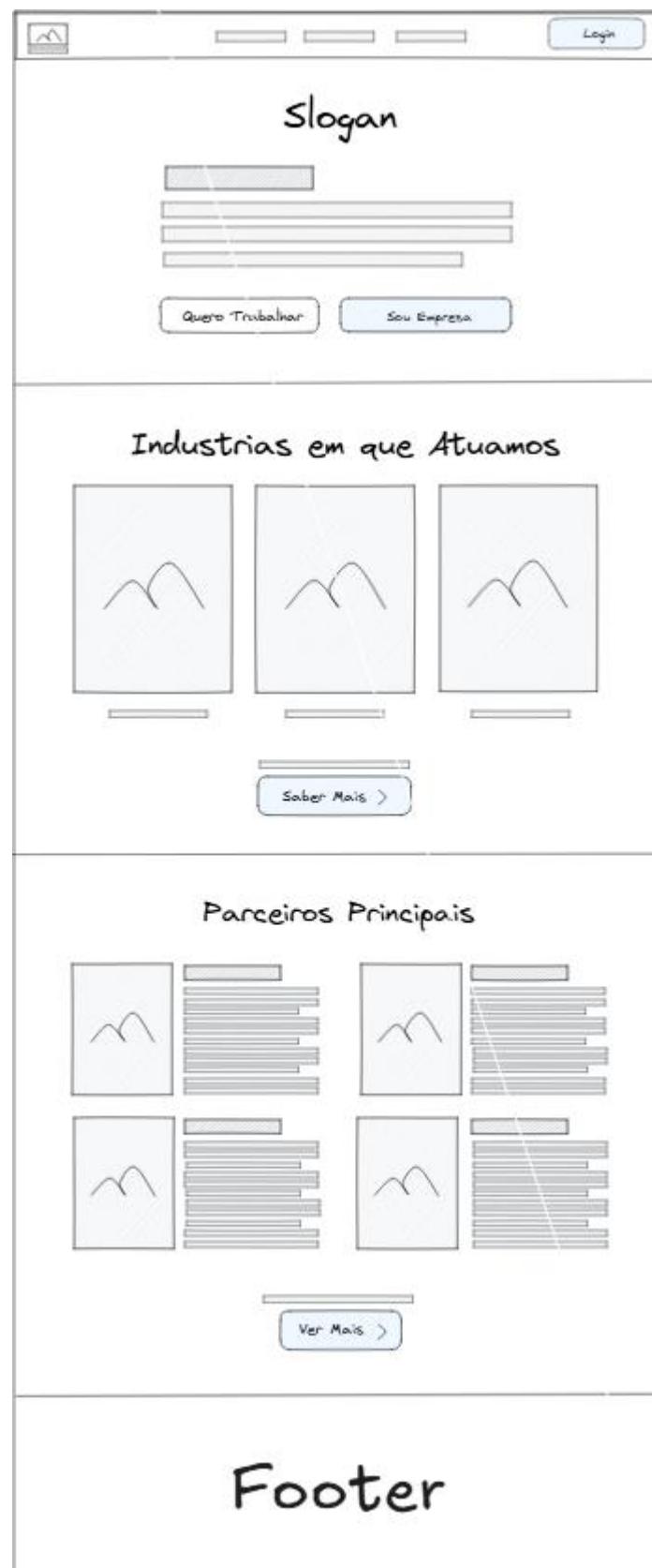


Figura 24-Wireframe Homepage



Input Field Text

Entrar

Quero criar conta

★ Google

★ LinkedIn

Figura 25-Wireframe página de login



Perfil

Alerta de Ofertas

Documentos

Dados Pessoais Formação Últimos Trabalhos Preferências

**Dados pessoais**

Name

Apelido

Data de Nascimento

Genêro

Nacionalidade

NIF

Nº de Telemovel

Figura 26-Wireframe Dados pessoais

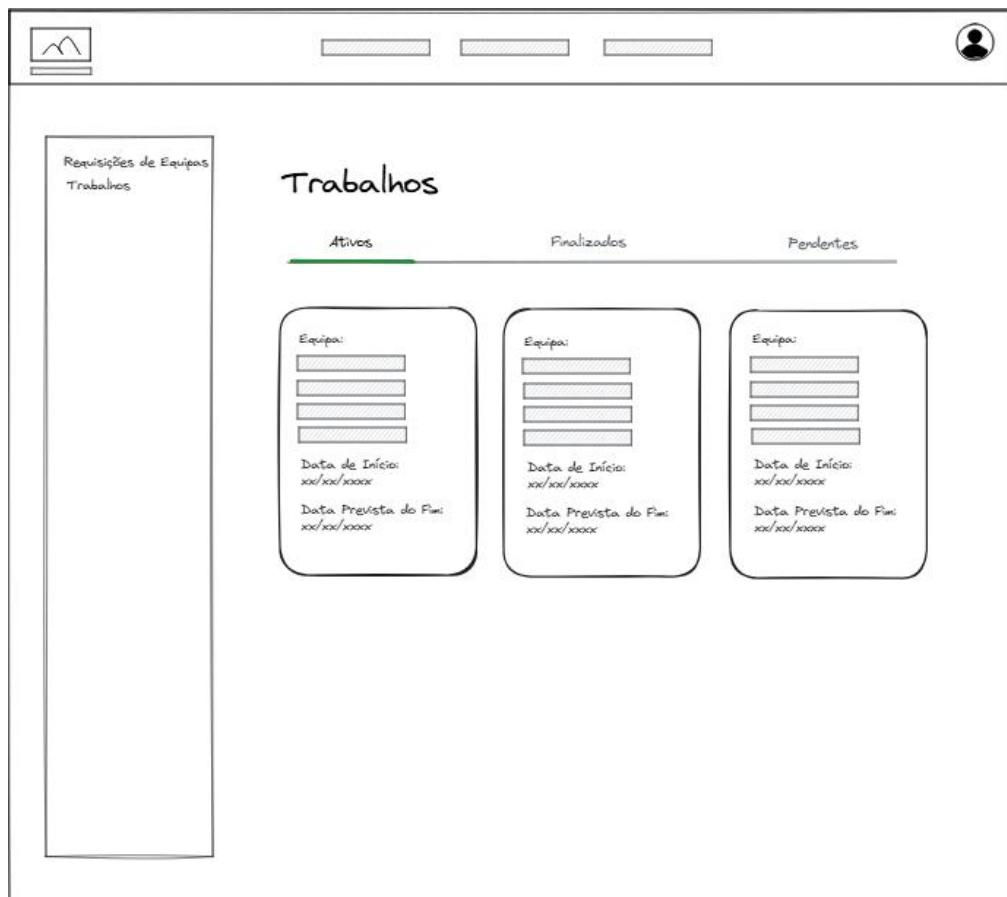


Figura 27-Wireframe trabalhos ativos

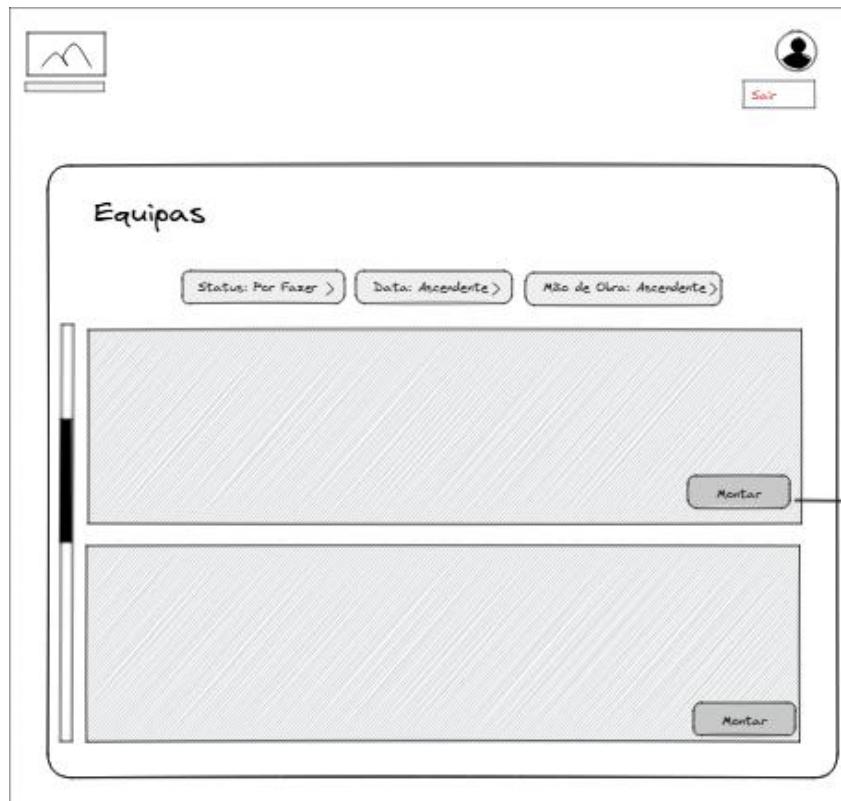
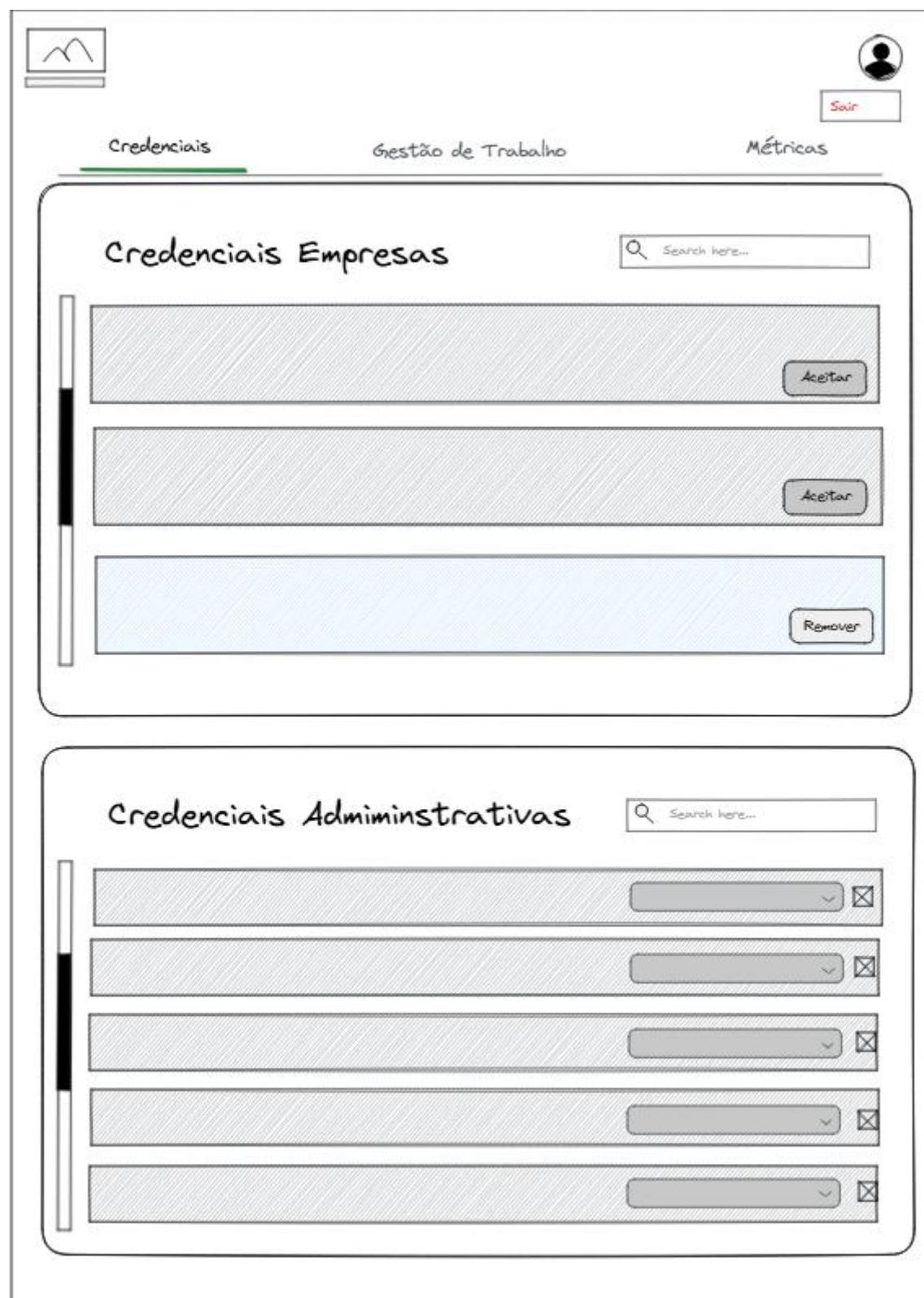


Figura 28-Wireframe Admin escolher equipas



The wireframe illustrates the 'Credenciais' (Credentials) management interface for a Super-Admin. It features a header with a profile icon, a search bar, and navigation tabs for 'Credenciais', 'Gestão de Trabalho', and 'Métricas'. The main content area is divided into two sections: 'Credenciais Empresas' (Enterprise Credentials) and 'Credenciais Administrativas' (Administrative Credentials). Each section contains a list of items, with the third item in each list highlighted in blue. Action buttons like 'Aceitar' (Accept) and 'Remover' (Remove) are shown next to the items.

Credenciais

Gestão de Trabalho

Métricas

Credenciais Empresas

Search here...

Aceitar

Aceitar

Remover

Credenciais Administrativas

Search here...

Remove

Remove

Remove

Remove

Remove

Figura 29-Wireframe Gestão de credenciais do Super-Admin

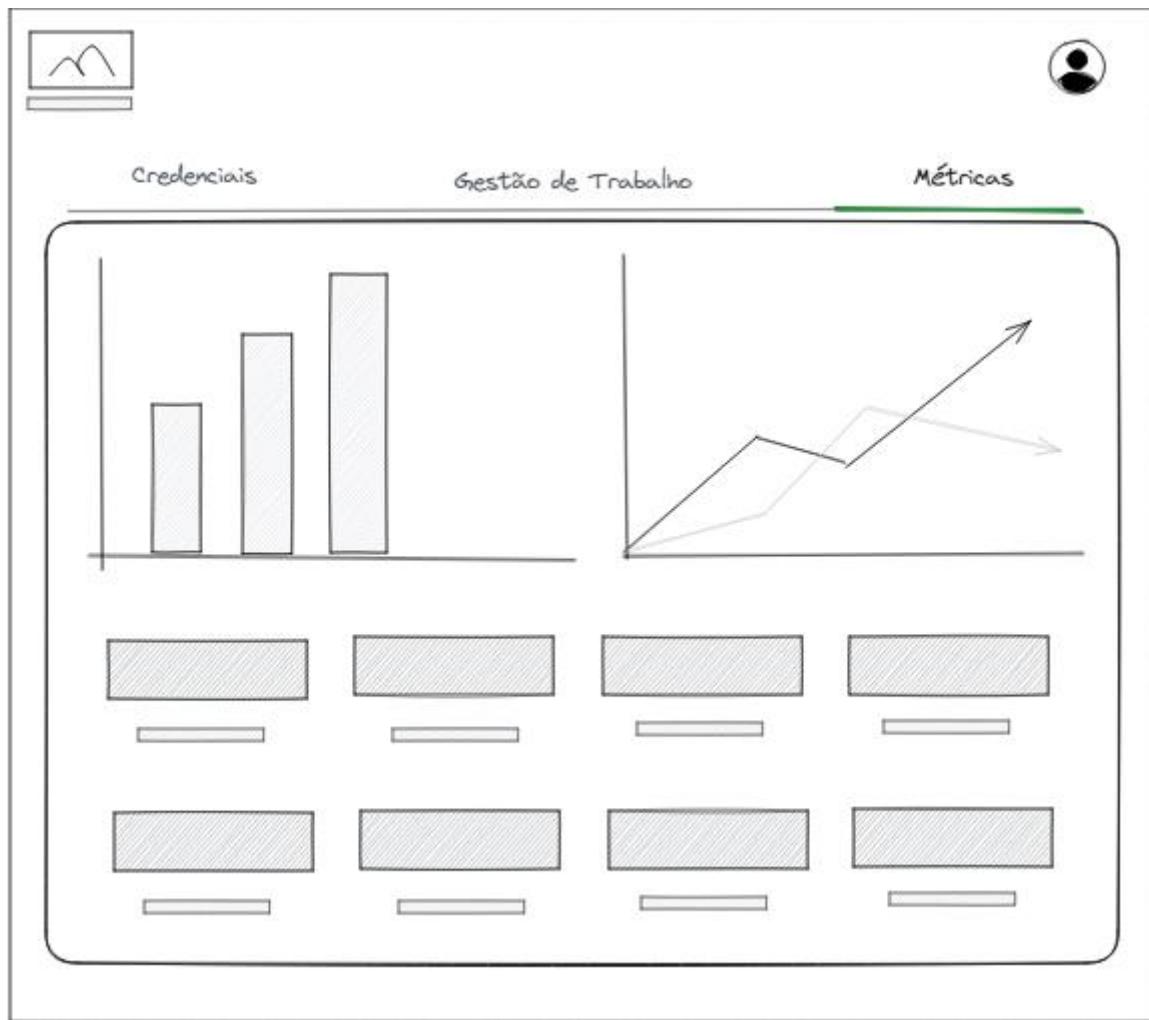


Figura 30-Wireframe Metricas do Super-Admin

### 3.2.1.1 Justificações das Decisões Tomadas

As decisões tomadas na construção dos wireframes de baixa fidelidade da aplicação tiveram como base critérios de usabilidade, acessibilidade e coerência visual procurando criar uma estrutura simples e fácil de navegar para todos os tipos de utilizadores.

Optou-se por um layout limpo e intuitivo com menu principal na parte superior e acesso rápido às áreas mais utilizadas. Esta escolha facilita a navegação e garante que o utilizador encontra facilmente as funcionalidades principais, sem sobrecarga visual. A organização dos ecrãs foi pensada para minimizar o número de cliques necessários para chegar às ações mais importantes, seguindo boas práticas de design centrado no utilizador.

As páginas iniciais apresentam apenas informação geral, como slogan, áreas de atuação e parceiros de forma a criar uma primeira impressão clara e objetiva. Após o login, cada perfil acede à sua área específica, com elementos adaptados às suas funções, o que melhora a eficiência e a personalização da experiência.

De forma geral, o design dos wireframes reflete a preocupação em criar uma estrutura funcional e clara, alinhada com os requisitos do projeto e as necessidades dos utilizadores da plataforma.

### 3.2.1.2 Avaliação do Protótipo com Pessoas Externas

O protótipo de baixa fidelidade da aplicação foi testado por um pequeno grupo de pessoas utilizadores de fora da equipa de desenvolvimento. O objetivo foi perceber se a estrutura da navegação, os ecrãs e a forma como a informação estava organizada faziam sentido e eram fáceis de entender. Esta fase foi importante para confirmar se o design inicial correspondia às expectativas dos utilizadores e se a interface era intuitiva e simples de usar.

A avaliação contou com colegas e potenciais utilizadores com perfis próximos do público-alvo da aplicação, como funcionários e representantes de empresas ligadas ao setor industrial. O método usado foi simples, observação direta e recolha de opiniões espontâneas. Durante uma pequena sessão, os participantes analisaram os wireframes e deram o seu feedback sobre a navegação, a disposição dos elementos e a facilidade em realizar ações básicas, como fazer login, atualizar o perfil ou criar uma requisição.

Os principais pontos analisados:

- Estrutura e lógica da navegação entre ecrãs.
- Clareza dos botões e ícones de ação.
- Organização das informações no ecrã.
- Facilidade de compreensão das opções apresentadas.
- Adequação da interface ao perfil de cada tipo de utilizador.

De forma geral, o feedback foi positivo, destacando-se a simplicidade da navegação, a organização clara dos conteúdos e a separação coerente das áreas por perfil. Alguns participantes sugeriram melhorar a visibilidade de determinados botões de ação e reduzir a quantidade de texto em certos ecrãs iniciais, de modo a tornar a interface ainda mais direta. Estas observações foram tidas em conta e serviram de base para o protótipo de alta fidelidade, apresentando na secção seguinte, onde foram ajustados elementos visuais e reforçada a hierarquia da informação.

### 3.2.1.3 Análise de Resultados

Com base no feedback recolhido durante a avaliação do protótipo de baixa fidelidade, foi possível identificar e implementar diversas melhorias no protótipo de alta fidelidade, tornando a interface mais clara, coerente e próxima da versão final da aplicação.

Os avaliadores destacaram como pontos positivos a organização lógica dos ecrãs, a separação por perfis de utilizador e a simplicidade da navegação, fatores que foram mantidos e reforçados na versão atual. A estrutura geral, com menus superiores e áreas específicas para cada tipo de utilizador, foi mantida, garantindo uma experiência consistente entre as secções da aplicação.

Entre as melhorias aplicadas, destacam-se:

- Ajuste do layout da homepage, com redução do texto e maior destaque visual para as áreas de atuação e parceiros principais.
- Maior visibilidade dos botões e ações principais, através do uso de contraste de cores e posicionamento centralizado.

- Utilização de ícones e separadores visuais, que facilitam a navegação e ajudam o utilizador a identificar rapidamente cada funcionalidade.
- Uniformização do design, assegura coerência entre os diferentes ecrãs.

O protótipo de alta-fidelidade reflete assim uma interface mais polida e funcional, onde o equilíbrio entre estética e usabilidade foi aprimorado. As alterações basearam-se diretamente nas observações recolhidas, o que comprova que o processo de avaliação foi fundamental para melhorar a clareza, a legibilidade e a experiência de utilização da aplicação.

### 3.2.2 Protótipo de Alta-Fidelidade

O protótipo de alta-fidelidade da aplicação representa a versão visual final da interface, desenvolvida com base nas melhorias identificadas durante a fase de baixa fidelidade. Esta versão já integra todos os elementos gráficos, isto inclui cores, tipografia, logótipo, ícones e componentes visuais que refletem o aspetto real da aplicação.

O protótipo foi criado na ferramenta Figma, que permite definir de forma precisa o layout final, a estrutura de navegação e o comportamento visual de cada ecrã. Foram concebidas as principais interfaces da aplicação, que abrangem os diferentes perfis de utilizado, cada uma adaptada às respetivas funcionalidades e permissões de acesso.

A homepage apresenta uma estrutura limpa e moderna, destaca o slogan da plataforma, as áreas de atuação e os parceiros principais, o que cria uma primeira impressão clara e profissional. Após o login, os utilizadores accedem à sua área pessoal, onde encontram informações relevantes, métricas, notificações e atalhos para as ações mais comuns. Os ecrãs de registo foram divididos em vários passos, isto torna o processo mais organizado e fácil de compreender, enquanto os painéis administrativos apresentam uma estrutura em tabelas e cartões, para otimizar a leitura e a gestão de dados.

As decisões de design tiveram como base a coerência visual e a usabilidade:

- A paleta de cores combina tons de azul e branco, isto transmite profissionalismo e confiança, em linha com o setor industrial.
- A tipografia simples e legível reforça a clareza e o foco na informação.
- O uso de ícones e separadores laterais facilita a navegação e diferencia as ações principais.

O protótipo de alta-fidelidade serviu como validação visual e funcional da aplicação antes do desenvolvimento técnico, permite antecipar eventuais ajustes de navegação e garantir uma experiência de utilizador fluida e consistente.

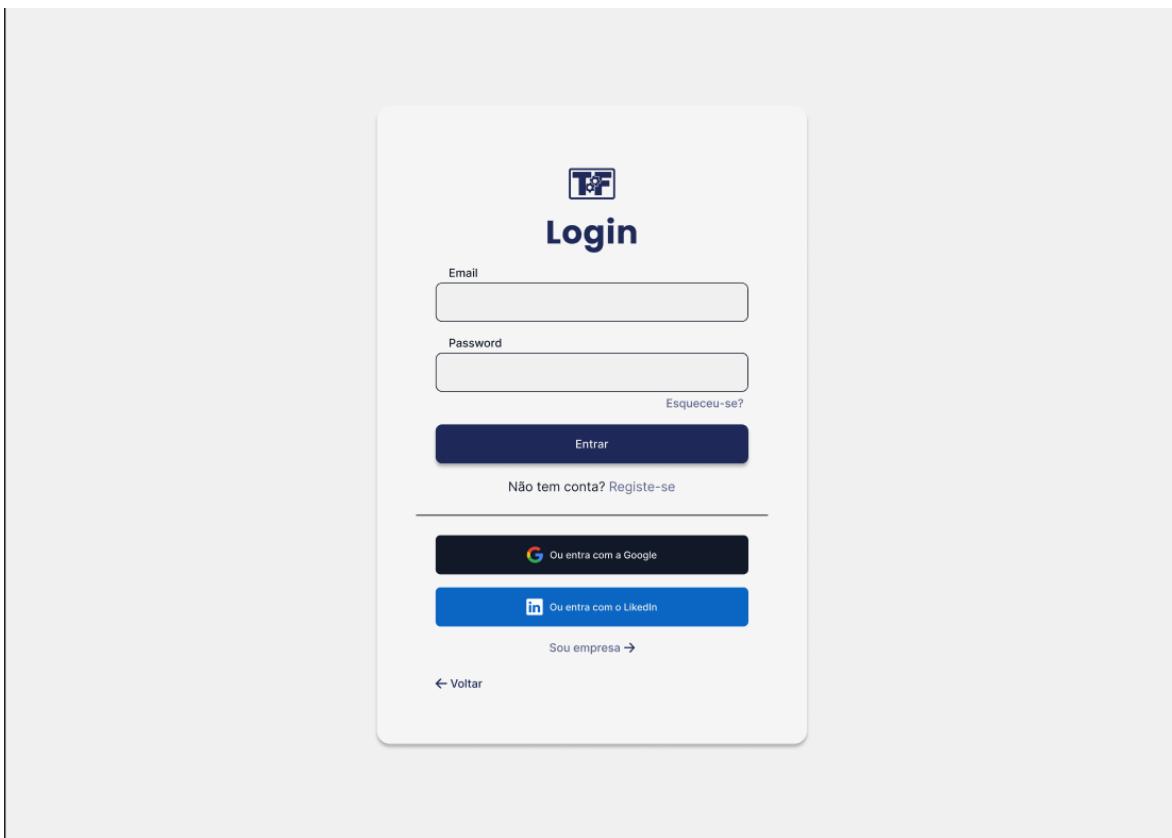
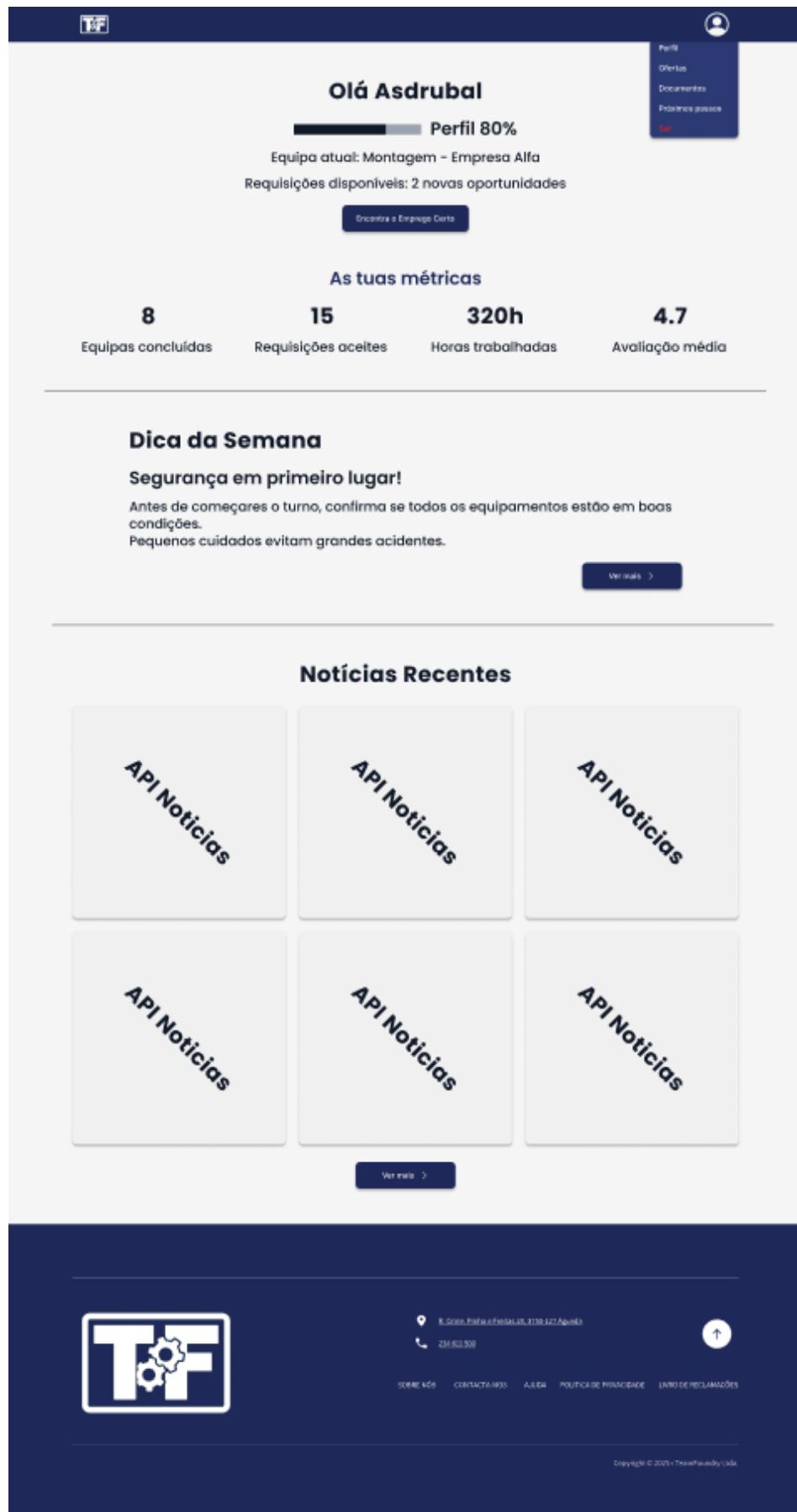


Figura 31-Protótipo de alta-fidelidade-página de login candidato



The image displays a high-fidelity prototype of a digital platform interface. At the top, there's a dark header bar with the 'estga' logo and a search bar containing the letters 'TF'. A user profile icon is on the right. A vertical menu on the right side includes 'Perfil', 'Oferias', 'Documentos', 'Prioridades pessoais', and a red-highlighted 'Sair'. Below the header, a greeting 'Olá Asdrubal' is followed by a progress bar labeled 'Perfil 80%'. It shows the user's current team ('Equipa atual: Montagem - Empresa Alfa') and available opportunities ('Requisições disponíveis: 2 novas oportunidades'). A button 'Encontra a Empreaga Conta' is present. A section titled 'As tuas métricas' displays four metrics: '8 Equipas concluídas', '15 Requisições aceites', '320h Horas trabalhadas', and '4.7 Avaliação média'. A horizontal line separates this from the next section. The 'Dica da Semana' section features the heading 'Segurança em primeiro lugar!' with the text 'Antes de começares o turno, confirma se todos os equipamentos estão em boas condições. Pequenos cuidados evitam grandes acidentes.' and a 'Ver mais >' button. Another horizontal line follows. The 'Notícias Recentes' section contains six placeholder cards, each with the text 'API Notícias' diagonally across it. A 'Ver mais >' button is located at the bottom of this section. The footer is a dark blue bar with the 'TF' logo, location information ('R. Gomes Pinto e Melo, 1010-127 Lisboa'), a phone number ('211 823 500'), and social media icons. Navigation links include 'SERVIÇOS', 'CONTACTA-NOS', 'AJUDA', 'POLÍTICA DE PRIVACIDADE', and 'LIVRO DE RECLAMAÇÕES'. A copyright notice 'Copyright © 2023 - Transfowandy Lda.' is also present.

Figura 32-Protótipo de alta-fidelidade

T

As Indústrias   Os Parceiros   Sobre nós   

 Perfil  
 Alerta de Emprego  
 Documentos  
 Proximos passos

  
**Nome Sobrenome**

[Dados Pessoais](#) [Formação](#) [Últimos Trabalhos](#) [Preferências](#)

Nome

Data de Nascimento

Nacionalidade

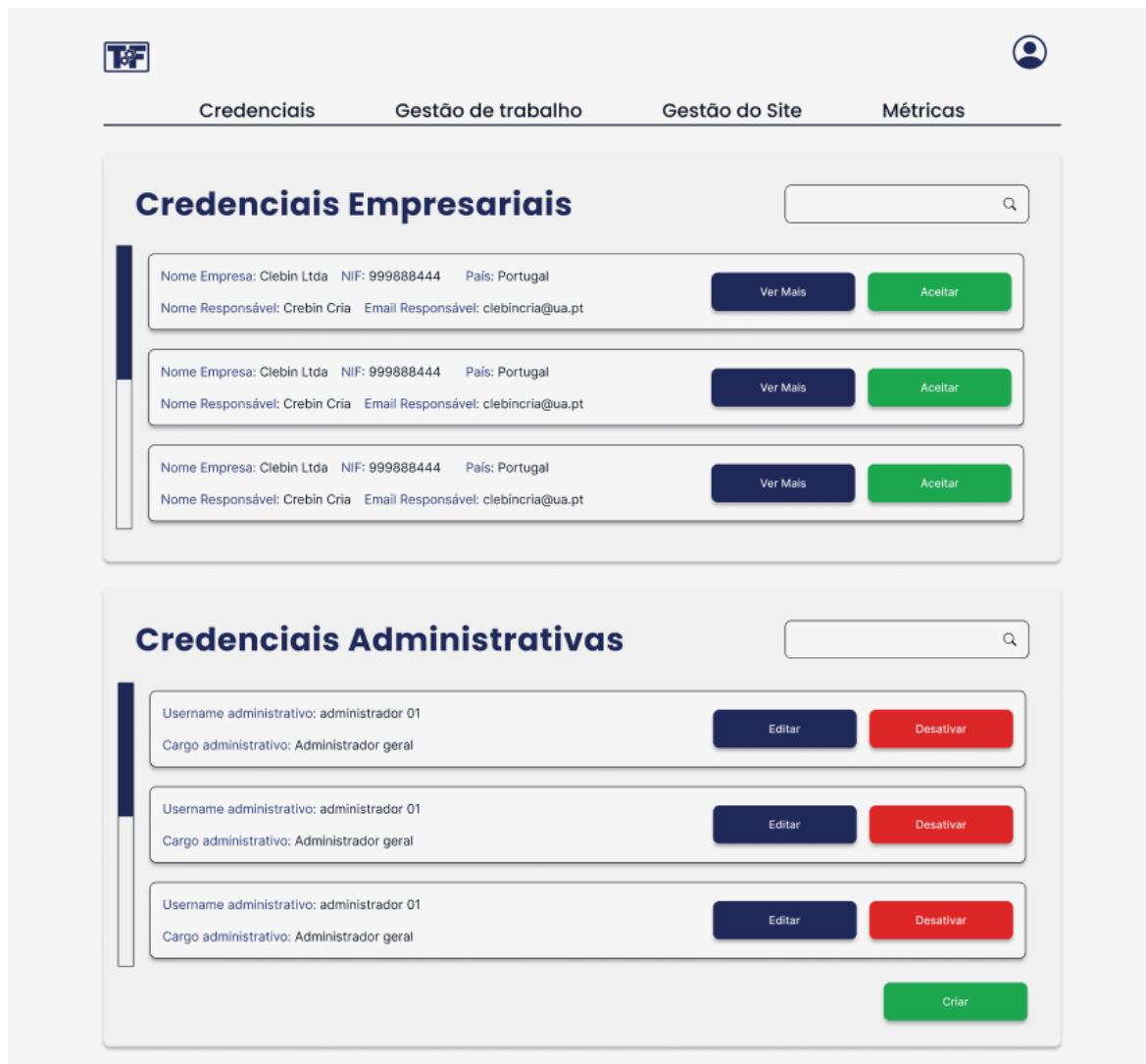
Contacto

Apelido

Genêro

NIF

Figura 33-Protótipo de alta-fidelidade-Página de dados pessoais



**Credenciais Empresariais**

Nome Empresa: Clebin Ltda NIF: 999888444 País: Portugal	Ver Mais	Aceitar
Nome Responsável: Crebin Cria Email Responsável: clebincria@ua.pt		

Nome Empresa: Clebin Ltda NIF: 999888444 País: Portugal	Ver Mais	Aceitar
Nome Responsável: Crebin Cria Email Responsável: clebincria@ua.pt		

Nome Empresa: Clebin Ltda NIF: 999888444 País: Portugal	Ver Mais	Aceitar
Nome Responsável: Crebin Cria Email Responsável: clebincria@ua.pt		

**Credenciais Administrativas**

Username administrativo: administrador 01	Editar	Desativar
Cargo administrativo: Administrador geral		

Username administrativo: administrador 01	Editar	Desativar
Cargo administrativo: Administrador geral		

Username administrativo: administrador 01	Editar	Desativar
Cargo administrativo: Administrador geral		

**Criar**

Figura 34-Protótipo de alta-fidelidade-Gestão de credenciais Super-Admin

### 3.2.2.1 Guia de Estilos

O guia de estilos define o conjunto de regras visuais e de identidade gráfica que orientam o design da interface, garantindo uma experiência coerente, moderna e profissional em todas as páginas e dispositivos. Serve de referência para o desenvolvimento visual, assegurando consistência na aplicação das cores, tipografia, espaçamentos e componentes.

A paleta de cores é composta por tons de azul-escuro e branco como cores principais, transmitindo confiança, estabilidade e profissionalismo - valores centrais da plataforma. As cores secundárias em tons de cinzento e azul-claro são usadas em elementos de navegação e realce, enquanto as cores de estado (verde, vermelho e laranja) representam, respetivamente, sucesso, erro e alerta, facilitando a percepção imediata do estado das ações do utilizador.

A tipografia segue uma hierarquia clara entre títulos, subtítulos e corpo de texto, assegurando legibilidade em diferentes resoluções. Foram definidas variantes para desktop e mobile, ajustando tamanhos e espaçamentos de forma proporcional, garantindo uma leitura fluida e consistente entre dispositivos.

Os componentes visuais reutilizáveis - como botões, tabelas, barras de navegação (navbar), modais e cartões - foram desenvolvidos com base em TailwindCSS e DaisyUI, permitindo uma construção modular, consistente e de rápida manutenção. Estes elementos seguem o mesmo sistema de cores e tipografia definidos no guia de estilos, reforçando a uniformidade visual do projeto.

A estrutura de grelha define margens, espaçamentos e proporções entre elementos, assegurando alinhamento e equilíbrio visual nas interfaces desktop e mobile. Esta grelha serve como base para o posicionamento dos componentes, garantindo uma disposição harmoniosa e responsiva.

Por fim, o guia de estilos estabelece as regras de consistência visual que devem ser respeitadas em todas as futuras implementações, promovendo coerência estética, usabilidade e reconhecimento visual da marca.

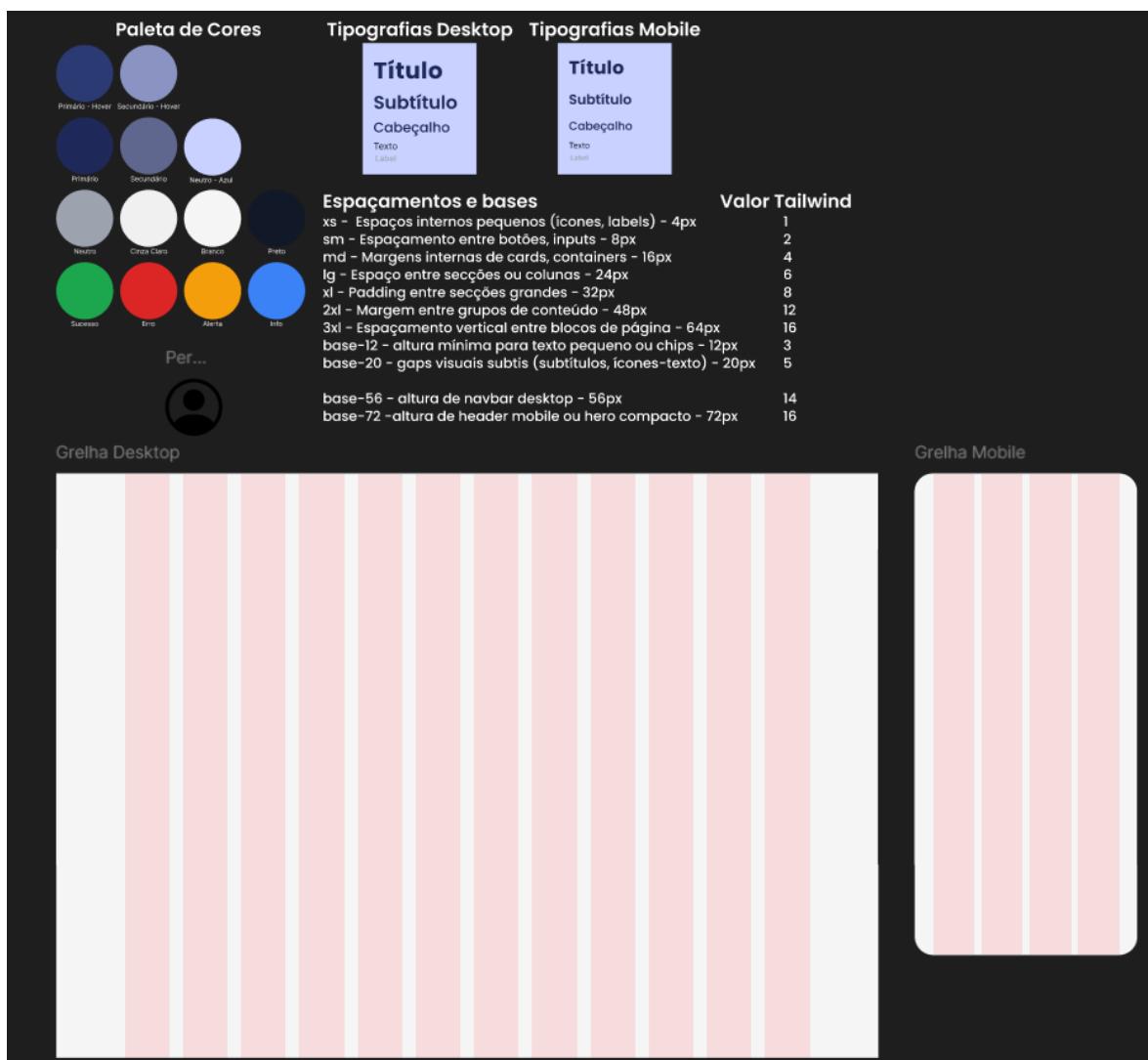


Figura 35-Paleta de cores/Tipografia/Espaçamentos e bases/Grelhas

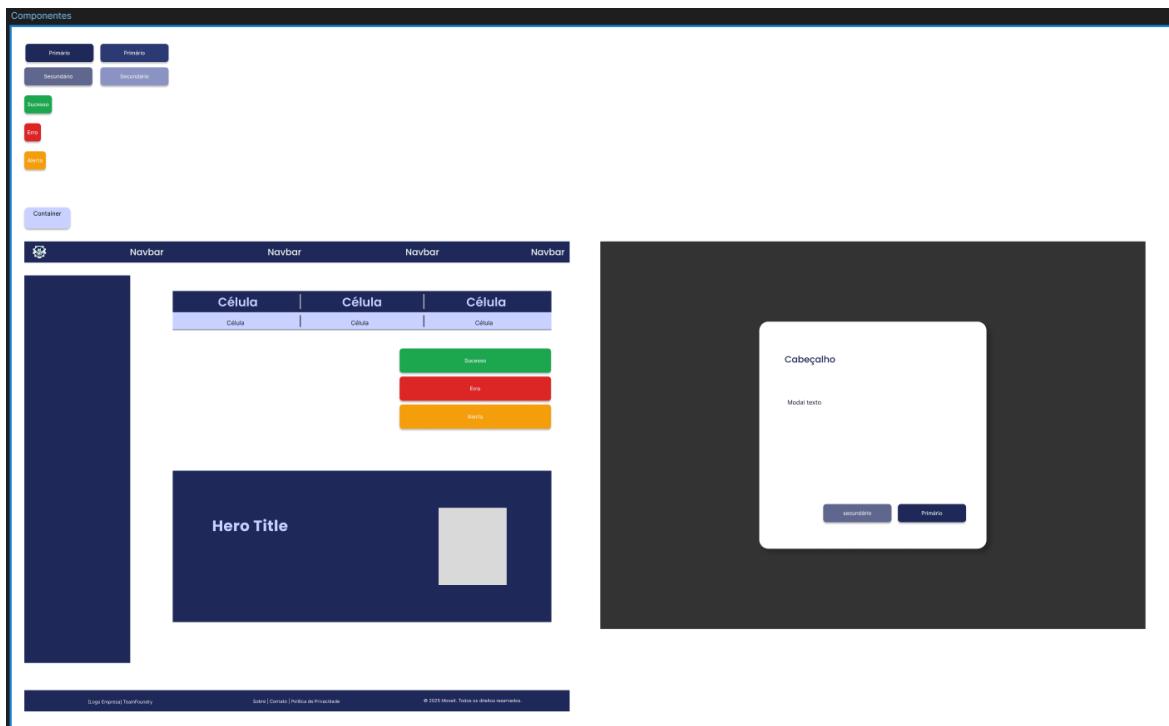


Figura 36-Componentes reutilizáveis

### 3.2.2.2 Descrição das Interfaces

As interfaces da aplicação foram feitas para serem simples e fáceis de usar. A ideia foi garantir que qualquer utilizador consiga usar a plataforma sem complicações, seja ele funcionário, empresa, administrador ou super administrador. O design foi criado no Figma, sempre com o foco em tornar a navegação intuitiva e agradável para quem a utiliza.

#### Design de informação

A informação na aplicação foi organizada de forma clara e prática, com o objetivo de tornar a navegação simples e intuitiva para todos os tipos de utilizadores. Em todos os ecrãs existe uma estrutura consistente, com o menu principal na parte superior que permite aceder às áreas mais importantes de forma rápida e direta. Dependendo do perfil, esse menu adapta-se, mostra apenas as opções relevantes para cada utilizador.

Cada área da aplicação apresenta uma disposição lógica dos conteúdos, com títulos bem visíveis, botões destacados e zonas de leitura ou ação bem delimitadas. Os painéis principais organizam a informação em blocos e cartões, isto facilita a consulta de dados, métricas e ações frequentes. Nos ecrãs de gestão, como os dos administradores e super administradores, as tabelas e listas de dados seguem um formato uniforme, permite visualizar e editar informação de forma rápida e eficiente.

#### Design gráfico

O design visual segue uma linha moderna e profissional, com uma paleta de cores centrada em tons de azul e branco, reforça a identidade da marca e transmite confiança e estabilidade, características associadas ao setor

industrial. A tipografia é simples e legível, isto garante boa leitura em diferentes tamanhos. Os ícones e elementos gráficos são consistentes em todos os ecrãs, contribui para uma identidade visual coesa. O espaçamento entre componentes foi cuidadosamente definido para evitar sobrecarga visual, e o uso de cartões e caixas permite uma apresentação organizada e equilibrada das informações.

### Design de Interação

A interação do utilizador com o sistema é intuitiva e previsível. Os botões principais estão destacados em tons de azul e incluem feedback visual (mudança de cor ou sombreamento) quando o utilizador passa o cursor ou realiza uma ação. Os formulários de login e registo foram divididos em passos curtos e sequenciais, com mensagens de validação que ajudam o utilizador a completar o processo sem erros. Nos dashboards administrativos, o utilizador pode filtrar, editar e visualizar dados através de botões e ícones uniformes.

#### Exemplos de ecrãs principais:

- Homepage: apresenta o slogan, áreas de atuação e parceiros principais, para introduzir o utilizador à aplicação.
- Ecrãs de Login e Registo: Permitem o acesso ao sistema, com um design limpo e foco na simplicidade e validação passo a passo.
- Área do Funcionário: Centraliza o perfil pessoal, formações, documentos e histórico de trabalhos, permite editar dados e visualizar notificações de emprego.
- Área da Empresa: Permite criar e gerir requisições, acompanhar equipas formadas e monitorizar trabalhos ativos, finalizados ou pendentes.
- Painel do Administrador: Apresenta as opções de gestão de equipas e funcionários, com filtros e visualização detalhada de perfis.
- Painel do Super-Administrador: Concentra as ferramentas de gestão global do sistema, inclui credenciais, métricas e controlo de administradores.

### Regras de consistência

Para garantir uma experiência uniforme, foram aplicadas regras de consistência em todas as interfaces:

- O menu superior e os títulos mantêm o mesmo estilo e posição.
- As ações principais utilizam sempre o mesmo formato de botão.
- As mensagens de erro, sucesso e aviso seguem o mesmo padrão visual e tonal.
- A navegação lateral e as estruturas em cartões repetem-se nos diferentes perfis, facilita a aprendizagem do utilizador.

#### 3.2.2.3 Estratégia de Promoção e Acessibilidade

Desde o início, quisemos que a TeamFoundry fosse simples de usar. A nossa ideia sempre foi criar uma aplicação acessível, que qualquer pessoa conseguisse perceber e usar sem esforço, seja qual for o dispositivo

ou as suas capacidades. Queríamos que tudo fosse intuitivo, direto e sem barreiras, que o utilizador conseguisse navegar naturalmente, sem ter de pensar muito no que fazer.

Durante o design e planeamento, seguimos várias boas práticas de acessibilidade, que definem regras para melhorar a usabilidade de aplicações web. Entre as principais medidas aplicadas são:

- Contraste adequado de cores, para facilitar a leitura e garantir boa visibilidade em todos os ecrãs.
- Tipografia limpa e legível, com tamanhos equilibrados e espaçamento confortável.
- Estrutura semântica correta, para que os leitores de ecrã interpretem os elementos na ordem certa.
- Feedback visual nas ações, como mudanças de cor e sombreamento, isto ajuda o utilizador a perceber quando algo foi clicado ou concluído.

Além da parte técnica, também pensámos na inclusão digital, queríamos que a aplicação fosse fácil de usar mesmo para quem tem pouca experiência com tecnologia. Por isso, optámos por uma estrutura simples, ícones intuitivos e linguagem direta, reduzindo a complexidade e tornando a navegação mais natural.

Durante o design, usámos ferramentas como o Lighthouse, para testar contraste e legibilidade, e as opções de verificação do Figma, que nos ajudaram a detetar e corrigir pequenos detalhes antes de avançar para a fase de implementação.

No fim, o objetivo foi criar uma aplicação clara e inclusiva, onde qualquer pessoa se sinta confortável a navegar, encontrar informação e realizar as suas tarefas sem complicações.

### *3.2.2.4 Avaliação e Testes dos Protótipo*

Depois de finalizado o protótipo de alta fidelidade, passámos à fase de testes de avaliação. O objetivo foi perceber se a experiência de utilização correspondia ao que os utilizadores esperavam e se o design era realmente fácil e agradável de usar. Esta etapa foi importante para garantir que a aplicação fosse intuitiva, visualmente consistente e simples de navegar antes de avançar para a parte técnica.

#### **Objetivo da avaliação**

O principal objetivo destes testes foi verificar a clareza da navegação, a consistência gráfica entre ecrãs e a facilidade de utilização das principais funcionalidades, como o acesso ao perfil, à gestão de requisições e à consulta de métricas.

#### **Perfil dos avaliadores e método**

Na avaliação participaram colegas e potenciais utilizadores com perfis próximos do público-alvo, principalmente pessoas ligadas à área industrial ou com funções de gestão de equipas. Os testes foram feitos no Figma, através de sessões de usabilidade acompanhadas por observação direta, onde os participantes exploraram o protótipo e foram dando a sua opinião sobre cada ecrã e as ações que conseguiam realizar.

#### **Aspectos analisado**

Durante os testes, foram analisados vários aspectos da interface:

- Navegação e hierarquia das páginas: verificar se o utilizador percebia facilmente onde estava e para onde podia ir.
- Feedback visual e coerência gráfica: observar se os botões, ícones e cores transmitiam bem o seu propósito.
- Organização da informação: garantir que os conteúdos estavam bem estruturados e legíveis.
- Acessibilidade: confirmar se o contraste e o tamanho dos textos eram adequados.

## Resultados e conclusões

De forma geral, o feedback foi bastante positivo. Os utilizadores consideraram o protótipo organizado, coerente e fácil de usar, destacando como pontos fortes a simplicidade visual, a navegação intuitiva e a boa separação entre perfis. Foram apontadas apenas pequenas sugestões, como aumentar o destaque de certos botões de ação e ajustar ligeiramente o contraste em algumas secções. Estas observações foram tidas em conta para otimizar o design final, reforçando a clareza visual e a consistência entre páginas.

No geral, os testes confirmaram que o protótipo de alta fidelidade cumpre os objetivos de usabilidade e acessibilidade definidos no projeto, estando preparado para a fase de desenvolvimento funcional.

### 3.2.2.5 Análise dos Resultados Obtidos

A análise dos testes realizados sobre o protótipo de alta fidelidade permitiu avaliar o desempenho da interface e recolher opiniões úteis para o aperfeiçoamento da aplicação antes da implementação. De forma geral, o feedback foi muito positivo, confirmando que o design segue uma linha simples, coerente e fácil de utilizar.

Os participantes destacaram como principais pontos fortes a clareza na navegação, a organização das páginas e a consistência visual entre ecrãs. A presença de menus intuitivos, botões bem posicionados e ícones facilmente reconhecíveis foi elogiada, assim como o equilíbrio entre texto e elementos gráficos, que torna a aplicação visualmente agradável e funcional ao mesmo tempo. A separação das áreas por perfil foi também valorizada, por permitir que cada utilizador aceda apenas às opções relevantes ao seu papel.

Apesar do feedback positivo, foram identificados alguns pequenos pontos a melhorar. Alguns avaliadores referiram que certos botões de ação poderiam estar mais visíveis, e que seria útil reforçar o contraste em algumas secções da interface. Outras sugestões incluíram melhorar a hierarquia de informação em páginas mais densas e adicionar pequenos detalhes visuais para tornar o design ainda mais apelativo.

De forma geral, a equipa concluiu que o protótipo de alta fidelidade cumpre os requisitos definidos e apresenta um nível de usabilidade e acessibilidade elevado, pronto a servir de base à implementação funcional da aplicação. O processo de testes foi essencial para consolidar o design, identificar melhorias e garantir que o produto final será mais intuitivo, consistente e centrado no utilizador.

### 3.3 Identificação das Tecnologias para Implementação

Para o desenvolvimento frontend foi escolhida a biblioteca React (JavaScript/TypeScript). Esta tecnologia permite criar aplicações dinâmicas e modulares através de componentes reutilizáveis, para facilitar a manutenção e a escalabilidade da aplicação. A escolha do React deveu-se à sua flexibilidade, elevada performance e à ampla comunidade e documentação disponível.

O TailwindCSS foi adotado para a estilização da interface, proporcionando maior produtividade e controlo preciso do design, evitando folhas de estilo extensas e difíceis de manter. Complementarmente, o DaisyUI foi utilizado para fornecer componentes prontos baseados em Tailwind, o que agilizou o processo de criação do protótipo de alta-fidelidade e garantiu uma interface visual consistente, moderna e responsiva.

No lado do servidor, foi selecionado o **Java Spring Boot (com Gradle)**, acompanhado de **Spring Security** e **JWT (JSON Web Tokens)** para garantir a **autenticação e autorização seguras** dos utilizadores. O Spring Boot foi escolhido pela sua **maturidade, estabilidade e compatibilidade com sistemas empresariais**, bem como pela **facilidade de integração com APIs REST e ferramentas de segurança**.

A base de dados PostgreSQL foi selecionada pela sua fiabilidade, desempenho e suporte a operações complexas, sendo uma solução relacional open source amplamente utilizada em ambientes de produção. Para o alojamento (hosting), optou-se pela Oracle Cloud, que oferece elevada disponibilidade, bom desempenho e facilidade de integração com Java, assegurando a escalabilidade da aplicação conforme o crescimento do projeto.

Para a gestão de containers foi utilizado o Docker, permitindo que a aplicação seja executada de forma isolada, portátil e consistente em diferentes ambientes de desenvolvimento e produção. Esta abordagem simplifica o deployment e reduz potenciais problemas de compatibilidade.

Usaremos também o GitHub como plataforma para as diferentes versões do código, facilitando o trabalho colaborativo, o controlo de versões, essenciais para garantir qualidade continua no desenvolvimento.

### 3.4 Modelo de Dados

A estrutura de dados foi concebida de forma a garantir a organização, integridade e eficiência da informação ao longo de todas as operações do sistema. Esta secção descreve o processo de modelação da base de dados, desde a fase conceptual até à implementação física, assegurando que cada entidade reflete os requisitos funcionais definidos anteriormente.

A definição desta estrutura permite que a aplicação mantenha dados fiáveis e coerentes, reduzindo redundâncias e facilitando a manutenção futura. Assim, a base de dados do projeto constitui o núcleo técnico que sustenta o correto funcionamento da aplicação e a integridade de todas as interações entre utilizadores, empresas e administradores.

### 3.4.1 Diagrama de Conceitos

O diagrama de conceitos ilustra as entidades principais da aplicação e as relações existentes entre elas. No centro do modelo encontra-se a entidade Utilizador, da qual derivam os diferentes perfis do sistema, trabalhador, empresa, administrador e super administrador. Cada tipo de utilizador está associado a entidades que representam as suas funções no sistema.

De forma geral, o diagrama resumo a estrutura conceptual do sistema e a interação entre os diferentes perfis e elementos que compõem o TeamFoundry.

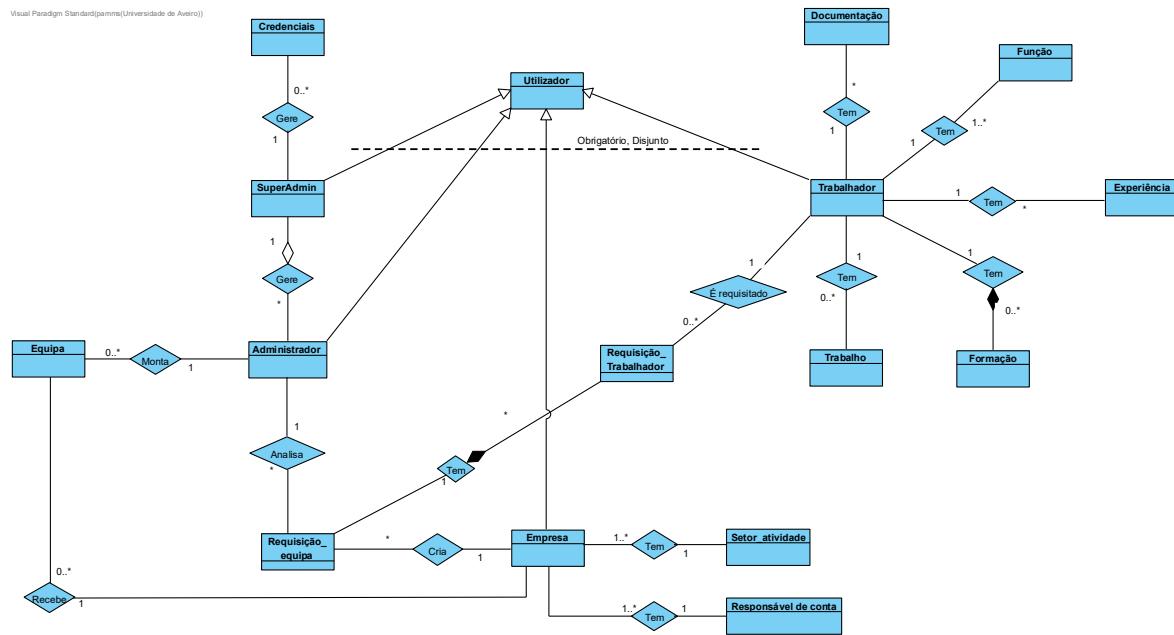
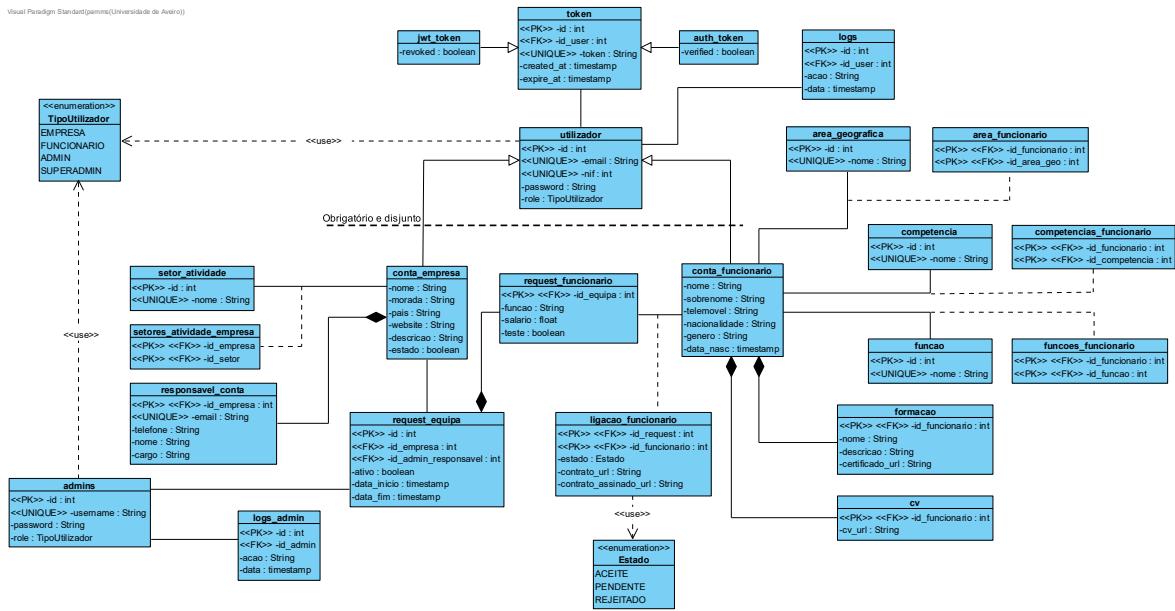


Figura 37-Diagrama de Conceitos

### 3.4.2 Diagrama de Entidades e Relações

A modelação lógica dos dados representa a estrutura base da informação que sustenta a aplicação. O diagrama de entidades e relações apresentado ilustra as principais entidades do sistema, os seus atributos e as relações que se estabelecem entre elas, definindo como os dados serão organizados antes da implementação física na base de dados.

Cada entidade – como Utilizador, Empresa, Funcionário, Equipa, Competência ou Formação – contém um conjunto de atributos específicos e respectivas chaves primárias (PK), que asseguram a identificação única de cada registo. As chaves estrangeiras (FK) criam ligações entre tabelas relacionadas, garantindo a integridade e coerência da informação.



*Figura 38-Diagrama de Entidades e Relações*

### 3.4.3 Esquema Físico

O modelo físico foi implementado em PostgreSQL, traduzindo o diagrama de entidades e relações para um conjunto de tabelas normalizadas, com chaves primárias/estrangeiras, restrições e índices que garantem integridade e desempenho.

Este modelo garante integridade referencial, consistência dos dados e um desempenho adequado às operações da aplicação, ficando pronto para a implementação e evolução futura.

Nesta subsecção deve ser apresentado o **modelo físico da base de dados**, ou seja, a implementação concreta em SQL ou noutra tecnologia de persistência escolhida.

## O que incluir:

- Estrutura das **tabelas** (atributos, tipos de dados, restrições).
  - Definição de **chaves primárias e estrangeiras**.
  - Índices e eventuais otimizações.
  - Ligações com tecnologias de persistência utilizadas no projeto (ex.: MySQL, PostgreSQL, MongoDB).

O objetivo é mostrar a **tradução prática do modelo de dados** num sistema de gestão de base de dados, pronto a ser utilizado na implementação da aplicação.

### 3.5 Distribuição e Estrutura do Site

A organização geral foi concebida de modo a garantir uma estrutura clara, modular e escalável, assegurando a coerência entre os requisitos funcionais definidos e a sua implementação técnica.

A aplicação foi dividida em páginas principais e áreas funcionais (como autenticação, gestão de utilizadores, e dashboards administrativos), de forma a otimizar a usabilidade e a navegação dos diferentes perfis de utilizador.

Ao nível técnico, a arquitetura baseia-se no padrão MVC (Model–View–Controller), o que permite uma separação lógica entre camadas, facilitando a manutenção, o trabalho colaborativo e futuras evoluções do projeto.

A camada Model gera os dados e integra-se com a base de dados PostgreSQL; a camada View corresponde à interface desenvolvida em React, estilizada com TailwindCSS e DaisyUI; e a camada Controller faz a ponte entre ambas, através de APIs REST implementadas em Spring Boot.

Esta estrutura modular garante que o sistema cumpre os requisitos funcionais principais — autenticação segura, gestão de perfis e equipas, e visualização de dados — mantendo ao mesmo tempo uma arquitetura flexível, escalável e fácil de manter.

### 3.5.1 Distribuição do Website

A aplicação foi estruturada de forma lógica e intuitiva, com o objetivo de proporcionar uma navegação simples, fluida e acessível a todos os tipos de utilizadores - funcionários, empresas, administradores e super-admin. A organização das páginas segue uma hierarquia clara, garantindo que o utilizador encontra facilmente as funcionalidades e informações de que necessita.

As principais páginas do sistema são:

- Login - permite o acesso seguro à plataforma, autenticando os utilizadores através de credenciais válidas (email e palavra-passe).
- Registo (Register) - página destinada à criação de novas contas, diferenciando entre perfis de funcionário e empresa, garantindo uma entrada personalizada no sistema.
- Homepage - serve como ponto de entrada principal, apresentando uma visão geral do projeto, acesso rápido a secções relevantes e ligações diretas para o login e registo.
- Dicas - área dedicada à apresentação de informações úteis e boas práticas sobre o uso da plataforma, apoiando novos utilizadores e melhorando a experiência de navegação.
- Sobre Nós - página institucional que descreve o objetivo, a missão e a equipa responsável pelo desenvolvimento do projeto TeamFoundry, reforçando a credibilidade e a identidade da aplicação.
- Área Pessoal - espaço reservado a cada utilizador, onde é possível gerir o perfil, visualizar requisições, atualizar dados e consultar o histórico de atividades.
- Dashboard do Administrador / Super-Admin - interface de gestão que permite monitorizar o sistema, gerir utilizadores e empresas, e acompanhar o funcionamento global da aplicação.

A organização visual do website baseia-se numa estrutura comum a todas as páginas, composta por:

- Menu de navegação (navbar) fixo no topo, garantindo acesso rápido às áreas principais.
- Corpo principal (content area) que apresenta os conteúdos e funcionalidades de forma clara e bem segmentada.
- Rodapé (footer) com informações complementares, como contactos, política de privacidade e direitos reservados.

O design segue critérios de usabilidade e acessibilidade, recorrendo a layouts responsivos, contrastes adequados de cor, e componentes reutilizáveis desenvolvidos com React e TailwindCSS, garantindo uma experiência consistente tanto em dispositivos desktop como mobile.

Esta estrutura lógica e visual assegura que a aplicação mantém uma navegação intuitiva e eficiente, permitindo que cada utilizador encontre rapidamente o que procura e interaja com o sistema de forma clara e segura.

### 3.5.2 Estrutura Implementada (Padrão MVC)

A arquitetura da aplicação web foi desenvolvida com base no padrão MVC, garantindo uma estrutura organizada, modular e de fácil manutenção. Este modelo separa claramente as responsabilidades de cada camada, promovendo uma maior escalabilidade e facilitando futuras evoluções do sistema.

- Model

A camada Model é responsável pela gestão da lógica de dados e pela comunicação com base de dados PostgreSQL. Aqui são definidos os modelos de dados e as entidades que representam as tabelas do sistema (como utilizador, empresa, funcionário, requisição, entre outras).

O Spring Boot gera esta camada através do String Data JPA, que simplifica a criação de repositórios e a execução de operações de persistência.

A autenticação e autorização dos utilizadores é garantida por Spring Security em conjunto com JWT, assegurando a integridade e a proteção das comunicações.

- View

A camada View corresponde à interface gráfica da aplicação, desenvolvida em React (JavaScript/TypeScript).

Esta tecnologia permite criar uma interface modular e dinâmica, baseada em componentes reutilizáveis, o que facilita a manutenção e garante uma experiência de utilização fluida. A estilização é assegurada por TailwindCSS, que oferece um controlo detalhado sobre o design e evita a criação de folhas de estilo extensas.

Adicionalmente, foi integrado o DaisyUI, que disponibiliza componentes visuais pré-definidos e consistentes, acelerando o processo de desenvolvimento e assegurando uma interface moderna, limpa e responsiva.

- Controller

A camada Controller atua como intermediária entre o frontend (View) e a lógica de negócio (Model). É nesta camada que se processam as requisições HTTP vindas da interface, que são depois tratadas

pelos controladores do Spring Boot, responsáveis por aceder aos serviços e devolver as respostas através das APIs REST.

Esta separação permite uma comunicação clara entre cliente e servidor, garantindo que cada parte do sistema mantém responsabilidades bem definidas.

Em suma, a adoção do padrão MVC, aliada a tecnologias modernas e bem integradas, proporcionando uma arquitetura robusta, escalável e de fácil manutenção adequada ao contexto empresarial e académico do projeto.

### 3.5.3 Mapa de Navegação

O mapa de navegação apresentado na figura seguinte ilustra a estrutura completa da aplicação e o percurso seguido por cada perfil de utilizador dentro da plataforma. O diagrama mostra a transição entre a homepage sem login, onde o utilizador tem acesso a informações gerais sobre o projeto (slogan, áreas de atuação e parceiros), e a homepage com login, que serve como ponto de entrada para as áreas internas do sistema.

Após o login, cada perfil é direcionado para a sua área correspondente:

- Funcionário: pode aceder ao perfil pessoal, consultar ofertas e documentos, acompanhar os próximos passos e atualizar as suas preferências.
- Empresa: tem acesso à gestão de requisições de equipas, visualização de equipas formadas e acompanhamento de trabalhos ativos, finalizados ou pendentes.
- Administrador: entra no painel administrativo, onde pode gerir equipas, perfis de utilizadores, e verificar métricas através do dashboard.
- Super administrador: dispõe de um painel com funcionalidades avançadas, isto inclui a gestão de credenciais, controlo de trabalhos e consulta de métricas globais.

O mapa reflete uma estrutura hierárquica clara e intuitiva, garante que cada tipo de utilizador encontra facilmente as funcionalidades relevantes ao seu papel no sistema.

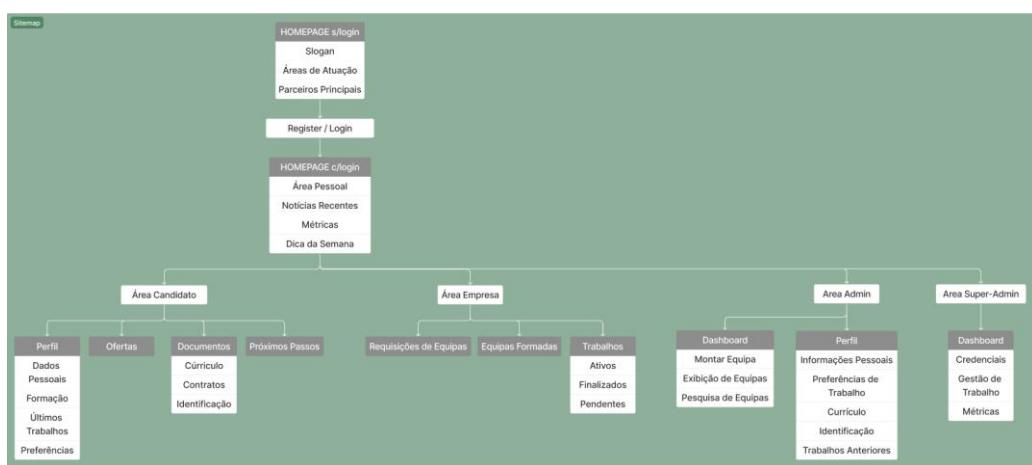


Figura 39-Mapa de Navegação

## 4. Implementação da Aplicação

### 4.1 Camada de Apresentação

Nesta secção deve ser descrita a **camada de apresentação (front-end)** da aplicação, ou seja, a parte visível com a qual o utilizador interage diretamente.

O que incluir:

- **Tecnologias utilizadas** (ex.: HTML, CSS, JavaScript, frameworks responsivas).
- A forma como foi implementado o **design definido nos protótipos** (cores, tipografia, estilos visuais, responsividade).
- Estrutura dos **layouts e componentes reutilizáveis** (menus, botões, formulários, tabelas, dashboards).
- **Acessibilidade e usabilidade** na interface (contrastos, textos alternativos, navegação simples).
- Adaptação da apresentação para diferentes **dispositivos e resoluções** (design responsivo).

O objetivo é demonstrar como o design previamente definido foi convertido numa **interface funcional, acessível e intuitiva**, garantindo uma boa experiência de utilização.

### 4.2 Camada de Lógica

Nesta secção deve ser descrita a **camada de lógica de negócio (back-end)**, responsável pelo processamento das regras e funcionalidades da aplicação.

O que incluir:

- **Tecnologias e linguagens utilizadas** (ex.: PHP, Node.js, Java, frameworks).
- Implementação das **regras de negócio** (ex.: validações, cálculos, controlo de permissões).
- Gestão da ligação entre a camada de apresentação e a base de dados.
- Estruturação do **código e organização em módulos/classes**.
- Como esta camada garante a **segurança, consistência e integridade** do sistema.

O objetivo é explicar como a aplicação processa e gera a lógica que suporta as funcionalidades visíveis no front-end.

### 4.3 Arquitetura e Organização da Aplicação

Nesta secção deve ser apresentada a forma como a aplicação está **arquitetada e organizada internamente**, mostrando como os diferentes componentes se interligam.

O que incluir:

- A **arquitetura escolhida** (ex.: padrão MVC, camadas, micro-serviços).
- A forma como foram **distribuídos os módulos** (ex.: autenticação, gestão de utilizadores, relatórios).
- Estrutura das **pastas e ficheiros** do projeto, explicando a sua função.
- Vantagens da arquitetura escolhida em termos de **organização, escalabilidade e manutenção**.
- Eventuais **ferramentas ou frameworks** usadas para apoiar a organização (ex.: Laravel, Spring, Express).

O objetivo é evidenciar que a aplicação foi construída sobre uma **estrutura sólida e bem organizada**, facilitando a compreensão, evolução e manutenção futura.

## 5. Testes e Validação

### 5.1 Estratégia de Testes e Validações

Nesta secção deve ser explicada a **abordagem seguida para garantir a qualidade da aplicação**, através de testes e processos de validação.

O que incluir:

- **Objetivo da estratégia de testes** (detetar erros, validar requisitos, assegurar usabilidade e acessibilidade).
- **Tipos de testes aplicados:**
  - *Testes funcionais* → verificar se as funcionalidades cumprem os requisitos definidos.
  - *Testes de usabilidade* → avaliar a facilidade de utilização da aplicação por utilizadores reais.
  - *Testes de compatibilidade* → verificar funcionamento em diferentes navegadores, dispositivos e sistemas operativos.
  - *Testes de acessibilidade* → garantir conformidade com normas como WCAG.
  - *Testes de desempenho* → tempo de resposta e comportamento sob carga (se aplicável).
- **Ferramentas utilizadas** (ex.: validadores W3C, Lighthouse, Postman, frameworks de testes).
- **Critérios de aceitação** → como a equipa define que um teste foi bem-sucedido.

O objetivo é mostrar que a validação não é feita apenas no final, mas é parte integrante do processo de desenvolvimento, garantindo que a aplicação entregue é **funcional, acessível e fiável**.

## 5.2 Testes Unitários

Nesta secção devem ser descritos os **testes unitários** realizados, que verificam o funcionamento correto de pequenas partes isoladas do código (funções, métodos, componentes).

O que incluir:

- O **objetivo** dos testes unitários (garantir que cada componente funciona de forma independente).
- As **ferramentas e frameworks** utilizadas (ex.: JUnit, PHPUnit, Jest, Mocha).
- Exemplos de **funcionalidades testadas** (ex.: autenticação, validação de formulários, operações sobre a base de dados).
- A forma como os testes foram integrados no **ciclo de desenvolvimento**.
- **Resultados** obtidos e sua influência na correção de erros.

O objetivo é mostrar que o código foi verificado desde cedo, reduzindo falhas e aumentando a confiabilidade do sistema.

## 5.3 Testes Estruturais

Nesta secção devem ser descritos os **testes unitários** realizados, que verificam o funcionamento correto de pequenas partes isoladas do código (funções, métodos, componentes).

O que incluir:

- O **objetivo** dos testes unitários (garantir que cada componente funciona de forma independente).
- As **ferramentas e frameworks** utilizadas (ex.: JUnit, PHPUnit, Jest, Mocha).
- Exemplos de **funcionalidades testadas** (ex.: autenticação, validação de formulários, operações sobre a base de dados).
- A forma como os testes foram integrados no **ciclo de desenvolvimento**.
- **Resultados** obtidos e sua influência na correção de erros.

O objetivo é mostrar que o código foi verificado desde cedo, reduzindo falhas e aumentando a confiabilidade do sistema.

## 5.4 Validação da Interface e Usabilidade

Nesta secção deve ser descrito o processo de **validação da interface gráfica** e da **usabilidade da aplicação**, garantindo que a experiência do utilizador é intuitiva, eficiente e agradável.

O que incluir:

- **Objetivos da validação** (avaliar clareza, simplicidade, consistência e acessibilidade).
- **Métodos utilizados** (ex.: testes com utilizadores, questionários, observação direta, heurísticas de Nielsen).
- **Aspectos avaliados:**
  - Clareza na disposição de menus, botões e formulários.
  - Facilidade de navegação entre páginas.
  - Consistência de design e feedback visual.
  - Tempo e esforço necessários para concluir tarefas.
- **Resultados obtidos** (pontos fortes, dificuldades encontradas).
- **Ajustes propostos** para melhorar a experiência do utilizador.

O objetivo é demonstrar que a aplicação foi concebida de forma **centrada no utilizador**, atendendo às suas necessidades reais.

## 5.5 Verificação do Cumprimento dos Requisitos

Nesta secção deve ser feita a **verificação final do sistema em relação aos requisitos definidos anteriormente**, garantindo que tudo o que foi planeado foi efetivamente implementado.

O que incluir:

- **Listagem dos requisitos funcionais** e confirmação do seu cumprimento.
- **Listagem dos requisitos não funcionais** (performance, segurança, usabilidade, compatibilidade, acessibilidade) e confirmação do seu cumprimento.
- Indicação de **quais requisitos foram totalmente, parcialmente ou não implementados**, com justificação.
- Eventuais **alterações ou adaptações** face ao levantamento inicial.

O objetivo é validar que a aplicação responde de forma consistente às **necessidades do cliente e dos utilizadores**, comprovando a qualidade e completude do trabalho desenvolvido.

## 6. Análise dos Resultados

### 6.1 Verificação do Cumprimento dos Requisitos Funcionais

Nesta secção deve ser analisado em detalhe se os **requisitos funcionais definidos no levantamento** foram efetivamente cumpridos pela aplicação final.

O que incluir:

- Uma **listagem dos requisitos funcionais** (UCs) levantados no início.
- Para cada requisito, indicar se foi:
  - **Totalmente implementado;**
  - **Parcialmente implementado;**
  - **Não implementado** (com justificação).
- Exemplos práticos ou evidências de funcionamento (prints, outputs, demonstrações).

O objetivo é comprovar que a aplicação final cumpre o que foi definido no **escopo funcional** do projeto.

### 6.2 Análise Crítica dos Resultados

Nesta secção deve ser feita uma **reflexão crítica** sobre os resultados obtidos, avaliando em que medida o projeto atingiu os seus objetivos e que pontos poderiam ser melhorados.

O que incluir:

- Comparação entre o **planeado** e o **executado** (cronograma, recursos, funcionalidades).
- **Aspectos positivos** alcançados (ex.: funcionalidades completas, boa performance, feedback positivo dos utilizadores).
- **Limitações ou dificuldades** enfrentadas (ex.: falta de tempo, restrições técnicas, requisitos alterados).
- Sugestões de **melhorias ou trabalhos futuros** (funcionalidades adicionais, otimizações técnicas, maior abrangência).

O objetivo é mostrar capacidade de **autoavaliação**, destacando tanto as conquistas como as fragilidades, de forma honesta e construtiva.

[apresentar os resultados e comentá-los]

[análise do cumprimento dos requisitos. Uma abordagem possível é apresentar novamente a tabela de requisitos e indicar numa coluna se faz ou não faz]

## 7. Reflexão e Autocrítica

### 7.1 Atividades Desenvolvidas

Nesta secção deve ser feita uma **reflexão sobre as atividades realizadas ao longo do projeto**, destacando o contributo de cada uma para o resultado final.

O que incluir:

- Breve resumo das **principais atividades desenvolvidas** (ex.: levantamento de requisitos, prototipagem, implementação, testes).
- A **importância de cada fase** para a evolução do projeto.
- Referência a eventuais **dificuldades superadas** durante o processo.
- O que correu **bem** e o que poderia ter sido **melhor planeado ou executado**.

O objetivo é mostrar uma visão crítica sobre o percurso realizado, indo além da simples descrição técnica.

### 7.2 Estratégias de Trabalho Adotadas

Nesta secção devem ser analisadas as **estratégias de organização e gestão do trabalho utilizadas pela equipa**, bem como a sua eficácia.

O que incluir:

- **Modelo de trabalho** seguido (ex.: reuniões semanais, divisão de tarefas, uso de metodologias ágeis ou em cascata).
- **Ferramentas de apoio** utilizadas (ex.: Trello, GitHub, Google Drive).
- Grau de **colaboração e comunicação interna** da equipa.
- **Pontos fortes** das estratégias aplicadas (ex.: boa coordenação, flexibilidade).
- **Aspetos a melhorar** em futuros projetos (ex.: melhor gestão de tempo, mais testes intermédios).

O objetivo é refletir sobre como a organização do trabalho impactou o sucesso do projeto, evidenciando aprendizagens para experiências futuras.

### 7.3 Tecnologias Utilizadas

Nesta secção deve ser feita uma reflexão crítica sobre as **tecnologias escolhidas e utilizadas no projeto**, avaliando a sua adequação e impacto no desenvolvimento.

O que incluir:

- **Lista das principais tecnologias adotadas**, divididas por camadas (front-end, back-end, base de dados, ferramentas de design, plataformas de gestão e colaboração).
- Justificação da **escolha dessas tecnologias** em função dos requisitos do projeto.
- **Vantagens observadas** durante a utilização (ex.: facilidade de uso, documentação disponível, performance).
- **Limitações ou dificuldades** encontradas (ex.: curva de aprendizagem, problemas de compatibilidade, falta de funcionalidades).
- Considerações sobre se as tecnologias foram a **melhor escolha possível** ou se poderiam ter sido exploradas alternativas.

O objetivo é demonstrar consciência crítica relativamente às tecnologias utilizadas, mostrando que a sua seleção foi pensada e que a equipa aprendeu com a experiência.

## 7.4 Planeamento vs Execução

Nesta secção deve ser feita uma **comparação entre o que foi planeado inicialmente e o que realmente foi executado** ao longo do projeto.

O que incluir:

- Diferenças entre o **cronograma previsto (planeamento)** e o **cronograma final (execução)**.
- Tarefas que **cumpriram os prazos** e aquelas que **sofreram atrasos ou antecipações**.
- **Causas** dessas diferenças (ex.: falta de tempo, dificuldades técnicas, alterações de requisitos).
- Impacto dessas variações no **resultado final** do projeto.

O objetivo é mostrar que houve uma reflexão crítica sobre a gestão do tempo e recursos, identificando aprendizagens para futuros projetos.

## 7.5 Sugestões de Melhoria

Nesta secção devem ser apresentadas **propostas de melhoria**, tanto para o sistema desenvolvido como para a forma de trabalhar em projetos semelhantes no futuro.

O que incluir:

- Melhorias **técnicas** na aplicação (novas funcionalidades, maior desempenho, reforço da segurança, acessibilidade, escalabilidade).
- Melhorias no **processo de trabalho** (ex.: mais testes intermédios, maior contacto com o cliente, melhor gestão de tarefas).

- **Ferramentas ou metodologias alternativas** que poderiam ter facilitado o desenvolvimento.

- Recomendações para **futuros trabalhos** que possam dar continuidade ou expandir o projeto.

O objetivo é evidenciar capacidade de **autoavaliação e visão de futuro**, mostrando como o projeto poderia evoluir e como a equipa pode melhorar a sua forma de trabalhar.

## 7.6 Síntese das Experiências

Nesta secção deve ser feita uma **recapitulação das experiências vividas ao longo do projeto**, tanto a nível técnico como organizacional.

O que incluir:

- Principais **aprendizagens adquiridas** (novas tecnologias, metodologias de trabalho, competências de comunicação).
- **Desafios enfrentados** e como foram superados.
- Momentos de **colaboração e trabalho em equipa** mais relevantes.
- Impacto global do projeto na **evolução académica e profissional** dos participantes.

O objetivo é sintetizar as experiências de forma crítica e reflexiva, valorizando os ganhos obtidos para além do produto final.

## 7.7 Perspetivas Futuras

Nesta secção devem ser apresentadas as **possibilidades de evolução do projeto** após a sua conclusão académica.

O que incluir:

- Funcionalidades que poderiam ser adicionadas ou melhoradas numa versão futura.
- Potencial de **aplicação prática** da solução no mercado ou em contextos reais.
- Oportunidades de **integração com outras tecnologias** ou sistemas.
- Continuidade do trabalho em contextos académicos ou profissionais (ex.: dissertações, spin-offs, startups).

O objetivo é demonstrar visão de futuro, mostrando que o projeto tem **potencial para crescer, evoluir e ser aplicado em cenários mais amplos**.

## 8. Conclusão

A conclusão deve retomar o tema/assunto principal do trabalho, abordado na introdução, destacando os aspectos essenciais que foram desenvolvidos no decurso do trabalho e integrando uma avaliação do trabalho realizado e do seu contributo para o aprofundamento dos conhecimentos relativos à área de formação. É também importante que realce as competências desenvolvidas e conhecimentos adquiridos (incluindo sobre conteúdos abordados na UC Metodologias e Gestão de Projetos, nomeadamente a temática do trabalho em equipa, gestão de conflitos, comunicação, etc. Por exemplo, deve ser feita uma reflexão que indique se houve sinergia positiva, se os membros do grupo trabalharam de forma articulada e comprometida para atingir o objetivo final do grupo, houve complementariedade e apoio entre os membros? Que aprendizagens podem ser retiradas desta experiência e de que forma esta poderá ser uma mais-valia para circunstâncias similares e em trabalhos futuros? Etc.).

Incluir uma reflexão crítica sobre o que foi planeado e o que foi executado.

Sempre que possível a conclusão deve integrar recomendações e propostas de ação e não deve ser introduzida informação que não decorra do conteúdo do corpo do trabalho.

# Referências Bibliográficas

Uma referência bibliográfica é um conjunto de elementos que permite a identificação no todo ou em parte, de documentos impressos ou registados em diversos tipos de suportes, e define uma sequência normalizada na apresentação desses mesmos elementos.

Os procedimentos e a apresentação gráfica das referências bibliográficas deverão seguir a norma do Institute of Electrical and Electronics Engineers (IEEE), cujas orientações se encontram disponíveis nos tutoriais existentes no sítio web da Biblioteca da UA em <https://www.ua.pt/sbidm/biblioteca/ReadObject.aspx?obj=36605>

Recomenda-se a utilização do gestor de referências bibliográficas Mendeley Desktop. Pode consultar a apresentação disponibilizada pela Biblioteca da UA sobre como instalar e usar o Mendeley (disponível em <http://portal.doc.ua.pt/gerirRefBibliogMendeley.pdf>).

# Apêndices

Os anexos/apêndices devem iniciar com uma página dedicada exclusivamente à sua apresentação. Esta página pode conter somente o título.

**ANEXOS/APÊNDICES.** Cada anexo/apêndice deve ser apresentado numa página distinta, e fazer-se acompanhar de um número e de um título explicativo na parte superior da primeira página. A ordenação dos anexos/apêndices segue a ordem segundo a qual são referenciados no texto, utilizando-se um número de série árabe.

Não confunda Anexos com Apêndices:

**Apêndices:** englobam materiais trabalhados e elaborados pelo autor do trabalho/relatório, tais como tabelas, ilustrações, brochuras, organogramas, manuais, cartas...

**Anexos:** englobam documentos auxiliares que não foram elaborados pelos autores, mas que facilitam a compreensão de algum assunto/temática abordada no trabalho.

Só deve ser colocada em anexo/apêndice informação à qual se faz referência ao longo do corpo do trabalho

## Apêndice – Protótipo de Alta Fidelidade

Este apêndice apresenta o protótipo de alta fidelidade completo da aplicação, desenvolvido na ferramenta Figma. As interfaces mostram o design final da aplicação, incluindo cores, ícones, tipografia, estrutura e navegação para os diferentes perfis de utilizador. As imagens seguintes representam os ecrãs finais da plataforma, complementando os exemplos descritos no ponto 3.2.2 do relatório.

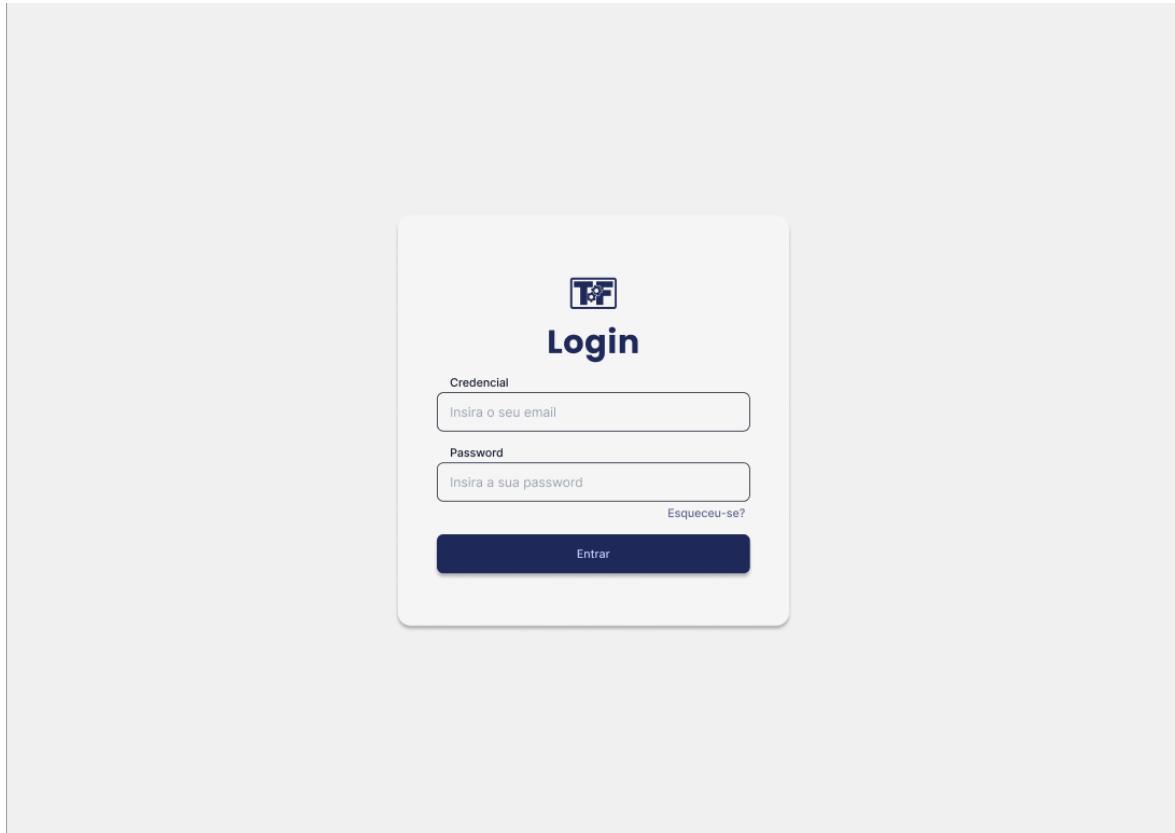


Figura 40 - Protótipo de Alta Fidelidade - Página de Login de Admin