

# **Plataforma de Gestão Integrada de Pedidos (PGIP)**

**Versão Definitiva Após Defesa Pública**

**Emanuel Grilo Oliveira**

Relatório de Estágio para obtenção do Grau de Mestre em  
**Engenharia Informática**  
(2<sup>o</sup> ciclo de estudos)

Orientador: Prof. Doutor Pedro Ricardo Morais Inácio  
Orientador da *Latitude*: Eng. João José Teles Gouveia

**Covilhã, Julho de 2025**



## **Declaração de Integridade**

Eu, Emanuel Grilo Oliveira, que abaixo assino, estudante com o número de inscrição M13780 de Engenharia Informática da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o Código de Integridades da Universidade da Beira Interior.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 10/07/2025



## Dedicatória

Dedico este relatório de Estágio à minha família, que sempre me apoiou incondicionalmente e continua a ser o meu pilar. Aos amigos que, apesar de não estarem presentes diariamente, me incentivaram a concluir esta etapa, seja com palavras de incentivo ou com avisos mais firmes. Ao meu orientador, o Prof. Doutor Pedro Inácio, pelo seu apoio constante e pela confiança em mim ao longo deste percurso académico. Ao meu orientador na *Latitudde*, João Gouveia, pela atenção, ajuda e suporte prestados ao longo destes meses.



## Agradecimentos

Ao longo deste percurso, senti muitas vezes que estava a escalar uma montanha. No início, cheio de energia e determinação, pensava apenas no objetivo final: alcançar o topo. No entanto, à medida que a subida avançava, a montanha revelou-se mais desafiante do que imaginava. Demorei a perceber que não conseguiria subi-la sozinho, e, quando finalmente me apercebi disso, parecia que estava numa situação ainda mais difícil do que quando comecei. Foram nos momentos mais difíceis que o verdadeiro valor da jornada se revelou. A minha família foi como um ponto de apoio seguro, sempre presente para me dar força quando as dificuldades pareciam insuperáveis. Aos meus amigos, agradeço por nunca me deixarem cair e por fixarem o primeiro espigão onde pude prender o arnês, sempre com um sorriso e uma palavra de incentivo.

E, claro, não posso deixar de reconhecer o papel essencial dos meus orientadores, tanto da universidade como da *Latitudde*. Foram como guias experientes que me apontaram os melhores caminhos e me ensinaram a enfrentar os desafios com confiança e determinação.

Agora que chego ao topo e olho para trás, percebo que esta jornada foi muito mais do que alcançar um destino. Foi sobre as pessoas que caminharam comigo, que me inspiraram e que me ajudaram a chegar até aqui. Porque, no final, o valor da montanha não está apenas no que encontramos no cume, mas na força, na aprendizagem e nas ligações que construímos ao longo do caminho.

Obrigado a todos que fizeram parte desta escalada.



## Resumo

Este relatório descreve de forma detalhada o percurso do estágio curricular realizado no âmbito do projeto Plataforma de Gestão Integrada de Pedidos, desenvolvido para o *Cliente* (por motivos de confidencialidade, a entidade para a qual foi elaborado este projeto será referido neste relatório apenas como Cliente). O documento está estruturado de forma a apresentar o contexto do projeto, as tecnologias utilizadas, os requisitos funcionais e não funcionais definidos, a metodologia de desenvolvimento adotada, as várias atividades desenvolvidas, funcionalidades implementadas, os testes realizados e, por fim, uma reflexão sobre os resultados obtidos e aprendizagens adquiridas.

O **Plataforma de Gestão Integrada de Pedidos** é uma plataforma web destinada à gestão integrada de pedidos, abrangendo os processos de agendamento, aprovação e execução de serviços. A solução permite a interação entre diferentes intervenientes, como requerentes e fornecedores, e integra-se com sistemas externos, como o SAP e o *Microsoft Azure*.

O foco principal do estágio incidiu sobre o desenvolvimento da interface de utilizador (*front-end*) da plataforma, com recurso à framework *Angular*, permitindo ao aluno adquirir competências na criação de interfaces web responsivas e funcionais. Embora a componente de *back-end* tenha sido maioritariamente desenvolvida por outros membros da equipa, o estudante contribuiu também na integração e validação conjunta entre interfaces.

## Palavras-chave

Análise de riscos, *front-end*, gestão de pedidos, interface de utilizador, *Latitudde*, *Microsoft Azure*, plataformas web, SAP, Plataforma de Gestão Integrada de Pedidos.



## Abstract

This report provides a detailed account of the curricular internship carried out within the scope of the *Plataforma de Gestão Integrada de Pedidos* project, developed for the *Client* (for confidentiality reasons, the entity for which this project was developed will be referred to throughout this report simply as the Client). The document is structured to present the project context, the technologies used, the defined functional and non-functional requirements, the adopted development methodology, the various activities carried out, the implemented features, the tests performed, and, finally, a reflection on the results achieved and the lessons learned.

**Plataforma de Gestão Integrada de Pedidos** is a web platform designed for the integrated management of service requests, encompassing the processes of scheduling, approval, and execution. The solution facilitates interaction between different stakeholders, such as requesters and suppliers, and integrates with external systems such as SAP and *Microsoft Azure*.

The main focus of the internship was the development of the platform's user interface (*front-end*), using the *Angular* framework, which enabled the student to acquire skills in creating responsive and functional web interfaces. Although the *back-end* component was primarily developed by other members of the team, the student also contributed to the integration and joint validation of the interfaces.

## Keywords

*Front-end*, interface de utilizador, *Latitudde*, *Microsoft Azure*, *request management*, *risk analysis*, SAP, *Plataforma de Gestão Integrada de Pedidos*, *user interface*, *web platforms*.



# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Caracterização da Empresa . . . . .	1
1.2	Caracterização do Estagiário . . . . .	1
1.3	Apresentação e Objetivos do Estágio . . . . .	1
1.4	Organização do Documento . . . . .	2
<b>2</b>	<b>Estado da Arte</b>	<b>5</b>
2.1	Introdução . . . . .	5
2.2	Enquadramento do Projeto . . . . .	5
2.3	Soluções e Tecnologias Relacionadas . . . . .	6
2.3.1	<i>Framework</i> de Desenvolvimento Angular . . . . .	6
2.3.2	Integração com o Sistema SAP . . . . .	6
2.3.3	Metodologias de Desenvolvimento . . . . .	6
2.3.4	Segurança em Aplicações Web . . . . .	7
2.4	Análise Comparativa . . . . .	7
2.4.1	Robustez e Escalabilidade . . . . .	7
2.4.2	Manutenção e Curva de Aprendizagem . . . . .	7
2.4.3	Tamanho do Bundle . . . . .	8
2.4.4	Flexibilidade . . . . .	8
2.5	Conclusão . . . . .	8
<b>3</b>	<b>Tecnologias e Ferramentas Utilizadas</b>	<b>9</b>
3.1	Introdução . . . . .	9
3.2	Ferramentas . . . . .	9
3.2.1	Visual Studio Code . . . . .	9
3.2.2	MySQL . . . . .	9
3.2.3	Figma . . . . .	10
3.3	Linguagens . . . . .	10
3.3.1	HTML . . . . .	11
3.3.2	CSS . . . . .	11
3.3.3	TypeScript . . . . .	11
3.3.4	PHP . . . . .	11
3.4	<i>Frameworks</i> . . . . .	11
3.4.1	Angular . . . . .	12
3.4.2	Laravel . . . . .	12
3.5	Conclusão . . . . .	13

<b>4</b>	<b>Planificação</b>	<b>15</b>
4.1	Introdução . . . . .	15
4.2	Enumeração das Tarefas . . . . .	15
4.3	Plano de Execução . . . . .	16
4.4	Planificação do Trabalho . . . . .	16
4.5	Requisitos do Sistema Desenvolvido . . . . .	17
4.5.1	Requisitos Funcionais . . . . .	17
4.5.2	Requisitos Técnicos . . . . .	18
4.5.3	Requisitos Não Funcionais . . . . .	18
4.5.4	Priorização dos Requisitos . . . . .	19
4.5.5	Suma de Requisitos . . . . .	19
4.6	Análise de Riscos e Plano de Mitigação . . . . .	19
4.6.1	Risco 1 - Sobrecarga de Tarefas . . . . .	19
4.6.2	Risco 2 - Dependência de Ferramentas Não <i>Open-Source</i> . . . . .	19
4.6.3	Risco 3 - Alterações nos Requisitos . . . . .	20
4.6.4	Risco 4 - Dificuldades de Comunicação . . . . .	20
4.6.5	Resumo da Análise de Riscos . . . . .	20
4.7	Execução do Plano de Trabalho . . . . .	21
4.8	Planeamento e <i>Design</i> . . . . .	22
4.8.1	Descrição das Entidades do Modelo de Dados . . . . .	22
4.8.2	Relações no Modelo de Dados . . . . .	24
4.8.3	Design da Interface do Utilizador . . . . .	25
4.9	Conclusão . . . . .	26
<b>5</b>	<b>Implementação e Testes</b>	<b>27</b>
5.1	Introdução . . . . .	27
5.2	Visão Geral do Sistema . . . . .	27
5.3	Arquitetura do Sistema . . . . .	27
5.4	Relacionamento entre <i>User Stories</i> e Funcionalidades Implementadas . . . . .	28
5.5	Fluxo de Utilização da Aplicação . . . . .	28
5.5.1	Criação de Pedido . . . . .	28
5.5.2	Submissão do Pedido . . . . .	28
5.5.3	Aprovação do Pedido . . . . .	28
5.5.4	Execução do Pedido . . . . .	29
5.5.5	Conclusão do Pedido . . . . .	29
5.6	Funcionalidades Desenvolvidas . . . . .	29
5.6.1	Interface do Utilizador . . . . .	29
5.6.2	Principais Funcionalidades Desenvolvidas pelo Estagiário . . . . .	30
5.7	Implementação Técnica . . . . .	30
5.7.1	Descrição do <i>Happy Flow</i> . . . . .	31
5.7.2	Considerações de Segurança . . . . .	31
5.8	Testes e Validação . . . . .	32
5.9	Conclusão . . . . .	33

<b>6 Conclusão</b>	<b>35</b>
6.1 Introdução . . . . .	35
6.2 Conclusões Principais . . . . .	35
6.3 Objetivos Iniciais e Resultados Obtidos . . . . .	36
6.4 Dificuldades e Aprendizagens . . . . .	36
6.4.1 Dificuldades . . . . .	36
6.4.2 Aprendizagens . . . . .	36
6.5 Reflexão sobre o Estágio . . . . .	37
6.6 Possíveis Melhorias Futuras e Continuação do Projeto . . . . .	37
6.7 Conclusão . . . . .	38

<b>Bibliografia</b>	<b>39</b>
---------------------	-----------

<b>A Anexos</b>	<b>41</b>
A.1 <i>Mockups</i> . . . . .	41
A.1.1 <i>Happy Flow</i> . . . . .	43
A.2 Modelo de Dados . . . . .	56
A.2.1 <i>User Stories</i> . . . . .	57
A.2.2 Arquitetura do Sistema . . . . .	58
A.2.3 Fluxo de Utilização da Aplicação . . . . .	59
A.3 Tarefas Realizadas . . . . .	60
A.3.1 <i>Tarefas Realizadas</i> . . . . .	60



## Lista de Figuras

4.1	Diagrama de <i>Gantt</i> da estimativa do plano de trabalho. . . . .	17
5.1	Página de <i>Login</i> . . . . .	29
A.1	Página Principal - Pedidos (Visão Administrador). Este <i>mockup</i> demonstra a visão administrativa, com a visão de todos os <i>sub-menus</i> , permitindo a gestão centralizada de pedidos e o acompanhamento do seu estado. . . . .	41
A.2	Formulário para Adicionar Novo Fornecedor. Este <i>mockup</i> detalha o processo de registo de novos fornecedores no sistema, garantindo a consistência dos dados recolhidos. . . . .	42
A.3	Lista de pedidos disponíveis para o Requerente, com possibilidade de criar novo pedido. . . . .	43
A.4	Seleção da tipologia do serviço a requisitar, por parte do Requerente. . . . .	43
A.5	Preenchimento dos dados detalhados do pedido, como quantidade, contentor e operação. . . . .	44
A.6	Confirmação final de todos os dados do pedido antes da submissão. . . . .	44
A.7	Notificação de sucesso após a submissão do pedido. . . . .	45
A.8	Visualização do estado atualizado do pedido submetido. . . . .	45
A.9	Lista de pedidos disponíveis para o Requerente, com possibilidade de ver o pedido criado. . . . .	45
A.10	Lista de pedidos visível ao Gestor de Rota, incluindo os que estão por agendar. . . . .	46
A.11	Seleção da tipologia e início do processo de validação do pedido. . . . .	46
A.12	Visualização dos detalhes do pedido e informação do Estabelecimento. . . . .	47
A.13	Seleção da zona e fornecedor disponível para o pedido. . . . .	47
A.14	Validação de disponibilidade do fornecedor e agendamento do serviço. . . . .	48
A.15	Confirmação de notificação ao fornecedor com todos os dados do pedido. . . . .	48
A.16	Confirmação de notificação com sucesso ao fornecedor com todos os dados do pedido. . . . .	49
A.17	Visualização do pedido após o agendamento com o fornecedor. . . . .	49
A.18	Lista de pedidos atribuídos visível ao Gestor de Fornecedor. . . . .	50
A.19	Lista de pedidos atribuídos visível ao Gestor de Fornecedor. . . . .	50
A.20	Lista de pedidos atribuídos visível ao Gestor de Fornecedor. . . . .	50
A.21	Lista de pedidos atribuídos visível ao Gestor de Fornecedor. . . . .	51
A.22	Lista de pedidos atribuídos visível ao Gestor de Fornecedor. . . . .	51
A.23	Lista de pedidos atribuídos visível ao Gestor de Fornecedor. . . . .	51
A.24	Resumo final do pedido na aba "Detalhe do Pedido" do ponto de vista do Requerente. . . . .	52
A.25	Resumo final do pedido na aba "Fornecedor" do ponto de vista do Requerente. . . . .	52
A.26	Resumo final do pedido na aba "Histórico" do ponto de vista do Requerente. . . . .	53

A.27	Resumo final do pedido na aba "Detalhe do Pedido" do ponto de vista do Gestor de Rota. . . . .	53
A.28	Resumo final do pedido na aba "Fornecedor" do ponto de vista do Gestor de Rota. . . . .	54
A.29	Resumo final do pedido na aba "Histórico" do ponto de vista do Gestor de Rota.	54
A.30	Resumo final do pedido na aba "Detalhe do Pedido" do ponto de vista do Gestor de Fornecedor. . . . .	55
A.31	Resumo final do pedido na aba "Fornecedor" do ponto de vista do Gestor de Fornecedor. . . . .	55
A.32	Resumo final do pedido na aba "Histórico" do ponto de vista do Gestor de Fornecedor. . . . .	55
A.33	Arquitetura geral do sistema PGIP, com destaque para as camadas de <i>frontend</i> , <i>backend</i> e integrações externas. . . . .	58
A.34	Diagrama de atividade do <i>Happy Flow</i> da aplicação PGIP, segmentado por perfis de utilizador. . . . .	59
A.35	Interface do menu de tipologias de pedidos. Esta funcionalidade permite ao administrador consultar, aceder e gerir as diferentes tipologias existentes no sistema. . . . .	60
A.36	Formulário para criação de nova tipologia de pedido. Esta interface permite definir o tipo de serviço e os campos necessários para o seu preenchimento. . . . .	60
A.37	Página de edição de tipologia de pedido. Permite alterar os campos, regras e validações associadas a uma tipologia já existente. . . . .	61
A.38	Vista detalhada de um pedido submetido. A página exhibe todas as informações associadas à requisição, incluindo datas, estados e dados adicionais. . . . .	61
A.39	Detalhes do fornecedor associado a um pedido. Esta página fornece uma visão consolidada do fornecedor atribuído, incluindo contactos e histórico. . . . .	61
A.40	Formulário para geração de guia de transporte. Esta funcionalidade permite criar uma guia associada ao pedido, incluindo os dados logísticos essenciais. . . . .	62
A.41	Visualização detalhada da guia de transporte gerada. Apresenta os dados transportados, horários e ligação direta ao pedido associado. . . . .	62
A.42	Página de histórico de pedidos. Regista as ações realizadas, alterações de estado e datas relevantes para rastreabilidade dos processos. . . . .	62
A.43	Parte 1 do formulário de registo de fornecedores. Esta secção permite inserir informações básicas do novo fornecedor, como nome e contacto. . . . .	63
A.44	Parte 2 do formulário de fornecedores. Inclui campos adicionais como localização, categoria de serviço e capacidade de resposta. . . . .	63
A.45	Parte 3 do formulário de fornecedores. Finaliza o processo de registo com validações e campos complementares. . . . .	63
A.46	Gestão de permissões (parte 1). Permite a atribuição de permissões específicas a diferentes roles, controlando o acesso às funcionalidades. . . . .	64
A.47	Gestão de permissões (parte 2). Complementa a interface anterior com a possibilidade de revisão e modificação granular das permissões atribuídas. . . . .	64

## Lista de Tabelas

2.1	Comparação entre <i>frameworks</i> de desenvolvimento. . . . .	7
4.1	Plano de execução do estágio. . . . .	16
4.2	Requisitos Funcionais do sistema PGIP . . . . .	17
4.3	Requisitos Não Funcionais do sistema PGIP . . . . .	18
4.4	Análise de Riscos - Resumo . . . . .	21
5.1	Sumário dos testes realizados no sistema PGIP. . . . .	32
6.1	Comparação entre os objetivos iniciais e os resultados obtidos. . . . .	36
A.1	Tabela com a descrição das User Stories do projeto PGIP . . . . .	57



## **Lista de Acrónimos**

- AOT** ahead-of-time
- API** Interface de Programação de Aplicações (*Application Programming Interface*)
- CI/CD** Integração Contínua/Entrega Contínua (*Continuous Integration/Continuous Delivery*)
- CRUD** Criar, Ler, Atualizar e Eliminar (*Create, Read, Update, Delete*)
- CSS** Folhas de Estilo em Cascata (*Cascading Style Sheets*)
- DB** Base de Dados (*Database*)
- HTML** Linguagem de Marcação de Hipertexto (*HyperText Markup Language*)
- ORM** *Object-Relational Mapping*
- PGIP** Plataforma de Gestão Integrada de Pedidos
- PHP** Pré-Processador de Hipertexto (*Hypertext PreProcessor*)
- TI** Tecnologia da Informação (*Information Technology*)
- SAP** *Systems, Applications, and Products in Data Processing*
- SQL** Linguagem de Consulta Estruturada (*Structured Query Language*)
- TS** *TypeScript*
- UBI** Universidade da Beira Interior
- UC** Unidade Curricular
- UI** Interface do Utilizador (*User Interface*)
- UX** Experiência do Utilizador (*User Experience*)



# Capítulo 1

## Introdução

Este documento é um relatório de estágio elaborado em cumprimento parcial para obtenção do grau de mestre em Engenharia Informática na Universidade da Beira Interior (UBI). O relatório segue o formato geral disponibilizado para este tipo de documento para a Unidade Curricular (UC) de *Dissertação ou Estágio* do 2º ciclo em Engenharia Informática.

No caso do estágio, o primeiro capítulo descreve a empresa e o estagiário, assim como os objetivos gerais do estágio. Aparte adaptações, os capítulos subsequentes descrevem tipicamente o projeto (nomeadamente requisitos, tecnologias, implementação e testes) e as tarefas desenvolvidas.

### 1.1 Caracterização da Empresa

A *Latitudde* é uma empresa pertencente ao grupo *Readiness IT*, especializada em consultoria e desenvolvimento de soluções na área das Tecnologia da Informação (*Information Technology*) (TI). Fundada em 2021, a *Latitudde* posiciona-se como uma referência na prestação de serviços de Engenharia de Software, abrangendo projetos *full-stack*.

Com uma presença crescente em três continentes, a *Latitudde* está sediada na Casa da Moagem, no Fundão, em Portugal, e conta com operações no Chile, Peru e Nova Zelândia, encontrando-se atualmente em processo de expansão para Amesterdão. A missão da empresa é oferecer soluções tecnológicas inovadoras e personalizadas que contribuam para o sucesso dos seus clientes nos mais variados setores.

### 1.2 Caracterização do Estagiário

O estágio foi realizado pelo estudante Emanuel Grilo Oliveira, no âmbito do curso de Mestrado em Engenharia Informática da UBI, tendo decorrido fisicamente nas instalações da empresa *Latitudde*, localizadas no Fundão. Com formação académica e interesses centrados na área do desenvolvimento de software, o estudante viu neste estágio uma oportunidade para aplicar e consolidar os conhecimentos adquiridos durante o percurso académico.

Durante o estágio, o estudante desempenhou funções no projeto **PGIP**, que envolvem o planeamento e desenvolvimento de uma solução tecnológica para otimizar a gestão de processos relacionados com a recolha e transporte de águas e resíduos.

### 1.3 Apresentação e Objetivos do Estágio

O estágio está integrado no desenvolvimento do projeto **PGIP**, cujo objetivo é otimizar a gestão de funcionários e a coordenação de pedidos relacionados com serviços de recolha e

transporte. Este projeto abrange a criação de fluxogramas detalhados para visualização dos processos do sistema, o mapeamento de regras de negócio e a elaboração de uma solução funcional para responder às necessidades do cliente.

No contexto da UC de Projeto de Estágio, este relatório centra-se na delimitação do problema, na definição das tecnologias e metodologias a utilizar e na execução prática do projeto PGIP. Este projeto visa otimizar a gestão de pedidos de serviços entre requerentes, gestores de rota e fornecedores. Assim, os principais objetivos do estágio foram:

- Compreender os requisitos funcionais e não funcionais do sistema PGIP;
- Selecionar e justificar a utilização de tecnologias como Angular e Laravel para o desenvolvimento da plataforma;
- Desenvolver a interface de utilizador (*front-end*) da aplicação, garantindo responsividade e integração com o *back-end*;
- Participar na definição e implementação de fluxos de trabalho e funcionalidades-chave, como agendamento, aprovação e execução de pedidos;
- Contribuir para os testes e validação das funcionalidades desenvolvidas.

## **1.4 Organização do Documento**

Este documento encontra-se estruturado em cinco capítulos principais, organizados da seguinte forma:

- O **Capítulo 1 — Introdução** — apresenta a caracterização da empresa, do estagiário, os objetivos do estágio e a organização do relatório;
- O **Capítulo 2 — Estado da Arte** — fornece uma análise de soluções e tecnologias semelhantes, servindo de base para o desenvolvimento do projeto;
- O **Capítulo 3 — Tecnologias e Ferramentas Utilizadas** — descreve e fundamenta as abordagens e ferramentas previstas para o desenvolvimento;
- O **Capítulo 4 — Planificação** — estrutura as tarefas a realizar e apresenta o cronograma previsto para a sua execução;
- O **Capítulo 5 — Implementação e Testes** — estrutura as tarefas realizadas e apresenta o fluxo de execução;
- O **Capítulo 6 — Conclusão** — resume os principais pontos do relatório e os resultados obtidos;
- Os **Anexos — Mockups** — apresentam os protótipos visuais das interfaces desenvolvidas, servindo como base de validação e orientação para a implementação;
- Os **Anexos — Modelo de Dados** — incluem os diagramas que descrevem a estrutura da base de dados e a relação entre entidades;

## **PGIP**

- Os **Anexos — Sistema Desenvolvido** — contêm os diagramas de arquitetura, diagramas de atividade e o registo completo das *User Stories* implementadas.



## Capítulo 2

### Estado da Arte

#### 2.1 Introdução

Neste capítulo são apresentados os conceitos, tecnologias (secção 2.1) e soluções existentes (secção 2.2) relevantes para o desenvolvimento do projeto **PGIP**. O objetivo desta secção é fornecer um enquadramento teórico e prático que suporte as escolhas metodológicas e técnicas efetuadas (secção 2.3) ao longo do estágio.

#### 2.2 Enquadramento do Projeto

O projeto **PGIP** consiste no desenvolvimento de uma plataforma digital orientada para a gestão operacional de serviços relacionados com a recolha e transporte de resíduos. Esta solução visa otimizar a comunicação e a coordenação entre diferentes intervenientes — requerentes, gestores e fornecedores — centralizando os pedidos de serviço e permitindo a sua gestão em tempo real. O projeto insere-se no contexto de plataformas empresariais orientadas à eficiência organizacional e interoperabilidade com sistemas externos.

Para exemplificar o seu funcionamento: um utilizador com permissões de Requerente pode aceder à plataforma, preencher um pedido de serviço com os dados exigidos (ex. tipo de contentor, localização, quantidade), submeter o pedido e acompanhar todo o seu estado até à conclusão. O Gestor de Rota valida, associa o fornecedor e agenda a execução, enquanto o Gestor de Fornecedor acede ao pedido e confirma a operação.

Adicionalmente, o projeto incorpora o desenvolvimento de uma interface moderna e adaptável usando tecnologias web e de integração, recorrendo a Interface de Programação de Aplicações (*Application Programming Interface*) (API)s e comunicação com bases de dados remotas. Neste contexto, destacam-se os seguintes conceitos-chave que fundamentam a abordagem adotada:

- **Gestão Integrada de Processos** — Sistemas que unificam diversas operações organizacionais, permitindo maior eficiência e redução de erros operacionais;
- **Regras de Negócio** — Conjunto de políticas e procedimentos que orientam a lógica de funcionamento das aplicações e garantem a sua aderência aos requisitos do cliente;
- **Interface de Utilizador Responsiva** — Interfaces adaptáveis a diferentes dispositivos e dimensões de ecrã, assegurando uma Experiência do Utilizador (*User Experience*) (UX) consistente.

## 2.3 Soluções e Tecnologias Relacionadas

De seguida, são descritas tecnologias e abordagens relevantes para o desenvolvimento de sistemas como o **PGIP**, agrupadas por categorias.

### 2.3.1 *Framework* de Desenvolvimento Angular

**Angular:** *Framework* de *front-end* baseada em *TypeScript*, conhecida pela criação de interfaces de utilizador dinâmicas e modulares. O Angular foi escolhido devido à sua robustez, escalabilidade e comunidade ativa, características fundamentais para o desenvolvimento de projetos de larga escala como o **PGIP** [1]. No contexto deste projeto, o Angular foi utilizado para garantir uma UX fluída, responsiva e compatível com múltiplos dispositivos, assegurando a interação eficiente com os serviços disponibilizados pela plataforma.

### 2.3.2 Integração com o Sistema SAP

O sistema *Systems, Applications, and Products in Data Processing* (SAP), utilizado pelo *Cliente*, desempenha um papel central na gestão dos dados necessários às operações da empresa. O projeto **PGIP** está a ser desenvolvido para comunicar diretamente com este sistema, permitindo a transferência eficiente de dados entre as duas plataformas. Esta integração visa:

- Enviar os dados recolhidos no **PGIP** para o sistema SAP, assegurando que a informação gerada pelo **PGIP** é incorporada no fluxo operacional existente do *Cliente*;
- Minimizar redundâncias e evitar a necessidade de inserção manual de dados no SAP;
- Garantir conformidade com os requisitos técnicos do *Cliente* [2].

A integração foi desenhada de forma a utilizar interfaces compatíveis (como APIs ou formatos específicos de ficheiros) para assegurar a transferência segura e eficiente de dados.

### 2.3.3 Metodologias de Desenvolvimento

Para garantir eficiência e flexibilidade no desenvolvimento, foi adotada a metodologia ágil **Scrum**. Esta abordagem foi escolhida devido à sua capacidade de adaptação a alterações nos requisitos e ao foco em entregas incrementais. Os principais benefícios incluem:

- Divisão do trabalho em ciclos iterativos (*sprints*), com metas bem definidas;
- *Feedback* contínuo das partes interessadas, permitindo ajustes ao longo do processo;
- Maior controlo sobre o progresso e a qualidade das entregas [3].

## PGIP

### 2.3.4 Segurança em Aplicações Web

A segurança é um fator crítico no desenvolvimento de aplicações web. Ataques como **SQL Injection** [4] e **Cross-Site Request Forgery (CSRF)** [5] são comuns em sistemas que não validam corretamente os dados introduzidos pelos utilizadores. O primeiro permite manipular a base de dados através da injeção de instruções maliciosas, enquanto o segundo explora a sessão de um utilizador autenticado para executar ações sem o seu consentimento. No contexto do projeto **PGIP**, foram aplicadas medidas para mitigar este tipo de vulnerabilidades, garantindo uma maior robustez da aplicação. O uso do *framework* Laravel permite, por defeito, a utilização de *prepared statements*, prevenindo ataques por injeção de SQL. Além disso, a *framework* inclui proteção automática contra *CSRF* através de *tokens*, que são gerados e validados em cada submissão de formulário, evitando a execução de ações sem autenticação válida.

## 2.4 Análise Comparativa

A Tabela 2.1 apresenta uma análise comparativa entre duas das principais *frameworks* de desenvolvimento *front-end* candidatas ao projeto — *Angular* e *Vue.js* — com base em critérios relevantes para o desenvolvimento do *PGIP*, tais como robustez, escalabilidade, facilidade de manutenção, curva de aprendizagem, peso do bundle e flexibilidade arquitetónica. Esta análise foi realizada com base em documentação oficial e artigos técnicos que comparam as características de ambas as ferramentas [6, 7].

Tabela 2.1: Comparação entre *frameworks* de desenvolvimento.

<b>Critério</b>	<b>Angular</b>	<b>Vue.js</b>
Robustez	Alta	Média
Escalabilidade	Alta	Alta
Facilidade de Manutenção	Média	Alta
Curva de Aprendizagem	Difícil	Fácil
Tamanho do Bundle	Médio/Alto	Baixo
Flexibilidade	Baixa	Alta

### 2.4.1 Robustez e Escalabilidade

O *Angular* é amplamente considerado mais robusto devido ao seu ecossistema completo e suporte corporativo pela Google. A sua arquitetura baseada em módulos e boas práticas integradas favorece aplicações empresariais de larga escala. O *Vue.js*, embora menos estruturado, também suporta escalabilidade, especialmente com o uso de ferramentas complementares como *Vuex* e *Vue Router*.

### 2.4.2 Manutenção e Curva de Aprendizagem

O *Vue.js* destaca-se pela facilidade de aprendizagem e manutenção, especialmente para equipas com menos experiência ou para projetos de menor dimensão. A sua curva de entrada é baixa, permitindo a criação de aplicações funcionais em pouco tempo. Já o *Angular* exige

maior esforço inicial, devido à complexidade do seu ecossistema e à forte integração com TypeScript, mas oferece ganhos de longo prazo em ambientes mais exigentes.

### 2.4.3 Tamanho do Bundle

Aplicações criadas com Vue.js tendem a gerar bundles finais significativamente mais leves (cerca de 30KB com Vuex e Vue Router), enquanto bundles criados com Angular CLI (mesmo com ahead-of-time (AOT) e *tree-shaking*) rondam os 65KB [7].

### 2.4.4 Flexibilidade

O *Vue.js* proporciona uma ampla liberdade na estruturação da aplicação, sem impor restrições excessivas, viabilizando assim diferentes trajetórias de desenvolvimento consoante os objetivos do utilizador. O *Angular*, por sua vez, segue uma abordagem mais rígida e estruturada, que pode vir a beneficiar projetos com equipas grandes e processos mais formais. Tendo em conta os requisitos do *PGIP* — nomeadamente integração com o SAP, segurança, consistência técnica e maturidade — a *framework* selecionada foi o *Angular*. Esta escolha deveu-se ao facto de esta tecnologia responder de forma mais adequada aos requisitos previamente identificados.

## 2.5 Conclusão

O levantamento do estado da arte apresentado neste capítulo permitiu identificar as tecnologias e abordagens mais adequadas para o desenvolvimento do *PGIP*. A escolha do *Angular* como *framework* principal e a integração eficiente com o sistema SAPs do *Cliente* foram fundamentadas na robustez, escalabilidade e adequação aos requisitos do cliente. Estas escolhas serão detalhadas e justificadas nos capítulos seguintes, focando nas ferramentas e metodologias aplicadas.

## Capítulo 3

# Tecnologias e Ferramentas Utilizadas

### 3.1 Introdução

Neste capítulo são apresentadas as tecnologias, ferramentas (secção 3.2), linguagens (secção 3.3) e *frameworks* (secção 3.4) utilizadas durante o desenvolvimento do projeto **PGIP**. Cada uma das tecnologias foi selecionada com base na sua adequação aos requisitos do sistema, tendo em conta fatores como escalabilidade, robustez, eficiência e integração com plataformas externas. A organização deste capítulo segue uma estrutura lógica que reflete as áreas principais do desenvolvimento.

### 3.2 Ferramentas

O desenvolvimento do **PGIP** envolveu a utilização de diversas ferramentas que suportaram o trabalho de programação, teste, integração e gestão de sistemas. As ferramentas escolhidas foram essenciais para garantir a qualidade e a eficiência no desenvolvimento, atendendo às necessidades específicas do projeto.

#### 3.2.1 Visual Studio Code

O **Visual Studio Code** [8] foi utilizado como editor de código principal para o desenvolvimento de componentes tanto do *front-end* como do *back-end*. Este editor, desenvolvido pela *Microsoft*, é amplamente reconhecido pela sua leveza, versatilidade e extensibilidade. As funcionalidades destacam-se por incluir:

- **Integração com Git** — Permitindo um controlo de versões eficiente e colaborativo diretamente na interface do editor;
- **Extensões Personalizáveis** — Foram utilizadas extensões como **Prettier**, para formatação automática do código, **ESLint**, para deteção de erros de sintaxe, e **Debugger for Chrome**, para testes e depuração de aplicações no navegador.

Estas capacidades ajudaram a melhorar a produtividade da equipa, garantindo a consistência e a qualidade do código ao longo do desenvolvimento. Segundo a documentação oficial, o *Visual Studio Code* é uma ferramenta robusta e amplamente adotada pela comunidade de programadores [8].

#### 3.2.2 MySQL

A base de dados **MySQL** [9] foi selecionada para armazenar as informações operacionais do sistema, como dados dos utilizadores, formulários e registos de pedidos. A escolha do

*MySQL* foi motivada por:

- **Desempenho e Confiabilidade** — O *MySQL* oferece alta performance em consultas e é amplamente utilizado para aplicações que requerem acesso rápido e confiável a dados;
- **Flexibilidade** — Suporte para armazenamento de dados estruturados e consultas complexas, adequadas às necessidades operacionais do sistema;
- **Compatibilidade** — Integração facilitada com as ferramentas e tecnologias utilizadas no desenvolvimento do *PGIP*.

Além disso, ferramentas como o *MySQL Workbench* foram utilizadas para a gestão da base de dados, incluindo modelação, criação de tabelas e monitorização de consultas Linguagem de Consulta Estruturada (*Structured Query Language*) (SQL). O *MySQL* é amplamente reconhecido como uma solução robusta e eficiente, especialmente para sistemas de médio e grande porte [9].

### 3.2.3 Figma

O *Figma* [10] foi utilizado para o *design* e prototipagem da interface do utilizador (*Interface do Utilizador (User Interface) (UI)*) do *PGIP*. Esta ferramenta destacou-se pela sua capacidade de:

- **Colaboração em Tempo Real** — Facilitando a interação entre membros da equipa, permitindo que designers e programadores trabalhassem de forma sincronizada;
- **Criação de Protótipos Interativos** — Para validar o *design* das páginas e simular a navegação no sistema;
- **Consistência no Design** — Através do uso de bibliotecas de componentes reutilizáveis e guias de estilo.

Os *designs* criados no *Figma* foram essenciais para garantir a usabilidade e a conformidade estética da aplicação, alinhando as expectativas do cliente e da equipa técnica. O *Figma* é amplamente utilizado pela comunidade de *design* como uma ferramenta confiável e eficiente [10].

## 3.3 Linguagens

A construção do *PGIP* recorreu a linguagens modernas e amplamente utilizadas no desenvolvimento de aplicações web, cada uma desempenhando um papel essencial nas diferentes camadas da aplicação. A escolha destas linguagens foi baseada na sua robustez, compatibilidade e integração com as tecnologias adotadas.

## PGIP

### 3.3.1 HTML

Linguagem de Marcação de Hipertexto (*HyperText Markup Language*) (HTML) foi utilizada como a base estrutural do conteúdo da interface do utilizador. Esta linguagem define a estrutura semântica das páginas web, permitindo a organização hierárquica dos elementos de conteúdo, como textos, imagens e tabelas. A compatibilidade do HTML com navegadores modernos foi crucial para garantir uma UX consistente em diferentes dispositivos [11].

### 3.3.2 CSS

Folhas de Estilo em Cascata (*Cascading Style Sheets*) (CSS) foi usado para estilizar a interface, assegurando que a aplicação apresentasse um design responsivo e atrativo. Através do CSS, foi possível criar uma identidade visual uniforme para a aplicação, utilizando técnicas como *media queries* para adaptar o *layout* a múltiplos dispositivos, desde computadores de secretária até *smartphones*. Adicionalmente, bibliotecas como *Bootstrap* foram integradas para facilitar a implementação de componentes reutilizáveis [12].

### 3.3.3 TypeScript

*TypeScript* (TS), uma extensão do JavaScript, foi a linguagem principal para o desenvolvimento do *front-end* da aplicação, devido à sua robustez e integração com *frameworks* como o **Angular**. A tipagem estática fornecida pelo TypeScript ajudou a prevenir erros durante o desenvolvimento, oferecendo maior segurança e previsibilidade no comportamento do código. Além disso, o *TypeScript* facilitou a criação de componentes modulares e escaláveis, fundamentais para o crescimento futuro da aplicação [13].

### 3.3.4 PHP

Pré-Processador de Hipertexto (*Hypertext PreProcessor*) (PHP) foi a linguagem principal para o desenvolvimento do *back-end*, sendo utilizada para a lógica do servidor e a interação com a base de dados. A escolha do PHP deveu-se à sua ampla adoção em projetos web, bem como à sua capacidade de integração com diferentes sistemas, como o **MySQL**. Adicionalmente, a utilização de *frameworks* como *Laravel* ajudou a estruturar o código de forma limpa e eficiente, facilitando a manutenção futura [14].

## 3.4 Frameworks

O projeto foi desenvolvido utilizando *frameworks* que facilitaram a criação de uma aplicação web moderna, escalável e de alta performance. Estas ferramentas foram escolhidas com base na sua robustez, comunidade ativa e compatibilidade com as tecnologias do projeto, o *Angular* e o *Laravel*.

### 3.4.1 Angular

A *framework* **Angular**, desenvolvida pela Google, foi utilizada no desenvolvimento do *front-end* do sistema, destacando-se pela sua modularidade e suporte para **TypeScript**. As principais vantagens, que justificam a escolha por esta ferramenta, incluem:

- **Componentização** — Facilita a organização do código em partes reutilizáveis;
- **Ligação de Dados Bidirecional** — Sincroniza automaticamente a interface com o modelo de dados;
- **Ferramentas Integradas** — Suporte nativo para rotas, testes e injeção de dependências.

O *Primeng* foi utilizado para implementar elementos visuais consistentes, garantindo uma interface moderna e acessível [15].

### 3.4.2 Laravel

A *framework* **Laravel** [16], baseada em PHP, foi utilizada para o desenvolvimento do *back-end* do sistema. O **Laravel** destaca-se pela sua simplicidade e eficiência, oferecendo ferramentas como:

- **Object-Relational Mapping (ORM) Eloquent**: Simplifica a interação com a base de dados através de um sistema de mapeamento objeto-relacional, facilitando a implementação das operações Criar, Ler, Atualizar e Eliminar (*Create, Read, Update, Delete*) (CRUD).
- **Roteamento Simples**: Facilita a comunicação entre o *front-end* e o *back-end*;
- **Segurança Incorporada**: Inclui hashing de senhas e proteção contra injeção de SQL.

O suporte integrado para desenvolvimento de APIs RESTful garantiu uma integração eficiente com o *front-end*, enquanto a documentação ampla facilitou a resolução de problemas [16].

Para além da sua flexibilidade no desenvolvimento de APIs REST e integração com bases de dados relacionais, o *Laravel* contribui significativamente para a segurança da aplicação. A *framework* utiliza, por padrão, *prepared statements* nas *queries* SQL, reduzindo significativamente o risco de injeções de código malicioso e reforçando a segurança da Base de Dados (*Database*) (DB). Adicionalmente, o *Laravel* implementa mecanismos automáticos de proteção contra CSRF, através da geração e verificação de *tokens* em cada submissão de formulário, assegurando que as ações são sempre executadas por utilizadores autenticados. Adicionalmente, foram utilizadas *pipelines* de Integração Contínua/Entrega Contínua (*Continuous Integration/Continuous Delivery*) (CI/CD) para validar automaticamente o código submetido e agilizar o processo de entrega contínua da aplicação.

### **3.5 Conclusão**

As tecnologias, ferramentas, linguagens e *frameworks* apresentadas neste capítulo foram selecionadas de forma criteriosa para garantir que o projeto **PGIP** cumprisse os seus objetivos. Ferramentas como o **Visual Studio Code**, o **MySQL** e o **Figma** proporcionaram suporte eficiente ao desenvolvimento e design da aplicação.

As linguagens **TypeScript** e **PHP**, combinadas com *frameworks* robustas como **Angular** e **Laravel**, permitiram criar um sistema modular, escalável e bem integrado com plataformas externas como o **SAP**. Esta escolha tecnológica assegurou uma base sólida para a construção de um sistema eficiente, seguro e preparado para futuras evoluções.



## Capítulo 4

### Planificação

#### 4.1 Introdução

A planificação é uma etapa crucial para a organização e execução eficiente do estágio. Este capítulo apresenta a estrutura do trabalho desenvolvido no âmbito do projeto **PGIP**, detalhando as tarefas previstas, o plano de execução, a planificação do trabalho, os requisitos do estágio e outros aspetos organizacionais que suportam a realização das atividades. Este planeamento tinha como objetivo assegurar uma abordagem rigorosa e alinhada com os objetivos estabelecidos. Em suma, a secção 4.2 inclui a identificação de todas as tarefas deste estágio, enquanto que o calendário está descrito na secção 4.3. A planificação mais detalhada do trabalho é depois discutida na secção 4.4, seguido pela a secção 4.5 que descreve os requisitos do sistema que foi desenvolvido, algo que já foi conseguido na fase de planeamento deste estágio.

#### 4.2 Enumeração das Tarefas

As tarefas previstas para o estágio foram organizadas em fases distintas, de modo a garantir a clareza e o foco em cada etapa do desenvolvimento. A lista de tarefas é apresentada de seguida:

- **Fase 1 - Análise e Levantamento de Requisitos** —
  - Reunião inicial com as partes interessadas para compreender os objetivos do sistema;
  - Levantamento e documentação dos requisitos funcionais e não-funcionais;
  - Identificação das regras de negócio associadas ao sistema.
- **Fase 2 - Planeamento e Design** —
  - Definição da arquitetura do sistema;
  - Criação de fluxogramas para mapear os processos internos;
  - Prototipagem inicial da interface de utilizador.
- **Fase 3 - Implementação** —
  - Desenvolvimento dos módulos principais do sistema, incluindo o *front-end* e *back-end*;
  - Configuração de integrações com ferramentas externas, como SAP.

- **Fase 4 - Testes e Validação** —
  - Realização de testes funcionais e de desempenho;
  - Ajustes com base no feedback dos stakeholders.
- **Fase 5 - Documentação** —
  - Elaboração do relatório final de estágio;
  - Preparação de documentação técnica para entrega ao cliente.

### 4.3 Plano de Execução

O plano de execução foi elaborado para distribuir as tarefas ao longo do período de estágio, assegurando que cada etapa tenha tempo suficiente para ser concluída de forma eficiente. A Tabela 4.1 apresenta o cronograma geral das fases.

Tabela 4.1: Plano de execução do estágio.

Fase	Descrição	Duração (semanas)
Fase 1	Análise e levantamento de requisitos	8
Fase 2	Planeamento e design	6
Fase 3	Implementação	10
Fase 4	Testes e validação	4
Fase 5	Documentação	2

### 4.4 Planificação do Trabalho

A planificação detalhada do trabalho foi realizada com recurso à metodologia **Scrum**, pela sua capacidade de gerir projetos de forma iterativa e adaptável. O desenvolvimento foi dividido em *sprints* semanais, de acordo com os seguintes objetivos principais:

- **Sprint 1-8 (Novembro a Janeiro)** — Levantamento de requisitos e identificação de regras de negócio (Fase 1) e desenvolvimento da arquitetura e protótipos iniciais (Fase 2);
- **Sprint 9-18 (Fevereiro a Abril)** — Implementação dos módulos principais do sistema (Fase 3);
- **Sprint 19-22 (Maio)** — Testes e validação das funcionalidades (Fase 4);
- **Sprint 23-24 (Junho)** — Finalização e documentação (Fase 5).

A metodologia *Scrum* foi aplicada com *sprints* semanais, promovendo um ciclo contínuo de desenvolvimento e validação das funcionalidades. Para além das tarefas planeadas inicialmente, foram realizadas reuniões diárias (*daily meetings*) com o objetivo de alinhar o progresso entre os elementos da equipa. Sempre que surgiam dúvidas ou necessidade de rever aspetos da implementação, eram agendadas reuniões adicionais de curta duração.

## PGIP

O trabalho foi realizado em colaboração com a equipa da **Latitудde**, através de reuniões regulares de revisão de progresso e planeamento, alinhadas com os princípios do *Scrum*. A gestão e organização das tarefas foi realizada através das ferramentas *GitLab* e *Fork*, que permitiram o controlo de versões, revisão de código e acompanhamento do progresso ao longo do estágio.

O Diagrama de *Gantt*, apresentado na Figura 4.1, ilustra a distribuição das atividades planeadas durante o estágio.

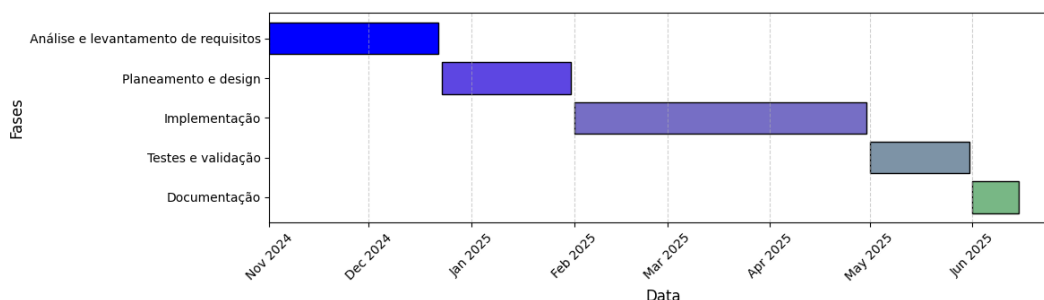


Figura 4.1: Diagrama de *Gantt* da estimativa do plano de trabalho.

## 4.5 Requisitos do Sistema Desenvolvido

Os requisitos do estágio foram organizados em três categorias principais: funcionais, técnicos e não-funcionais. Estes requisitos foram definidos para orientar o desenvolvimento do sistema **PGIP**, garantindo que atende às necessidades dos utilizadores e aos objetivos definidos pela organização.

### 4.5.1 Requisitos Funcionais

Os requisitos funcionais descrevem as principais funcionalidades que o sistema deve oferecer para satisfazer as necessidades dos utilizadores. Foram definidos com base nas *User Stories* e funcionalidades implementadas no decorrer do projeto.

ID	Descrição do Requisito Funcional
RF01	O sistema deve permitir aos utilizadores agendar pedidos de recolha e transporte de resíduos, especificando localização, tipo de resíduo e horários.
RF02	Os utilizadores devem poder consultar o estado dos pedidos em tempo real (ex.: pendente, em processamento, concluído).
RF03	O sistema deve enviar notificações automáticas em eventos críticos (ex.: alterações de estado ou atrasos no transporte).
RF04	O sistema deve permitir a exportação de relatórios com detalhes dos pedidos submetidos.
RF05	O sistema deve permitir a gestão de tipologias de pedidos e respetivos formulários dinâmicos.
RF06	O sistema deve permitir a gestão de utilizadores, estabelecimentos e fornecedores.
RF07	O sistema deve permitir a geração de guias de transporte associadas aos pedidos.
RF08	O sistema deve garantir a autenticação via <i>Active Directory</i> para utilizadores internos.

Tabela 4.2: Requisitos Funcionais do sistema PGIP

Adicionalmente, a funcionalidade de monitorização de pedidos em tempo real (RFo2) foi uma prioridade desde a fase inicial do projeto, tendo impacto direto na experiência de utilizador. Esta funcionalidade foi implementada com recurso a atualizações periódicas e estados bem definidos.

Outro requisito de destaque foi o RFo3, que exigiu a configuração de um sistema de notificações por email e, futuramente, SMS, com base nas transições de estado dos pedidos. Este requisito garantiu que os intervenientes fossem informados em tempo útil.

#### 4.5.2 Requisitos Técnicos

Os requisitos técnicos garantem que o sistema é construído sobre uma base tecnológica robusta e eficiente. Os requisitos técnicos elicitados para o sistema a ser desenvolvido são os seguintes:

- **Tecnologias utilizadas** – Utilização do **Angular** para o *front-end*, e sincronização de dados e utilização da base de dados **MySQL** para armazenamento de informações;
- **Integração segura com sistemas externos** – A comunicação entre o **PGIP** e sistemas como o **SAP** deve ser realizada utilizando protocolos seguros, garantindo a proteção de dados sensíveis;
- **Desempenho e escalabilidade** – O sistema deve ser capaz de suportar aumentos no volume de utilizadores e pedidos sem comprometer o desempenho.

#### 4.5.3 Requisitos Não Funcionais

Os requisitos não funcionais estabelecem critérios de qualidade e desempenho que devem ser assegurados para garantir a fiabilidade, segurança e usabilidade do sistema PGIP.

ID	Descrição do Requisito Não Funcional
RNFo1	A aplicação deve ser responsiva e funcional em dispositivos <i>desktop</i> e móveis.
RNFo2	A interface deve ser intuitiva, com navegação clara e centrada no utilizador.
RNFo3	O sistema deve processar operações críticas (ex.: submissão de pedidos) em menos de 2 segundos.
RNFo4	Os dados devem estar protegidos contra ameaças como SQL Injection, CSRF e acessos não autorizados.
RNFo5	A aplicação deve estar em conformidade com o RGPD, protegendo dados pessoais dos utilizadores.
RNFo6	O sistema deve garantir rastreabilidade completa das ações efetuadas pelos utilizadores.

Tabela 4.3: Requisitos Não Funcionais do sistema PGIP

Em particular, o requisito RNFo4 esteve presente em todas as fases do desenvolvimento. A proteção contra ataques como SQL Injection e CSRF foi assegurada com o uso de prepared statements no back-end Laravel e com tokens automáticos nos formulários. A conformidade com o RGPD (RNFo5) foi também considerada, nomeadamente através da separação de dados sensíveis e autenticação segura via Microsoft Azure.

## PGIP

### 4.5.4 Priorização dos Requisitos

Para orientar o desenvolvimento e garantir o cumprimento dos prazos, os requisitos foram classificados por prioridade:

- **Essenciais** — Agendamento de pedidos, monitorização em tempo real, notificações de eventos, integração com o *SAP* e garantia de segurança de dados;
- **Desejáveis** — Implementação de relatórios exportáveis, suporte para múltiplos idiomas (português e inglês) e funcionalidades adicionais de responsividade avançada.

### 4.5.5 Suma de Requisitos

A definição clara dos requisitos do estágio assegura que o desenvolvimento do sistema **PGIP** estará alinhado com as metas da organização e as expectativas dos utilizadores. Estes requisitos servem como uma base sólida para o planeamento, desenvolvimento e monitorização do progresso ao longo do estágio.

## 4.6 Análise de Riscos e Plano de Mitigação

Nesta secção, são analisados os riscos identificados durante o planeamento e desenvolvimento do estágio, bem como as respetivas estratégias de mitigação. A análise é apresentada com base na descrição, probabilidade, impacto e plano de mitigação para cada risco.

### 4.6.1 Risco 1 - Sobrecarga de Tarefas

**Descrição:** Algumas tarefas previstas no planeamento podem coincidir temporalmente, o que pode gerar sobrecarga de trabalho.

**Impacto:** Possíveis atrasos na entrega de atividades e comprometimento na qualidade do trabalho.

**Probabilidade:** Média.

**Plano de Mitigação:** Este risco pode ser mitigado através de:

- Realização de um planeamento semanal detalhado das tarefas;
- Reavaliação e reorganização de prioridades sempre que necessário.

**Critério de Ativação:** Este plano de mitigação deve ser imediatamente acionado caso se verifique um atraso superior a dois dias nas tarefas planeadas.

### 4.6.2 Risco 2 - Dependência de Ferramentas Não *Open-Source*

**Descrição:** Algumas ferramentas necessárias para o projeto podem não ser de código aberto, dificultando o acesso e utilização.

**Impacto:** Redução na eficiência e no progresso das atividades.

**Probabilidade:** Média.

**Plano de Mitigação:** Para mitigar este risco, as seguintes medidas devem ser adotadas:

- Substituir ferramentas não *open-source* por alternativas de código aberto;

**Critério de Ativação:** A mitigação deve ser iniciada sempre que for identificado que ferramentas críticas não estão disponíveis.

#### 4.6.3 Risco 3 - Alterações nos Requisitos

**Descrição:** Mudanças nos requisitos por parte do cliente ou do supervisor podem impactar o desenvolvimento.

**Impacto:** Necessidade de refazer tarefas concluídas, resultando em atrasos e desperdício de esforço.

**Probabilidade:** Alta.

**Plano de Mitigação:** Para lidar com este risco, deve-se:

- Manter uma comunicação contínua com as partes interessadas;
- Documentar todas as alterações nos requisitos e ajustá-las ao planeamento.

**Critério de Ativação:** Sempre que o cliente ou o supervisor solicitar alterações significativas.

#### 4.6.4 Risco 4 - Dificuldades de Comunicação

**Descrição:** Restrições associadas ao trabalho remoto podem dificultar a comunicação com supervisores ou colegas.

**Impacto:** Lentidão na resolução de dúvidas ou problemas, podendo atrasar o desenvolvimento.

**Probabilidade:** Baixa.

**Plano de Mitigação:** Este risco pode ser minimizado através de:

- Agendamento de reuniões semanais regulares com supervisores;
- Utilização de ferramentas colaborativas como Microsoft Teams ou Slack.

**Critério de Ativação:** Este plano deve ser acionado caso sejam identificados atrasos frequentes na obtenção de respostas importantes.

#### 4.6.5 Resumo da Análise de Riscos

A Tabela 4.4 apresenta uma visão resumida dos riscos identificados durante a fase de planeamento, acompanhados das respetivas probabilidades, impactos e estratégias de mitigação previstas.

Tabela 4.4: Análise de Riscos - Resumo

Risco	Probabilidade	Impacto	Mitigação
Sobrecarga de Tarefas	Média	Médio	Planeamento semanal detalhado e reorganização de prioridades
Ferramentas Não Open-Source	Média	Alto	Uso de alternativas de código aberto
Alterações nos Requisitos	Alta	Alto	Comunicação contínua e documentação das alterações
Dificuldades de Comunicação	Baixa	Médio	Reuniões regulares e uso de ferramentas colaborativas

Durante o decorrer do estágio, alguns destes riscos vieram efetivamente a manifestar-se. Nomeadamente, os riscos de **Alterações nos Requisitos** e de **Dificuldades de Comunicação** revelaram-se relevantes, devido a mudanças pontuais nos requisitos e à demora nas respostas, o que resultou em tarefas pendentes de aprovação. As medidas de mitigação definidas foram aplicadas com sucesso, contribuindo para a minimização dos impactos no desenvolvimento do projeto. Os restantes riscos não se materializaram, mas mantiveram-se sob monitorização ativa ao longo de todo o processo.

## 4.7 Execução do Plano de Trabalho

A execução do estágio seguiu, de forma geral, o plano de trabalho inicialmente delineado, com adaptações naturais decorrentes das necessidades reais do projeto e da dinâmica da equipa. Esta secção apresenta uma síntese de como cada tarefa foi realizada, destacando os principais marcos, desafios e resultados.

- **Tarefa 1 — Análise e levantamento de requisitos** — Realizada entre novembro e dezembro (*Sprints* 1 a 4), esta fase decorreu como o previsto, consistiu em reuniões com os responsáveis da empresa para compreender o funcionamento do sistema **PGIP** e as suas necessidades operacionais. O principal desafio foi a assimilação dos fluxos internos do cliente. No final desta etapa, foi possível redigir uma primeira versão da lista de requisitos funcionais e não funcionais.
- **Tarefa 2 — Estudo das tecnologias e ferramentas** — Decorreu em paralelo com a Tarefa 1 e prolongou-se até janeiro (*Sprints* 1 a 8). Incluiu a análise comparativa de *frameworks* e bibliotecas para o desenvolvimento da plataforma, destacando-se a comparação entre Angular e Vue.js. A escolha recaiu sobre Angular, tendo em conta a robustez, escalabilidade e apoio da comunidade (ver Secção 2.4).
- **Tarefa 3 — Planeamento e arquitetura do sistema** — Desenvolvida nos meses de dezembro e janeiro (*Sprints* 5 a 8), consistiu na elaboração do modelo de dados, nos primeiros diagramas de arquitetura e na definição da estrutura modular da aplicação. Esta tarefa garantiu uma base sólida e escalável, compatível com a integração SAP para a continuidade do projeto. O principal resultado está ilustrado no Anexo A.33.
- **Tarefa 4 — Desenvolvimento da interface** — Esta foi a tarefa de maior duração, tendo decorrido entre fevereiro e abril (*Sprints* 9 a 18). Incluiu a criação de formulá-

rios dinâmicos, componentes reutilizáveis, autenticação e integração com serviços da API. O desenvolvimento seguiu uma abordagem iterativa. Os principais desafios estiveram relacionados com alterações frequentes de requisitos e a necessidade de manter coerência entre os componentes, provocando alguns atrasos no desenvolvimento de algumas tarefas. Dos requisitos que foram alterados, alguns voltaram ao seu estado inicial, enquanto que outros foram necessários alterar e provocou algumas mudanças à estrutura inicial.

- **Tarefa 5 — Testes e validação** — Devido a alterações na tarefa anterior, durante o mês de maio foi iniciada a (*Sprints* 19 a 22), esta fase envolveu testes manuais e funcionais às principais áreas do sistema, verificação de consistência de dados, e correção de erros das tarefas que se encontravam finalizadas. A estabilização da aplicação e a validação das funcionalidades descritas nas User Stories (ver Anexo A.2.1) foram os resultados centrais desta fase e decorreram como o esperado.
- **Tarefa 6 — Documentação e relatório** — Desenvolvida em paralelo com os testes e a primeira versão finalizada em junho (*Sprints* 23 e 24). Incluiu a criação da documentação técnica da plataforma e a redação do presente relatório, consolidando as aprendizagens adquiridas e sistematizando os contributos do estágio.

## 4.8 Planeamento e *Design*

A fase de planeamento e design do projeto **PGIP** incluiu a elaboração de um modelo de dados detalhado para suportar todas as funcionalidades previstas para a aplicação. Este modelo foi desenvolvido de forma a garantir escalabilidade, eficiência e facilidade de manutenção.

### 4.8.1 Descrição das Entidades do Modelo de Dados

De seguida, são descritas todas as entidades presentes no modelo de dados, detalhando a sua funcionalidade e propósito no sistema.

Os principais resultados obtidos nesta fase incluem:

- A definição de entidades principais que suportam funcionalidades críticas do sistema;
- A estruturação das relações entre as entidades, garantindo integridade referencial;
- A criação de um diagrama claro e completo para facilitar a implementação e comunicação com a equipa técnica.

### **Users**

A tabela **Users** armazena os dados essenciais dos utilizadores da aplicação, incluindo informações de autenticação, como o *e-mail* e a palavra-passe, bem como o seu papel no sistema. Esta tabela também indica se o utilizador está associado a uma empresa específica.

## **PGIP**

### ***Permissions***

A tabela **Permissions** contém todas as permissões associadas às funcionalidades da aplicação. Cada **permission** é associada a uma ou mais roles, definindo os acessos permitidos para cada tipo de utilizador.

### ***Roles***

A tabela **Roles** define os diferentes tipos de utilizadores existentes na aplicação. Cada role é configurável e possui um conjunto de permissões associado, exceto a role **Administrador**, que é a role padrão e não pode ser alterada.

### ***Solicitations***

A tabela **Solicitations** regista os dados base de todos os pedidos realizados na aplicação, como o autor, o responsável, o estado e as datas associadas ao pedido. Esta tabela centraliza a gestão de pedidos no sistema.

### ***Solicitation\_Statutes***

Esta tabela armazena os possíveis estados de um pedido, como *Pendente*, *Aprovado* ou *Concluído*. Cada estado está associado a um código único para facilitar a sua identificação.

### ***Solicitation\_Types***

A tabela **Solicitation\_Types** categoriza os diferentes tipos de pedidos, permitindo uma melhor organização e filtragem.

### ***Forms***

A tabela **Forms** é responsável por armazenar os dados relativos aos formulários utilizados na aplicação. Estes formulários podem ser associados a pedidos, utilizadores ou roles.

### ***Form\_Types***

Esta tabela distingue os diferentes tipos de formulários disponíveis no sistema, indicando se estão relacionados com pedidos, utilizadores ou outra entidade.

### ***Form\_Fields***

A tabela **Form\_Fields** define os campos individuais de cada formulário, armazenando informações como o título do campo, a ordem e o tipo de campo.

### ***Form\_Field\_Options***

Para campos complexos, como seleções múltiplas ou escolhas únicas, a tabela **Form\_Field\_Options** armazena as opções disponíveis para cada campo.

### ***Form\_Inputs***

A tabela **Form\_Inputs** guarda os valores introduzidos pelos utilizadores nos formulários, assegurando a rastreabilidade dos dados inseridos.

### ***Field\_Types***

A tabela **Field\_Types** descreve os diferentes tipos de campos disponíveis nos formulários, como *texto*, *data* ou *seleção múltipla*.

### ***Location***

A tabela **Location** armazena informações sobre os estabelecimentos associados a cada pedido, incluindo o seu tipo e localização.

### ***Location\_Types***

A tabela **Location\_Types** classifica os diferentes tipos de estabelecimentos.

### ***Companies***

Por fim, a tabela **Companies** identifica as empresas registadas no sistema, incluindo o nome e um código único.

## 4.8.2 Relações no Modelo de Dados

O modelo de dados do sistema **PGIP** foi concebido com um forte enfoque relacional, de forma a garantir a consistência, a integridade referencial e a escalabilidade da aplicação. A estrutura do modelo assegura que cada entidade se relaciona de forma clara com as restantes, respeitando as dependências funcionais e promovendo a normalização da base de dados.

Por exemplo:

- A tabela **Users** está diretamente ligada à tabela **Roles**, através de uma chave estrangeira, permitindo definir o perfil de acesso de cada utilizador. Esta relação é fundamental para assegurar a aplicação das permissões corretas, com base nas funções atribuídas.
- As tabelas **Forms**, **Form\_Fields**, **Form\_Inputs** e **Field\_Types** trabalham em conjunto para suportar a definição e o preenchimento de formulários dinâmicos. Esta estrutura modular permite a personalização de formulários consoante a tipologia do pedido, reforçando a flexibilidade do sistema.
- A entidade **Solicitations** está relacionada com várias tabelas auxiliares, como **Solicitation\_Statuses**, **Solicitation\_Types** e **Location**. Estas relações permitem classificar cada pedido com base no seu tipo, estado atual e localização associada, assegurando uma gestão eficaz do seu ciclo de vida.

## PGIP

- A associação entre **Users**, **Companies** e **Location** permite identificar o contexto organizacional e geográfico de cada pedido, sendo crucial para a atribuição correta de permissões e para a rastreabilidade dos processos.
- A tabela **Form\_Field\_Options** está relacionada com **Form\_Fields** e é responsável por definir as opções disponíveis em campos de seleção, como listas suspensas ou caixas de múltipla escolha, contribuindo para a consistência e padronização dos dados introduzidos.

Este modelo foi projetado para ser escalável e flexível, permitindo a expansão da aplicação para atender a novas necessidades no futuro. Para além disso, a estrutura relacional permite garantir integridade transacional, uma vez que as ações sobre uma entidade (como a remoção de um utilizador ou alteração de um pedido) respeitam as dependências existentes, evitando a perda de informação ou referências inválidas. Esta abordagem modular e relacional traduz-se num sistema robusto, adaptável às necessidades do Cliente e alinhado com as boas práticas de modelação de bases de dados relacionais.

### 4.8.3 Design da Interface do Utilizador

O design da UI do **PGIP** foi concebido para garantir uma experiência intuitiva e eficiente, alinhada aos objetivos da aplicação. O foco esteve na simplicidade e clareza, garantindo que os diferentes perfis de utilizadores, como administradores e fornecedores, conseguissem interagir com a aplicação de forma eficaz.

#### **Página de Login**

A Figura 5.1 (incluída no anexo 1) apresenta o *mockup* da página de **Login**. Este é o ponto de entrada para todos os utilizadores da aplicação, sendo projetado com os seguintes objetivos:

- Garantir um acesso seguro através da introdução de e-mail e palavra-passe;
- Proporcionar uma interface limpa e profissional, com foco nos campos essenciais;
- Inclui um link para recuperação de palavra-passe, promovendo acessibilidade e suporte ao utilizador.

#### **Página Principal - Pedidos**

A Figura 5.1 (incluída no anexo 1) mostra o *mockup* da **Página Principal** na visão do administrador. Esta página centraliza as informações e funcionalidades relacionadas aos pedidos, permitindo uma gestão eficiente. As características principais incluem:

- Exibição de uma lista detalhada de pedidos, incluindo informações como tipo, estado e responsável;
- Filtros e opções de ordenação para facilitar a navegação e a localização de pedidos específicos;

- Acesso rápido às ações principais, como adicionar novos pedidos ou alterar o estado de pedidos existentes.

### Formulário - Adicionar Novo Fornecedor

A Figura A.2 (incluída no anexo 1) apresenta o *mockup* do **Formulário para Adicionar Novo Fornecedor**. Este formulário foi desenhado para simplificar o processo de registo de fornecedores no sistema. As características principais incluem:

- Campos claros e organizados, como nome, telefone e informações adicionais do fornecedor;
- Uso de validações para garantir que os dados inseridos estão completos e corretos;
- Um botão de submissão destacado, promovendo uma navegação intuitiva.

### Diretrizes de Design

O design das interfaces foi guiado pelas seguintes diretrizes:

- **Consistência Visual:** Todas as páginas e componentes seguem uma identidade visual uniforme, com cores, fontes e espaçamentos consistentes;
- **Responsividade:** As interfaces foram desenhadas para se adaptarem a diferentes dispositivos, desde computadores até dispositivos móveis;
- **Usabilidade:** Priorizou-se a simplicidade e a clareza na apresentação das informações, minimizando a curva de aprendizagem para novos utilizadores.

Este design foi concebido com foco na experiência do utilizador (UX), garantindo que as operações na aplicação sejam realizadas de forma rápida e sem complicações. Os *mockups* servem como base para a implementação das interfaces desenvolvidas utilizando as tecnologias descritas no capítulo anterior.

## 4.9 Conclusão

A planificação descrita neste capítulo fornece uma estrutura clara e detalhada para a realização do estágio. A organização das tarefas, o plano de execução e a definição de requisitos permitem uma abordagem rigorosa e eficiente ao desenvolvimento do projeto **PGIP**. Este planeamento foi ajustado, sempre que necessário, durante o estágio, de forma a responder a imprevistos ou novas necessidades identificadas.

## Capítulo 5

# Implementação e Testes

### 5.1 Introdução

Este capítulo apresenta o processo de implementação e validação do sistema **PGIP**, desenvolvido durante o estágio. A implementação das funcionalidades foi orientada pelas *User Stories* definidas e pelo modelo arquitetônico previamente estabelecido. Para além da descrição técnica da arquitetura e das funcionalidades implementadas, é também abordado o fluxo de utilização da aplicação, bem como os testes realizados para garantir a robustez e fiabilidade do sistema.

O conteúdo deste capítulo reflete a transição do planeamento para a execução prática, documentando os principais desenvolvimentos técnicos, a abordagem seguida, os desafios enfrentados e as medidas adotadas para assegurar o correto funcionamento da aplicação.

### 5.2 Visão Geral do Sistema

O sistema **PGIP** foi desenvolvido com o objetivo de otimizar a gestão de pedidos, facilitando a comunicação entre os diferentes intervenientes do processo. Durante o estágio, o foco principal esteve na implementação da interface do utilizador e na integração com os sistemas externos, garantindo uma experiência intuitiva e eficiente para os utilizadores finais.

A implementação seguiu uma abordagem modular, permitindo uma escalabilidade futura do sistema. A plataforma foi concebida para operar em ambiente web, assegurando compatibilidade com diferentes dispositivos e navegadores.

### 5.3 Arquitetura do Sistema

A arquitetura da plataforma foi concebida em camadas distintas, conforme está ilustrado na Figura A.33 o Anexo A.33.

- **Front-end** — Implementado em *Angular* e *TypeScript*, garantindo um desenvolvimento estruturado e eficiente da interface gráfica;
- **Back-end** — Desenvolvido com *Laravel* (PHP), servindo como camada de lógica de negócio e responsável pela comunicação com a base de dados;
- **Base de Dados** — Utilização de *MySQL*, permitindo a persistência estruturada das informações do sistema;

- Integração com Sistemas Externos — Comunicação com *SAP* e *Microsoft Azure*, garantindo a sincronização dos dados empresariais, e utilização do *SMSP* da *NOS* para envio de SMS e SMTP do cliente para envio de emails.

## 5.4 Relacionamento entre *User Stories* e Funcionalidades Implementadas

O desenvolvimento da plataforma PGIP seguiu um conjunto de *User Stories*, que definiram os requisitos funcionais do sistema. Essas *User Stories* foram fundamentais para a estruturação e implementação das funcionalidades, permitindo um desenvolvimento iterativo e alinhado com as necessidades do utilizador final.

Cada *User Story* detalha uma funcionalidade específica que foi implementada na plataforma, cobrindo áreas como autenticação, gestão de utilizadores, operações sobre pedidos e integração com sistemas externos. A relação entre as *User Stories* e as funcionalidades implementadas pode ser consultada no Anexo A.1, onde cada história está associada à funcionalidade correspondente na plataforma.

Dessa forma, assegurou-se que cada funcionalidade desenvolvida atendia a um requisito real, garantindo uma cobertura completa das necessidades do sistema e proporcionando um fluxo de utilização coerente para os utilizadores.

## 5.5 Fluxo de Utilização da Aplicação

O PGIP segue um fluxo bem definido para o tratamento de pedidos. O fluxo principal da aplicação pode ser descrito da seguinte forma:

### 5.5.1 Criação de Pedido

O utilizador (requerente) inicia um novo pedido através da interface gráfica, seleccionando a tipologia desejada e preenchendo as informações relevantes no formulário de criação de pedido.

### 5.5.2 Submissão do Pedido

Após o preenchimento, o pedido é submetido para aprovação. O sistema verifica a validade das informações e exibe um feedback para o utilizador.

### 5.5.3 Aprovação do Pedido

O gestor de rota ou responsável analisa o pedido e decide se aprova ou rejeita. Caso aprovado, o pedido passa para o estado de "Agendado", ficando pronto para execução.

## PGIP

### 5.5.4 Execução do Pedido

Nesta fase, o pedido é tratado e acompanhado pelo sistema. Caso necessário, pode ser reagendaado ou atualizado conforme o progresso da execução.

### 5.5.5 Conclusão do Pedido

Após a execução bem-sucedida do serviço, o pedido é finalizado e marcado como "Concluído". O sistema pode gerar um relatório final com os detalhes do serviço prestado.

## 5.6 Funcionalidades Desenvolvidas

Nesta secção são descritas as principais funcionalidades desenvolvidas durante o estágio. A comunicação do sistema com a plataforma SAP para obtenção de dados relevantes, garantindo um fluxo contínuo de informações. A autenticação dos utilizadores internos é realizada via *Microsoft Azure Active Directory*.

### 5.6.1 Interface do Utilizador

A interface do utilizador foi desenhada para ser responsiva e intuitiva. Foi utilizado o *Figma* para prototipagem, garantindo um alinhamento com as melhores práticas de UI/UX. Algumas das páginas implementadas incluem:

- Página de Login (ver Figura 5.1) — Permite autenticação segura através de *OAuth2*;
- Gestão de Pedidos — Exibição, criação e edição de pedidos de serviço;
- Notificações — Sistema de alertas por e-mail e SMS para eventos críticos.



Figura 5.1: Página de *Login*.

## 5.6.2 Principais Funcionalidades Desenvolvidas pelo Estagiário

Dentro destas várias funcionalidades, o foco de algumas atividades desenvolvidas foi focado no desenvolvimento de tarefas como as seguintes:

- **Menu de Tipologias de Pedidos** (ver Figura A.35, incluída no Anexo A.3) — Interface para gestão e consulta das diferentes tipologias no sistema;
- **Criação de uma Tipologia de Pedido** (ver Figura A.36, incluída no Anexo A.3) — Formulário de criação com campos configuráveis;
- **Edição de uma Tipologia de Pedido** (ver Figura A.37, incluída no Anexo A.3) — Permite modificar regras e dados associados a tipologias existentes;
- **Detalhes do Pedido** (ver Figura A.38, incluída no Anexo A.3) — Página com as informações completas do pedido;
- **Detalhes do Fornecedor do Pedido** (ver Figura A.39, incluída no Anexo A.3) — Visualização consolidada dos dados do fornecedor;
- **Geração de Guia de Transporte** (ver Figura A.40, incluída no Anexo A.3) — Interface para gerar a guia associada ao pedido;
- **Detalhes da Guia do Pedido** (ver Figura A.41, incluída no Anexo A.3) — Visualização detalhada da guia gerada;
- **Histórico do Pedido** (ver Figura A.42, incluída no Anexo A.3) — Registo cronológico de alterações e ações executadas;
- **Formulário de Fornecedores** (ver Figuras A.43, A.44 e A.45, incluídas no Anexo A.3) — Formulário dividido em três partes para inserção completa dos dados dos fornecedores;
- **Gestão de Permissões** (ver Figuras A.46 e A.47, incluídas no Anexo A.3) — Interface para atribuição e revisão de permissões por perfil;
- **Ficheiro de Tradução** — Não acompanhado de figura, mas incluído na entrega final do sistema como recurso auxiliar de internacionalização.

## 5.7 Implementação Técnica

A implementação seguiu boas práticas de desenvolvimento, incluindo:

- Utilização de componentes reutilizáveis em *Angular*;
- Estruturação do back-end em *Laravel*, seguindo o padrão MVC (*Model - View - Controller*);
- Implementação de APIs REST para comunicação entre as diferentes camadas do sistema;

## PGIP

- Adoção de práticas de segurança, incluindo encriptação de dados sensíveis.

### 5.7.1 Descrição do *Happy Flow*

O *Happy Flow* representa a sequência ideal de ações realizadas na plataforma **PGIP**, desde a criação de um pedido até à sua conclusão, envolvendo os perfis de Requerente, Gestor de Rota e Gestor de Fornecedor. O fluxo ideal de execução pode ser visualizado no diagrama de atividade presente no Anexo A.34, onde se representa a sequência de ações entre os diferentes intervenientes.

O processo inicia-se com o Requerente, que acede à lista de pedidos disponíveis e inicia a criação de um novo pedido, selecionando a tipologia pretendida. De seguida, o formulário de pedido é preenchido com os dados necessários, como o tipo de contentor, código LER, estabelecimento, entre outros campos relevantes. Após a validação e confirmação dos dados inseridos, o pedido é submetido com sucesso e apresentado com o respetivo estado na interface do Requerente (ver Figuras A.3 a A.9, no Anexo A, Secção A.1, Subsecção A.1.1).

Posteriormente, o Gestor de Rota acede aos pedidos submetidos e consulta os seus detalhes para validação e continuação do processo. Após a validação, o Gestor seleciona a zona correspondente e o fornecedor a associar ao pedido, verifica a disponibilidade do fornecedor e procede ao agendamento do serviço. O sistema confirma o envio da notificação ao fornecedor. Após este processo, o estado atualizado do pedido fica visível na interface do Gestor de Rota (ver Figuras A.10 a A.17, no Anexo A, Secção A.1, Subsecção A.1.1).

O Gestor de Fornecedor visualiza os pedidos atribuídos à sua entidade através da interface dedicada, podendo proceder ao agendamento e execução do serviço (ver Figuras A.18 a A.23, no Anexo A, Secção A.1, Subsecção A.1.1).

Por fim, cada interveniente tem acesso ao resumo do estado final do pedido. O Requerente visualiza o pedido como agendado e validado nas diferentes abas: detalhe, fornecedor e histórico. O Gestor de Rota confirma o sucesso da operação com o detalhe do pedido, fornecedor e histórico. Por sua vez, o Gestor de Fornecedor consulta toda a informação necessária para execução do serviço nas mesmas abas (ver Figuras A.24 a A.32, no Anexo A, Secção A.1, Subsecção A.1.1).

Tendo em conta a complexidade funcional e a interação entre perfis, torna-se essencial garantir que o sistema permanece robusto e seguro, conforme explorado na secção seguinte.

### 5.7.2 Considerações de Segurança

Durante o desenvolvimento, foram validadas as medidas de segurança aplicadas, nomeadamente no que diz respeito à proteção contra ataques como *SQL Injection* e *Cross-Site Request Forgery (CSRF)*. As funcionalidades implementadas foram sujeitas a testes manuais com entradas maliciosas, de forma a verificar a integridade da aplicação perante tentativas de injeção de código e submissões externas não autorizadas.

A utilização de *prepared statements*, providenciada pelo *framework* Laravel, assegura a execução segura de operações com base de dados, enquanto a geração automática de *tokens CSRF* protege os formulários da aplicação contra requisições forjadas. Estas medidas de-

monstraram-se eficazes durante os testes, reforçando a fiabilidade e robustez da plataforma PGIP.

## 5.8 Testes e Validação

Para garantir a fiabilidade do sistema, foram realizados diversos testes funcionais e de integração, com o objetivo de assegurar o correto funcionamento de cada funcionalidade desenvolvida. A Tabela 5.1 apresenta a descrição detalhada dos testes realizados durante o estágio.

Tabela 5.1: Sumário dos testes realizados no sistema PGIP.

Teste	Descrição	Resultado
Realização do Login	Validação do mecanismo de autenticação para utilizadores internos (via AD) e externos (via e-mail/password)	Sucesso
Criação de um Pedido	Verificação da criação de pedidos a partir do preenchimento de formulários dinâmicos	Sucesso
Agendamento de um Pedido	Teste à atribuição de datas para execução do pedido por parte do gestor	Sucesso
Execução de um Pedido	Validação da funcionalidade de registo da execução do pedido e suas atualizações de estado	Sucesso
Finalização de um Pedido	Verificação do processo de encerramento do pedido com estado "Concluído"	Sucesso
Criação da Guia de Transporte	Teste à geração automática da guia de transporte associada ao pedido	Sucesso
Criação de uma Tipologia de Pedido	Validação do registo de novas tipologias, com respetivos formulários associados	Sucesso
Edição de uma Tipologia de Pedido	Teste à alteração de tipologias existentes e dos seus respetivos campos	Sucesso
Criação de um Fornecedor	Teste ao registo de um novo fornecedor na plataforma	Sucesso
Edição de um Fornecedor	Validação da edição de dados de um fornecedor existente	Sucesso
Criação de um Estabelecimento	Verificação do registo de um novo estabelecimento associado a uma entidade	Sucesso
Edição de um Estabelecimento	Teste à edição de dados de estabelecimentos já registados	Sucesso
Notificações	Verificação do envio de notificações automáticas (e-mail e/ou SMS) aos intervenientes do processo	Sucesso
Criação de um Utilizador	Teste ao registo de novos utilizadores (internos e externos) com definição de permissões	Sucesso

Continua na próxima página

Tabela 5.1 – continuação da página anterior

<b>Teste</b>	<b>Descrição</b>	<b>Resultado</b>
Edição de um Utilizador	Validação da edição de dados e permissões de utilizadores existentes	Sucesso
Geração de uma Nota de Encomenda	Teste à criação automática da nota de encomenda com base na execução do pedido	Dificuldade

Os testes demonstraram a robustez do sistema, garantindo que todas as funcionalidades críticas operam conforme o esperado, com a exceção do teste referente à “Geração de uma Nota de Encomenda” em que não foi possível conseguir testar corretamente devido a dados em falta.

## **5.9 Conclusão**

O desenvolvimento do sistema **PGIP** representou um desafio enriquecedor, permitindo aplicar conhecimentos adquiridos ao longo do percurso académico. A implementação das funcionalidades e a integração com sistemas externos proporcionaram um ambiente de trabalho alinhado com as necessidades do Cliente, garantindo um produto final eficiente e escalável.



## Capítulo 6

### Conclusão

#### 6.1 Introdução

Este capítulo apresenta uma síntese dos principais resultados obtidos durante o estágio, refletindo sobre os objetivos definidos, as dificuldades enfrentadas e as aprendizagens adquiridas. Procura ainda contextualizar o impacto do trabalho desenvolvido e apontar possíveis caminhos para a continuação e melhoria do projeto **PGIP**.

#### 6.2 Conclusões Principais

O presente relatório descreveu o decorrer do estágio curricular realizado na empresa **Latitude**, no âmbito do projeto **PGIP**. Esta fase final do estágio complementou a delimitação do problema, a seleção de metodologias, o levantamento de requisitos e a planificação detalhada das atividades a desenvolver, com a implementação e validação das funcionalidades desenvolvidas.

As principais conclusões desta fase são as seguintes:

- O levantamento de requisitos funcionais e técnicos permitiu delimitar o âmbito do projeto, garantindo um alinhamento com as necessidades do cliente;
- A escolha de tecnologias como *Angular* e *Laravel* demonstrou-se adequada às exigências do sistema, permitindo a criação de uma aplicação web modular e escalável;
- A metodologia ágil *Scrum* revelou-se eficaz na adaptação a novas exigências, garantindo um desenvolvimento iterativo e uma melhor comunicação com a equipa;
- A análise de riscos revelou desafios que impactaram o desenvolvimento, mas a definição de estratégias de mitigação permitiu minimizar atrasos e assegurar o progresso das tarefas;
- O estágio proporcionou um ambiente de aprendizagem contínua, onde a resolução de problemas e a colaboração com a equipa foram elementos centrais.

Com base no trabalho desenvolvido, conclui-se que a implementação do **PGIP** atingiu os objetivos definidos, garantindo uma plataforma funcional e alinhada com as necessidades do cliente.

## 6.3 Objetivos Iniciais e Resultados Obtidos

Desde o início do estágio, foram estabelecidos objetivos claros relativamente às funcionalidades a serem implementadas. A Tabela 6.1 apresenta um resumo da correspondência entre os objetivos iniciais e os resultados efetivamente alcançados.

ID	Objetivos Iniciais	Resultados Obtidos
USo1	Autenticação através do AD	Implementado conforme especificado
USo2	Registo de utilizadores	Implementado com validação
USo3	Gestão de permissões de utilizadores	Implementado com interface intuitiva
USo6	Gestão de tipologias de pedidos	Implementado com suporte a personalização
US10	Associação de utilizadores a fornecedores	Implementado e testado com sucesso
US16	Criação de pedidos de serviço	Implementado e integrado com SAP
US22	Conclusão de pedidos	Implementado e testado

Tabela 6.1: Comparação entre os objetivos iniciais e os resultados obtidos.

A análise demonstra que os principais objetivos definidos foram cumpridos, com algumas melhorias introduzidas ao longo do desenvolvimento.

## 6.4 Dificuldades e Aprendizagens

O desenvolvimento do projeto não esteve isento de desafios. Durante o estágio, surgiram algumas dificuldades que exigiram adaptações e redefinições estratégicas:

### 6.4.1 Dificuldades

Podem-se identificar as maiores dificuldades enfrentadas neste estágio da seguinte forma:

- **Falta de resposta atempada por parte do cliente** — Algumas decisões de implementação foram atrasadas devido à necessidade de validação externa;
- **Alterações tardias nos requisitos** — Algumas funcionalidades tiveram de ser ajustadas para responder a mudanças nos requisitos, o que exigiu reavaliações frequentes;
- **Integração com sistemas externos** — A comunicação com a plataforma SAP exigiu uma adaptação às restrições impostas pelos serviços disponíveis;
- **Gestão de prioridades** — A distribuição de tarefas nem sempre refletiu a complexidade real de cada funcionalidade, resultando na necessidade de ajustes na planificação.

### 6.4.2 Aprendizagens

As principais lições aprendidas no final deste estágio foram:

- **Trabalho em equipa e metodologias ágeis** — A adoção do *Scrum* facilitou a organização das tarefas e a comunicação entre os membros da equipa;
- **Capacidade de adaptação** — O desenvolvimento de software exige uma flexibilidade constante para responder a mudanças nos requisitos e imprevistos técnicos;

## PGIP

- **Resolução de problemas** — A necessidade de integrar múltiplos sistemas e garantir a segurança dos dados levou à exploração de novas abordagens para superar desafios técnicos;
- **Importância da documentação** — A documentação clara das funcionalidades e requisitos provou ser essencial para manter a coesão do projeto.

### 6.5 Reflexão sobre o Estágio

A realização deste estágio representou uma oportunidade de crescimento profissional e acadêmico, proporcionando uma experiência prática em ambiente real de desenvolvimento de software.

Este percurso permitiu consolidar conhecimentos adquiridos ao longo da formação acadêmica e aplicar novas competências, particularmente na utilização de frameworks modernas, no desenvolvimento de interfaces responsivas e na comunicação com sistemas externos.

Além disso, o estágio revelou a importância da gestão de tempo e da colaboração entre equipes, destacando a necessidade de um planejamento cuidadoso e de um acompanhamento contínuo para garantir o sucesso de um projeto tecnológico.

Esta experiência contribuiu para o desenvolvimento da minha autonomia, sentido de responsabilidade e espírito crítico. Lidar com restrições externas e tomar decisões em contexto de incerteza reforçou a importância da comunicação eficaz e da proatividade no seio da equipe. Em resultado deste percurso, considero-me mais preparado para integrar equipes de desenvolvimento profissional, não apenas pelo domínio técnico consolidado, mas pela experiência adquirida na resolução de problemas reais ao longo de um ciclo de desenvolvimento completo.

### 6.6 Possíveis Melhorias Futuras e Continuação do Projeto

Apesar dos progressos alcançados, há sempre margem para evolução. Algumas melhorias e possíveis desenvolvimentos futuros incluem:

- **Otimização do desempenho** — Algumas consultas à base de dados podem ser melhoradas para reduzir tempos de resposta;
- **Expansão das funcionalidades** — Poderiam ser introduzidas novas opções de personalização dos pedidos e relatórios mais detalhados para os utilizadores;
- **Automação de processos** — A implementação de fluxos automatizados para validação e execução de pedidos poderia reduzir a necessidade de intervenção manual;
- **Melhor integração com o SAP** — A comunicação entre os sistemas pode ser refinada para minimizar redundâncias e otimizar o fluxo de informação;
- **Maior usabilidade** — Pequenos ajustes no design da interface podem melhorar ainda mais a UX.

## **6.7 Conclusão**

O desenvolvimento do projeto **PGIP** demonstrou a viabilidade de uma solução integrada para a gestão de pedidos, permitindo a automatização de processos e a melhoria da eficiência organizacional.

Os desafios enfrentados ao longo do estágio reforçaram a importância de uma abordagem flexível e bem estruturada, onde a adaptação a novas exigências e a capacidade de resolver problemas técnicos desempenham um papel crucial.

A experiência adquirida durante esta fase será essencial para desafios futuros, consolidando conhecimentos na área do desenvolvimento de software e integração de sistemas empresariais. O projeto tem um grande potencial de evolução e a sua continuidade poderá trazer benefícios adicionais à organização, promovendo uma experiência de utilizador mais eficaz e uma gestão otimizada dos processos internos.

## Bibliografia

- [1] Google. (2023) Angular documentation. Último acesso a 14 de dezembro de 2024. [Online]. Available: <https://angular.io/docs> 6
- [2] SAP. (2023) Sap integration suite. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://www.sap.com/portugal/products/technology-platform/integration-suite.html> 6
- [3] K. Schwaber and J. Sutherland. (2020) The scrum guide. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://www.scrum.org/resources/scrum-guide> 6
- [4] T. P. Group. (2025) Php documentation. Último acesso a 27 de Maio de 2025. [Online]. Available: [https://www.w3schools.com/sql/sql\\_injection.asp](https://www.w3schools.com/sql/sql_injection.asp) 7
- [5] OWASP. (2025) Cross site request forgery (csrf). Último acesso a 27 de Maio de 2025. [Online]. Available: <https://owasp.org/www-community/attacks/csrf> 7
- [6] A. Kareliya. (2025) Vue vs angular: Which one is the better framework? Último acesso a 26 de Maio de 2025. [Online]. Available: <https://radixweb.com/blog/vuejs-vs-angular> 7
- [7] V. Community. (2025) Comparison with other frameworks - angular (formerly known as angular 2). Último acesso a 26 de Maio de 2025. [Online]. Available: <https://v2.vuejs.org/v2/guide/comparison?redirect=true#AngularJS-Angular-1> 7, 8
- [8] Microsoft. (2025) Visual studio code documentation. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://code.visualstudio.com> 9
- [9] O. Corporation. (2025) Mysql documentation. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://dev.mysql.com/doc/> 9, 10
- [10] I. Figma. (2025) Figma: The collaborative interface design tool. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://www.figma.com> 10
- [11] M. D. Network. (2025) Html: Hypertext markup language documentation. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML> 11
- [12] ——. (2025) Css: Cascading style sheets documentation. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS> 11
- [13] Microsoft. (2025) Typescript documentation. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://www.typescriptlang.org/docs/> 11

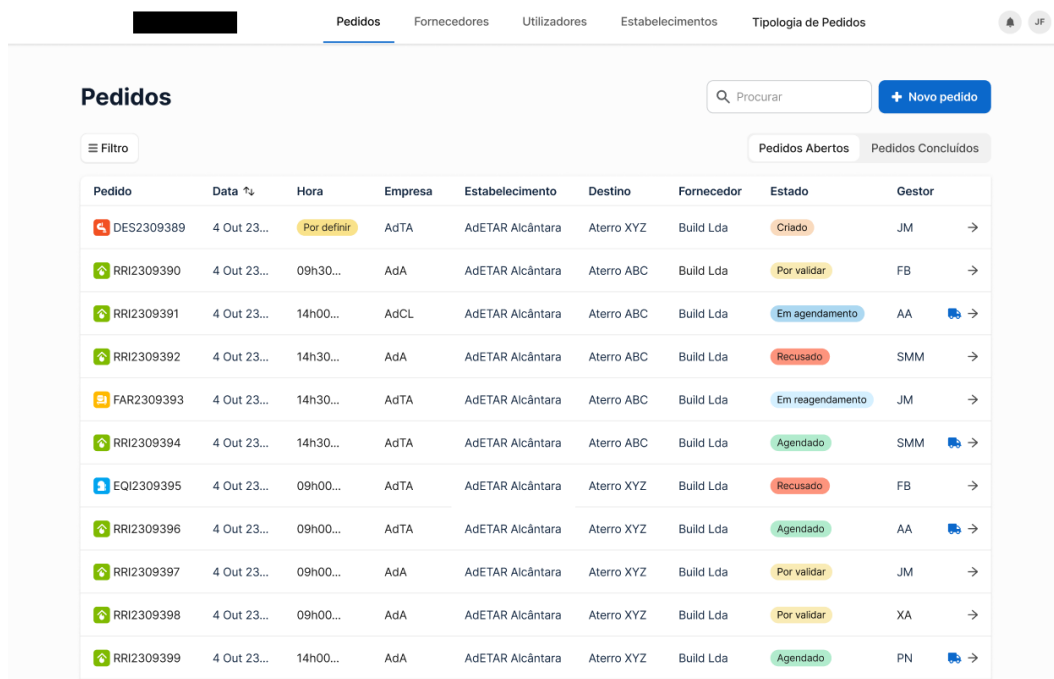
- [14] T. P. Group. (2025) Php documentation. Último acesso a 13 de dezembro de 2024. [Online]. Available: <https://www.php.net/docs.php> 11
- [15] P. Informatics. (2023) Primeng documentation. Último acesso a 14 de dezembro de 2024. [Online]. Available: <https://primeng.org/installation> 12
- [16] T. Otwell. (2025) Laravel documentation. Último acesso a 14 de dezembro de 2024. [Online]. Available: <https://laravel.com/docs> 12

# Apêndice A

## Anexos

### A.1 Mockups

Os *mockups* apresentados nesta secção estão a ser desenvolvidos durante a fase de design do projeto **PGIP** e são referidos na Secção 5.3 do relatório principal. Estes representam algumas das interfaces principais da aplicação e funcionam como referência visual para a implementação do sistema.



Pedido	Data	Hora	Empresa	Estabelecimento	Destino	Fornecedor	Estado	Gestor
DES2309389	4 Out 23...	Por definir	AdTA	AdETAR Alcântara	Aterro XYZ	Build Lda	Criado	JM →
RRI2309390	4 Out 23...	09h30...	AdA	AdETAR Alcântara	Aterro ABC	Build Lda	Por validar	FB →
RRI2309391	4 Out 23...	14h00...	AdCL	AdETAR Alcântara	Aterro ABC	Build Lda	Em agendamento	AA →
RRI2309392	4 Out 23...	14h30...	AdA	AdETAR Alcântara	Aterro ABC	Build Lda	Recusado	SMM →
FAR2309393	4 Out 23...	14h30...	AdTA	AdETAR Alcântara	Aterro ABC	Build Lda	Em reagendamento	JM →
RRI2309394	4 Out 23...	14h30...	AdTA	AdETAR Alcântara	Aterro ABC	Build Lda	Agendado	SMM →
EQI2309395	4 Out 23...	09h00...	AdTA	AdETAR Alcântara	Aterro XYZ	Build Lda	Recusado	FB →
RRI2309396	4 Out 23...	09h00...	AdTA	AdETAR Alcântara	Aterro XYZ	Build Lda	Agendado	AA →
RRI2309397	4 Out 23...	09h00...	AdA	AdETAR Alcântara	Aterro XYZ	Build Lda	Por validar	JM →
RRI2309398	4 Out 23...	09h00...	AdA	AdETAR Alcântara	Aterro XYZ	Build Lda	Por validar	XA →
RRI2309399	4 Out 23...	14h00...	AdA	AdETAR Alcântara	Aterro XYZ	Build Lda	Agendado	PN →

Figura A.1: Página Principal - Pedidos (Visão Administrador). Este *mockup* demonstra a visão administrativa, com a visão de todos os *sub-menus*, permitindo a gestão centralizada de pedidos e o acompanhamento do seu estado.

The image shows a web application interface. On the left, there is a table titled 'Fornecedores' with columns: Código APA, Código SAP, Empresa, Estabelecimento, and UR. The table contains 10 rows of data. On the right, there is a modal form titled 'Adicionar Fornecedor' with the subtitle 'Preencha os dados do fornecedor'. The form contains several input fields and dropdown menus for adding a new supplier.

Código APA	Código SAP	Empresa	Estabelecimento	UR
PEMGI	JD7269	AdTA	AdETAR Alcântara	5
VMFKV	KB8261	AdA	AdETAR Alcântara	10
KENAB	AH8274	AdCL	AdETAR Alcântara	7
DPQND	PQ7251	AdA	AdETAR Alcântara	23
HPWAH	YV8462	AdTA	AdETAR Alcântara	60
DPWNT	NH8274	AdTA	AdETAR Alcântara	8
WUFN	AY7361	AdTA	AdETAR Alcântara	9
GQYAP	PQ8471	AdTA	AdETAR Alcântara	17
EOAHD	CNS785	AdA	AdETAR Alcântara	11
OPHWY	UI9173	AdA	AdETAR Alcântara	19

**Adicionar Fornecedor** ×  
Preencha os dados do fornecedor

**Código\***

**NIF\***

**Nome\***

**Telefone\***

**Morada\***

**Código postal\***

**Responsável**

**Contratos\***

**Estado\***

Figura A.2: Formulário para Adicionar Novo Fornecedor. Este *mockup* detalha o processo de registo de novos fornecedores no sistema, garantindo a consistência dos dados recolhidos.

# PGIP

## A.1.1 Happy Flow

Esta secção anexa apresenta uma visão prática do *happy flow* da aplicação **PGIP**, ou seja, o percurso ideal de utilização sem erros ou interrupções. As imagens documentam, de forma sequencial, os passos realizados por diferentes perfis de utilizador — Requerente, Gestor de Rota e Gestor de Fornecedor — desde a criação de um pedido até à sua conclusão.

### Perspetiva Requerente

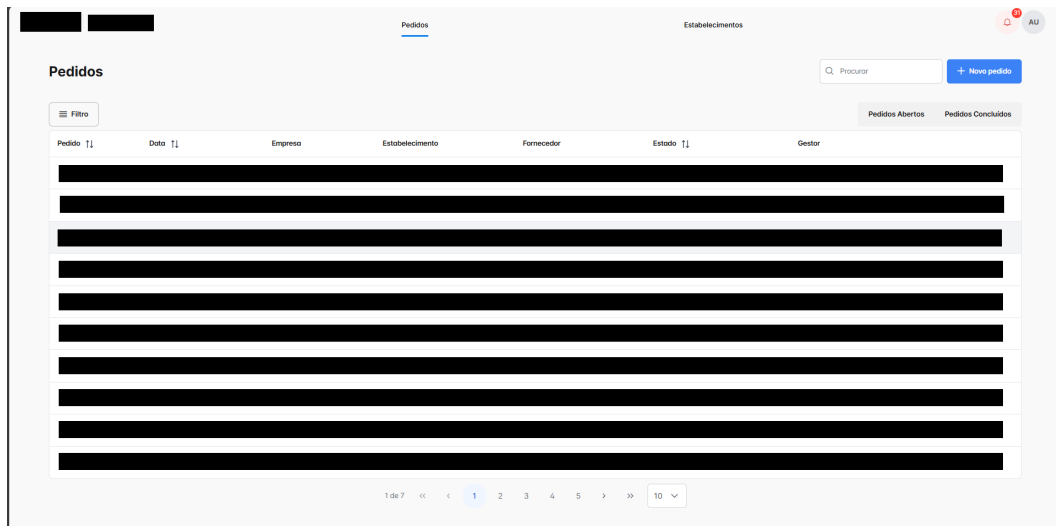


Figura A.3: Lista de pedidos disponíveis para o Requerente, com possibilidade de criar novo pedido.

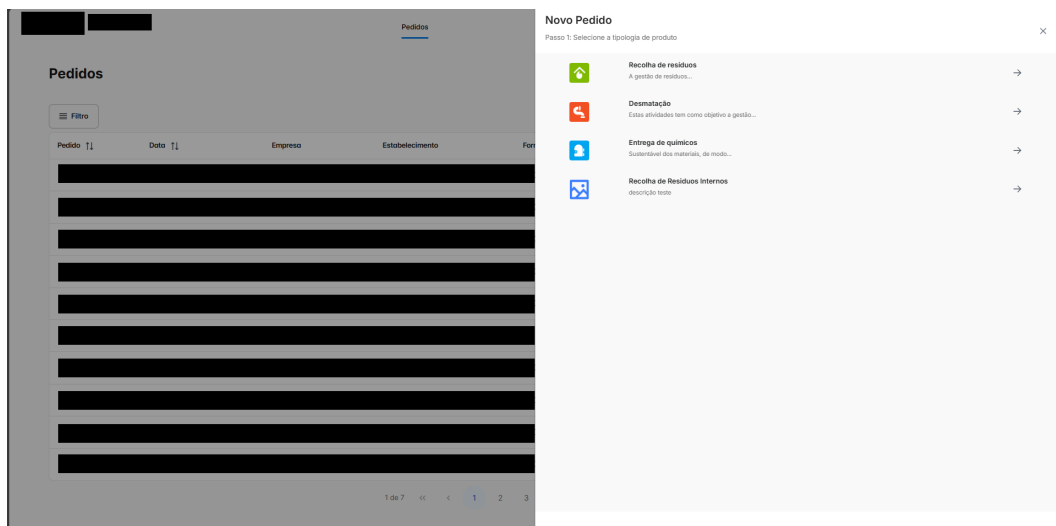


Figura A.4: Seleção da tipologia do serviço a requisitar, por parte do Requerente.

**Novo Pedido**  
Passo 2: Preencha as informações

**Recolha de resíduos**  
Estabelecimento:\*  
Selecione um estabelecimento

Data de Recolha:\*      Ate:\*  
[Redacted]      [Redacted]

**Informação do Serviço**  
Quantidade Kg  
[Redacted]

Código LER\*  
Selecione um

Tipo de Contendor  
Selecione um

Código de operação  
Selecione um

Destino Externo  
Selecione um

1 de 7    <<    <    1    2    3

[← Voltar](#)      [Seguinte →](#)

Figura A.5: Preenchimento dos dados detalhados do pedido, como quantidade, contendor e operação.

**Novo Pedido**  
Passo 2: Preencha as informações

**Recolha de resíduos**  
Estabelecimento:\*  
Estabelecimento AN1

Data de Recolha:\*      Ate:\*  
02/04/2025      04/04/2025

**Informação do Serviço**  
Quantidade Kg  
12345

Código LER\*  
010504 - Lamas e outros resíduos de perfuração, contendo água doce

Tipo de Contendor  
Saco

Código de operação  
R2 - Recuperação/Regeneração de solventes

Destino Externo  
APA00047987 - Estabelecimento Externo

1 de 7    <<    <    1    2    3

[← Voltar](#)      [Seguinte →](#)

Figura A.6: Confirmação final de todos os dados do pedido antes da submissão.

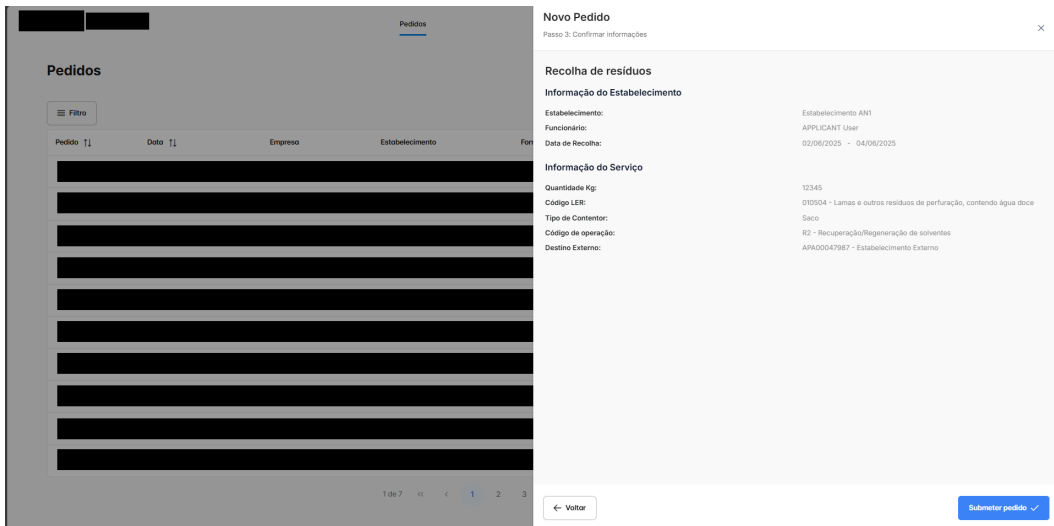


Figura A.7: Notificação de sucesso após a submissão do pedido.

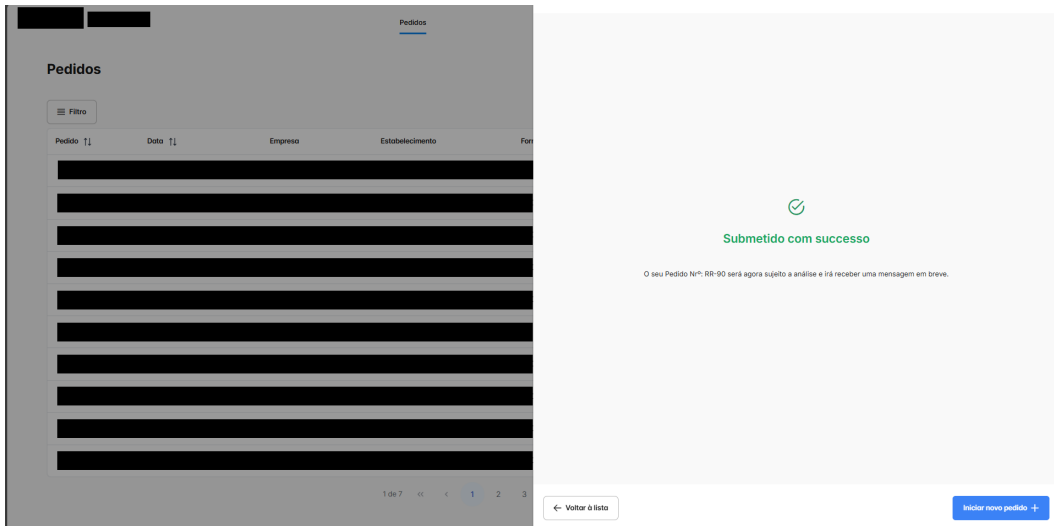


Figura A.8: Visualização do estado atualizado do pedido submetido.

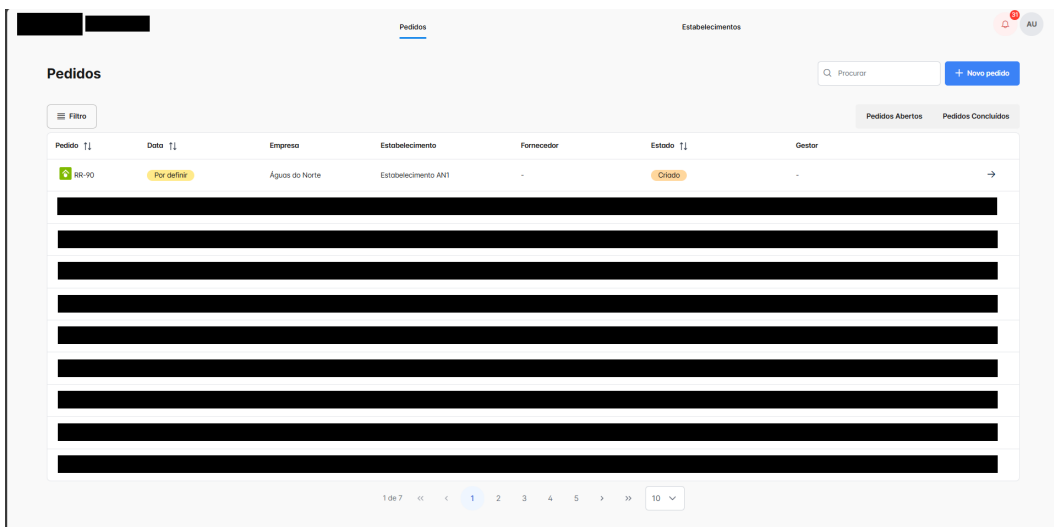


Figura A.9: Lista de pedidos disponíveis para o Requerente, com possibilidade de ver o pedido criado.

Perspetiva Gestor de Rota

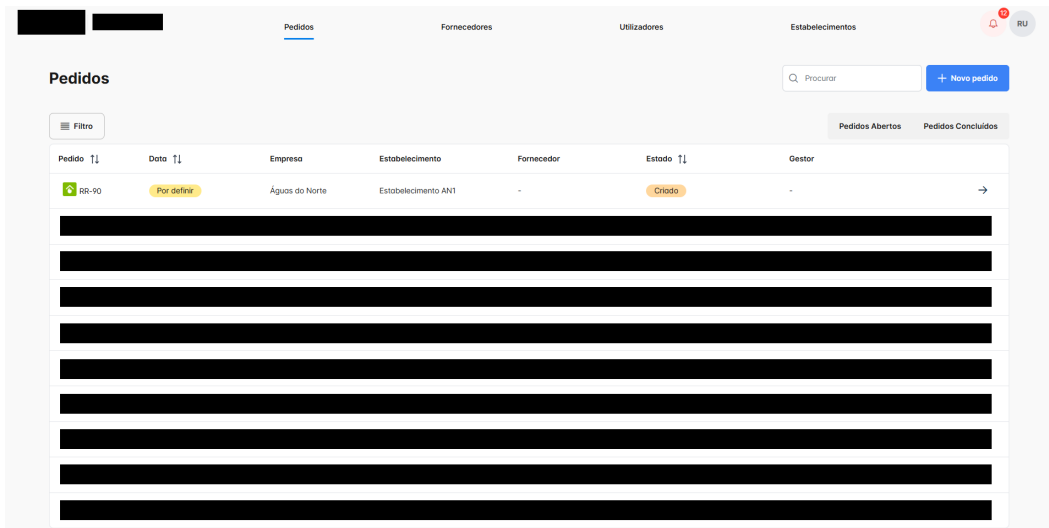


Figura A.10: Lista de pedidos visível ao Gestor de Rota, incluindo os que estão por agendar.

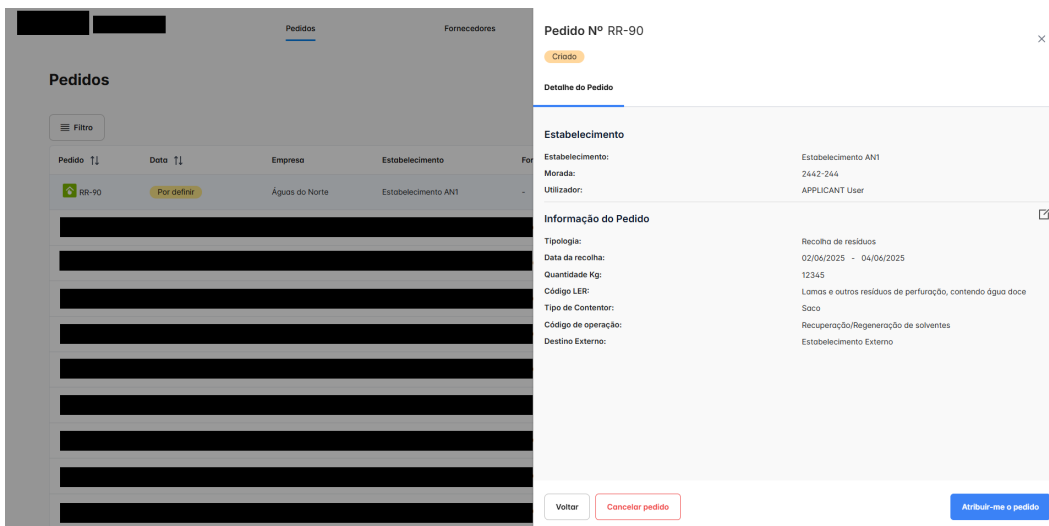


Figura A.11: Seleção da tipologia e início do processo de validação do pedido.

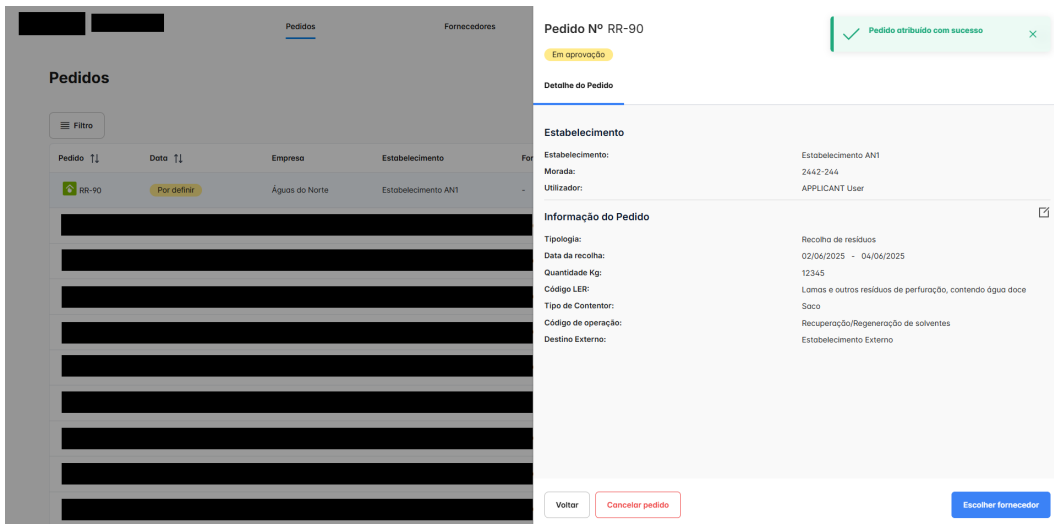


Figura A.12: Visualização dos detalhes do pedido e informação do Estabelecimento.

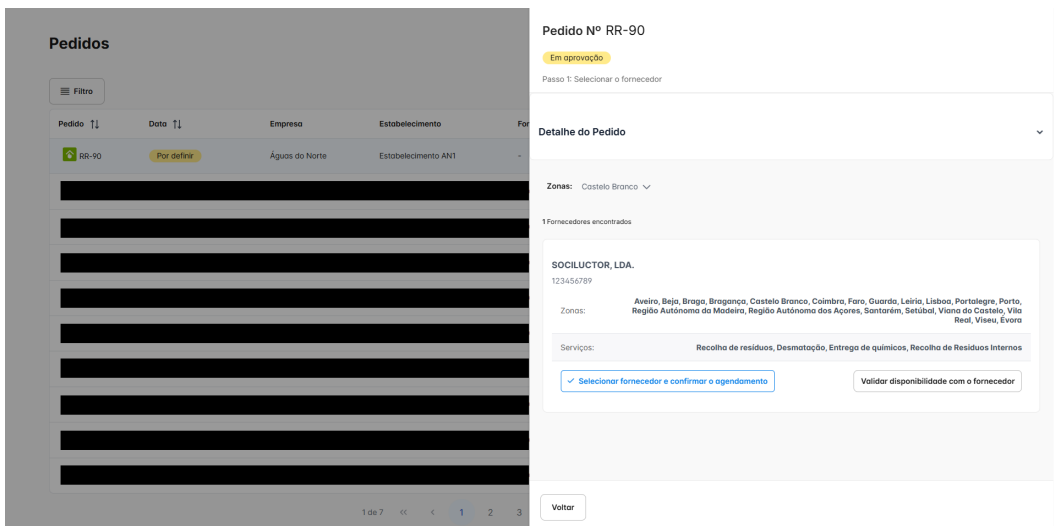


Figura A.13: Seleção da zona e fornecedor disponível para o pedido.

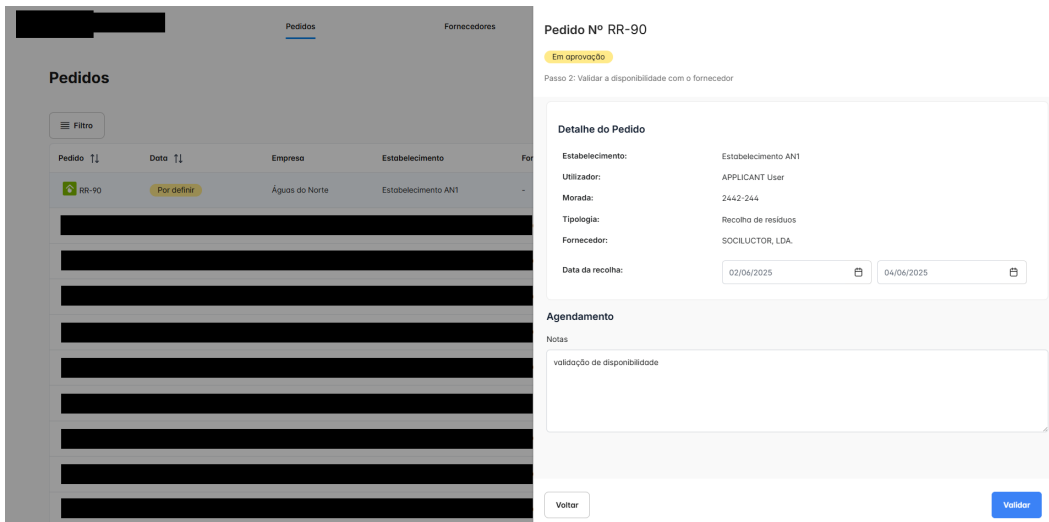


Figura A.14: Validação de disponibilidade do fornecedor e agendamento do serviço.

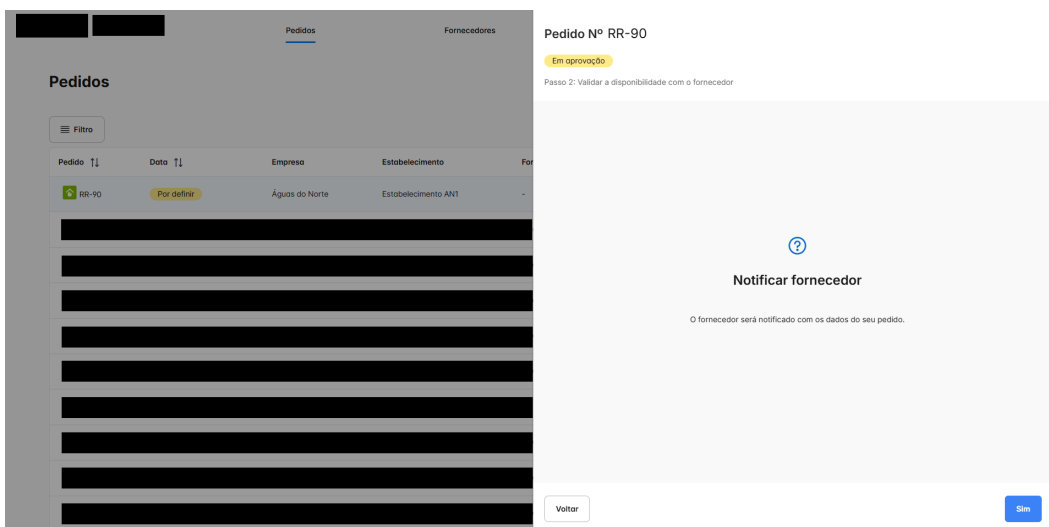


Figura A.15: Confirmação de notificação ao fornecedor com todos os dados do pedido.

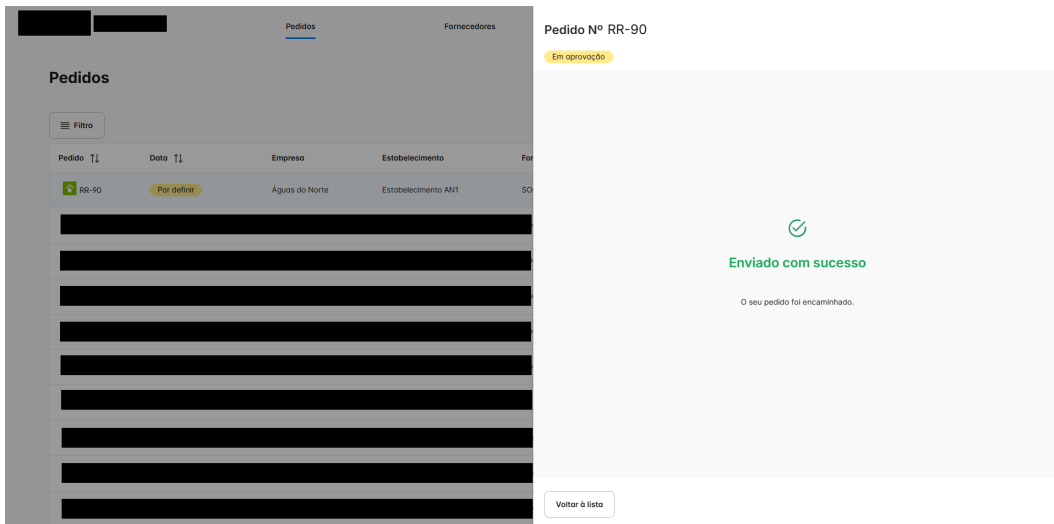


Figura A.16: Confirmação de notificação com sucesso ao fornecedor com todos os dados do pedido.

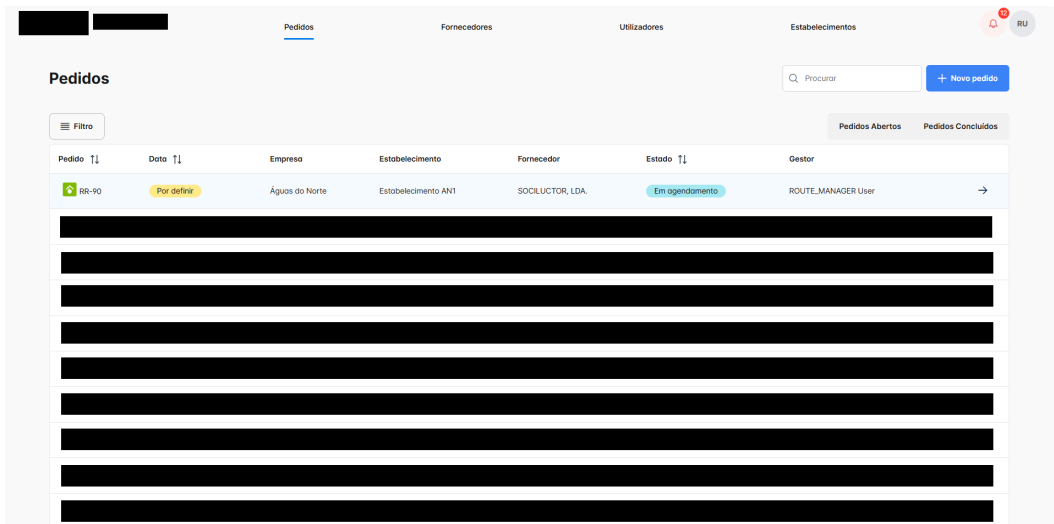


Figura A.17: Visualização do pedido após o agendamento com o fornecedor.

Perspetiva Gestor de Fornecedor

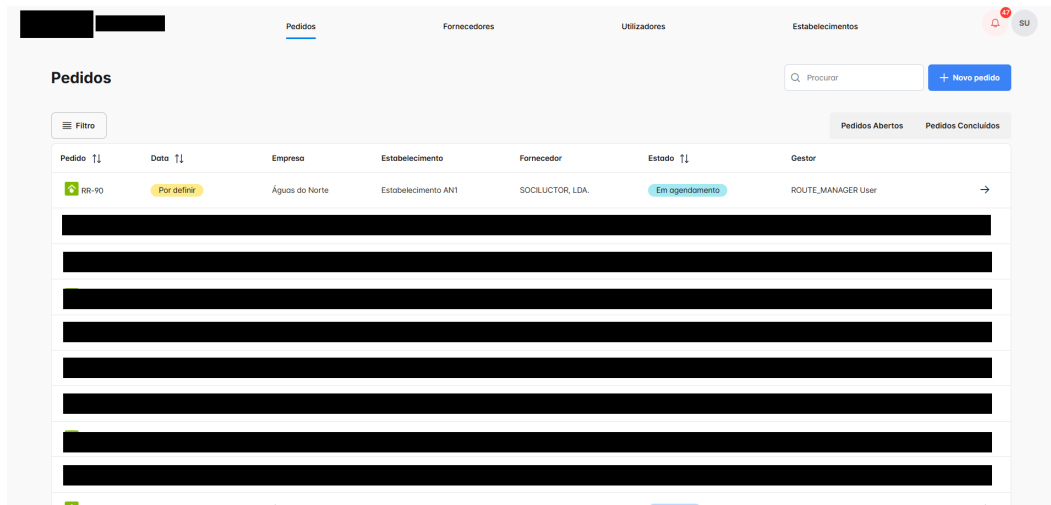


Figura A.18: Lista de pedidos atribuídos visível ao Gestor de Fornecedor.

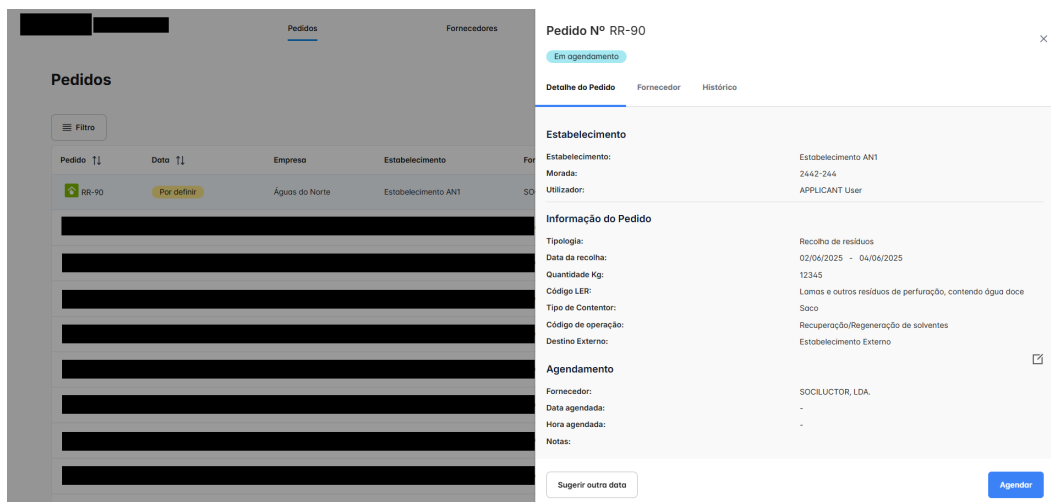


Figura A.19: Lista de pedidos atribuídos visível ao Gestor de Fornecedor.

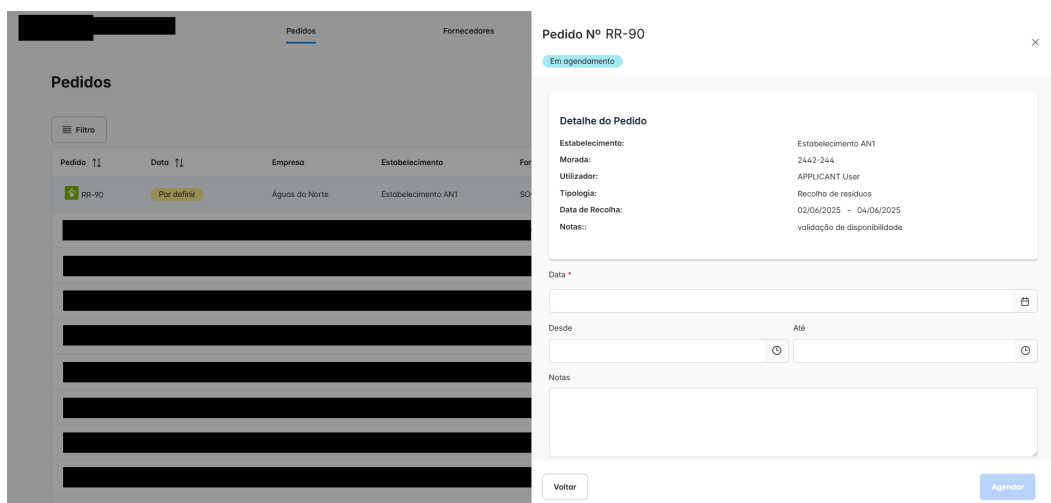


Figura A.20: Lista de pedidos atribuídos visível ao Gestor de Fornecedor.

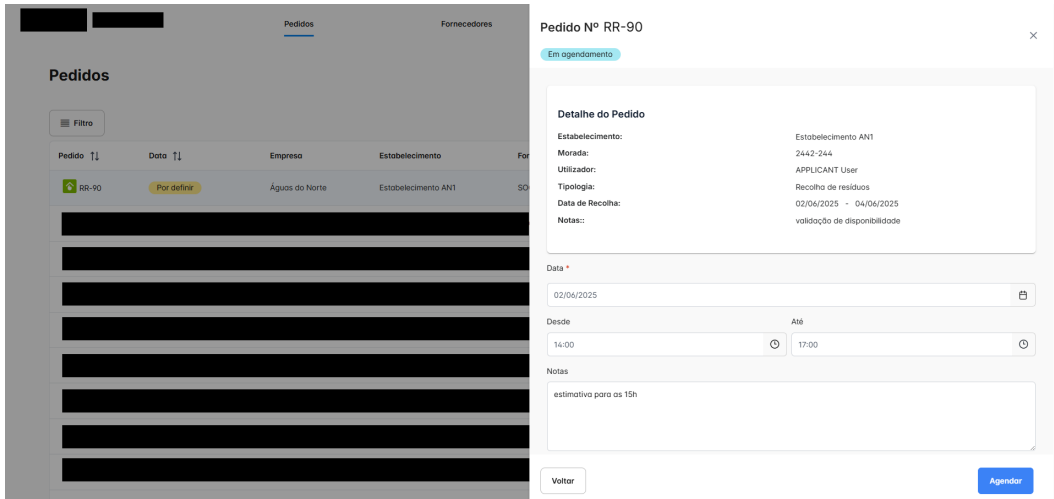


Figura A.21: Lista de pedidos atribuídos visível ao Gestor de Fornecedor.

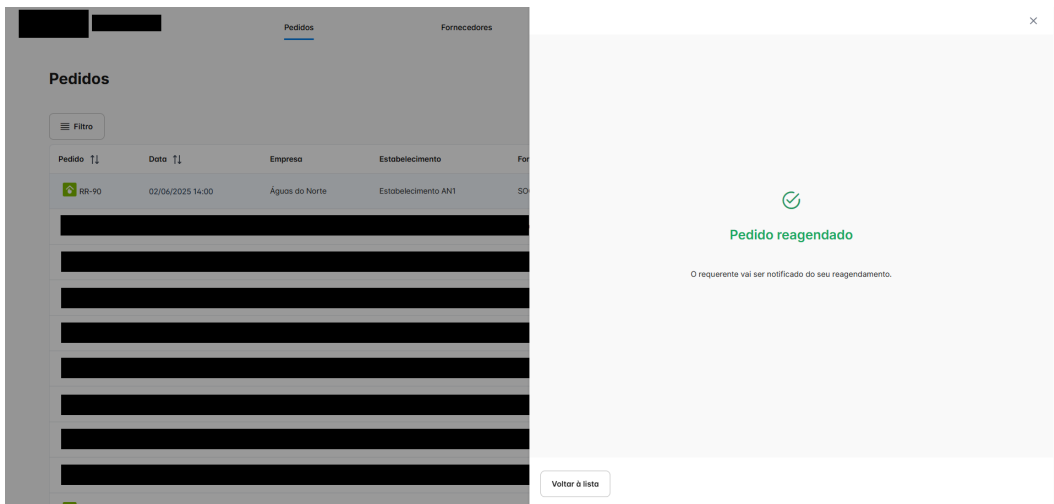


Figura A.22: Lista de pedidos atribuídos visível ao Gestor de Fornecedor.

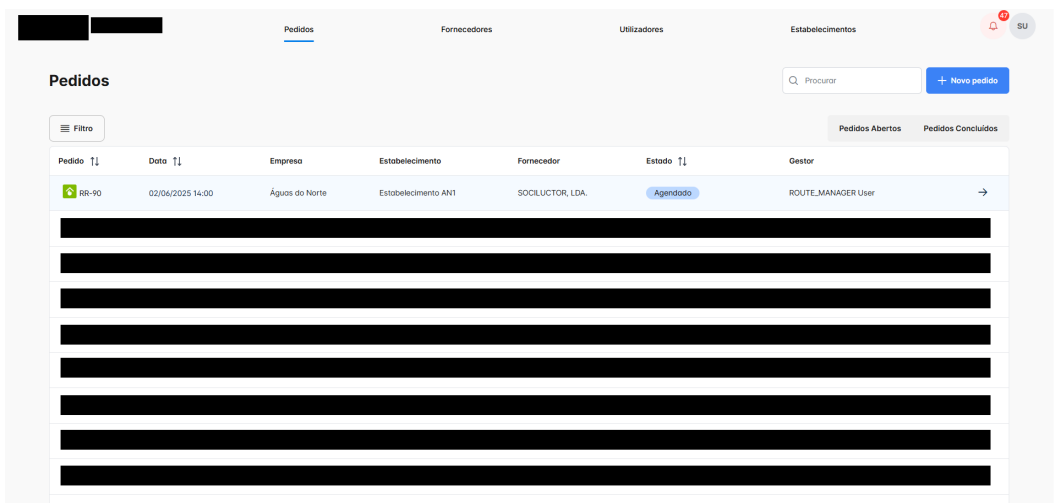


Figura A.23: Lista de pedidos atribuídos visível ao Gestor de Fornecedor.

Perspetivas Finais

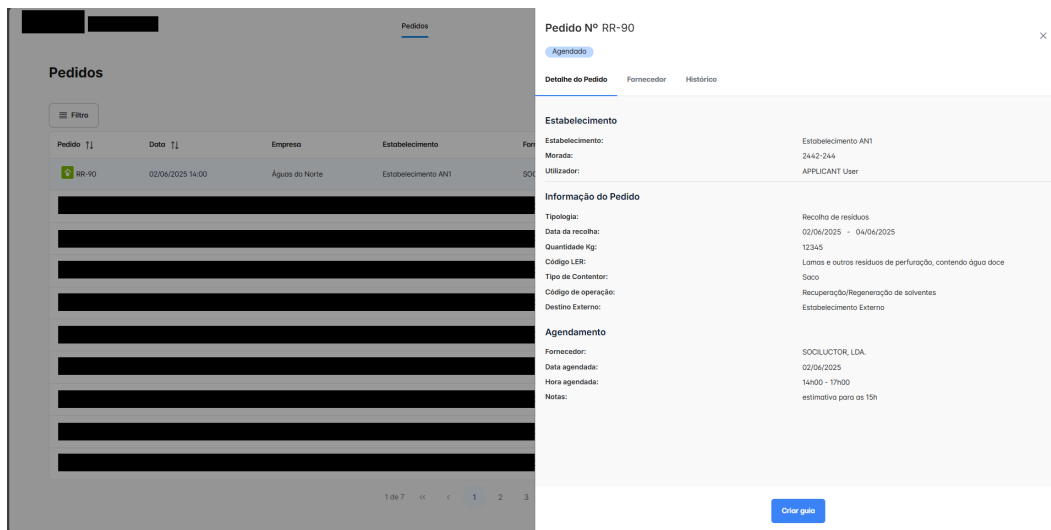


Figura A.24: Resumo final do pedido na aba "Detalhe do Pedido" do ponto de vista do Requerente.

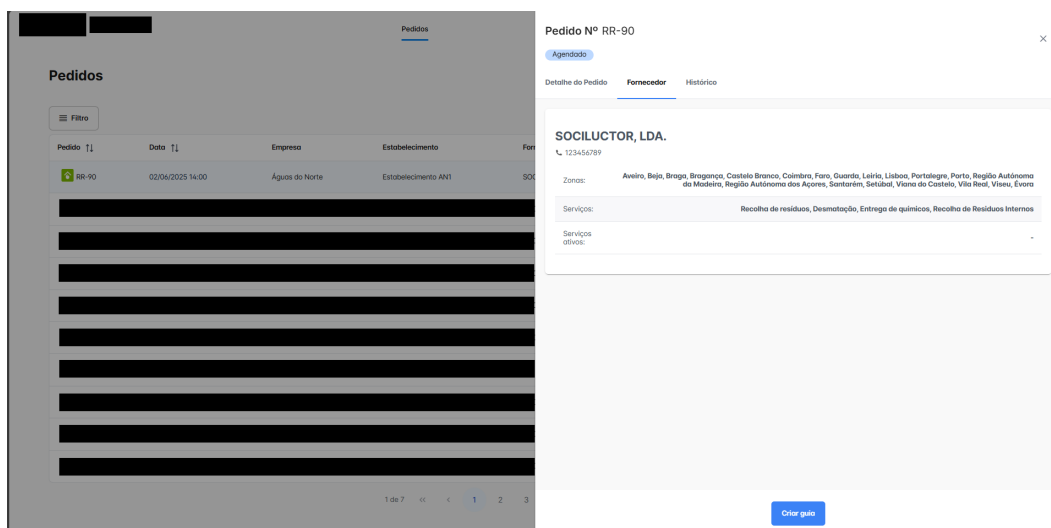


Figura A.25: Resumo final do pedido na aba "Fornecedor" do ponto de vista do Requerente.

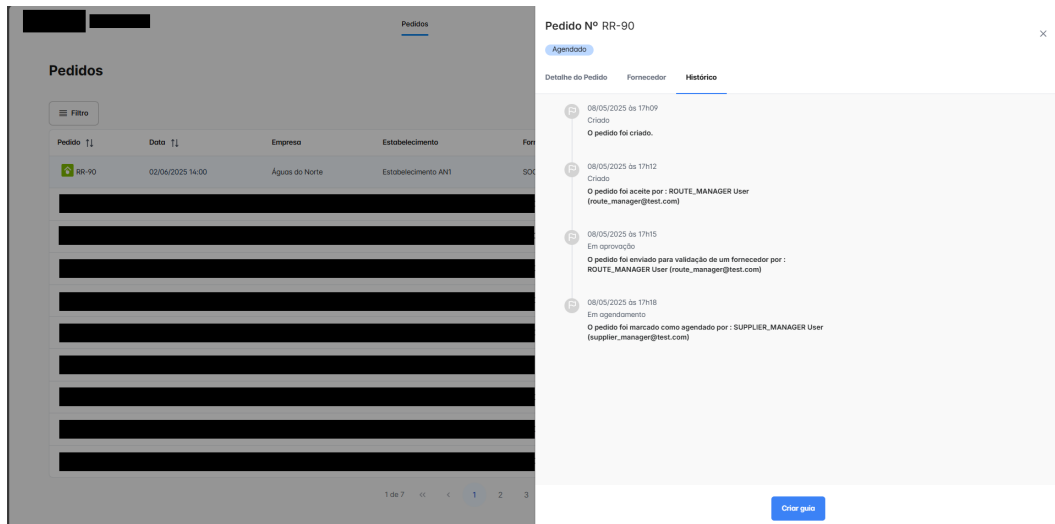


Figura A.26: Resumo final do pedido na aba "Histórico" do ponto de vista do Requerente.

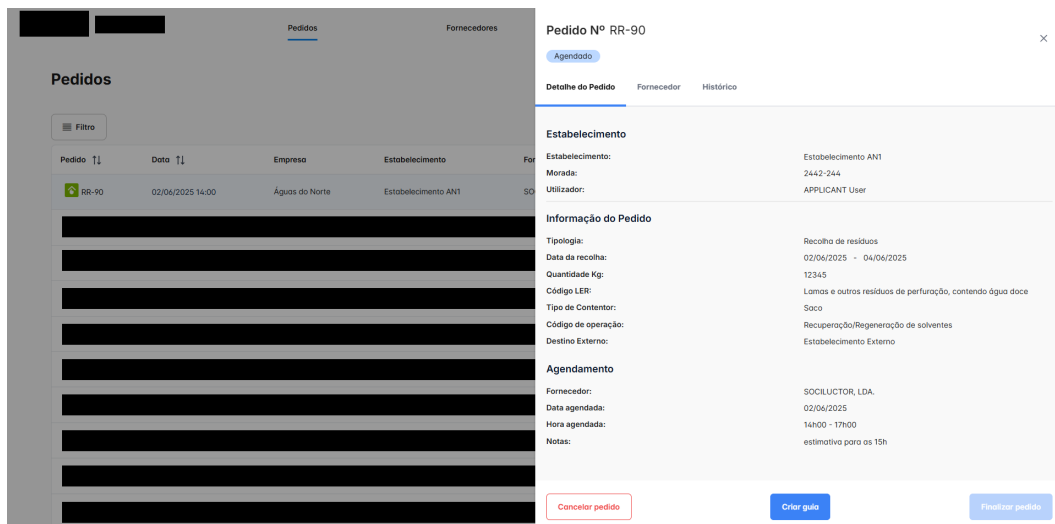


Figura A.27: Resumo final do pedido na aba "Detalhe do Pedido" do ponto de vista do Gestor de Rota.

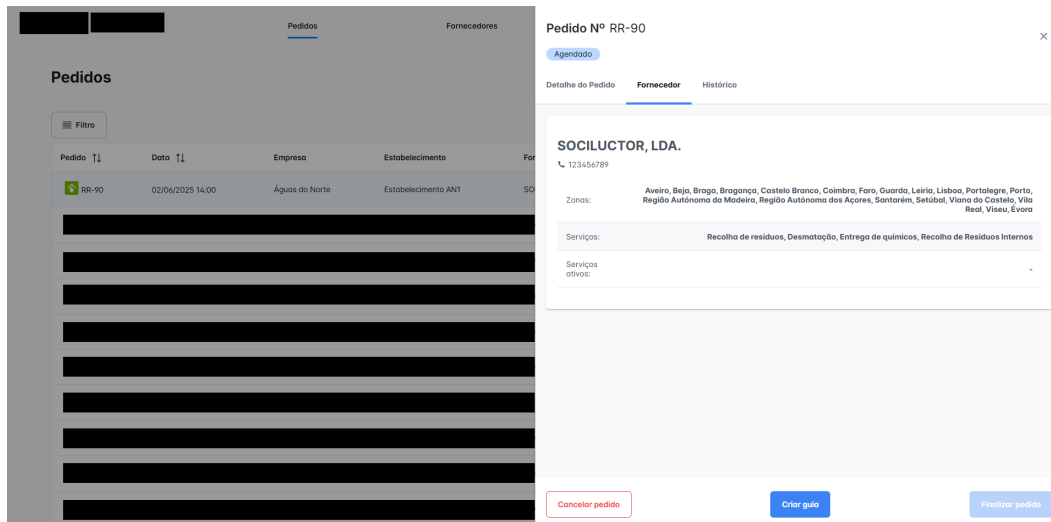


Figura A.28: Resumo final do pedido na aba "Fornecedor" do ponto de vista do Gestor de Rota.

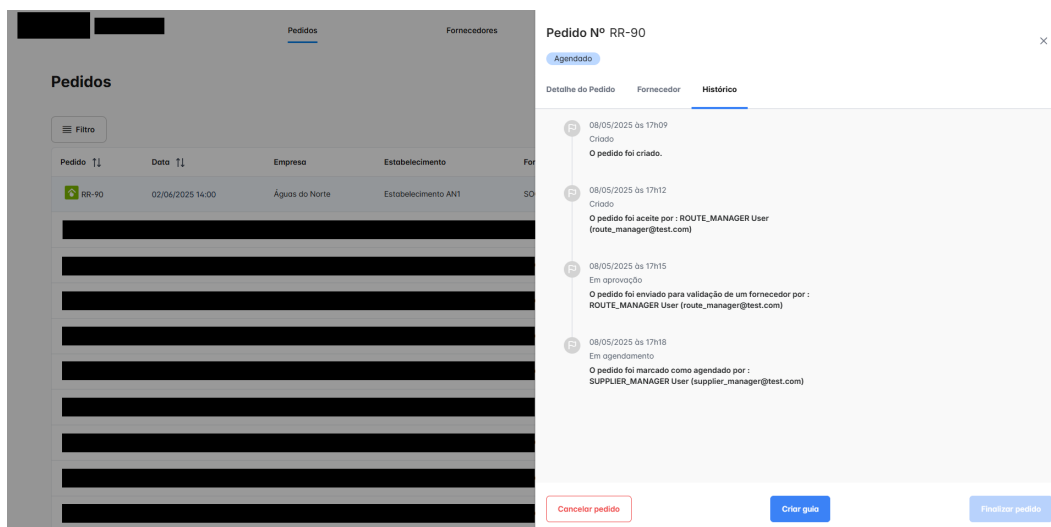


Figura A.29: Resumo final do pedido na aba "Histórico" do ponto de vista do Gestor de Rota.

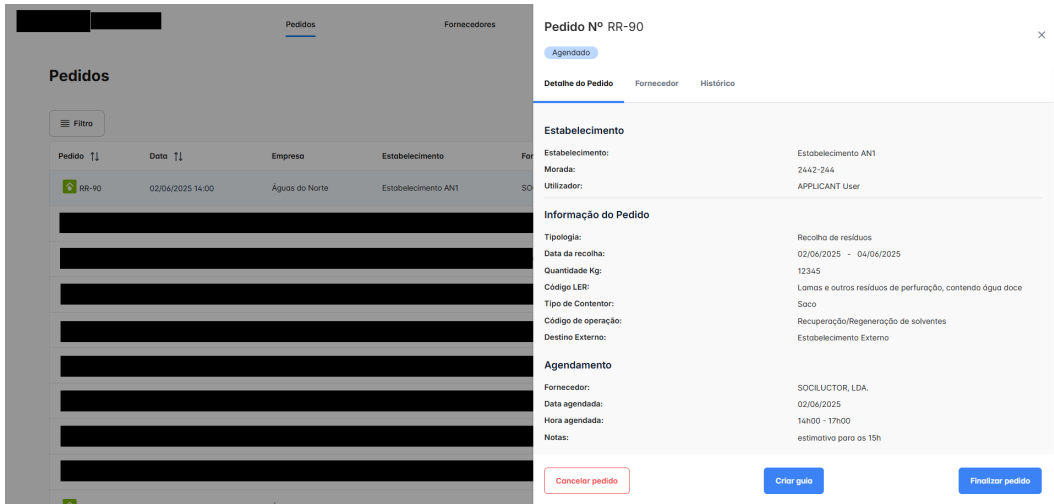


Figura A.30: Resumo final do pedido na aba "Detalhe do Pedido" do ponto de vista do Gestor de Fornecedor.

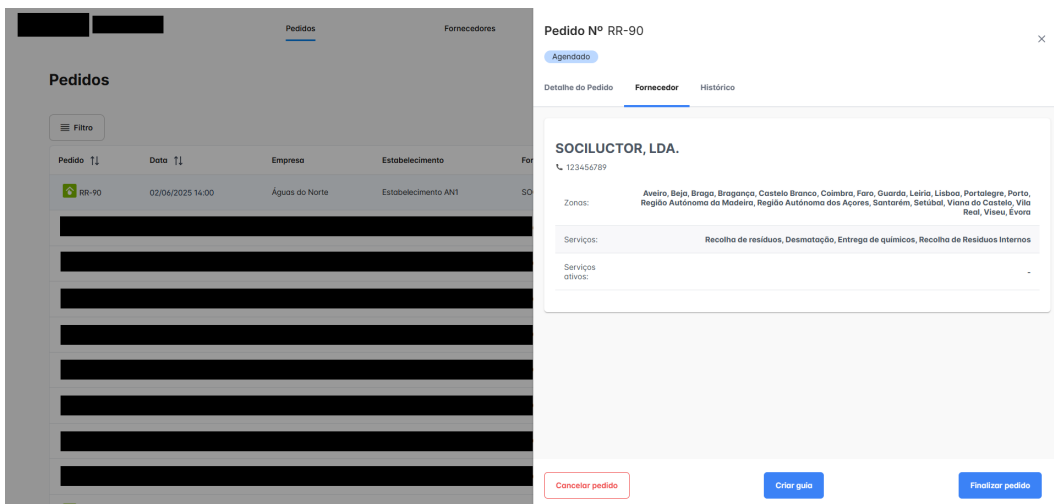


Figura A.31: Resumo final do pedido na aba "Fornecedor" do ponto de vista do Gestor de Fornecedor.

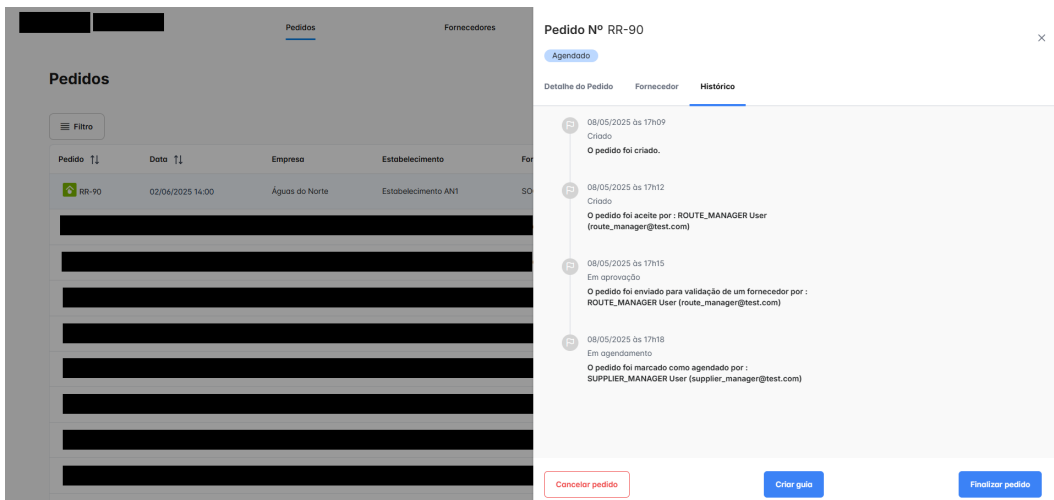


Figura A.32: Resumo final do pedido na aba "Histórico" do ponto de vista do Gestor de Fornecedor.

## A.2 Modelo de Datos

## PGIP

### A.2.1 User Stories

A seguinte tabela apresenta todas as User Stories identificadas no projeto, descrevendo os requisitos e funcionalidades implementadas.

ID	Descrição da User Story
USo1	Como utilizador, quero poder fazer login na plataforma utilizando o AD para autenticação segura.
USo2	Como administrador, quero poder registar novos utilizadores e definir se são internos ou externos.
USo3	Como administrador, quero poder alterar a role de um utilizador para gerir permissões.
USo4	Como administrador, quero poder criar novas roles na plataforma.
USo5	Como administrador, quero modificar permissões de roles existentes.
USo6	Como administrador, quero poder gerir tipologias de pedidos/serviços.
USo7	Como administrador, quero definir formulários dinâmicos associados às tipologias de pedidos.
USo8	Como administrador, quero poder adicionar novos fornecedores.
USo9	Como administrador, quero poder editar informações de fornecedores.
US10	Como administrador, quero poder associar utilizadores a fornecedores.
US11	Como administrador, quero poder remover fornecedores do sistema.
US12	Como administrador, quero poder adicionar novos estabelecimentos.
US13	Como administrador, quero poder editar informações dos estabelecimentos.
US14	Como administrador, quero poder associar utilizadores a estabelecimentos.
US15	Como administrador, quero poder remover estabelecimentos do sistema.
US16	Como requerente, quero criar pedidos de serviço na plataforma.
US17	Como utilizador, quero consultar a lista de pedidos submetidos.
US18	Como gestor, quero aprovar ou rejeitar pedidos submetidos.
US19	Como gestor, quero selecionar fornecedores para os pedidos.
US20	Como gestor, quero aprovar as rotas dos pedidos.
US21	Como gestor, quero reagendar pedidos quando necessário.
US22	Como gestor, quero concluir pedidos após a sua execução.
US23	Como sistema, quero enviar notificações automáticas sobre pedidos.

Tabela A.1: Tabela com a descrição das User Stories do projeto PGIP

### A.2.2 Arquitetura do Sistema

O diagrama apresentado nesta secção ilustra a arquitetura da plataforma **PGIP**, composta por três camadas principais: a camada de apresentação (*frontend*), a camada lógica (*backend*) e as integrações externas com sistemas como SAP e Azure AD. Esta arquitetura modular foi concebida para garantir escalabilidade, manutenibilidade e segurança no desenvolvimento da aplicação.

## ARQUITETURA DO SISTEMA PGIP

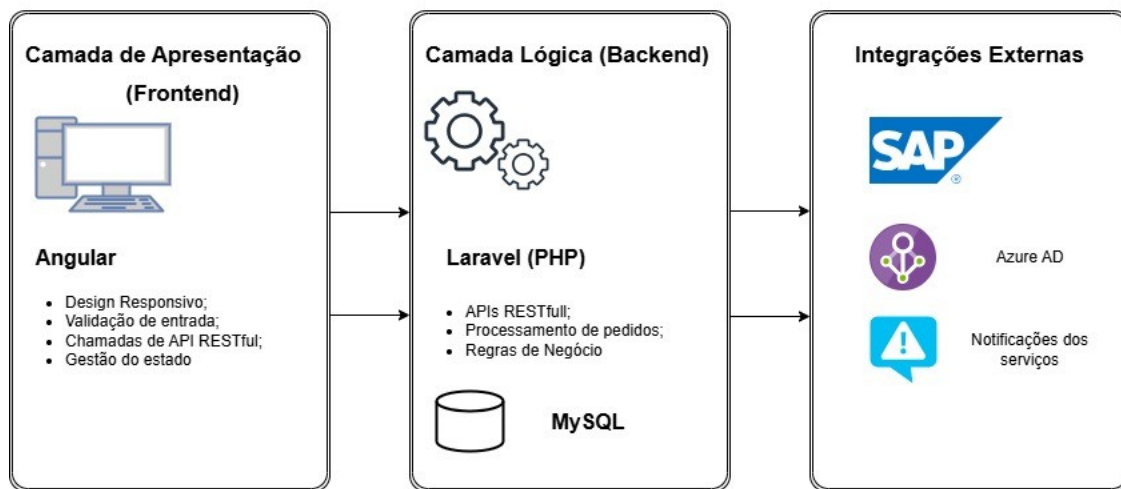


Figura A.33: Arquitetura geral do sistema PGIP, com destaque para as camadas de *frontend*, *backend* e integrações externas.

# PGIP

## A.2.3 Fluxo de Utilização da Aplicação

O diagrama de atividade apresentado representa o *Happy Flow* da plataforma **PGIP**. Nele estão descritas, de forma sequencial e organizada por perfis de utilizador, as ações realizadas desde a criação de um pedido até à sua conclusão. Este diagrama ajuda a visualizar a colaboração entre Requerente, Gestor de Fornecedor e Gestor de Rota.

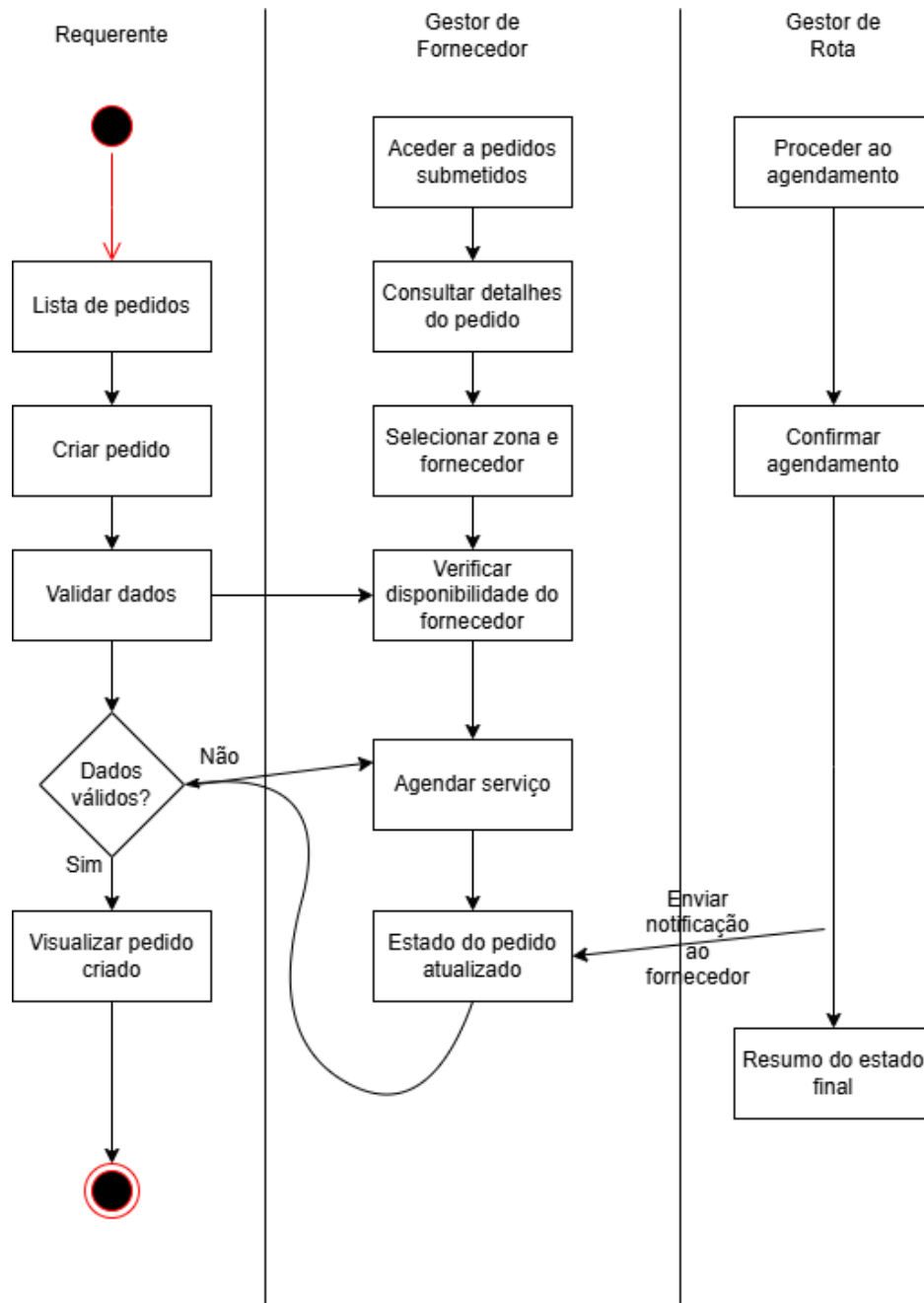


Figura A.34: Diagrama de atividade do *Happy Flow* da aplicação PGIP, segmentado por perfis de utilizador.

## A.3 Tarefas Realizadas

### A.3.1 Tarefas Realizadas

Nesta secção são apresentadas as interfaces desenvolvidas durante o estágio no âmbito do projeto **PGIP**. Estas representam funcionalidades implementadas e validadas ao longo do processo de desenvolvimento, abordando desde a configuração de entidades, como tipologias e fornecedores, até à gestão de permissões e guias de transporte. Cada imagem documenta visualmente uma etapa significativa do sistema, ilustrando o resultado prático das tarefas atribuídas.

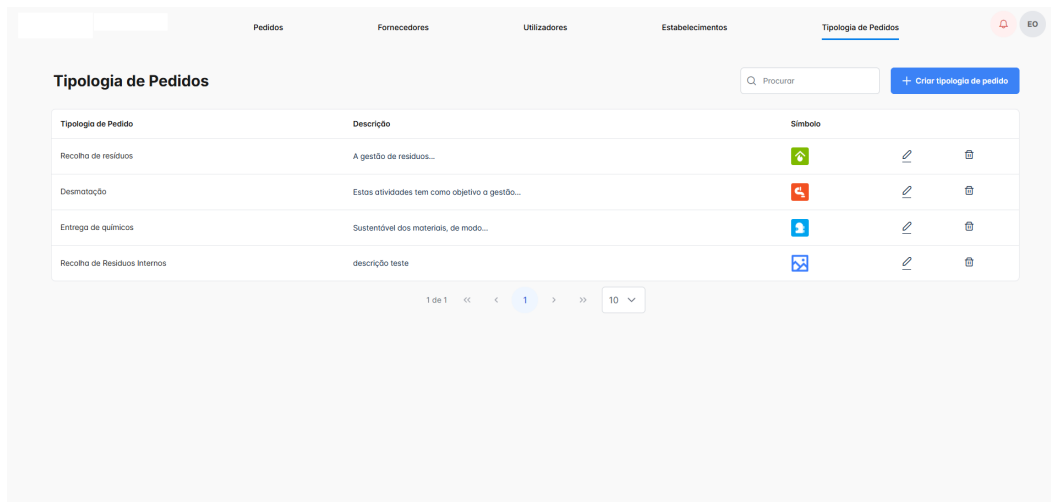


Figura A.35: Interface do menu de tipologias de pedidos. Esta funcionalidade permite ao administrador consultar, aceder e gerir as diferentes tipologias existentes no sistema.

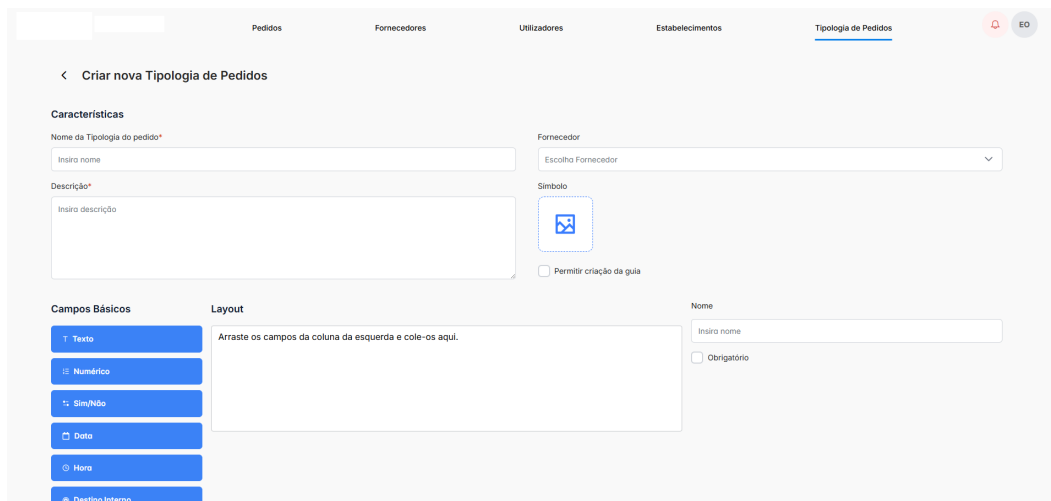


Figura A.36: Formulário para criação de nova tipologia de pedido. Esta interface permite definir o tipo de serviço e os campos necessários para o seu preenchimento.

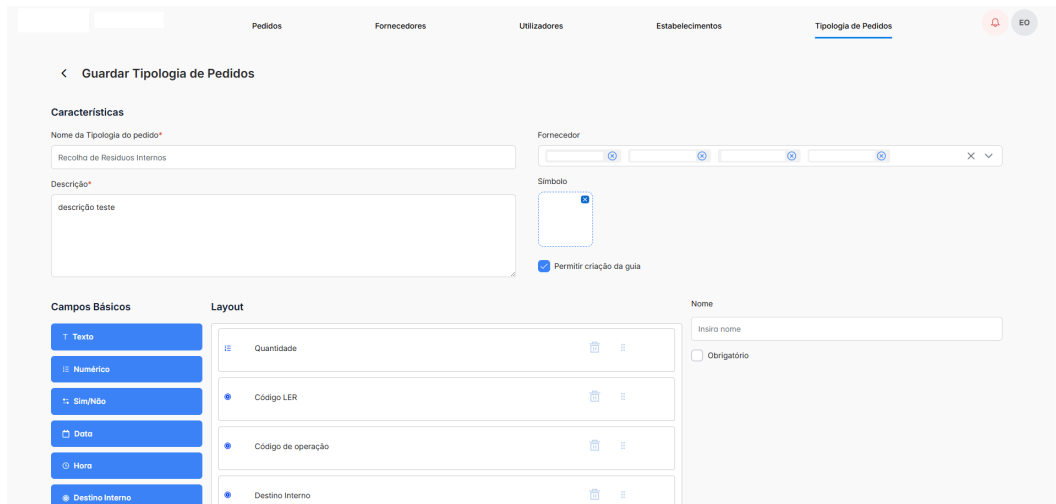


Figura A.37: Página de edição de tipologia de pedido. Permite alterar os campos, regras e validações associadas a uma tipologia já existente.

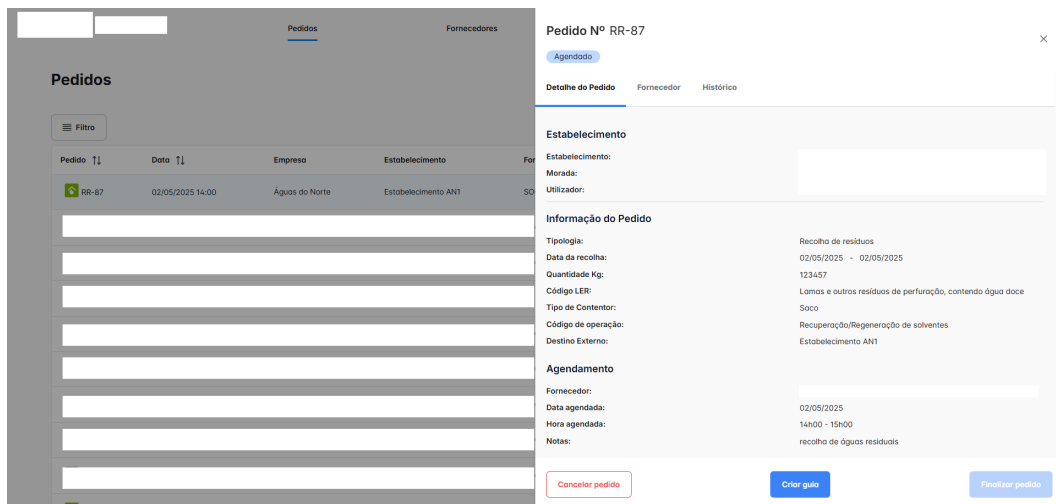


Figura A.38: Vista detalhada de um pedido submetido. A página exhibe todas as informações associadas à requisição, incluindo datas, estados e dados adicionais.

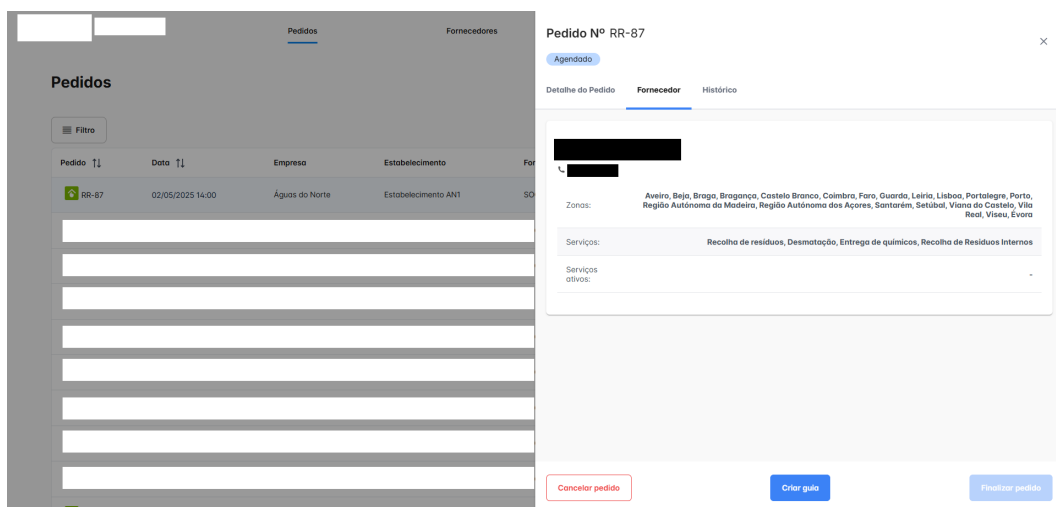


Figura A.39: Detalhes do fornecedor associado a um pedido. Esta página fornece uma visão consolidada do fornecedor atribuído, incluindo contactos e histórico.

Figura A.40: Formulário para geração de guia de transporte. Esta funcionalidade permite criar uma guia associada ao pedido, incluindo os dados logísticos essenciais.

Figura A.41: Visualização detalhada da guia de transporte gerada. Apresenta os dados transportados, horários e ligação direta ao pedido associado.

Figura A.42: Página de histórico de pedidos. Regista as ações realizadas, alterações de estado e datas relevantes para rastreabilidade dos processos.

Figura A.43: Parte 1 do formulário de registo de fornecedores. Esta secção permite inserir informações básicas do novo fornecedor, como nome e contacto.

Figura A.44: Parte 2 do formulário de fornecedores. Inclui campos adicionais como localização, categoria de serviço e capacidade de resposta.

Figura A.45: Parte 3 do formulário de fornecedores. Finaliza o processo de registo com validações e campos complementares.

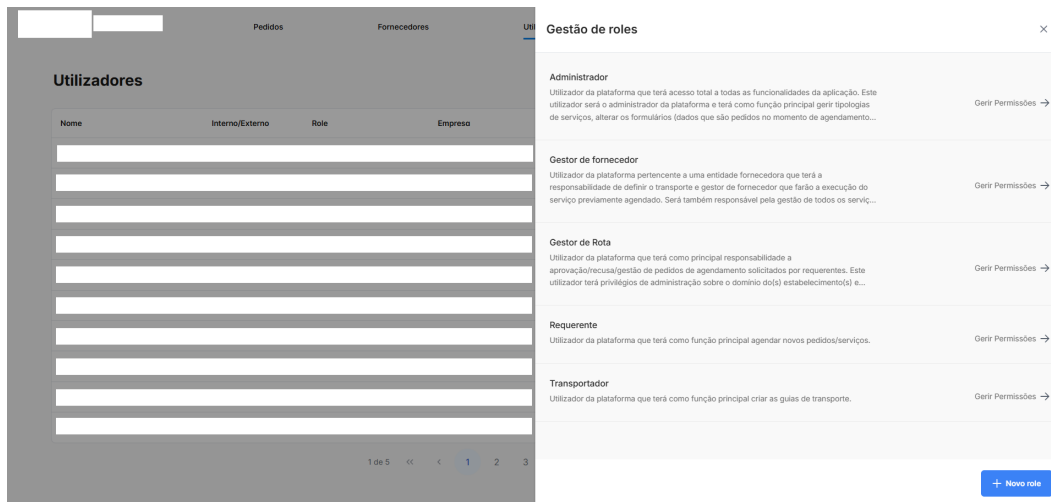


Figura A.46: Gestão de permissões (parte 1). Permite a atribuição de permissões específicas a diferentes roles, controlando o acesso às funcionalidades.

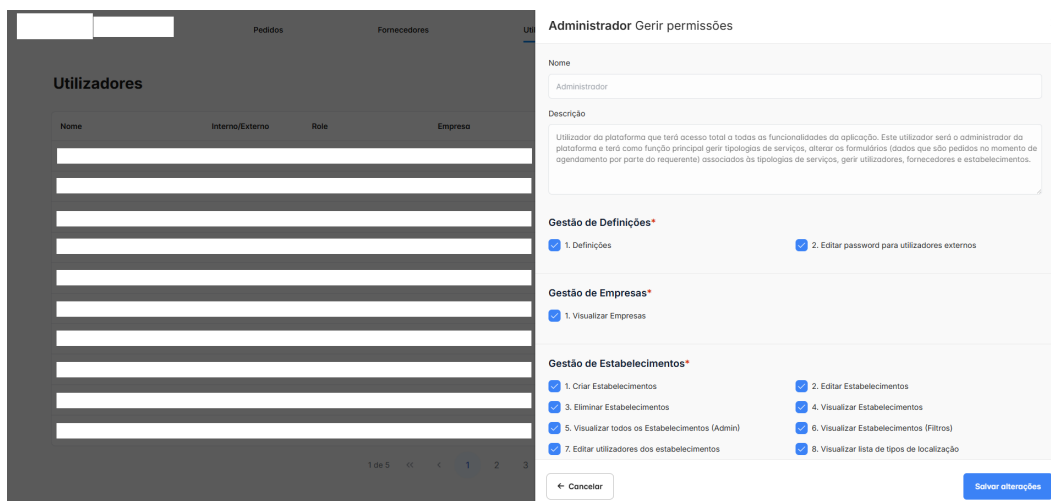


Figura A.47: Gestão de permissões (parte 2). Complementa a interface anterior com a possibilidade de revisão e modificação granular das permissões atribuídas.