POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Relatório de Estágio na ValGrupo

Desenvolvimento de Aplicações Seguras

João Rodrigo Caetano Neves

Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

Leiria, Junho 2026

Página intencionalmente deixada em branco.

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Relatório de Estágio na ValGrupo

Desenvolvimento de Aplicações Seguras

João Rodrigo Caetano Neves

Nº de Estudante 2024101497

Supervisor: Maria Micaela Gonçalves Pinto Dinis Esteves

Escola Superior de Tecnologia e Gestão

Departamento de Engenharia Informática

Mestrado em Cibersegurança e Informática Forense

Estágio Curricular

Leiria, Junho 2026

Página intencionalmente deixada em branco.

Relatório de Estágio na ValGrupo

O presente trabalho beneficiou da utilização do modelo *IPLeiria-Thesis*.

Página intencionalmente deixada em branco.

Agradecimentos

A realização deste estágio e a elaboração do presente relatório representaram uma etapa importante do meu percurso académico e pessoal, marcada por aprendizagem, desafios e crescimento.

Em primeiro lugar, agradeço à minha orientadora, Professora Maria Micaela Gonçalves Pinto Dinis Esteves, pelo acompanhamento próximo, disponibilidade, orientação e apoio prestado ao longo deste processo. O seu contributo foi fundamental para a estruturação, revisão e melhoria deste relatório.

Agradeço à ValFinance e ao ValGrupo pela oportunidade de realizar o estágio em contexto empresarial, bem como à equipa de desenvolvimento pela integração, apoio e disponibilidade demonstrados ao longo do período de estágio. O contacto com projetos reais, sistemas internos e desafios técnicos concretos foi essencial para consolidar conhecimentos e desenvolver novas competências.

Agradeço ainda aos meus colegas, amigos e família pelo apoio, paciência e incentivo ao longo deste percurso. O seu suporte foi fundamental para ultrapassar os momentos mais exigentes e manter a motivação até à conclusão desta etapa.

Agradeço também à Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, instituição que fez parte do meu percurso académico ao longo dos últimos sete anos, desde o CTeSP em Programação de Sistemas de Informação, passando pela Licenciatura em Engenharia Informática e finalizando neste Mestrado. Este percurso permitiu-me crescer a nível académico, técnico e pessoal, sendo uma parte importante da minha formação.

A todos, o meu sincero obrigado.

Página intencionalmente deixada em branco.

Resumo

Este relatório foi elaborado no âmbito do estágio curricular do Mestrado em Cibersegurança e Informática Forense da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria. O estágio foi realizado na ValFinance, empresa integrada no ValGrupo, e decorreu na equipa de desenvolvimento de *software*, tendo como objetivo principal a aplicação de conhecimentos técnicos em contexto empresarial real.

Ao longo do estágio foram desenvolvidas e adaptadas funcionalidades no sistema interno da organização, com especial destaque para o projeto ValSabor, que constitui o principal caso de estudo deste relatório. Este projeto teve como finalidade apoiar processos industriais e logísticos associados à gestão de unidades logísticas, incluindo a leitura de códigos de barras, criação e agrupamento de caixas, montagem de paletes e grades, geração de etiquetas, controlo de peso e integração com serviços externos associados ao SAP.

O desenvolvimento foi realizado com recurso a Laravel no *backend* e Vue.js no *frontend*, envolvendo a implementação de regras de negócio, validações no *backend*, controlo de acessos, mecanismos de rastreabilidade e testes funcionais. Para além do desenvolvimento aplicacional, o estágio incluiu atividades complementares na área da cibersegurança, nomeadamente a identificação de uma situação de exposição indevida de informação técnica num serviço *web* e a realização de uma campanha controlada de *phishing*.

Este relatório descreve o enquadramento do estágio, o estado da arte associado ao desenvolvimento de aplicações empresariais seguras, o trabalho desenvolvido, o caso de estudo ValSabor, as principais implementações técnicas, a análise crítica das dificuldades encontradas e as aprendizagens obtidas. O estágio permitiu consolidar competências em desenvolvimento *web*, integração entre sistemas, validação de dados, segurança aplicacional e trabalho em equipa em contexto empresarial.

Palavras-Chave: Desenvolvimento seguro, Aplicações *web* empresariais, Laravel, Vue.js, SAP, Rastreabilidade, Cibersegurança.

Página intencionalmente deixada em branco.

Abstract

This report was prepared within the scope of the curricular internship of the Master's Degree in Cybersecurity and Digital Forensics at the School of Technology and Management of the Polytechnic Institute of Leiria. The internship was carried out at ValFinance, a company integrated in ValGrupo, within the software development team, with the main objective of applying technical knowledge in a real business context.

During the internship, several functionalities were developed and adapted in the organisation's internal systems, with particular emphasis on the ValSabor project, which is the main case study of this report. This project aimed to support industrial and logistics processes related to the management of logistics units, including barcode reading, box creation and grouping, pallet and crate assembly, label generation, weight control and integration with external services associated with SAP.

The development work was carried out using Laravel on the backend and Vue.js on the frontend, involving the implementation of business rules, backend validations, access control, traceability mechanisms and functional testing. In addition to application development, the internship included complementary cybersecurity activities, namely the identification of a situation involving the exposure of technical information in a web service and the execution of a controlled phishing campaign.

This report describes the internship context, the state of the art related to the development of secure enterprise applications, the work carried out, the ValSabor case study, the main technical implementations, the critical analysis of the difficulties encountered and the lessons learned. The internship contributed to the consolidation of skills in web development, system integration, data validation, application security and teamwork in a business environment.

Keywords: Secure development, Enterprise web applications, Laravel, Vue.js, SAP, Traceability, Cybersecurity.

Página intencionalmente deixada em branco.

Declaração sobre o Uso de Inteligência Artificial

Declaro que utilizei ferramentas de inteligência artificial generativa e de apoio à pesquisa bibliográfica, nomeadamente o ChatGPT (https://chatgpt.com) e o Elicit (https://elicit.com). O ChatGPT foi utilizado como apoio à organização de ideias, estruturação textual, reescrita de algumas frases, melhoria da clareza do português, revisão linguística e adequação do tom académico. O Elicit foi utilizado como apoio à pesquisa e identificação de referências bibliográficas relevantes para o enquadramento teórico do relatório.

A utilização destas ferramentas foi realizada com base em informação fornecida por mim relativamente ao trabalho desenvolvido durante o estágio. A descrição das funcionalidades implementadas, as decisões técnicas apresentadas, os resultados obtidos, a seleção final das referências bibliográficas e a validação final do conteúdo são da minha responsabilidade.

O conteúdo final foi revisto, ajustado e validado por mim, garantindo a sua correspondência com o trabalho efetivamente realizado em contexto empresarial e com as fontes bibliográficas consultadas.

Página intencionalmente deixada em branco.

Conteúdo

Página intencionalmente deixada em branco.

Lista de Figuras

Página intencionalmente deixada em branco.

Lista de Tabelas

Página intencionalmente deixada em branco.

Lista de Listagens

Página intencionalmente deixada em branco.

Página intencionalmente deixada em branco.

Glossário

Auditabilidade Capacidade de reconstruir e analisar operações realizadas num sistema através de registos, histórico de ações e evidências verificáveis. (*p. 19, 20*)

DevSecOps Abordagem de desenvolvimento que integra práticas de segurança ao longo de todo o ciclo de vida do software, promovendo a automatização e validação contínua em ambientes DevOps. (*p. 9, 17, 73, 76*)

GS1 Conjunto de normas globais (códigos) que identificam, capturam e partilham informação de ativos – produtos, locais, pessoas, entre outros. (*p. 2, 11, 12, 19, 26, 31, 35, 43, 45, 67, 70, 72, 86, 88*)

Idempotência Propriedade de uma operação que permite a sua repetição sem produzir efeitos adicionais indesejados quando executada com os mesmos parâmetros. (*p. 15, 16, 59*)

Integridade dos dados Garantia de que a informação se mantém correta, completa, coerente e não adulterada ao longo do seu ciclo de vida. (*p. 13, 14, 18, 19, 22, 50, 68, 73, 76*)

Logging Registo sistemático de eventos, erros ou ações relevantes de um sistema, permitindo diagnóstico, monitorização e análise posterior. (*p. 16, 20, 34, 35, 55–58, 99*)

rastreabilidade Capacidade de acompanhar a história, localização, estado e relações de um produto, componente ou unidade logística ao longo de um processo. (*p. 1–4, 11, 13, 14, 19, 20, 22, 23, 25, 26, 30, 31, 34, 38, 45, 46, 50, 59, 60, 70, 73, 74, 76, 86, 88, 89, 101*)

Security Misconfiguration Categoria de vulnerabilidade associada a configurações incorretas ou inseguras de aplicações, servidores ou serviços. (*p. 17, 68*)

Stack trace Sequência detalhada de chamadas de funções apresentada após uma exceção ou erro numa aplicação, podendo revelar informação interna sensível. (*p. 36, 67*)

staging Ambiente intermédio que replica o ambiente de produção, utilizado para validação de funcionalidades antes da sua disponibilização final. (*p. 9, 11, 12, 26, 38, 58, 60, 67, 69, 71, 75, 86, 90*)

Tolerância a falhas Capacidade de um sistema lidar com erros, interrupções ou comportamentos inesperados de forma controlada, limitando o seu impacto. (*p. 16*)

Página intencionalmente deixada em branco.

Siglas

API Application Programming Interface. (*p. 8, 9, 11, 12, 14–18, 22, 28, 29, 33–35, 38, 43, 61, 63, 64, 69, 70, 74, 85, 87, 99*)

ASVS Application Security Verification Standard. (*p. 14, 16, 17, 58, 59, 62, 64, 65, 68, 76, 98*)

CSRF Cross-Site Request Forgery. (*p. 63, 64*)

ERP Enterprise Resource Planning. (*p. 1, 4, 13, 14, 22, 31, 34, 35, 42, 43, 47, 51, 54, 55*)

EWM Extended Warehouse Management. (*p. 55, 56*)

GTIN Global Trade Item Number. (*p. 19, 46*)

HTTP Hypertext Transfer Protocol. (*p. 15, 16, 37*)

IIoT Industrial Internet of Things. (*p. 20*)

IP Internet Protocol. (*p. 36*)

MQTT Message Queuing Telemetry Transport. (*p. 20, 43*)

NIST National Institute of Standards and Technology. (*p. 16, 59, 65, 76*)

OWASP Open Web Application Security Project. (*p. 15–18, 37, 58, 59, 62, 64–66, 68, 70, 76, 98*)

RBAC Role-Based Access Control. (*p. 18*)

SPA Single-Page Application. (*p. 63*)

SSCC Serial Shipping Container Code. (*p. 19, 31*)

SSDF Secure Software Development Framework. (*p. 16, 59, 65, 76*)

ZPL Zebra Programming Language. (*p. 2, 11, 12, 26, 31–33, 67, 70, 72, 86, 89*)

Página intencionalmente deixada em branco.

Página intencionalmente deixada em branco.

1 Introdução

A digitalização dos processos industriais e logísticos tem aumentado a dependência das organizações relativamente a sistemas de informação internos capazes de suportar operações críticas, integrar dados provenientes de diferentes fontes e assegurar a rastreabilidade das atividades realizadas. Em contextos produtivos, a informação registada nos sistemas aplicacionais deve refletir, com rigor, o estado real das operações físicas, uma vez que erros na identificação de produtos, na associação de lotes, na pesagem, na movimentação de unidades logísticas ou na integração com sistemas empresariais podem originar inconsistências operacionais, perda de rastreabilidade e dificuldades na tomada de decisão.

Este problema torna-se particularmente relevante em organizações que recorrem a sistemas Enterprise Resource Planning (ERP), como o SAP, uma plataforma de gestão empresarial que permite integrar e centralizar informação proveniente de diferentes áreas da organização, incluindo as áreas logística, produtiva e administrativa (Butarbutar et al., 2023). Nestes cenários, uma aplicação operacional desenvolvida internamente pode desempenhar um papel intermédio entre o ambiente físico de produção e os sistemas de gestão empresarial. Assim, a sua fiabilidade não depende apenas da implementação de funcionalidades, mas também da correta validação dos dados, do controlo de acessos, da prevenção de duplicações, da gestão de estados e da capacidade de integração com serviços externos (OWASP Foundation, 2023; Fielding et al., 2022).

O presente relatório surge no âmbito do estágio curricular realizado na ValFinance, empresa integrada no ValGrupo, no contexto do Mestrado em Cibersegurança e Informática Forense. O estágio decorreu na equipa de desenvolvimento de *software* da organização e incidiu, numa fase inicial, sobre várias tarefas de desenvolvimento e manutenção de sistemas internos. Contudo, devido à sua complexidade técnica, impacto operacional e relação direta com temas como segurança aplicacional, validação de dados, rastreabilidade e integração com sistemas externos, o projeto desenvolvido para a ValSabor foi selecionado como principal caso de estudo deste relatório.

O projeto ValSabor envolveu o desenvolvimento e evolução de funcionalidades associadas à gestão de unidades logísticas em ambiente industrial. Entre os principais

elementos do sistema destacam-se a leitura de códigos de barras segundo as normas GS1 e a geração de etiquetas em Zebra Programming Language (ZPL), utilizadas na identificação de produtos e unidades logísticas (GS1, 2025; GS1, 2019). Destacam-se ainda a criação e agrupamento de caixas, o fecho de paletes intermédias, a montagem de unidades logísticas, o controlo de peso através de balanças industriais, a utilização de leitores de códigos de barras em contexto operacional e a integração com serviços externos associados ao SAP. Estas funcionalidades exigiram a implementação de regras de negócio específicas, mecanismos de validação no *backend*, controlo de permissões e registo de operações, contribuindo para a consistência e a rastreabilidade dos dados ao longo do processo.

Para além da componente de desenvolvimento *full stack*, o trabalho realizado possui uma dimensão relevante no domínio da cibersegurança. A segurança foi considerada não apenas como uma atividade complementar, mas como uma preocupação transversal ao desenvolvimento do sistema, refletida na validação de dados críticos, na centralização da lógica de negócio no *backend*, no controlo de acessos, no tratamento de erros e na proteção das operações de integração. Esta abordagem encontra-se alinhada com boas práticas de desenvolvimento seguro, segundo as quais a segurança deve ser incorporada ao longo do ciclo de vida do *software*, e não apenas avaliada numa fase posterior (Souppaya et al., 2022; OWASP Foundation, 2025; Rajapakse et al., 2022).

1.1 Enquadramento do Estágio

O estágio curricular foi realizado na ValFinance, entidade pertencente ao ValGrupo, no âmbito do Mestrado em Cibersegurança e Informática Forense. A ValFinance desempenha funções transversais de suporte ao grupo, incluindo áreas como gestão financeira, apoio administrativo, informática e desenvolvimento de sistemas de informação. A integração do estágio na equipa de desenvolvimento de *software* permitiu o contacto direto com aplicações internas utilizadas por diferentes unidades do grupo, bem como com processos reais de desenvolvimento, validação e disponibilização de funcionalidades em ambiente empresarial.

Embora o estágio tenha incluído tarefas de natureza diversa, nomeadamente desenvolvimento de *software*, testes, validação funcional e atividades complementares relacionadas com segurança da informação, o presente relatório centra-se sobretudo no projeto ValSabor. Esta delimitação justifica-se pelo facto de este projeto reunir, de forma integrada, os principais desafios técnicos e científicos relevantes para o âmbito do mestrado: desenvolvimento de aplicações *web* empresariais, integração com sistemas externos, tratamento de dados operacionais sensíveis, rastreabilidade, controlo de acessos e segurança aplicacional.

1.2 Motivação e Objetivos do Problema

A motivação principal deste trabalho resulta da necessidade de desenvolver sistemas internos capazes de suportar processos industriais e logísticos com elevados requisitos de fiabilidade, rastreabilidade e segurança. Num ambiente em que as operações físicas dependem de registos digitais e em que esses registos são posteriormente integrados com sistemas empresariais, torna-se essencial garantir que os dados introduzidos, processados e enviados para sistemas externos são válidos, completos e coerentes.

Neste contexto, o problema central abordado no relatório pode ser formulado da seguinte forma: de que modo pode uma aplicação *web* empresarial, desenvolvida em contexto interno, suportar processos logísticos e industriais críticos, assegurando simultaneamente validação robusta de dados, controlo de acessos, rastreabilidade operacional e integração fiável com sistemas empresariais externos, nomeadamente o SAP?

A partir deste problema, definiram-se os seguintes objetivos:

- analisar o contexto tecnológico e operacional em que o sistema ValSabor se insere;
- identificar os principais requisitos funcionais, operacionais, de segurança e de integridade associados à gestão de unidades logísticas;
- desenvolver e adaptar funcionalidades *backend* e *frontend* que suportem os fluxos de criação, agrupamento, validação e movimentação de unidades logísticas;
- implementar mecanismos de validação de dados e regras de negócio no *backend*, reduzindo a dependência de validações executadas apenas no *frontend*;
- assegurar mecanismos de controlo de acessos e permissões adequados às operações críticas do sistema;
- apoiar a integração com serviços externos associados ao SAP, garantindo validações prévias, prevenção de duplicações e registo das respostas obtidas;
- contribuir para a rastreabilidade das operações, permitindo acompanhar o ciclo de vida das unidades logísticas ao longo dos diferentes processos;
- refletir criticamente sobre as decisões técnicas adotadas, relacionando-as com boas práticas de desenvolvimento seguro e com os desafios identificados na literatura.

1.3 Metodologia e Delimitação do Trabalho

O trabalho desenvolvido seguiu uma abordagem prática e iterativa, ajustada ao contexto real da equipa de desenvolvimento. Os requisitos foram transmitidos sobretudo através da equipa técnica e do responsável de informática, que funcionavam como intermediários entre as necessidades operacionais e a implementação técnica. Estes requisitos foram sendo clarificados e refinados ao longo do desenvolvimento, à medida que surgiam novas regras de negócio, exceções operacionais ou necessidades de adaptação do sistema.

Esta abordagem não correspondeu a um processo formal completo de engenharia

de requisitos, com levantamento sistemático junto de todos os utilizadores finais, mas refletiu a realidade de um contexto empresarial em evolução, onde as funcionalidades eram discutidas, implementadas, testadas e ajustadas de forma progressiva. Esta delimitação é assumida no relatório de forma explícita, permitindo apresentar o trabalho com rigor académico sem artificializar o processo seguido.

O relatório centra-se, assim, no projeto ValSabor enquanto caso de estudo principal, não procurando descrever exaustivamente todas as tarefas realizadas durante o estágio. As restantes atividades, incluindo tarefas gerais de desenvolvimento e ações complementares de cibersegurança, são apresentadas apenas quando contribuem para compreender o percurso realizado, as competências desenvolvidas ou o enquadramento global do estágio. A organização temporal das principais atividades desenvolvidas é apresentada na Secção 3.4.

1.4 Estrutura do Relatório

O presente relatório encontra-se organizado em nove capítulos. O **Capítulo 1** (**Introdução**) introduz o problema, o contexto do estágio, a motivação e os objetivos do trabalho. O **Capítulo 2** (**Apresentação da Empresa**) apresenta a empresa acolhedora, descrevendo a ValFinance, o ValGrupo e o enquadramento organizacional em que o estágio decorreu.

O **Capítulo 3** (**Enquadramento do Estágio**) enquadra o estágio do ponto de vista tecnológico e metodológico, apresentando a integração na equipa de desenvolvimento, as metodologias de trabalho adotadas e as principais tecnologias utilizadas. O **Capítulo 4** (**Estado da Arte**) corresponde ao Estado da Arte, onde são analisados os conceitos fundamentais relacionados com sistemas de informação empresariais, ERP, integração entre sistemas, segurança em aplicações *web*, validação de dados e rastreabilidade em contextos industriais e logísticos.

O **Capítulo 5** (**Trabalho Desenvolvido**) descreve o trabalho desenvolvido, incluindo a recolha e refinamento iterativo de requisitos, o planeamento das tarefas, o desenvolvimento *full stack* e os processos de teste e validação. O **Capítulo 6** (**Caso de Estudo**) apresenta o caso de estudo do sistema de gestão de unidades logísticas da ValSabor, analisando os principais fluxos operacionais, validações, mecanismos de rastreabilidade e integração com SAP.

O **Capítulo 7** (**Implementações Técnicas**) aprofunda as implementações técnicas realizadas, com enfoque na arquitetura da aplicação, *backend*, *frontend*, autenticação, autorização e medidas de segurança. O **Capítulo 8** (**Análise Crítica e Aprendizagens**) apresenta uma análise crítica do trabalho desenvolvido, discutindo dificuldades, soluções adotadas, limitações e aprendizagens. Por fim, o **Capítulo 9** (**Conclusão**) apresenta a conclusão, avaliando o cumprimento dos objetivos, os contributos do trabalho, as limitações identificadas e possíveis linhas de trabalho futuro.

2

Apresentação da Empresa

O presente capítulo tem como objetivo apresentar a empresa acolhedora do estágio, fornecendo um enquadramento organizacional e tecnológico que permita compreender o contexto em que o estágio decorreu. São abordadas a caracterização geral da empresa, a sua área de atuação, a estrutura organizacional e a infraestrutura tecnológica existente.

2.1 Caracterização da Empresa

A ValFinance integra o ValGrupo, um grupo económico nacional com presença em diversos setores de atividade, com especial destaque para o setor agropecuário. O grupo apresenta uma estrutura organizacional assente num modelo de verticalização das suas atividades, no qual diferentes fases do processo produtivo e logístico são geridas internamente. Este modelo permite um controlo rigoroso da qualidade, da rastreabilidade e da eficiência dos seus produtos e serviços, ao mesmo tempo que aumenta a complexidade dos processos de gestão e coordenação entre as diversas empresas que o integram.

A diversidade de áreas de atuação do grupo implica a necessidade de mecanismos eficazes de planeamento, controlo e suporte à decisão. Neste contexto, a existência de estruturas internas dedicadas à gestão administrativa, financeira e tecnológica assume um papel fundamental para garantir a coerência dos processos, a uniformização de práticas e a fiabilidade da informação utilizada ao nível da gestão.

A ValFinance surge como resposta à necessidade de centralizar funções críticas de suporte ao grupo, nomeadamente nas áreas administrativa, financeira, contabilística, informática e de controlo de gestão. A concentração destas atividades numa única entidade permite reduzir redundâncias, promover a normalização de procedimentos e melhorar a capacidade de controlo e monitorização das operações desenvolvidas pelas diferentes empresas do grupo.

Ao assumir este papel transversal, a ValFinance contribui para uma visão integrada dos processos internos, facilitando a implementação de políticas comuns e a adoção de soluções tecnológicas partilhadas. Neste enquadramento, a empresa assume uma função estratégica, funcionando como elemento agregador entre as áreas operacionais

do grupo e os sistemas de informação que suportam a sua atividade.

Embora a ValFinance integre o ValGrupo, o estágio decorre fisicamente nas instalações da ValSabor, uma das principais empresas do grupo, localizada na região de Alcanede, no distrito de Santarém. Esta proximidade física facilita a articulação entre os serviços corporativos e as restantes unidades de negócio do grupo.

Relativamente ao tipo de clientes, a ValFinance presta essencialmente serviços às empresas pertencentes ao ValGrupo, atuando, maioritariamente, como uma entidade de suporte interno. Indiretamente, os serviços prestados contribuem para o funcionamento e suporte aos clientes finais e empresariais das diferentes áreas de atividade do grupo.

2.2 Área de Atuação e Serviços

A área de atuação da ValFinance centra-se na prestação de serviços corporativos transversais às empresas do ValGrupo, desempenhando um papel estratégico no suporte às suas atividades operacionais. Entre os principais serviços prestados destacam-se a gestão contabilística e financeira, o desenvolvimento e manutenção de sistemas de informação, o apoio informático e a consultoria interna.

Este modelo de centralização de serviços permite ao grupo otimizar recursos, garantir maior controlo sobre os processos internos e promover a uniformização de procedimentos. No contexto do mercado, esta abordagem constitui uma vantagem competitiva, especialmente num grupo empresarial com múltiplas unidades de negócio e elevada complexidade operacional.

A tecnologia assume um papel fundamental na atividade da ValFinance, sendo indispensável para a gestão eficiente da informação, para o suporte às operações do grupo e para a tomada de decisão. O sistemo de informação desenvolvido e mantido pela empresa suporta diretamente os processos administrativos e operacionais das empresas do grupo, tornando a sua integração, fiabilidade e segurança fatores críticos para o funcionamento eficiente da organização.

2.3 Estrutura Organizacional

A estrutura organizacional da ValFinance encontra-se orientada para a prestação de serviços especializados às empresas do grupo, estando organizada por áreas funcionais. Entre os principais departamentos destacam-se as áreas de contabilidade e finanças, informática e sistemas de informação, consultoria interna e apoio administrativo.

No âmbito da área tecnológica, a ValFinance dispõe de uma equipa de desenvolvimento de *software*, atualmente composta por cerca de 10 elementos, responsável pelo desenvolvimento do sistema interno utilizado pelas várias empresas do grupo. Esta equipa desempenha um papel essencial na adaptação do sistema às necessidades específicas das diferentes unidades de negócio.

Para além da equipa de desenvolvimento, existe ainda um departamento de assistência informática, responsável pelo suporte técnico aos utilizadores, manutenção de equipamentos, resolução de incidentes e apoio à infraestrutura tecnológica da organização.

O estágio encontra-se integrado na equipa de desenvolvimento de *software* da ValFinance e decorre em regime híbrido, combinando trabalho presencial e remoto. A componente presencial realiza-se dois dias por semana nas instalações da empresa, nomeadamente na sede da ValSabor, permitindo um contacto direto com a equipa e com outros departamentos. A componente remota assegura a continuidade das atividades de desenvolvimento e suporte tecnológico, em conformidade com os procedimentos internos definidos pela empresa.

2.4 Infraestrutura Tecnológica

A infraestrutura tecnológica da ValFinance suporta as necessidades operacionais e administrativas das empresas do ValGrupo, recorrendo a um conjunto de sistemas de informação que permitem a gestão integrada de processos financeiros, administrativos e operacionais.

O ambiente tecnológico inclui sistemas internos desenvolvidos à medida, bem como plataformas de apoio à gestão e à comunicação entre departamentos. O desenvolvimento e a manutenção destes sistemas são assegurados internamente pela equipa de desenvolvimento, permitindo uma maior flexibilidade e adaptação às necessidades específicas do grupo.

Relativamente ao ambiente de desenvolvimento, são utilizados contextos controlados que permitem a implementação, teste e validação de novas funcionalidades antes da sua disponibilização em ambiente produtivo, contribuindo para a estabilidade e fiabilidade dos sistemas.

No que respeita às práticas relevantes para a segurança da informação, são adotadas medidas como o controlo de acessos, a separação de ambientes, a monitorização de sistemas e a realização de testes de segurança, com o objetivo de proteger a informação sensível e mitigar riscos associados à utilização de sistemas de informação em contexto empresarial. Estas práticas encontram-se alinhadas com modelos de gestão de cibersegurança, como o *NIST Cybersecurity Framework*, que estrutura a gestão do risco em torno de funções como governação, identificação, proteção, deteção, resposta e recuperação (National Institute of Standards and Technology, 2024).

3
Enquadramento do Estágio

Neste capítulo é apresentado o enquadramento detalhado do estágio, com enfoque nos seus objetivos, no contexto tecnológico em que se insere e nas metodologias de trabalho adotadas. São ainda descritas as principais ferramentas e tecnologias utilizadas ao longo do período de estágio.

3.1 Objetivos do Estágio

O estágio teve como objetivo geral a integração em contexto profissional, permitindo a aplicação prática dos conhecimentos adquiridos ao longo do Mestrado. Procurou proporcionar contacto com ambientes reais de desenvolvimento de *software* e de operação de sistemas de informação, promovendo a consolidação de competências técnicas e profissionais.

Como objetivos específicos, destacam-se a participação no desenvolvimento e manutenção do sistema interno da organização, recorrendo a tecnologias de desenvolvimento *web* utilizadas pela equipa, nomeadamente Laravel no *backend*, Vue.js no *frontend* e APIs internas para comunicação entre as diferentes camadas da aplicação. O estágio teve ainda como objetivo proporcionar contacto com práticas de segurança da informação em ambiente empresarial, incluindo o apoio a campanhas de sensibilização, integradas nas necessidades da empresa. Pretendeu-se também desenvolver capacidades de trabalho em equipa, organização e planeamento de tarefas, bem como a adaptação a metodologias de trabalho utilizadas em contexto profissional.

3.2 Contexto Tecnológico

O estágio decorreu num contexto tecnológico assente no desenvolvimento e manutenção de um sistema interno, desenvolvido pela própria equipa e utilizado por vários departamentos da organização. Este sistema, designado internamente como Portal, suporta processos administrativos, financeiros, operacionais e logísticos, tratando diferentes tipos de informação, incluindo dados sensíveis.

A aplicação desenvolvida é uma aplicação *web* organizada segundo uma arquitetura que separa a camada de apresentação da lógica de negócio e do acesso a dados. O desenvolvimento do *backend* é realizado em Laravel, uma *framework* utilizada para a criação de *endpoints* de API, implementação de controladores, modelos, validações e regras de negócio. O *frontend* é desenvolvido em Vue.js, permitindo a criação de interfaces modulares, componentes reutilizáveis, formulários dinâmicos e páginas adaptadas aos diferentes processos suportados pelo Portal.

A comunicação entre o *frontend* e o *backend* é realizada através de APIs internas, sendo implementados mecanismos de autenticação e autorização baseados em utilizadores, grupos de permissões e níveis de acesso distintos. Para apoio ao desenvolvimento e validação foram utilizadas ferramentas como Git, ambiente local de desenvolvimento e ambiente de *staging*.

Atendendo à natureza do sistema e ao tipo de dados tratados, a componente de segurança assume particular relevância, sendo considerados aspetos como o controlo de acessos, a proteção da informação, a validação dos dados no *backend* e a fiabilidade das comunicações entre os diferentes componentes da aplicação.

3.3 Metodologias de Trabalho

O trabalho desenvolvido ao longo do estágio foi organizado segundo uma abordagem inspirada em metodologias ágeis, adaptada à realidade da equipa. As atividades de desenvolvimento foram planeadas em *sprints* com duração aproximada de quinze dias de calendário, durante as quais foram definidas prioridades e objetivos para cada período de trabalho, com acompanhamento regular do progresso. Esta organização permitia ajustar as tarefas às necessidades da equipa, à evolução dos projetos e a eventuais problemas identificados durante o desenvolvimento.

Embora esta organização tenha sido mantida ao longo do estágio, não foi realizada uma contabilização consolidada do número total de *sprints*. Algumas tarefas transitaram entre ciclos ou foram temporariamente interrompidas para responder a situações urgentes, pelo que a indicação de um número exato não representaria com rigor o trabalho realizado. A distribuição temporal das principais atividades encontra-se representada no cronograma apresentado na Secção 3.4.

A integração progressiva de práticas de segurança no ciclo de desenvolvimento encontra-se alinhada com os princípios defendidos na literatura no âmbito do DevSecOps, que preconiza a incorporação da segurança desde as fases iniciais do desenvolvimento, reduzindo a probabilidade de introdução de vulnerabilidades estruturais (Rajapakse et al., 2022; Mohammed et al., 2025). Esta abordagem é igualmente coerente com o *Secure Software Development Framework* do NIST, que recomenda a integração de práticas de segurança ao longo do ciclo de vida do desenvolvimento de *software* (Souppaya et al., 2022).

As tarefas foram atribuídas por um elemento com funções de coordenação técnica, existindo reuniões diárias de acompanhamento para discussão do trabalho realizado,

identificação de dificuldades e planeamento das atividades seguintes. Sempre que necessário, eram realizadas reuniões adicionais para apresentação de funcionalidades desenvolvidas, validação de soluções e esclarecimento de requisitos.

A gestão das tarefas incluía atividades de natureza diversa, abrangendo tanto o desenvolvimento e manutenção de *software* como tarefas relacionadas com a segurança da informação. As atividades complementares de cibersegurança, incluindo a campanha controlada de sensibilização para o *phishing* e a identificação observacional de uma vulnerabilidade, foram realizadas de forma integrada no estágio, embora não fizessem parte do fluxo regular de desenvolvimento.

O controlo de versões foi realizado com recurso ao Git, sendo o código-fonte alojado e gerido através do Bitbucket, plataforma utilizada pela equipa para organizar os repositórios do sistema interno. Cada tarefa definida no *backlog* do Jira correspondia a uma ramificação específica. Esta abordagem permitiu uma melhor organização do trabalho, a colaboração entre elementos da equipa e a manutenção de um histórico consistente das alterações efetuadas.

3.4 Cronograma do Estágio

O estágio decorreu entre 15 de setembro de 2025 e 28 de maio de 2026, correspondendo a uma duração total de 1400 horas, distribuídas ao longo de aproximadamente nove meses. As atividades foram organizadas de forma progressiva e ajustadas às necessidades da equipa de desenvolvimento, seguindo ciclos de trabalho com duração aproximada de quinze dias e reuniões diárias de acompanhamento.

Atividade	Set.	Out.	Nov.	Dez.	Jan.	Fev.	Mar.	Abr.	Mai.
Integração e adaptação inicial									
Tarefas iniciais no sistema interno									
Início e enquadramento ValSabor									
Desenvolvimento inicial do portal ValSabor									
Reavaliação e refinamento de requisitos									
Integrações ERP/SAP e consolidação ValSabor									
Campanha controlada de *phishing*									
Elaboração e revisão do relatório									

Figura 3.1: *Cronograma das principais fases do estágio*

Numa fase inicial, o trabalho incidiu sobretudo na integração na equipa, na configuração do ambiente de desenvolvimento, na compreensão da arquitetura do sistema já desenvolvido e na realização de tarefas de menor complexidade. Posteriormente, o estágio evoluiu para a participação em projetos de maior dimensão, com especial destaque para o projeto ValSabor, que constituiu o principal caso de estudo deste relatório.

O projeto ValSabor teve início aproximadamente em meados de dezembro de 2025. Numa primeira fase, foram realizadas reuniões de enquadramento e identificação dos requisitos iniciais, seguindo-se uma fase de desenvolvimento mais acelerado, orientada para a disponibilização de uma primeira versão funcional no portal. Posteriormente, o projeto entrou numa fase de reavaliação, teste e consolidação, permitindo ajustar regras de negócio, reforçar validações e melhorar a fiabilidade das funcionalidades implementadas.

A Figura 3.1 sintetiza, por mês, as principais fases do estágio.

O cronograma deve ser entendido como uma representação aproximada, uma vez que algumas atividades decorreram em paralelo e foram sendo ajustadas ao longo do estágio.

3.5 Tecnologias e Ferramentas Utilizadas

Ao longo do estágio foram utilizadas diferentes tecnologias e ferramentas de apoio ao desenvolvimento, teste, integração e validação das funcionalidades implementadas. Embora tenham sido realizadas várias tarefas no sistema interno da organização, a presente secção centra-se nas tecnologias mais relevantes para o projeto ValSabor, por este constituir o principal caso de estudo do relatório.

O sistema foi desenvolvido com base numa arquitetura *web full stack*, recorrendo a Laravel no *backend* e Vue.js no *frontend*. O *backend* ficou responsável pela lógica de negócio, validação de dados, controlo de acessos, persistência da informação e comunicação com serviços externos. O *frontend* assegurou a interação com os utilizadores, através de páginas, componentes, formulários e fluxos adaptados ao contexto operacional.

Para além das tecnologias de desenvolvimento *web*, o projeto envolveu a utilização de APIs internas, leitura de códigos associados à norma GS1, leitores de códigos de barras, balanças industriais, geração de etiquetas em ZPL, controlo de versões com Git, validação em ambiente de *staging* e testes de APIs com recurso ao Postman. A Tabela 3.1 apresenta uma síntese das principais áreas tecnológicas envolvidas, sendo a descrição detalhada apresentada no Apêndice A.

A utilização destas tecnologias foi relevante para garantir que o sistema não se limitava à criação de interfaces aplicacionais, mas suportava processos operacionais concretos. Em particular, a articulação entre *backend*, *frontend*, equipamentos industriais e serviços externos permitiu desenvolver funcionalidades orientadas para validação, rastreabilidade, controlo de acessos e integração com sistemas empresariais.

A utilização destas tecnologias resultou, em grande parte, do ecossistema tecnológico já adotado pela empresa no desenvolvimento do sistema. O *backend* encontrava-se

Tabela 3.1: *Síntese das tecnologias e ferramentas utilizadas*

Área	Tecnologias e função no projeto
Desenvolvimento aplicacional	Laravel no *backend* e Vue.js no *frontend*, utilizados para implementar lógica de negócio, validações, interfaces dinâmicas e comunicação entre componentes.
Integração entre sistemas	APIs internas e serviços externos associados ao SAP, utilizados para comunicação entre *frontend*, *backend* e sistemas empresariais externos.
Identificação e operação industrial	Códigos GS1, leitores de códigos de barras, balanças industriais e etiquetas em ZPL, utilizados para ligar operações físicas aos respetivos registos digitais.
Segurança e integridade	Validações no *backend*, controlo de acessos, prevenção de duplicados e registo de operações, utilizados para proteger dados e garantir coerência operacional.
Desenvolvimento e validação	Git, ambiente local, ambiente de *staging* e Postman, utilizados para controlo de versões, testes, validação de APIs e verificação antes da passagem para produção.

desenvolvido em Laravel e o *frontend* em Vue.js, pelo que a continuidade destas tecnologias permitiu manter coerência com a arquitetura existente, seguir os padrões da equipa de desenvolvimento e integrar as novas funcionalidades com os módulos já implementados. As ferramentas de apoio ao desenvolvimento, como Git, Postman e ambiente de *staging*, foram utilizadas por fazerem parte do processo habitual de desenvolvimento, teste e validação da equipa. Já as tecnologias associadas à operação, como leitores de códigos de barras, balanças industriais, ZPL e GS1, decorreram das necessidades específicas dos processos logísticos e industriais suportados pelo projeto ValSabor.

4
Estado da Arte

O presente capítulo apresenta o enquadramento teórico e técnico necessário à compreensão dos principais conceitos associados ao trabalho desenvolvido durante o estágio. Atendendo à natureza do projeto analisado, são abordados temas relacionados com sistemas de informação empresariais em contexto industrial, integração entre sistemas, segurança em aplicações *web*, validação de dados, controlo de acessos, Integridade dos dados e rastreabilidade em processos logísticos.

A finalidade deste capítulo não é descrever a solução desenvolvida, mas enquadrar os conceitos e desafios que ajudam a compreender o caso de estudo apresentado posteriormente. Assim, são analisadas práticas, documentos de referência e problemas associados ao desenvolvimento de aplicações empresariais que interagem com processos operacionais e sistemas externos.

4.1 Sistemas de Informação Empresariais e ERP

Os sistemas de informação empresariais desempenham um papel relevante no suporte às atividades das organizações, permitindo recolher, processar, armazenar e disponibilizar informação necessária à execução e gestão de processos internos. Em contextos industriais e logísticos, estes sistemas tornam-se particularmente importantes, uma vez que suportam operações relacionadas com produção, gestão de *stocks*, expedição, receção de mercadorias, controlo de qualidade e acompanhamento de unidades logísticas (Butarbutar et al., 2023; Zhou et al., 2022).

Entre estes sistemas, os sistemas ERP assumem uma função central, ao integrarem informação proveniente de diferentes áreas da organização, como produção, logística, finanças, compras e gestão de armazém. A utilização de sistemas ERP permite reduzir redundâncias, melhorar a coerência dos dados e apoiar a tomada de decisão com base numa visão mais integrada dos processos organizacionais (Butarbutar et al., 2023).

Contudo, os sistemas ERP não funcionam, na prática, de forma isolada. Em muitos ambientes empresariais, coexistem com aplicações desenvolvidas internamente, sistemas operacionais especializados, dispositivos industriais e serviços de integração. Es-

tas aplicações podem funcionar como pontos de recolha, validação e preparação de dados antes da sua comunicação ao sistema central. Esta realidade torna a integração entre sistemas um aspeto relevante na arquitetura tecnológica das organizações.

No âmbito deste relatório, a referência a sistemas ERP é enquadrada sobretudo na perspetiva da integração aplicacional. Em contextos empresariais, as aplicações operacionais desempenham frequentemente um papel complementar aos sistemas ERP, ao recolher, validar e estruturar dados provenientes dos processos de negócio antes da sua comunicação a plataformas empresariais externas. Esta perspetiva permite centrar a análise na interação entre aplicações operacionais e sistemas empresariais, sem aprofundar aspetos de administração ou configuração interna das plataformas ERP.

A coexistência de vários sistemas pode introduzir desafios ao nível da consistência da informação, da sincronização de dados e da representação correta do estado das operações. Em processos logísticos e industriais, estes desafios assumem maior relevância porque os dados digitais estão frequentemente associados a operações físicas, como movimentações de mercadoria, identificação de lotes, criação de caixas, composição de paletes ou alterações de localização. Esta relação entre dados digitais e operações físicas reforça a necessidade de mecanismos de validação, controlo, Integridade dos dados e rastreabilidade ao longo dos processos (International Organization for Standardization, 2007; Zhou et al., 2022).

Deste modo, a fiabilidade de um sistema empresarial não depende apenas do ERP enquanto plataforma central, mas também da qualidade dos dados recebidos, dos mecanismos de validação existentes nas aplicações periféricas e da forma como a integração entre sistemas é desenhada e controlada.

4.2 Integração entre Sistemas, APIs e Serviços Externos

A integração entre sistemas constitui um elemento essencial em aplicações empresariais modernas. À medida que as organizações utilizam diferentes aplicações para suportar processos específicos, torna-se necessário assegurar a comunicação entre componentes internos, serviços externos, sistemas de gestão e, em alguns casos, equipamentos industriais. Esta comunicação deve garantir que os dados trocados entre sistemas são compreensíveis, completos e adequados ao processo de negócio em causa.

As Application Programming Interface (API) são um dos mecanismos mais utilizados para permitir esta comunicação. Através de APIs, diferentes sistemas podem trocar informação de forma estruturada, permitindo separar responsabilidades entre componentes e facilitar a evolução independente de diferentes partes da arquitetura. Em aplicações *web* empresariais, as APIs são frequentemente utilizadas para comunicação entre *frontend* e *backend*, bem como para integração com serviços externos.

Apesar das vantagens associadas à utilização de APIs, a sua adoção introduz desafios de segurança, validação e controlo. Uma API que suporte operações críticas deve garantir que os pedidos recebidos são autenticados, autorizados e validados antes da execução da operação. O Application Security Verification Standard (ASVS), na versão

5.0.0, desenvolvido pela Open Web Application Security Project (OWASP), identifica requisitos técnicos relacionados com autenticação, controlo de acessos, validação de entradas, tratamento de erros e segurança de APIs, constituindo uma referência útil para a verificação de controlos de segurança em aplicações *web* (OWASP Foundation, 2025; Wen et al., 2023).

A segurança de APIs é também abordada pelo OWASP API Security Top 10, que sistematiza riscos específicos associados a este tipo de interface, incluindo falhas de autorização, autenticação insuficiente, exposição excessiva de dados e consumo inseguro de APIs externas (OWASP Foundation, 2023). Estes riscos são particularmente relevantes em aplicações empresariais que comunicam com serviços externos ou disponibilizam operações críticas através de interfaces aplicacionais.

Em integrações com serviços externos, a comunicação entre sistemas pode ser afetada por falhas de rede, indisponibilidade temporária, respostas inesperadas, *timeouts* ou repetição de pedidos. Estas situações podem originar erros operacionais ou inconsistências se não forem tratadas de forma controlada. Por esse motivo, torna-se relevante prever mecanismos de validação prévia, registo de erros, controlo de respostas e prevenção de operações duplicadas. A semântica do HTTP, definida no RFC 9110, é relevante neste contexto por clarificar propriedades de métodos, incluindo o conceito de idempotência em operações HTTP (Fielding et al., 2022).

Assim, a integração entre sistemas deve ser entendida não apenas como uma questão de comunicação técnica, mas também como um problema de fiabilidade, segurança e qualidade dos dados. A robustez da integração depende da capacidade de validar a informação antes do envio, controlar o resultado das operações e impedir que falhas pontuais originem inconsistências persistentes entre sistemas.

4.3 Consistência, Idempotência e Tolerância a Falhas

A consistência dos dados é um aspeto fundamental em sistemas que envolvem múltiplos componentes ou aplicações. Quando diferentes sistemas participam no mesmo processo de negócio, é necessário garantir que a informação registada permanece coerente ao longo das várias etapas. A ausência de consistência pode originar duplicação de registos, estados contraditórios, perda de informação ou dificuldade em reconstruir o histórico de uma operação.

Em contextos industriais e logísticos, esta questão é particularmente sensível, uma vez que os dados digitais procuram representar operações físicas. A leitura de um código de barras, a associação de um lote, a criação de uma unidade logística, a pesagem de uma carga ou a expedição de mercadoria são exemplos de operações em que o estado registado no sistema deve corresponder ao estado real observado no processo operacional. A rastreabilidade na cadeia de abastecimento depende precisamente da capacidade de relacionar eventos, entidades e localizações ao longo do processo (International Organization for Standardization, 2007; Zhou et al., 2022).

A Idempotência é um conceito relevante quando uma operação pode ser repetida,

por exemplo devido a falhas de comunicação ou incerteza quanto ao resultado de um pedido anterior. De forma geral, uma operação idempotente é aquela cuja repetição não produz efeitos adicionais indesejados quando executada com os mesmos parâmetros. Este conceito é particularmente importante em APIs e integrações entre sistemas, pois ajuda a reduzir o risco de duplicação de operações. No contexto da semântica HTTP, o RFC 9110 define métodos idempotentes e clarifica a importância desta propriedade em pedidos que podem ser repetidos em caso de falha de comunicação (Fielding et al., 2022).

A Tolerância a falhas refere-se à capacidade de um sistema lidar com erros ou interrupções de forma controlada. Em aplicações empresariais, esta capacidade pode envolver validações antes da execução de operações críticas, tratamento adequado de exceções, registo de Logging, mensagens de erro apropriadas, mecanismos de repetição controlada e procedimentos de análise ou correção posterior. Estes mecanismos não eliminam a possibilidade de erro, mas permitem limitar o seu impacto e facilitar o diagnóstico.

A preocupação com o tratamento de falhas está também relacionada com a segurança aplicacional. O OWASP ASVS na versão 5.0.0 inclui requisitos relacionados com gestão de erros, validação de entradas, registo de eventos e proteção das interfaces aplicacionais, aspetos que contribuem para reduzir a exposição de informação sensível e melhorar a capacidade de análise posterior de incidentes (OWASP Foundation, 2025). Em aplicações que integram dados operacionais com sistemas externos, estes mecanismos são relevantes para evitar que erros pontuais se transformem em inconsistências persistentes.

Deste modo, consistência, Idempotência e Tolerância a falhas constituem preocupações relevantes no desenvolvimento de aplicações empresariais integradas. Estes conceitos são especialmente importantes quando as operações realizadas numa aplicação têm impacto noutros sistemas ou representam movimentos físicos dentro de um processo industrial ou logístico.

4.4 Segurança em Aplicações *Web* Empresariais

As aplicações *web* empresariais são utilizadas para suportar processos internos, gerir informação sensível e disponibilizar funcionalidades a diferentes perfis de utilizadores. Em muitos casos, estas aplicações comunicam com outros sistemas e executam operações com impacto direto na atividade da organização. Por esse motivo, a segurança deve ser considerada desde as fases iniciais do desenvolvimento e não apenas como uma verificação posterior.

O Secure Software Development Framework (SSDF), proposto pelo National Institute of Standards and Technology (NIST), recomenda a incorporação de práticas de segurança ao longo do ciclo de vida do desenvolvimento de *software*, incluindo preparação da organização, proteção do *software*, produção de *software* seguro e resposta a vulnerabilidades identificadas (Souppaya et al., 2022). Esta perspetiva é coerente com

abordagens de desenvolvimento seguro e com a integração progressiva de práticas de segurança nos processos de desenvolvimento.

A literatura sobre DevSecOps reforça esta preocupação, defendendo a integração da segurança nos processos de desenvolvimento e operação. Estudos recentes identificam o DevSecOps como uma abordagem que procura reduzir a separação entre desenvolvimento, operações e segurança, promovendo validação contínua, automatização e identificação mais precoce de vulnerabilidades (Rajapakse et al., 2022; Mohammed et al., 2025). A consciência e maturidade dos programadores relativamente a práticas de segurança continuam também a ser fatores relevantes na redução de vulnerabilidades em *software* (Alqahtani, 2025).

O Open Web Application Security Project (OWASP) Top 10:2021 é um documento de sensibilização que apresenta uma seleção das categorias de riscos de segurança mais críticas para aplicações *web*. Elaborado com base em contributos da comunidade e na análise de dados sobre vulnerabilidades, este permite identificar problemas recorrentes e apoiar equipas de desenvolvimento na definição de prioridades de segurança. A sua importância resulta da ampla adoção no setor e da capacidade de fornecer uma linguagem comum para comunicar riscos entre programadores, profissionais de segurança e organizações. Entre os riscos identificados encontram-se falhas de controlo de acessos, falhas criptográficas, injeção, desenho inseguro, Security Misconfiguration e utilização de componentes vulneráveis (OWASP Foundation, 2021). Embora não substitua uma análise de risco completa nem constitua um conjunto exaustivo de requisitos técnicos, o OWASP Top 10:2021 é útil para enquadrar categorias de vulnerabilidades frequentemente observadas em aplicações *web*.

Para uma abordagem mais técnica, o OWASP ASVS na versão 5.0.0 apresenta requisitos verificáveis para diferentes áreas de segurança aplicacional, incluindo autenticação, gestão de sessões, controlo de acessos, validação de dados, tratamento de erros, proteção de dados e segurança de APIs (OWASP Foundation, 2025). Trabalhos recentes têm explorado a utilização do ASVS como base para avaliação quantitativa da segurança em aplicações *web*, demonstrando a sua relevância enquanto referencial técnico (Wen et al., 2023).

A avaliação de segurança em aplicações *web* pode ainda recorrer a diferentes abordagens e ferramentas, incluindo testes estáticos, testes dinâmicos e validações manuais. A literatura recente evidencia que a escolha e combinação destas abordagens influencia a capacidade de identificar vulnerabilidades de forma eficaz (Qadir et al., 2025). Em contexto empresarial, estas práticas devem ser articuladas com o ciclo de desenvolvimento e com os procedimentos internos de validação antes da disponibilização de funcionalidades.

Em aplicações empresariais, uma vulnerabilidade pode afetar não apenas a confidencialidade dos dados, mas também a integridade das operações. Uma falha de autorização pode permitir que um utilizador execute ações fora do seu perfil; uma validação insuficiente pode permitir a introdução de dados inválidos; e uma configuração inadequada pode expor informação técnica sensível. Estes riscos justificam a adoção

de mecanismos de segurança aplicados de forma transversal ao desenvolvimento.

4.5 Controlo de Acessos, Validação e Integridade de Dados

O controlo de acessos é um dos elementos centrais da segurança em aplicações empresariais. A sua função é garantir que cada utilizador apenas acede aos recursos e executa as operações compatíveis com o seu perfil, função ou nível de responsabilidade. Este aspeto é particularmente importante em sistemas internos utilizados por diferentes departamentos, onde nem todos os utilizadores devem ter acesso às mesmas funcionalidades ou aos mesmos dados.

Falhas de controlo de acessos encontram-se entre os riscos destacados pelo OWASP Top 10:2021, sendo especialmente relevantes em aplicações com múltiplos perfis de utilizador (OWASP Foundation, 2021). A investigação recente continua a apontar o controlo de acessos como uma área crítica em aplicações *web*, em particular quando existem regras de autorização complexas ou dependentes do contexto da operação (Xu et al., 2025).

Modelos baseados em papéis, frequentemente designados por Role-Based Access Control (RBAC), são utilizados para organizar permissões de acordo com funções atribuídas aos utilizadores. Em aplicações empresariais, este tipo de modelo pode facilitar a gestão de permissões e a separação de responsabilidades, mas deve ser complementado por validações adicionais ao nível da operação, sobretudo quando existem restrições associadas a entidades, departamentos, estados de processo ou propriedade dos dados. A relevância e complexidade do controlo de acessos em aplicações *web* é reforçada por estudos recentes sobre deteção e reparação de vulnerabilidades em modelos baseados em papéis (Xu et al., 2025).

A validação de dados constitui outro aspeto essencial para a segurança e Integridade dos dados das aplicações. Embora a validação no *frontend* contribua para melhorar a experiência do utilizador e reduzir erros de introdução, não deve ser considerada suficiente para proteger a aplicação. O *frontend* pode ser manipulado e os pedidos podem ser enviados diretamente para a API, pelo que as validações críticas devem ser realizadas no *backend* ou em componentes controlados pelo servidor (OWASP Foundation, 2025; Souppaya et al., 2022).

A Integridade dos dados diz respeito à garantia de que a informação se mantém correta, completa e coerente ao longo do seu ciclo de vida. Em sistemas empresariais, esta integridade depende da validação de entradas, da aplicação consistente de regras de negócio, da gestão de estados, da prevenção de duplicações e da existência de registos que permitam analisar operações realizadas. Em contextos logísticos, a integridade dos dados é particularmente relevante porque a informação registada suporta a reconstrução do percurso dos produtos e unidades ao longo da cadeia (International Organization for Standardization, 2007; Zhou et al., 2022).

O tratamento de erros também influencia a segurança da aplicação. Mensagens de erro demasiado detalhadas podem revelar informação técnica sensível, enquanto

a ausência de registos internos dificulta o diagnóstico de problemas. Por este motivo, boas práticas de segurança recomendam que a informação apresentada ao utilizador seja controlada e que os detalhes técnicos relevantes sejam registados internamente para análise (OWASP Foundation, 2021; OWASP Foundation, 2025).

Assim, controlo de acessos, validação e Integridade dos dados devem ser entendidos como mecanismos complementares. O controlo de acessos limita quem pode executar determinadas operações; a validação confirma se os dados e condições da operação são aceitáveis; e os mecanismos de integridade procuram assegurar que a informação permanece coerente ao longo do processo.

4.6 Rastreabilidade em Contextos Industriais e Logísticos

A rastreabilidade é um conceito relevante em contextos industriais e logísticos, permitindo acompanhar produtos, componentes ou unidades logísticas ao longo das diferentes fases de um processo. Esta capacidade é importante para controlo de qualidade, gestão de *stock*, Auditabilidade, análise de anomalias e reconstrução histórica de operações. No domínio alimentar, a norma ISO 22005 define princípios e requisitos básicos para a conceção e implementação de sistemas de rastreabilidade na cadeia alimentar, sendo útil para enquadrar a rastreabilidade enquanto capacidade de identificar a história, localização e elementos relevantes de um produto ao longo da cadeia (International Organization for Standardization, 2007).

Em cadeias produtivas, a rastreabilidade depende da identificação correta das entidades envolvidas, como artigos, lotes, caixas, paletes, unidades de transporte, armazéns e movimentos. A relação entre estas entidades permite reconstruir o percurso de um produto ou unidade logística e compreender as operações realizadas ao longo do seu ciclo de vida. Revisões recentes sobre rastreabilidade em cadeias alimentares destacam a importância da identificação, recolha e partilha de informação ao longo da cadeia de abastecimento (Zhou et al., 2022).

As normas de identificação, como as definidas pela GS1, são frequentemente utilizadas para representar informação sobre produtos e unidades logísticas através de códigos de barras. A utilização de identificadores como o Global Trade Item Number (GTIN) e o Serial Shipping Container Code (SSCC) contribui para reduzir ambiguidades na identificação de produtos, lotes e unidades de transporte. As especificações gerais da GS1 estabelecem regras para a utilização de chaves de identificação e dados associados, enquanto a GS1 *Logistic Label Guideline* apresenta orientações específicas para etiquetas logísticas e identificação de unidades logísticas através do SSCC (GS1, 2025; GS1, 2019).

A rastreabilidade não depende apenas da existência de códigos de identificação. É igualmente necessário que os sistemas registem os eventos relevantes associados a cada entidade, como criação, associação, movimentação, alteração de estado, expedição, receção ou integração com sistemas externos. Sem estes registos, torna-se mais difícil reconstruir o histórico de uma operação ou identificar a origem de uma inconsistência

(International Organization for Standardization, 2007; Zhou et al., 2022).

Em ambientes industriais, a rastreabilidade pode ser reforçada através da combinação entre identificação automática e validações físicas. A leitura de códigos de barras reduz a dependência de introdução manual de dados, enquanto mecanismos como a pesagem permitem confirmar determinados aspetos da operação física. Esta articulação entre identificação, registo e validação contribui para aproximar o estado digital registado no sistema do estado físico observado no processo operacional.

A rastreabilidade apresenta também uma dimensão relevante para a segurança e Auditabilidade. O registo de operações, a associação de ações a utilizadores e a manutenção de histórico de estados contribuem para a análise posterior de incidentes, para a responsabilização e para a deteção de comportamentos anómalos. Esta perspetiva aproxima a rastreabilidade de conceitos como Logging, Auditabilidade e *accountability*, frequentemente associados à governação e segurança dos sistemas de informação (National Institute of Standards and Technology, 2024; OWASP Foundation, 2025).

Deste modo, em contextos industriais e logísticos, a rastreabilidade deve ser entendida como uma capacidade transversal. Esta capacidade depende da correta identificação das entidades, da qualidade dos dados registados, da consistência dos estados e da existência de mecanismos que permitam reconstruir o histórico das operações.

4.7 Segurança em Ambientes IIoT e Comunicações MQTT

A integração de equipamentos físicos com aplicações e redes empresariais enquadra-se no domínio da Industrial Internet of Things (IIoT). Este conceito refere-se à aplicação de instrumentos, sensores e outros dispositivos interligados a máquinas, veículos e infraestruturas industriais. Embora esta conectividade permita apoiar a monitorização e a automatização dos processos, introduz também desafios relacionados com a integridade das comunicações, autenticação, controlo de acessos e proteção dos dispositivos interligados (McCarthy et al., 2022).

Em ambientes industriais, uma falha de comunicação ou a manipulação dos dados transmitidos pode afetar não apenas a informação registada, mas também a representação digital das operações físicas. Por esse motivo, a segurança destes ambientes deve considerar mecanismos como autenticação dos dispositivos, autorização, proteção das comunicações, validação dos dados recebidos, registo de eventos e segmentação da rede (McCarthy et al., 2022).

Neste tipo de ambientes, o Message Queuing Telemetry Transport (MQTT) é um protocolo de comunicação baseado no modelo de publicação e subscrição, frequentemente utilizado na troca de mensagens entre dispositivos e aplicações. A especificação do protocolo não impõe uma tecnologia de segurança específica, cabendo aos responsáveis pela implementação selecionar mecanismos adequados de autenticação, autorização, privacidade e proteção das comunicações. A especificação recomenda a utilização destas capacidades em contextos que envolvam infraestruturas críticas ou informação sensível (Banks et al., 2019).

Neste contexto, os dados provenientes de dispositivos industriais não devem ser considerados automaticamente confiáveis. A aplicação que os recebe deve validar o seu formato, origem e coerência antes de os utilizar em operações críticas. Estas preocupações são relevantes para compreender a interação entre o sistema analisado neste relatório e os equipamentos utilizados nos processos operacionais (McCarthy et al., 2022).

4.8 Sensibilização para a Segurança e Campanhas Simuladas de *Phishing*

O fator humano constitui uma dimensão relevante da segurança da informação, uma vez que ataques de engenharia social procuram explorar comportamentos dos utilizadores, como a confiança em comunicações aparentemente legítimas ou a interação com ligações fraudulentas. Por esse motivo, os programas de sensibilização procuram promover mudanças comportamentais e desenvolver uma cultura de segurança nas organizações.

O NIST SP 800-50 Rev. 1 (Hansche et al., 2024) recomenda que os programas de aprendizagem em cibersegurança sejam planeados e geridos ao longo de um ciclo contínuo, incluindo identificação de necessidades, definição de objetivos, implementação das atividades e avaliação dos resultados. O referencial destaca ainda a importância da utilização de métricas para avaliar a eficácia das iniciativas e orientar a sua melhoria contínua.

As campanhas simuladas de *phishing* permitem avaliar o comportamento dos utilizadores perante comunicações fraudulentas num ambiente controlado. Entre as métricas que podem ser analisadas encontram-se a taxa de abertura das mensagens, a taxa de interação com ligações, a submissão de informação, a comunicação da mensagem suspeita e a evolução destes comportamentos entre campanhas sucessivas. Contudo, estas métricas devem ser interpretadas considerando o contexto da campanha, o grau de dificuldade do cenário e os perfis dos participantes (Hansche et al., 2024).

Para além da avaliação, as campanhas simuladas devem ser acompanhadas por ações de sensibilização e medidas de melhoria. O seu objetivo não deve consistir apenas na identificação de utilizadores que interagem com mensagens fraudulentas, mas na utilização dos resultados para reforçar conhecimentos, reduzir comportamentos de risco e melhorar progressivamente a cultura de segurança da organização (Hansche et al., 2024).

Os resultados obtidos em campanhas simuladas podem variar significativamente em função do contexto organizacional, da experiência prévia dos participantes e da credibilidade do cenário utilizado. Num estudo realizado em seis instituições de saúde, que envolveu aproximadamente 2,9 milhões de mensagens simuladas, foi observada uma taxa global de interação com ligações de 14,2% e uma mediana global de 16,7%. Contudo, as medianas registadas nas diferentes instituições variaram entre 7,4%

e 30,7%, demonstrando que os resultados devem ser interpretados considerando as características específicas de cada campanha e organização (Gordon et al., 2019).

Num contexto de infraestrutura crítica, um estudo realizado junto de 735 trabalhadores de uma empresa ferroviária tailandesa observou uma taxa de interação de 10,9% na primeira campanha simulada. A realização de uma segunda campanha com um cenário diferente produziu resultados distintos, demonstrando que o conteúdo e o contexto das mensagens podem influenciar o comportamento dos utilizadores (Sirawongphatsara et al., 2024).

4.9 Síntese do Capítulo

A revisão apresentada permitiu enquadrar os principais conceitos relevantes para o desenvolvimento de aplicações empresariais em contexto industrial e logístico. Os sistemas ERP assumem um papel central na gestão integrada das organizações, sendo a sua integração contínua com os processos organizacionais relevante após a implementação (Butarbutar et al., 2023). Neste contexto, aplicações operacionais e serviços externos podem recolher, validar e preparar dados antes da sua comunicação ao sistema central.

A integração entre sistemas, especialmente quando realizada através de APIs ou serviços externos, introduz desafios relacionados com segurança, consistência, tratamento de erros e prevenção de duplicações. Estes desafios tornam-se mais relevantes quando os dados integrados correspondem a operações físicas ou processos logísticos, nos quais a informação digital deve refletir corretamente o estado operacional (International Organization for Standardization, 2007; Zhou et al., 2022).

No domínio da segurança aplicacional, os referenciais analisados destacam a importância do desenvolvimento seguro, do controlo de acessos, da validação de dados no lado do servidor, do tratamento adequado de erros e da proteção das APIs (Souppaya et al., 2022; OWASP Foundation, 2021; OWASP Foundation, 2025). Estes aspetos são particularmente relevantes em aplicações empresariais que tratam dados operacionais e disponibilizam funcionalidades a diferentes perfis de utilizadores.

A Integridade dos dados e a rastreabilidade surgem também como preocupações essenciais. Em processos industriais e logísticos, é necessário assegurar que os dados registados são coerentes, completos e associados às entidades corretas, permitindo acompanhar o ciclo de vida das operações e apoiar auditorias ou análises posteriores. A utilização de normas de identificação e de princípios formais de rastreabilidade contribui para enquadrar estas preocupações em sistemas que dependem da correspondência entre eventos físicos e registos digitais (GS1, 2025; GS1, 2019; International Organization for Standardization, 2007; Zhou et al., 2022).

A segurança dos ambientes industriais envolve também a proteção das comunicações com dispositivos interligados e a validação dos dados por estes transmitidos. Paralelamente, a sensibilização dos utilizadores e a avaliação do seu comportamento através de campanhas simuladas constituem componentes relevantes de uma estratégia de segurança abrangente.

Este enquadramento teórico prepara a análise do caso de estudo apresentado nos capítulos seguintes, no qual serão discutidas decisões técnicas relacionadas com validação, controlo de acessos, integração com sistemas externos, rastreabilidade e segurança no desenvolvimento de uma aplicação empresarial em contexto industrial.

5 Trabalho Desenvolvido

Este capítulo encontra-se estruturado de forma a refletir a progressão natural do trabalho realizado, iniciando-se com a análise dos requisitos do projeto ValSabor e a respetiva sistematização, seguindo-se o planeamento e organização das atividades, o desenvolvimento *full stack*, a integração de aspetos de segurança e, por fim, os processos de testes e validação das soluções implementadas.

Ao longo do estágio, o trabalho incidiu maioritariamente no desenvolvimento e evolução do sistema interno da organização, recorrendo a tecnologias modernas de desenvolvimento *web*. Para além das atividades de desenvolvimento, o estágio incluiu tarefas relacionadas com a segurança da informação, integradas de acordo com as necessidades da empresa, permitindo uma abordagem prática e aplicada à cibersegurança em contexto empresarial.

Em linha com os desafios identificados no Capítulo 4, nomeadamente ao nível da integração entre sistemas, da consistência dos dados, do controlo de acessos e da rastreabilidade em contexto industrial, o trabalho desenvolvido procurou responder a problemas concretos observados no ambiente organizacional. Assim, este capítulo não se limita a descrever as tarefas realizadas, procurando também evidenciar de que forma as decisões técnicas adotadas contribuíram para mitigar limitações frequentemente apontadas na literatura.

Entre as várias atividades e projetos desenvolvidos, destaca-se o sistema de gestão de unidades logísticas, associado ao projeto ValSabor, devido à sua complexidade, impacto operacional e requisitos de segurança. Este sistema será analisado em detalhe no Capítulo 6.

5.1 Análise de Requisitos do Projeto ValSabor

A análise dos requisitos do projeto ValSabor não resultou de um processo formal completo de engenharia de requisitos, com entrevistas sistemáticas aos utilizadores finais ou documentação inicial exaustiva. O processo seguido refletiu a realidade do contexto empresarial em que o estágio decorreu, no qual as necessidades foram transmi-

tidas principalmente pela equipa técnica e pelo responsável da área de informática, que funcionavam como intermediários entre os processos operacionais e a equipa de desenvolvimento.

Numa fase inicial do estágio, as tarefas atribuídas estavam sobretudo relacionadas com correções, melhorias pontuais e adaptação a sistemas existentes. Por esse motivo, a análise de requisitos tinha inicialmente um caráter mais genérico e transversal às várias atividades realizadas. No entanto, com a evolução do estágio e a consolidação do projeto ValSabor como principal caso de estudo do relatório, tornou-se necessário identificar de forma mais concreta os requisitos associados a este sistema.

No projeto ValSabor, os requisitos foram sendo transmitidos de forma progressiva, frequentemente através de indicações práticas sobre o comportamento esperado do sistema. Em muitos casos, as necessidades eram apresentadas sob a forma de regras operacionais concretas, tais como validações a aplicar, estados permitidos, sequência correta de operações, restrições associadas a artigos ou lotes, condições para criação de caixas, regras para agrupamento de caixas ou critérios necessários para o fecho de unidades logísticas.

Este modelo exigiu um processo contínuo de clarificação. À medida que as funcionalidades eram implementadas e testadas, surgiam novas exceções, regras de negócio adicionais e ajustes necessários ao funcionamento real da operação. Assim, os requisitos não se mantiveram estáticos ao longo do desenvolvimento, tendo sido ajustados em função das necessidades identificadas pela equipa técnica, dos testes realizados e das particularidades dos processos industriais e logísticos envolvidos.

A ausência de contacto direto sistemático com os utilizadores finais constitui uma limitação do processo de análise de requisitos, uma vez que parte da informação funcional foi transmitida de forma mediada. Ainda assim, este modelo permitiu manter alinhamento com a equipa responsável pelo sistema e responder de forma progressiva às necessidades operacionais identificadas. A validação das funcionalidades foi realizada através de testes, revisão pela equipa técnica e utilização de ambientes controlados antes da disponibilização das alterações em produção.

Desta forma, a análise de requisitos assumiu uma natureza prática e evolutiva, adequada ao contexto do estágio e ao ambiente empresarial em que o projeto foi desenvolvido. O objetivo principal consistiu em transformar necessidades operacionais concretas em funcionalidades aplicacionais fiáveis, assegurando simultaneamente validação de dados, controlo de acessos, rastreabilidade e integração com sistemas externos.

5.2 Requisitos do Projeto ValSabor

Tendo em conta o foco do relatório no projeto ValSabor, tornou-se necessário sistematizar os principais requisitos associados ao sistema. Estes requisitos resultam da análise das funcionalidades desenvolvidas, das regras transmitidas pela equipa técnica e das necessidades operacionais identificadas ao longo do estágio.

No corpo principal do relatório, os requisitos são apresentados de forma agrupada

na Tabela 5.1, de modo a evitar uma listagem excessivamente extensa nesta fase da leitura. A sistematização detalhada encontra-se na Tabela B.1, apresentada no Apêndice B.

Durante o desenvolvimento, não foi utilizada uma classificação formal dos requisitos segundo níveis de prioridade, como alta, média ou baixa. As prioridades eram definidas de forma iterativa pela equipa, considerando as dependências entre funcionalidades, as necessidades operacionais e a urgência associada à disponibilização do sistema. Numa primeira fase, foram privilegiadas as funcionalidades essenciais ao funcionamento do processo, seguindo-se o refinamento das regras de negócio e o desenvolvimento das integrações com serviços externos. Os requisitos de segurança e integridade foram considerados transversais às diferentes fases do desenvolvimento.

Tabela 5.1: *Síntese dos principais tipos de requisitos do projeto ValSabor*

Tipo de requisito	**Exemplos no projeto ValSabor**
Funcionais	Leitura de códigos GS1, criação de caixas, agrupamento de caixas, fecho de paletes intermédias, montagem de unidades logísticas, geração de etiquetas em ZPL e integração com serviços externos associados ao SAP.
Operacionais	Utilização de leitores de códigos de barras, integração com balanças industriais, adequação da interface ao contexto industrial e suporte a fluxos rápidos de operação.
Segurança	Controlo de acessos, validação de autenticação e autorização em operações sensíveis, e proteção de *endpoints* críticos.
Integridade	Validação no *backend*, prevenção de duplicados, controlo de estados, coerência entre dados registados e operações físicas, e manutenção de histórico das operações.
Não funcionais	Validação em ambiente de *staging*, tratamento adequado de erros e redução do risco de impacto em ambiente produtivo.

Esta organização evidencia que o projeto ValSabor não se limitou à implementação de funcionalidades isoladas. O sistema precisava de responder a problemas concretos de identificação, validação, integração e rastreabilidade, num contexto em que os dados digitais estavam diretamente ligados a operações físicas. Por esse motivo, os requisitos funcionais, operacionais, de segurança, integridade e validação foram tratados de forma interdependente ao longo do desenvolvimento.

A leitura de códigos e a utilização de equipamentos industriais permitiram aproximar a aplicação dos processos executados no terreno. Por outro lado, a validação no *backend*, o controlo de acessos, a prevenção de duplicados e a validação em ambiente de *staging* contribuíram para aumentar a fiabilidade das operações e reduzir o risco de inconsistências antes da disponibilização das funcionalidades em produção.

5.3 Planeamento e Organização

O planeamento e a organização do trabalho foram assegurados através de uma abordagem estruturada, baseada em práticas de desenvolvimento ágil. As atividades foram organizadas em ciclos de trabalho com uma duração aproximada de quinze dias, durante os quais as tarefas a realizar eram definidas e acompanhadas através de uma ferramenta de gestão de projetos. Todas as tarefas eram registadas num *backlog*, permitindo uma visão global do trabalho em curso e das prioridades definidas para cada período.

O planeamento das tarefas foi realizado pelo elemento da equipa com funções de coordenação técnica, em articulação com o responsável da área de informática. Em alguns momentos, nomeadamente quando foi necessário esclarecer requisitos ou ajustar prioridades, foram realizadas reuniões adicionais com o objetivo de discutir as necessidades do projeto e criar ou ajustar tarefas de forma colaborativa. Esta flexibilidade permitia adaptar o planeamento às exigências reais do desenvolvimento, sem comprometer a organização do trabalho.

As tarefas atribuídas apresentavam naturezas distintas, variando entre atividades previamente planeadas e situações que exigiam intervenção imediata. Sempre que surgiam correções urgentes, nomeadamente problemas detetados em ambiente de produção, estas assumiam prioridade máxima, sendo tratadas de acordo com a sua criticidade. As restantes tarefas foram retomadas após a resolução das situações urgentes, garantindo que a estabilidade do sistema era mantida.

O trabalho foi desenvolvido de forma focada em projetos específicos, não existindo a necessidade de gerir múltiplos projetos em simultâneo. Esta abordagem permitiu uma melhor concentração nas tarefas atribuídas e facilitou a estimativa do tempo necessário para a sua execução. As estimativas foram realizadas com base na perceção individual da complexidade das tarefas, considerando não apenas a implementação, mas também a fase de testes e validação das funcionalidades desenvolvidas.

Com o aumento da complexidade dos projetos e da responsabilidade associada às tarefas atribuídas, tornou-se necessário adotar uma organização mais rigorosa do trabalho. Neste contexto, passou a ser dada maior importância à definição clara dos passos de implementação e dos testes necessários, de forma a reduzir a probabilidade de erros e facilitar eventuais correções. A necessidade de responder rapidamente a falhas ou ajustes reforçou a importância de um planeamento adequado, uma vez que intervenções apressadas tendem a comprometer a qualidade do código desenvolvido.

O acompanhamento do trabalho foi realizado através de reuniões diárias, durante as quais foram partilhadas as atividades realizadas no dia anterior e discutidos os objetivos para o período seguinte. Para além destas reuniões, foram efetuadas sessões de validação e testes mais intensivos quando os elementos da equipa se encontravam presencialmente, permitindo testar funcionalidades em conjunto, identificar problemas e validar soluções de forma colaborativa.

A adoção deste modelo de planeamento e organização teve um impacto direto na

qualidade do trabalho desenvolvido. Um planeamento mais estruturado contribuiu para a redução de erros, para uma melhor compreensão dos requisitos e para uma execução mais eficiente das tarefas. À medida que a experiência aumentava, tornou-se mais evidente a relação entre a definição clara de objetivos, a estimativa adequada do tempo necessário e a qualidade final das soluções implementadas.

De forma global, o processo de planeamento evidenciou a importância da organização e da comunicação em ambiente profissional. A definição clara das tarefas, aliada a uma transmissão eficaz das ideias e dos objetivos pretendidos, revelou-se determinante para garantir que os resultados obtidos correspondiam ao que havia sido inicialmente idealizado.

De forma geral, o modelo de organização adotado demonstrou que, em contextos reais de desenvolvimento, a qualidade das soluções não depende exclusivamente da competência técnica, mas também da capacidade de estruturar o trabalho, gerir prioridades e responder a eventos imprevistos. Esta constatação encontra-se alinhada com a literatura analisada, que aponta a integração entre processos de desenvolvimento, validação e controlo como fator determinante para a robustez dos sistemas empresariais.

5.4 Desenvolvimento Full Stack

O desenvolvimento realizado incidiu tanto na manutenção e evolução de código existente como na implementação de novas funcionalidades, num sistema interno que se encontrava em contínua evolução. Este sistema, desenvolvido com recurso a Laravel no *backend* e Vue.js no *frontend*, tinha sido alvo de uma migração tecnológica recente, o que implicava a coexistência entre componentes legados e novas implementações, exigindo especial atenção à coerência da arquitetura e à integração entre módulos.

Numa fase inicial, o trabalho incidiu maioritariamente em correções pontuais, pequenas melhorias funcionais e adaptação de páginas existentes, permitindo uma compreensão progressiva da arquitetura do sistema e dos padrões de desenvolvimento adotados pela equipa. Posteriormente, foram atribuídos desenvolvimentos mais significativos, incluindo a criação de novas páginas, reestruturação de tabelas de base de dados e migração de funcionalidades previamente implementadas noutras tecnologias para a *stack* atual.

5.4.1 *Backend*

No desenvolvimento *backend*, o trabalho envolveu a implementação de novas funcionalidades de acordo com os requisitos funcionais definidos, bem como a adaptação de módulos existentes. Sempre que era necessário criar um novo módulo, procedia-se à definição das respetivas tabelas, modelos, controladores e *endpoints* de APIs, garantindo o cumprimento das regras de negócio e a integridade dos dados.

A implementação seguiu uma estrutura base definida pela equipa, recorrendo a controladores e modelos base reutilizáveis que facilitavam a criação de funcionalida-

des comuns, como listagens, filtros, criação e edição de dados. Esta abordagem permitia reduzir redundâncias e assegurar consistência entre os diferentes módulos do sistema. Foram também utilizados recursos (*resources*) para controlar a forma como os dados eram expostos pelas APIs, variando entre estruturas simples e estruturas mais complexas quando envolviam relações entre entidades.

A validação dos dados foi assegurada através de ficheiros de *requests*, garantindo que os dados recebidos respeitavam os formatos e restrições definidos. O controlo de acessos e permissões foi integrado no desenvolvimento dos módulos, recorrendo a mecanismos já existentes no sistema, assegurando que apenas utilizadores autorizados podiam aceder ou modificar determinados recursos.

A abordagem adotada no *backend* permitiu concentrar a lógica crítica da aplicação numa camada mais controlada e previsível, reduzindo a dependência de validações realizadas no cliente. Esta decisão encontra-se alinhada com o enquadramento teórico apresentado no Capítulo 4, onde se evidenciou que a centralização da validação e do controlo de acessos no *backend* constitui uma prática fundamental para assegurar a integridade dos dados e reduzir a exposição a comportamentos indevidos (OWASP Foundation, 2025).

5.4.2 *Frontend*

No *frontend*, o trabalho iniciou-se com a adaptação de páginas existentes e a implementação de pequenas funcionalidades, como exportação de dados, correção de erros e melhoria da experiência de utilização. Com o aumento da complexidade das tarefas atribuídas, passou a ser desenvolvido um maior número de páginas novas, recorrendo a uma estrutura modular previamente definida pela equipa.

Cada módulo *frontend* seguia uma organização consistente, composta por componentes distintos para listagem, filtros, edição, visualização de detalhes e configuração. Esta estrutura, aliada à utilização de *mixins* e serviços genéricos para comunicação com o *backend*, permitiu acelerar o desenvolvimento e manter um comportamento uniforme entre páginas. A validação dos formulários foi realizada de forma sistemática, recorrendo a bibliotecas específicas, assegurando a coerência e fiabilidade dos dados introduzidos.

Devido à maior experiência prévia no desenvolvimento *backend* e à maior familiaridade com a implementação de lógica de negócio, modelos e controladores, o desenvolvimento *frontend* revelou-se particularmente desafiante em páginas com formulários extensos e altamente dinâmicos. A gestão de estados, a visibilidade condicional de componentes e a implementação de validações complexas exigiram uma abordagem cuidadosa, tendo a utilização de componentes reutilizáveis e *mixins* desempenhado um papel essencial na superação destas dificuldades.

Embora o *frontend* tenha desempenhado um papel importante ao nível da usabilidade e da eficiência operacional, a sua implementação foi sempre pensada como complemento da lógica de validação e controlo centralizada no *backend*. Esta separação de

responsabilidades revelou-se essencial para manter a coerência da aplicação e está em linha com as boas práticas discutidas no Estado da Arte, segundo as quais o *frontend* não deve ser encarado como uma camada de confiança do ponto de vista da segurança e integridade da informação (OWASP Foundation, 2025).

5.4.3 Projetos Desenvolvidos

Entre os projetos desenvolvidos, destaca-se a reestruturação da área de parâmetros do sistema, que consistiu na migração de uma funcionalidade anteriormente implementada noutra tecnologia para uma nova página desenvolvida em Vue.js. Este trabalho envolveu a criação de uma nova estrutura de base de dados, a migração de dados existentes e a implementação de uma interface mais flexível, permitindo a gestão centralizada de parâmetros utilizados por diferentes módulos do sistema.

Outro projeto relevante foi o desenvolvimento do módulo de gestão de viaturas da empresa. O objetivo principal consistiu na criação de uma página centralizada que permitisse consultar e gerir as diferentes viaturas, incluindo as suas características específicas e documentação associada. Este projeto colocou desafios específicos ao nível de complexidade dos formulários, uma vez que cada tipo de viatura apresentava características distintas, exigindo uma interface altamente dinâmica e cuidadosamente validada.

Entre os projetos desenvolvidos, o módulo associado ao projeto ValSabor assumiu particular relevância, não só pela sua complexidade técnica, mas também pelo impacto direto nos processos operacionais da organização. Responsável pela gestão de processos relacionados com produção, congelados e montagem de unidades de transporte, este módulo envolveu validações complexas, comunicação com equipamentos industriais e integração com sistemas externos. O projeto foi inicialmente desenvolvido com um prazo reduzido para entrada em produção, sendo posteriormente alvo de melhorias e reestruturações à medida que os requisitos se tornaram mais claros.

Este módulo encontra-se dividido em duas áreas funcionais principais. A primeira é dedicada ao processamento de produtos congelados, permitindo a criação e identificação de unidades a partir de artigos frescos, com suporte a leitura de códigos de barras e mecanismos de validação associados. Este processo assegura a rastreabilidade dos produtos ao longo das diferentes etapas.

A segunda área centra-se na criação e gestão de unidades logísticas finais, como paletes ou grades, permitindo a associação de diferentes elementos de acordo com regras de negócio específicas. O sistema inclui mecanismos que garantem a consistência dos dados e a correta sequência das operações ao longo do processo.

A implementação deste módulo permitiu automatizar processos operacionais anteriormente realizados de forma manual, contribuindo para uma maior rastreabilidade das unidades logísticas, melhoria da eficiência dos processos e redução do risco de erro humano durante as operações de produção e expedição.

Entre os diferentes projetos, o módulo ValSabor destacou-se por concentrar, num

único sistema, vários dos desafios identificados na literatura: integração com sistemas externos, necessidade de validação robusta, articulação com dispositivos físicos e exigência de rastreabilidade contínua (OWASP Foundation, 2025; McCarthy et al., 2022; International Organization for Standardization, 2007; Zhou et al., 2022).

5.4.4 Integrações e Tecnologias

O desenvolvimento do módulo ValSabor, integrado no sistema interno da empresa, implicou a comunicação com diversos equipamentos industriais e sistemas externos utilizados nos processos operacionais da empresa.

Estas integrações permitiram automatizar operações críticas associadas à identificação de produtos, pesagem, geração de etiquetas e comunicação com sistemas de gestão empresarial, contribuindo para a fiabilidade dos processos e redução de erros humanos.

Entre as principais integrações, destacam-se a comunicação com equipamentos industriais, o desenvolvimento de etiquetas utilizando a linguagem Zebra Programming Language (ZPL) e a integração com sistemas ERP baseados em SAP.

Integração com Equipamentos Industriais

Uma das componentes centrais do sistema desenvolvido consiste na integração com equipamentos industriais utilizados nas operações de produção e logística, nomeadamente leitores de códigos de barras e balanças eletrónicas.

Os leitores de códigos de barras são utilizados como dispositivos de entrada, permitindo a captura automática de informação no sistema através da leitura de códigos na norma GS1.

Esta norma permite codificar múltiplos tipos de informação num único código de barras, incluindo identificadores de produto, lotes de produção e unidades logísticas. No contexto da aplicação foram utilizados diferentes tipos de códigos, entre os quais:

- Códigos GS1 contendo identificação de artigo e lote para leitura de artigos frescos;
- Códigos Serial Shipping Container Code (SSCC) utilizados na identificação de caixas, paletes ou grades;
- Códigos de agrupamento gerados internamente pela aplicação para representar conjuntos de caixas durante processos de montagem de unidades logísticas.

Após a leitura de um código de barras, os dados são capturados pelo *frontend* da aplicação e enviados para o *backend* através de pedidos à API interna. O *backend* procede então à validação do artigo, lote ou unidade logística correspondente, aplicando as regras de negócio definidas para o processo em execução e apresentando ao utilizador a informação validada.

Para além da leitura de códigos de barras, o sistema integra-se com balanças industriais, permitindo a receção automática de dados de peso e a sua disponibilização

em tempo real na aplicação. Estes mecanismos contribuem para a automatização dos processos e para a redução de erros associados à introdução manual de dados.

Esta integração evidenciou a importância de assegurar uma correspondência fiável entre os eventos físicos ocorridos no terreno e o seu reflexo no sistema de informação. Tal como discutido no Estado da Arte, a ligação entre sistemas digitais e operações industriais exige mecanismos robustos de validação e controlo, uma vez que pequenas discrepâncias podem propagar-se ao longo do processo e comprometer a consistência global da informação (McCarthy et al., 2022; International Organization for Standardization, 2007; Zhou et al., 2022).

Desenvolvimento de Etiquetas ZPL

Outro elemento fundamental do sistema consiste na geração automática de etiquetas utilizadas na identificação de produtos e unidades logísticas ao longo do processo produtivo.

As etiquetas são impressas em impressoras industriais Zebra ligadas em rede à infraestrutura da empresa. Cada utilizador pode ter uma impressora associada ao seu posto de trabalho, sendo as etiquetas enviadas automaticamente para a impressora configurada no sistema quando um processo é concluído.

A geração das etiquetas é realizada através da linguagem ZPL, uma linguagem específica utilizada por impressoras Zebra para definição de layouts de impressão. Os modelos de etiquetas foram definidos com recurso a ferramentas de apoio e posteriormente integrados no sistema através de *templates* dinâmicos em ZPL, permitindo a geração automática de etiquetas adaptadas ao contexto operacional. De forma a tornar o sistema mais flexível e reutilizável, foi adotada uma abordagem baseada em *templates* dinâmicos. Cada modelo de etiqueta contém variáveis definidas sob a forma de *placeholders*, como por exemplo o da Listagem 1:

```
1  {{ PRODUCT_CODE }}
2  {{ NET_WEIGHT }}
3  {{ BATCH_CODE }}
4  {{ PRODUCTION_DATE }}
5  {{ LABEL_NAME }}
```

Listagem 1: *Exemplo variáveis dinâmicas em ZPL*

Durante o processo de geração da etiqueta, o *backend* substitui estes *placeholders* pelos valores reais associados ao processo em execução. Esta abordagem permitiu criar um sistema genérico capaz de suportar múltiplos tipos de etiquetas sem necessidade de duplicação de código.

Foram desenvolvidos diversos tipos de etiquetas, entre os quais:

- Etiquetas de identificação de produto;
- Etiquetas associadas à criação de caixas;

- Etiquetas de agrupamento de caixas;
- Etiquetas de paletes intermédias;
- Etiquetas finais de paletes e grades.

Algumas destas etiquetas são utilizadas exclusivamente para processos internos de logística e controlo operacional, enquanto outras acompanham os produtos ao longo da cadeia de distribuição.

A utilização de *templates* dinâmicos permitiu ainda adaptar facilmente o formato das etiquetas a diferentes contextos operacionais, incluindo variações associadas à empresa responsável pelo processo ou ao destino final dos produtos.

A adoção de *templates* dinâmicos em ZPL permitiu responder simultaneamente a requisitos de flexibilidade operacional e de manutenção do sistema, reduzindo duplicações e facilitando a adaptação a novos cenários. Esta abordagem demonstra como decisões aparentemente localizadas ao nível da implementação podem ter impacto direto na escalabilidade e robustez da solução.

Integração com Sistemas ERP

Para além das integrações com equipamentos industriais, o sistema desenvolvido integra-se também com o sistema de gestão empresarial SAP, utilizado pelas empresas do grupo para registo de operações logísticas e movimentações de *stock*.

A comunicação com SAP é realizada através de APIs disponibilizadas por serviços de integração externos, nomeadamente desenvolvidas pelas empresas FutureView e ActonIT, sendo a escolha do serviço dependente da empresa associada ao processo em execução.

O fluxo geral de comunicação entre o Portal e os serviços externos de integração encontra-se representado na Figura F.1, apresentada no Apêndice F. A comunicação entre estes serviços e os sistemas empresariais externos encontra-se representada de forma abstrata, uma vez que não foi analisada.

As integrações são iniciadas a partir da aplicação desenvolvida em Laravel, que prepara os dados necessários e envia pedidos para os respetivos *endpoints* de integração. Os *payloads* enviados variam consoante o tipo de operação, podendo incluir informação como o *token* de autenticação, identificação de veículos de transporte, armazéns de origem e destino, bem como listas de linhas contendo artigos e lotes associados ao processo.

Entre os principais tipos de integração implementados destacam-se:

- Registo de produções associadas à criação de agrupamentos de caixas na área de congelados;
- Transferências internas de paletes entre diferentes salas;
- Transferências de *stock* entre armazéns distintos;
- Operações de entrega direta associadas à venda de requisições de paletes.

As integrações com os serviços externos associados ao sistema SAP foram implementadas através de uma abordagem modular, garantindo a separação de responsabilidades e facilitando a validação e construção dos dados enviados para os serviços externos.

A comunicação com os serviços de integração utiliza mecanismos de autenticação baseados em *tokens* temporários, contribuindo para a proteção das operações realizadas.

Após cada integração, a resposta devolvida pelo serviço externo é armazenada na base de dados da aplicação. Esta informação inclui o número do documento criado no sistema ERP, bem como as linhas afetadas pela operação, permitindo assegurar rastreabilidade das operações realizadas.

Em caso de erro durante a comunicação com o serviço externo, o sistema regista automaticamente informação através de mecanismos de Logging, que permitem documentar os eventos e erros ocorridos durante a execução da operação. Estes registos incluem os identificadores dos elementos envolvidos e o pedido que originou o erro, permitindo apoiar o diagnóstico de problemas e a monitorização das integrações entre sistemas.

Esta componente revelou, de forma particularmente clara, os desafios discutidos no Estado da Arte relativamente à integração entre sistemas distribuídos, nomeadamente ao nível da consistência dos dados, validação prévia das operações e prevenção de estados inconsistentes.

5.5 Integração de Segurança

A integração de mecanismos de segurança ao longo do ciclo de desenvolvimento constitui uma prática essencial para a redução de vulnerabilidades estruturais em aplicações *web*. A ausência de validações sistemáticas e controlos de segurança durante o desenvolvimento pode aumentar o risco de introdução de vulnerabilidades no *software* (Souppaya et al., 2022; OWASP Foundation, 2025). Esta abordagem encontra-se igualmente alinhada com o *Secure Software Development Framework*, que recomenda a incorporação de práticas de segurança ao longo de todo o ciclo de vida do *software* (Souppaya et al., 2022).

Em sistemas que suportam processos operacionais críticos, como é o caso da aplicação desenvolvida durante o estágio, a proteção da informação e o controlo adequado de acessos assumem particular relevância. No desenvolvimento do sistema, a segurança não foi tratada como um processo isolado, mas sim integrada de forma transversal nas diferentes fases de implementação das funcionalidades.

Ao longo do desenvolvimento, foram considerados mecanismos de controlo de acessos, validação de dados e proteção das APIs, integrando a segurança de forma transversal nas diferentes funcionalidades do sistema.

O sistema implementa controlo de acessos baseado em permissões, recorrendo ao pacote Spatie Laravel Permission, permitindo uma gestão granular de papéis e acessos.

Esta abordagem revelou-se fundamental para garantir a segregação de responsabilidades e a proteção de funcionalidades sensíveis.

A gestão de permissões encontra-se integrada no sistema de administração da aplicação, sendo possível atribuir a cada utilizador diferentes grupos de permissões através de uma interface específica de gestão de utilizadores. Estas permissões determinam tanto o acesso às páginas do *frontend* como aos respetivos *endpoints* da API, garantindo que apenas utilizadores autorizados podem executar determinadas operações.

Para além do controlo de acessos, foram implementados mecanismos de validação de dados que garantem a integridade da informação processada, reduzindo o risco de inconsistências e comportamentos inesperados.

Adicionalmente, foram implementados *middlewares* de permissões, responsáveis por verificar se o utilizador autenticado possui autorização para aceder a determinados recursos. Estes controlos permitem, por exemplo, restringir o acesso a dados associados a empresas específicas, garantindo que os utilizadores apenas conseguem visualizar ou manipular informação relacionada com as entidades às quais estão associados.

A autenticação dos utilizadores é realizada através do mecanismo padrão do *framework* Laravel, baseado em sessões autenticadas. Após o processo de autenticação através de credenciais válidas, o utilizador recebe uma sessão ativa que é utilizada para autenticar os pedidos subsequentes realizados à aplicação. As APIs internas verificam a validade desta sessão antes de permitir a execução de qualquer operação.

As APIs internas são ainda protegidas através de múltiplos mecanismos complementares, incluindo validação dos pedidos recebidos, verificação de permissões associadas ao utilizador e controlo das operações permitidas para cada tipo de recurso. Esta abordagem permite reduzir significativamente o risco de acesso indevido a funcionalidades sensíveis do sistema.

No contexto específico das integrações externas, como a comunicação com os serviços de integração associados ao sistema ERP SAP, foram igualmente implementados mecanismos adicionais de validação e registo de erros. Antes do envio de qualquer pedido para os serviços externos, são realizadas validações internas que verificam a consistência dos dados que irão compor o *payload*. Caso seja detetada alguma inconsistência, o processo é interrompido e é registada informação detalhada através de mecanismos de Logging, permitindo identificar a origem do problema.

Adicionalmente, sempre que ocorre um erro durante o processo de integração, a aplicação adapta a resposta devolvida ao utilizador, evitando a exposição direta de informação técnica sensível no *frontend*. Desta forma, os utilizadores recebem mensagens genéricas de erro, enquanto a informação detalhada é registada internamente para efeitos de diagnóstico técnico.

Outro aspeto relevante prende-se com a validação de dados provenientes de equipamentos industriais, nomeadamente leitores de códigos de barras utilizados nos processos operacionais da aplicação. Estes códigos seguem a norma GS1, sendo aplicadas validações específicas associadas à estrutura dos códigos lidos. A aplicação verifica, por exemplo, o comprimento dos códigos e os identificadores associados a diferentes

tipos de informação, como artigos, lotes ou unidades logísticas, rejeitando automaticamente códigos que não respeitem o formato esperado.

De forma global, estas práticas contribuíram para a implementação de um sistema mais robusto e resiliente, reduzindo a probabilidade de ocorrência de erros ou comportamentos inesperados em ambiente produtivo. A adoção de mecanismos de validação sistemática, controlo de acessos e tratamento adequado de erros constitui um conjunto de boas práticas amplamente recomendado na literatura para o desenvolvimento seguro de aplicações *web* (OWASP Foundation, 2025; Souppaya et al., 2022).

A experiência obtida ao longo do estágio demonstrou que a segurança é mais eficaz quando incorporada de forma transversal no desenvolvimento, em vez de ser tratada como uma preocupação posterior. Esta constatação reforça o enquadramento apresentado no Estado da Arte, segundo o qual a integração de mecanismos de validação, controlo de acessos e tratamento adequado de erros constitui um elemento essencial para a construção de aplicações empresariais mais robustas (Souppaya et al., 2022; OWASP Foundation, 2025).

No contexto do estágio, estas medidas foram complementadas pela identificação observacional de uma vulnerabilidade relevante num serviço *web* associado ao grupo empresarial, cuja análise é apresentada na secção seguinte.

5.5.1 Identificação Observacional de Vulnerabilidade em Aplicação *Web*

No âmbito das atividades desenvolvidas, e em complemento às tarefas associadas ao desenvolvimento do portal interno, foi identificada uma situação relevante relacionada com a exposição indevida de informação técnica num serviço *web* pertencente ao grupo empresarial. Esta identificação ocorreu no contexto de uma abordagem observacional e não intrusiva, alinhada com práticas éticas de cibersegurança.

A situação foi detetada de forma acidental durante uma navegação funcional normal a um dos *websites* públicos do grupo. Ao aceder à secção *Produtos*, funcionalidade disponibilizada aos utilizadores finais, foi observado um redirecionamento automático para um endereço IP distinto do domínio original, associado a um serviço *backend*. Este comportamento revelou-se inesperado do ponto de vista funcional e motivou uma análise mais cuidada do serviço exposto.

Após o redirecionamento, a página apresentou uma mensagem de erro visível no browser, indicando uma falha na ligação à base de dados. A mensagem incluía informação técnica detalhada, nomeadamente um Stack trace completo da aplicação, permitindo inferir diversos aspetos da arquitetura interna do sistema. Através da análise do erro apresentado, foi possível identificar que o serviço correspondia a uma instância da plataforma *Magento*, configurada em modo de desenvolvimento, bem como a utilização de uma base de dados *MySQL* inacessível ou indisponível no momento do acesso.

Com o objetivo de validar tecnicamente o serviço exposto e obter uma visão geral da sua superfície de ataque, foi realizada uma enumeração básica e controlada recorrendo

à ferramenta *nmap*, utilizando a opção `-sV`, de forma a identificar os serviços ativos e as respetivas versões. Esta validação teve um caráter estritamente não intrusivo, não envolvendo qualquer tentativa de autenticação, exploração de vulnerabilidades ou interação com os mecanismos internos da aplicação.

Adicionalmente, foram efetuados pedidos HTTP simples a diferentes *endpoints* da aplicação, recorrendo à ferramenta *curl*, com o objetivo de observar o comportamento global do serviço. Os resultados obtidos demonstraram que o erro apresentado não se encontrava limitado a um único ponto de acesso, sendo reproduzível em múltiplos caminhos da aplicação, incluindo a página principal, áreas administrativas e diretórios comuns da plataforma. Em todos os casos, as respostas do servidor incluíam mensagens de erro detalhadas e *stack traces*, evidenciando a inexistência de mecanismos adequados de gestão de erros e de isolamento entre ambientes.

A informação recolhida ao longo desta análise confirmou a exposição sistemática de informação técnica sensível a partir de um contexto acessível ao público, sem necessidade de autenticação ou interação avançada. Esta situação evidencia uma falha de configuração relevante, relacionada com a gestão de ambientes, a proteção de serviços *backend* e a correta implementação de mecanismos de tratamento de erros em aplicações *web*. Este tipo de exposição enquadra-se em falhas de configuração amplamente documentadas na literatura, estando frequentemente associadas à ausência de mecanismos adequados de tratamento de erros e segregação de ambientes (OWASP Foundation, 2021).

Face à natureza da informação exposta e ao contexto empresarial em que o sistema se insere, não foram realizadas ações adicionais de exploração, de forma a respeitar os limites definidos para o estágio, preservar a integridade dos sistemas envolvidos e cumprir princípios éticos fundamentais da prática de cibersegurança. Ainda assim, a situação identificada constituiu um exemplo prático e relevante da importância da segurança por configuração, da segregação adequada de serviços e da validação rigorosa de integrações entre sistemas antes da sua disponibilização em ambientes públicos.

A situação identificada pode ser enquadrada em categorias de risco amplamente documentadas pelo OWASP Top 10:2021 (OWASP Foundation, 2021), em particular no que diz respeito a falhas de configuração e exposição indevida de informação sensível, sendo esta análise aprofundada no Capítulo 8.

5.6 Testes e Validação

A realização de testes e processos de validação constitui uma etapa essencial no desenvolvimento de sistemas de informação, permitindo verificar o correto funcionamento das funcionalidades implementadas, identificar erros e garantir que as soluções desenvolvidas cumprem os requisitos definidos.

No contexto do estágio, os testes realizados tiveram essencialmente um caráter funcional, sendo executados de forma contínua ao longo do processo de desenvolvimento. Sempre que uma nova funcionalidade era implementada ou uma alteração significa-

tiva era introduzida no sistema, realizavam-se testes locais com o objetivo de verificar o comportamento da aplicação e validar as regras de negócio associadas.

Estes testes incluíam a verificação de diferentes cenários de utilização, nomeadamente a introdução de dados válidos e inválidos, a execução de operações com diferentes perfis de utilizador e a validação das respostas devolvidas pela API. Este processo permitiu identificar erros de implementação, inconsistências na lógica de negócio e problemas relacionados com a interface de utilizador.

Para além dos testes realizados em ambiente local, as funcionalidades desenvolvidas foram posteriormente disponibilizadas em ambientes de *staging*, que replicam de forma controlada as condições do ambiente de produção. Esta etapa permitiu realizar uma validação mais abrangente das funcionalidades, incluindo a verificação da integração com sistemas externos e equipamentos industriais.

Em determinados momentos, especialmente quando a equipa se encontrava presencialmente nas instalações da empresa, foram realizadas sessões de teste colaborativas. Estas sessões permitiam testar funcionalidades em conjunto, simular diferentes cenários de utilização e validar o comportamento do sistema em condições próximas das utilizadas pelos operadores em ambiente industrial.

A integração com equipamentos industriais, como leitores de códigos de barras e balanças eletrónicas, exigiu também a realização de testes específicos em ambiente real. Estes testes permitiram validar a correta captura dos dados provenientes dos dispositivos, bem como a sua integração com os processos implementados na aplicação.

Sempre que se identificavam erros ou comportamentos inesperados, procedia-se à análise das causas e à implementação das respetivas correções. Em muitos casos, estas correções envolviam ajustes na validação de dados, melhorias na gestão de estados da aplicação ou alterações na lógica de negócio implementada no *backend*.

De forma global, o processo de testes e validação revelou-se essencial para garantir a estabilidade e fiabilidade das funcionalidades desenvolvidas, permitindo identificar falhas de implementação, validar regras de negócio e assegurar a adequação das soluções ao contexto operacional da organização.

Em síntese, os testes realizados confirmaram que a validação contínua das funcionalidades é indispensável em sistemas com forte dependência de regras de negócio, integrações externas e interação com dispositivos físicos. Esta prática revelou-se coerente com os desafios identificados no Capítulo 4, nomeadamente no que respeita à necessidade de assegurar consistência, fiabilidade e rastreabilidade em ambientes empresariais e industriais (Souppaya et al., 2022; McCarthy et al., 2022; International Organization for Standardization, 2007).

Para além das atividades diretamente associadas ao desenvolvimento e validação de sistemas internos, o estágio incluiu também iniciativas no domínio da sensibilização para a segurança da informação. Entre estas atividades, destaca-se a realização de uma campanha de *phishing* controlada, concebida para avaliar o comportamento dos utilizadores perante cenários de engenharia social e reforçar a cultura de segurança dentro da organização.

5.7 Campanha de *Phishing* e Sensibilização de Utilizadores

Conforme discutido no Capítulo 4, as campanhas simuladas de *phishing* permitem avaliar o comportamento dos utilizadores perante comunicações fraudulentas e identificar necessidades de sensibilização. Neste contexto, durante o estágio foi realizada uma campanha de *phishing* controlada com o objetivo de avaliar o nível de sensibilização dos utilizadores para ataques de engenharia social, bem como identificar comportamentos de risco associados à interação com comunicações fraudulentas. Estas campanhas constituem uma componente relevante das estratégias de cibersegurança, uma vez que permitem promover boas práticas, influenciar comportamentos e reforçar a cultura de segurança nas organizações (European Union Agency for Cybersecurity, 2021; Hansche et al., 2024).

A campanha foi conduzida recorrendo à ferramenta *GoPhish*, a qual permitiu a gestão centralizada do envio de emails, criação de páginas simuladas e recolha de métricas associadas ao comportamento dos utilizadores.

5.7.1 Planeamento da Campanha

A campanha foi desenhada com base num cenário realista, simulando um processo interno da organização relacionado com aprovação de faturas. Para tal, foi utilizado um *template* de *email* semelhante aos utilizados internamente, incluindo linguagem, estrutura e elementos visuais consistentes com comunicações legítimas.

Adicionalmente, foi registado um domínio com um nome visualmente semelhante ao portal interno da organização, com o objetivo de reforçar a credibilidade do cenário. Foi também desenvolvida uma página de autenticação simulada, replicando de forma controlada a interface do portal interno, permitindo avaliar o comportamento dos utilizadores perante um cenário altamente realista.

A seleção dos utilizadores alvo incidiu sobre aproximadamente 250 colaboradores, com especial foco em perfis com maior nível de responsabilidade dentro da organização, tendo em consideração o potencial impacto deste tipo de ataque em contexto real.

Durante o planeamento, foram considerados aspetos éticos e de privacidade, garantindo que os dados recolhidos seriam utilizados exclusivamente para análise interna e sensibilização, sendo tratados de forma agregada e anonimizada.

No contexto da campanha, foi elaborado um *email* com características semelhantes às comunicações internas da organização. Um exemplo da mensagem utilizada encontra-se apresentado na Figura G.1, no Apêndice G. As imagens foram selecionadas e, quando necessário, ajustadas de forma a evitar a exposição de informação sensível da organização.

5.7.2 Execução da Campanha

A campanha foi executada através do envio de emails simulados contendo uma ligação para a página de autenticação controlada. Toda a infraestrutura foi configurada de forma a garantir que não existia qualquer risco para os utilizadores, não sendo efetuada qualquer integração com sistemas reais da organização.

A plataforma *GoPhish* permitiu monitorizar em tempo real o comportamento dos utilizadores, incluindo abertura de emails, cliques em ligações e submissão de dados na página simulada.

A página de autenticação simulada foi desenvolvida com base na interface do portal interno. Um exemplo desta página encontra-se apresentado na Figura G.2, no Apêndice G.

5.7.3 Resultados Obtidos

A campanha permitiu recolher métricas relevantes sobre o comportamento dos utilizadores, destacando-se os seguintes resultados:

- Aproximadamente 70 utilizadores interagiram com o *email* e acederam à ligação disponibilizada;
- Cerca de 20 utilizadores interagiram com a página de autenticação simulada, submetendo informação no formulário disponibilizado.

Considerando os aproximadamente 250 destinatários, cerca de 28% acederam à ligação disponibilizada e aproximadamente 8% submeteram informação através da página simulada. A taxa de interação com a ligação foi superior à mediana de 16,7% observada no estudo realizado em instituições de saúde e à taxa de 10,9% registada na primeira campanha realizada numa organização de infraestrutura crítica (Gordon et al., 2019; Sirawongphatsara et al., 2024).

Esta comparação deve ser interpretada com prudência, uma vez que a campanha realizada utilizou uma mensagem, um domínio e uma interface adaptados ao contexto da organização. Os resultados das campanhas simuladas dependem das características dos participantes, do contexto organizacional e da dificuldade do cenário utilizado. Adicionalmente, por ter sido realizada apenas uma campanha, não foi possível avaliar a evolução do comportamento dos utilizadores ao longo do tempo.

Estes resultados evidenciam a eficácia do cenário construído e demonstram que, mesmo em ambientes organizacionais, o fator humano continua a representar uma superfície de ataque significativa.

A taxa de interação observada reforça a necessidade de implementar continuamente medidas de sensibilização e formação, especialmente junto de perfis com maior responsabilidade e acesso a informação crítica.

5.7.4 Medidas de Mitigação e Sensibilização

Com base nos resultados obtidos, foram propostas ações de sensibilização direcionadas aos utilizadores, com foco na identificação de tentativas de *phishing*, verificação de remetentes e análise crítica de ligações recebidas por *email*.

Adicionalmente, foi recomendada a realização de campanhas periódicas de *phishing* controlado, permitindo acompanhar a evolução do comportamento dos utilizadores e reforçar a cultura de segurança dentro da organização.

A campanha foi conduzida de forma controlada, garantindo que não houve recolha de credenciais reais nem integração com sistemas produtivos. Os dados recolhidos foram utilizados exclusivamente para análise interna, sendo posteriormente anonimizados.

Esta abordagem permitiu equilibrar a necessidade de realismo na simulação com a proteção da privacidade dos utilizadores, assegurando que a campanha cumpriu os seus objetivos sem comprometer a confiança dentro da organização.

5.7.5 Síntese

A campanha realizada evidenciou a relevância do fator humano na segurança da informação, demonstrando que ataques de engenharia social continuam a ser eficazes mesmo em ambientes organizacionais.

Esta atividade reforçou a importância da combinação entre medidas técnicas e ações de sensibilização, evidenciando que a segurança não depende apenas da tecnologia, mas também do comportamento dos utilizadores, constituindo este último um dos principais vetores de risco em sistemas organizacionais (European Union Agency for Cybersecurity, 2021; National Institute of Standards and Technology, 2024).

6

Caso de Estudo: Sistema de Gestão de Unidades Logísticas

O presente capítulo apresenta um estudo do sistema desenvolvido durante o estágio, com enfoque nos mecanismos de validação, controlo e integração que contribuem para um desenvolvimento seguro em contexto empresarial. O sistema foi concebido para suportar processos operacionais críticos relacionados com produção, transformação e logística de produtos em ambiente industrial.

São analisadas as principais componentes do sistema, incluindo os fluxos de produção e tratamento de congelados, a montagem de unidades logísticas, a gestão de pedidos operacionais e a integração com sistemas externos, nomeadamente o SAP. É dada especial atenção às regras de negócio e validações implementadas, evidenciando o seu papel na prevenção de inconsistências e na garantia da coerência da informação.

A abordagem segue uma perspetiva de caso de estudo, permitindo descrever o funcionamento do sistema e analisar de que forma as decisões técnicas adotadas contribuíram para a validação dos dados, a integração entre componentes e a fiabilidade das operações.

6.1 Visão Geral do Sistema

O sistema desenvolvido durante o estágio tem como principal objetivo suportar digitalmente os processos operacionais associados à produção, transformação e logística de produtos em contexto industrial. Neste tipo de sistemas, a integração de práticas de desenvolvimento seguro assume um papel relevante ao longo do ciclo de vida da aplicação (Souppaya et al., 2022).

Em particular, destaca-se o módulo de gestão de unidades logísticas, responsável por acompanhar o ciclo completo desde a criação de caixas de produto até à sua consolidação em paletes ou grades e respetiva integração com sistemas externos, nomeadamente sistemas de ERP, como o SAP.

O sistema foi desenvolvido com base numa arquitetura *full stack*, recorrendo a Laravel no *backend* e Vue.js no *frontend*, promovendo uma separação clara entre a lógica de negócio e a interface de utilizador. A comunicação entre estas camadas é realizada através de API, permitindo modularidade, escalabilidade e controlo centralizado das operações.

Para além disso, o sistema integra-se com diversos dispositivos e serviços externos, nomeadamente leitores de códigos de barras baseados na norma GS1, balanças industriais com comunicação via MQTT e serviços externos responsáveis por operações de integração associadas ao sistema ERP SAP, como é mostrado na Figura 6.1.

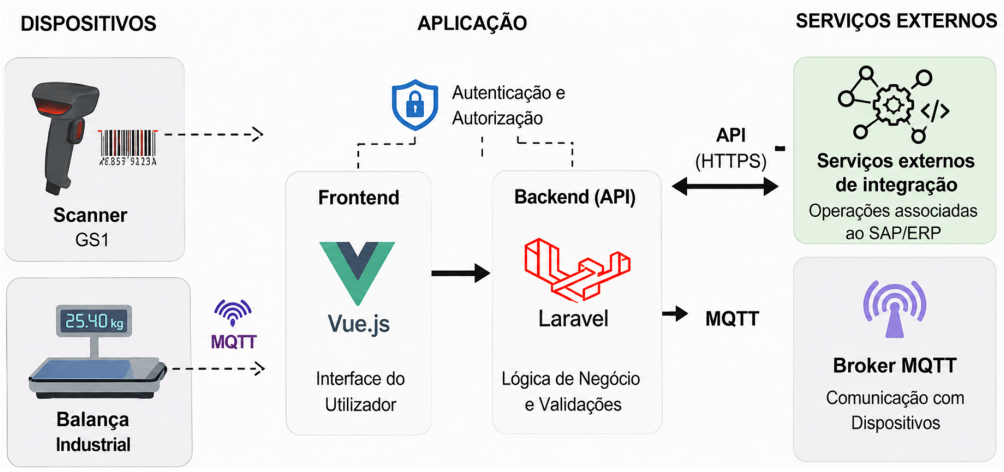

Figura 6.1: *Arquitetura geral do módulo de gestão de unidades logísticas*

O processo operacional suportado pelo sistema encontra-se dividido em três fases principais, que refletem a evolução das unidades logísticas ao longo do ciclo produtivo e logístico.

Numa primeira fase, designada por tratamento de congelados, são criadas caixas de produto com base em artigos frescos, sendo aplicadas validações relacionadas com peso, lote, datas e regras de produção.

Numa segunda fase, essas caixas são agrupadas e consolidadas em unidades logísticas intermédias, sendo posteriormente organizadas em paletes ou grades finais.

Por fim, numa terceira fase, as unidades logísticas são associadas a pedidos operacionais que representam movimentos de entrada, transferência ou saída de mercadoria, podendo posteriormente originar operações de integração associadas ao sistema SAP.

A Figura 6.2 apresenta de forma simplificada este fluxo operacional, evidenciando as principais etapas do processo.

Uma das características centrais do sistema reside na forte dependência de validações, regras de negócio e mecanismos de rastreabilidade, aplicados de forma transversal desde a leitura de códigos GS1 até à integração com sistemas externos. Estes mecanismos contribuem para a consistência dos dados, para a integridade dos processos e para a capacidade de auditoria, assumindo particular relevância em contexto industrial.

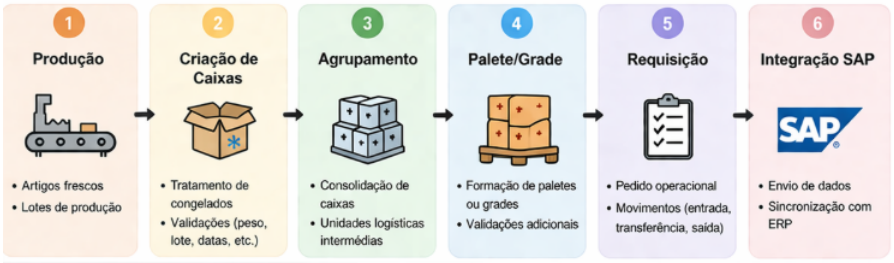

Figura 6.2: *Fluxo operacional do módulo de gestão de unidades logísticas*

Neste contexto, o presente capítulo tem como objetivo analisar detalhadamente o funcionamento do sistema, com especial enfoque nos mecanismos de validação, controlo e integração que contribuem para um desenvolvimento seguro. Para tal, serão descritos os principais fluxos operacionais, as regras de negócio implementadas e as estratégias de integração com sistemas externos, evidenciando as decisões técnicas adotadas ao longo do desenvolvimento.

A Figura 6.3 apresenta o *dashboard* principal do sistema, onde são disponibilizados os principais módulos e funcionalidades acessíveis aos utilizadores.

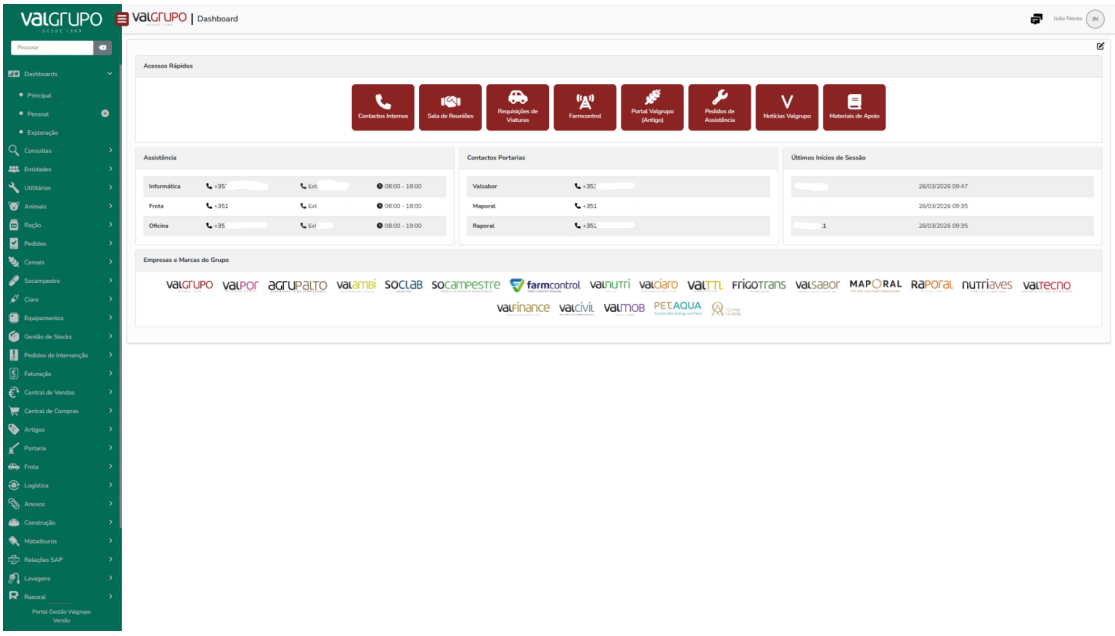

Figura 6.3: *Dashboard principal do Portal*

6.2 Processo de Produção e Tratamento de Congelados

O módulo de tratamento de congelados representa a fase inicial do processo operacional suportado pelo sistema, sendo responsável pela transformação de produtos frescos em unidades logísticas estruturadas, garantindo simultaneamente a rastreabilidade e a consistência dos dados ao longo de todo o processo.

Este módulo permite a criação de caixas de produto com base em artigos previ-

amente definidos, assegurando a validação de múltiplos parâmetros críticos, como o lote, o peso, as datas de produção e congelação, bem como as regras específicas definidas na ficha técnica de cada artigo. A interface associada a este processo encontra-se ilustrada na Figura 6.4.

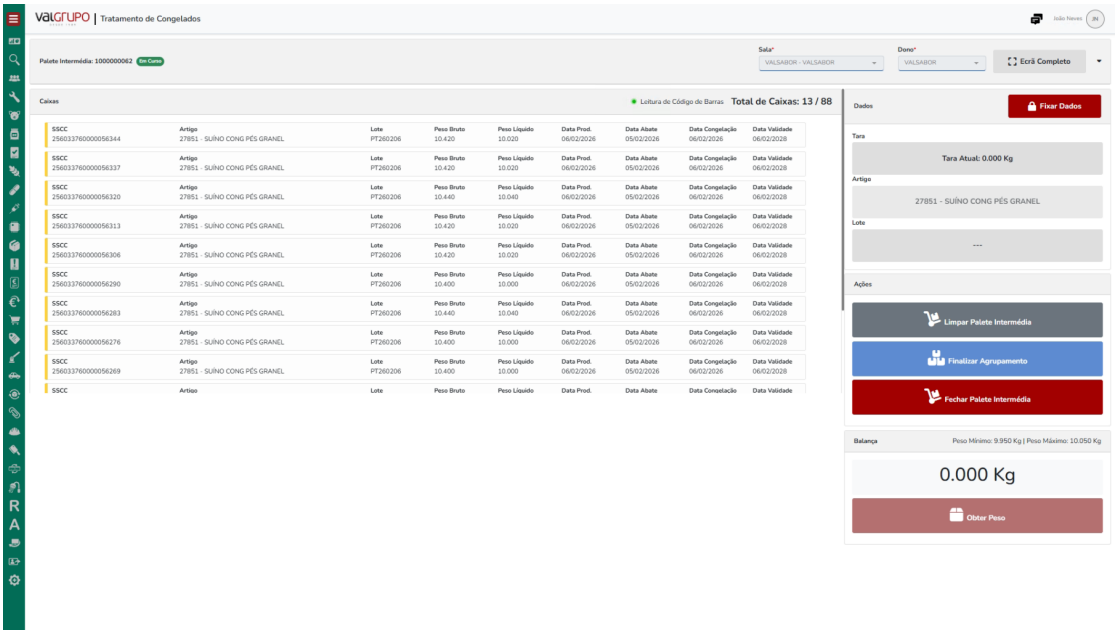

Figura 6.4: *Interface do módulo de tratamento de congelados.*

6.2.1 Criação de Caixas e Validações

No contexto do sistema desenvolvido, a fase de criação de caixas assume particular relevância, uma vez que constitui o ponto de entrada de dados que irão propagar-se ao longo de todo o processo produtivo e logístico. Qualquer erro nesta etapa, como a associação incorreta de lotes, pesos fora dos limites definidos ou incoerências temporais, pode comprometer a rastreabilidade das unidades e introduzir inconsistências nos módulos subsequentes e nas integrações externas.

Por este motivo, a criação de caixas foi concebida com forte dependência de mecanismos de validação, garantindo que apenas dados coerentes e autorizados são aceites pelo sistema antes da persistência da informação.

No sistema desenvolvido, o processo de criação de caixas inicia-se com a leitura de códigos de barras através de dispositivos compatíveis com o padrão GS1, permitindo identificar automaticamente o artigo, o lote e outros parâmetros relevantes. Esta informação é processada pelo sistema e sujeita a um conjunto rigoroso de validações antes da criação efetiva da caixa.

As validações implementadas nesta fase assumem um papel central, garantindo que apenas dados coerentes e autorizados são persistidos no sistema. Entre as principais validações destacam-se:

- Verificação de permissões do utilizador relativamente ao dono e à sala selecio-

nada;
- Validação da existência de um código Global Trade Item Number (GTIN) válido associado ao artigo;
- Garantia de coerência temporal entre datas de abate, produção e congelação;
- Verificação do cumprimento dos limites mínimo e máximo de peso definidos na ficha técnica;
- Validação da existência e consistência do lote associado ao artigo.

Estas validações são implementadas ao nível do *backend*, recorrendo a mecanismos como *Form Requests* e lógica de negócio centralizada, em alinhamento com recomendações de desenvolvimento seguro que defendem a validação de entradas e a aplicação de regras críticas em componentes controlados do lado do servidor (Souppaya et al., 2022; OWASP Foundation, 2025).

O Código 2, apresentado no Apêndice C, exemplifica algumas das regras aplicadas aos dados recebidos antes da criação de uma caixa. Estas regras verificam campos obrigatórios, referências a entidades existentes, formatos recebidos e restrições básicas de integridade. Pedidos que não respeitem estas condições são rejeitados antes de avançarem para a lógica de negócio.

Para além das regras estruturais, são aplicadas verificações dependentes do contexto da operação. O Código 3, apresentado no Apêndice C, exemplifica o controlo de acesso às entidades selecionadas e a validação do peso segundo a ficha técnica do artigo. Caso estas condições não sejam cumpridas, a operação é interrompida antes da persistência dos dados.

A adoção desta abordagem permite garantir que todas as operações são verificadas independentemente da origem do pedido, reduzindo significativamente o risco de inconsistências e vulnerabilidades associadas à introdução de dados inválidos. Para além disso, assegura a consistência entre diferentes componentes do sistema e facilita a manutenção da lógica de validação.

Após a validação, o sistema procede à criação da caixa, associando-lhe o lote correspondente e registando automaticamente os dados necessários para a rastreabilidade, garantindo assim a integridade e fiabilidade da informação ao longo de todo o processo.

6.2.2 Agrupamento de Caixas

Após a criação, as caixas são organizadas em agrupamentos lógicos que representam unidades intermédias dentro do processo produtivo. Estes agrupamentos permitem estruturar a produção e preparar a consolidação posterior em paletes ou grades.

A criação de um agrupamento está sujeita a um conjunto de restrições que garantem a consistência dos dados, nomeadamente:

- Todas as caixas devem pertencer ao mesmo artigo;
- As caixas devem estar associadas à mesma unidade intermédia em curso;

- Não podem existir caixas previamente finalizadas ou já associadas a outras estruturas.

Quando um agrupamento é concluído, o sistema altera o seu estado interno e procede à emissão de uma etiqueta de identificação, permitindo a sua utilização nas fases seguintes do processo.

6.2.3 Fecho de Palete Intermédia

A fase final deste módulo corresponde ao fecho da palete intermédia, que representa a consolidação de todos os agrupamentos criados durante o processo de produção. O fecho só é permitido quando todas as condições de integridade são verificadas, incluindo:

- Todos os agrupamentos devem estar finalizados;
- Todas as caixas devem pertencer ao mesmo artigo;
- Não podem existir inconsistências nos dados associados à produção.

Após validação, o sistema atualiza o estado da palete intermédia para finalizado, garantindo que esta se encontra pronta para ser utilizada na fase de montagem de unidades logísticas finais.

6.2.4 Integração com SAP (Produção - ActonIT)

No módulo de tratamento de congelados, a integração com o SAP constitui uma etapa crítica, uma vez que permite refletir no sistema ERP os resultados das operações realizadas no terreno. Nesta fase, a fiabilidade da integração depende não apenas da comunicação com o sistema externo, mas sobretudo da garantia de que os dados enviados correspondem a operações válidas, completas e consistentes.

Dado que qualquer incoerência nesta fase pode originar discrepâncias entre o sistema operacional e o sistema de gestão empresarial, a integração foi desenhada com mecanismos específicos de validação e estruturação dos dados antes do envio. O Código 7, apresentado no Apêndice C, exemplifica a sequência aplicada à integração das produções, na qual os dados são validados antes da construção e envio dos respetivos *payloads*.

No sistema desenvolvido, a integração com o serviço externo relacionado que comunica com SAP é realizada através de um mecanismo estruturado que assegura a sincronização dos dados operacionais com o sistema externo. Esta integração ocorre em dois momentos distintos:

- No fecho de cada agrupamento, sendo enviada a informação de produção associada;
- No fecho da palete intermédia, sendo enviada a confirmação de conclusão da operação.

A informação enviada inclui dados como o artigo produzido, o lote, as quantidades e os consumos associados, garantindo que o sistema externo reflete corretamente o processo produtivo realizado.

De forma a assegurar a robustez da integração, o sistema recorre a serviços dedicados responsáveis pela validação dos dados, construção do *payload* e comunicação com o sistema SAP. Esta separação de responsabilidades segue boas práticas de arquitetura em sistemas empresariais, permitindo maior modularidade, reutilização de componentes e facilidade de manutenção.

Antes do envio de qualquer informação, são verificadas a existência da entidade e do respetivo identificador, a associação do dono a uma empresa, a configuração do armazém para integração e a existência dos artigos, lotes e linhas de produção e consumo necessários. Nos agrupamentos, é ainda validada a existência de caixas associadas e das respetivas fichas técnicas. Caso alguma destas condições não seja cumprida, a operação é interrompida antes da construção do *payload*. Esta abordagem é particularmente relevante em sistemas distribuídos, onde a propagação de dados incorretos pode originar inconsistências difíceis de corrigir. Algumas destas verificações encontram-se exemplificadas no Código 6, apresentado no Apêndice C.

Adicionalmente, o sistema implementa controlos que impedem integrações duplicadas, assegurando que cada operação é processada uma única vez no sistema externo. Este princípio, frequentemente designado por idempotência, é particularmente relevante em sistemas distribuídos, onde falhas de comunicação podem levar à repetição de pedidos, tornando essencial garantir que o sistema mantém um comportamento determinístico.

Desta forma, a estratégia de integração adotada procura reforçar a integridade e a consistência da informação comunicada entre o sistema desenvolvido e os serviços externos associados ao SAP.

6.3 Montagem de Unidades Logísticas

A montagem de unidades logísticas corresponde à fase de consolidação final do processo operacional, onde as caixas ou agrupamentos previamente produzidos são organizados em unidades de transporte, como paletes ou grades. Esta etapa assume particular relevância, uma vez que estabelece a ligação entre o processo produtivo e a componente logística do sistema.

O processo é suportado por uma interface dedicada, que permite ao operador definir os parâmetros iniciais da unidade logística e proceder à leitura dos elementos a associar. Um exemplo desta interface encontra-se apresentado na Figura H.1, no Apêndice H.

6.3.1 Inicialização da Unidade Logística

O processo de montagem inicia-se com a definição dos parâmetros da unidade logística, incluindo a sala, o dono e o tipo de tara a utilizar (palete ou grade). Estes parâmetros são essenciais para garantir a correta contextualização da operação e influenciam diretamente as validações subsequentes.

Durante esta fase, o sistema permite ainda recuperar unidades em curso, possibilitando a continuidade de operações previamente iniciadas. Esta funcionalidade é particularmente relevante em ambientes industriais, onde interrupções no processo são comuns.

6.3.2 Leitura e Associação de Caixas

A associação de elementos à unidade logística é realizada através da leitura de códigos de barras, podendo o operador introduzir caixas individuais ou agrupamentos completos provenientes do módulo de tratamento de congelados.

O sistema distingue automaticamente o tipo de código lido (caixa ou agrupamento), aplicando validações específicas para cada caso. Entre as principais verificações destacam-se:

- Existência da caixa ou agrupamento no sistema;
- Garantia de que o elemento não foi previamente utilizado numa unidade final;
- Verificação de que não está associado a uma estrutura intermédia ainda em curso;
- Confirmação de que a palete intermédia de origem se encontra finalizada;
- Validação da correspondência entre o dono da unidade e o dono das caixas.

Adicionalmente, quando a unidade logística já contém elementos, são aplicadas validações adicionais que garantem a consistência da estrutura:

- Homogeneidade de artigo na unidade logística;
- Respeito pelo número máximo de caixas definido na ficha técnica;
- No caso de artigos monolote, verificação da consistência do lote entre todas as caixas.

Estas validações são implementadas no *backend*, assegurando que qualquer tentativa de associação incompatível é interrompida antes da persistência dos dados. O Código 8, apresentado no Apêndice C, exemplifica algumas das verificações aplicadas durante a associação, incluindo o estado da unidade logística, o limite máximo de caixas e a consistência dos artigos associados.

6.3.3 Controlo de Peso e Validação Física

No processo de montagem de unidades logísticas, o controlo de peso funciona como um mecanismo de validação física complementar às verificações lógicas já realizadas pelo sistema. A sua função principal consiste em confirmar que a composição digital da unidade logística corresponde efetivamente ao estado físico observado no terreno.

Esta verificação torna-se particularmente importante nesta fase, uma vez que erros de contagem, associação indevida de caixas ou discrepâncias na tara utilizada podem não ser imediatamente visíveis apenas através da validação lógica, comprometendo a fiabilidade da unidade antes do seu fecho e posterior integração nos restantes processos logísticos.

Durante o processo de montagem, o sistema integra-se com uma balança industrial, permitindo a leitura contínua do peso da unidade logística. Esta integração é realizada através de comunicação via MQTT, garantindo atualização em tempo real dos valores de peso.

O controlo de peso constitui um mecanismo adicional de validação, sendo comparado com o peso esperado com base nas caixas associadas e na tara selecionada. O sistema apenas permite o fecho da unidade logística quando o peso real se encontra dentro da margem admissível definida na ficha técnica do artigo.

A adoção deste mecanismo introduz uma validação cruzada entre os dados lógicos do sistema e a verificação física da unidade logística, permitindo reforçar a consistência da informação e reduzir a probabilidade de erros não detetados. Ao combinar validações digitais com validação física, o sistema aumenta significativamente a fiabilidade do processo.

Desta forma, o controlo de peso não atua apenas como um mecanismo operacional, mas também como um elemento complementar de garantia da Integridade dos dados, contribuindo para aproximar os registos digitais do estado físico observado no processo e reforçar a rastreabilidade das operações (McCarthy et al., 2022; International Organization for Standardization, 2007).

6.3.4 Fecho da Unidade Logística

O fecho da unidade logística representa a conclusão do processo de montagem, sendo condicionado pela verificação de um conjunto de requisitos:

- Existência de caixas associadas à unidade;
- Validação do peso dentro dos limites definidos;
- Coerência dos dados associados à unidade logística.

Após validação, o sistema atualiza o estado da unidade para armazenado, regista os movimentos de entrada em armazém e associa a tara utilizada. Adicionalmente, são emitidas etiquetas identificativas da unidade logística, permitindo a sua identificação e utilização nas fases seguintes do processo.

O sistema suporta também o fecho parcial de unidades, permitindo gerar estados intermédios sem finalizar completamente a operação, funcionalidade útil em cenários de produção contínua.

6.3.5 Rastreabilidade e Controlo de Estados

Ao longo de todo o processo de montagem, o sistema mantém registos detalhados das operações realizadas, garantindo a rastreabilidade das caixas e agrupamentos associados a cada unidade logística.

A Figura 6.5 apresenta uma das consultas disponibilizadas pelo Portal para acompanhamento das caixas e unidades logísticas por lote. Esta consulta permite visualizar informação como o artigo, lote, datas relevantes, pesos registados, número de caixas e tipo de unidade logística.

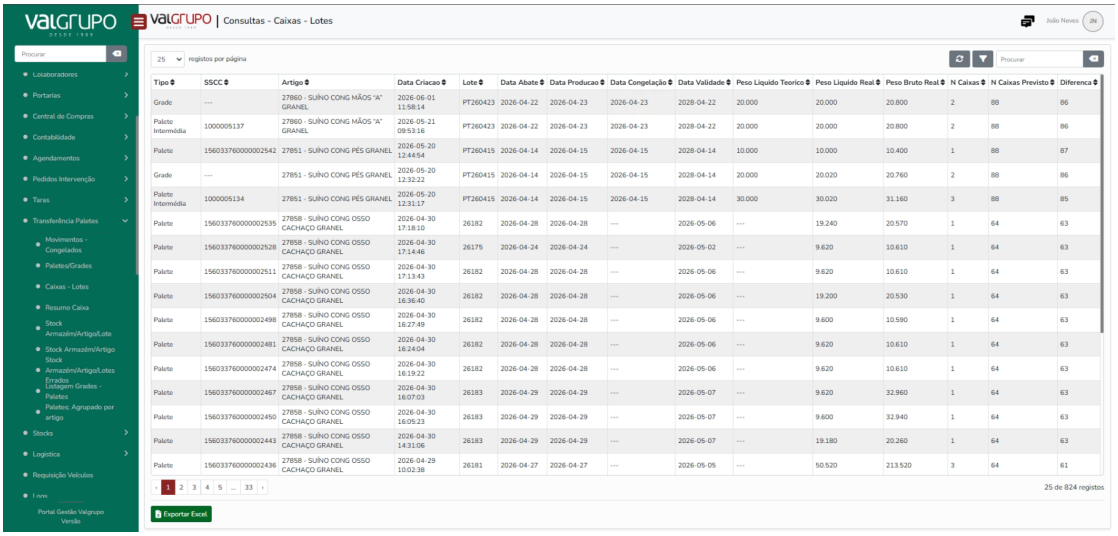

Figura 6.5: *Consulta de caixas e unidades logísticas por lote*

Cada alteração relevante, como a associação ou remoção de caixas, é registada, permitindo reconstruir o histórico da unidade. Este mecanismo é essencial para auditoria e controlo, especialmente em ambientes industriais com elevados requisitos de conformidade.

Adicionalmente, o sistema utiliza uma máquina de estados para controlar o ciclo de vida das unidades logísticas, assegurando que as transições entre estados ocorrem de forma controlada e consistente. Estados como "em curso" e "armazenado" refletem diferentes fases do processo e condicionam as operações permitidas em cada momento.

6.4 Gestão de Pedidos de Paletes e Integração com SAP

O módulo de gestão de pedidos de paletes constitui a camada de orquestração entre o processo produtivo, a logística interna e a integração com sistemas externos. Nesta fase, as unidades logísticas previamente criadas são associadas a pedidos operacionais que representam movimentos de entrada, transferência ou saída de mercadoria.

Este módulo assume um papel central no sistema, uma vez que transforma unidades físicas em entidades documentais, permitindo a sua gestão, validação e posterior integração com sistemas ERP, nomeadamente o SAP.

6.4.1 Criação e Gestão de Pedidos

O processo inicia-se com a criação de um pedido de paletes, que contém informação relativa ao tipo de movimento (entrada, transferência ou saída), ao dono, aos armazéns de origem e destino, às datas operacionais e ao tipo de serviço associado.

A Figura 6.6 apresenta um exemplo de criação de um pedido de transferência entre armazéns. Neste exemplo, são definidos os dados gerais da operação e selecionadas as unidades logísticas a associar ao pedido.

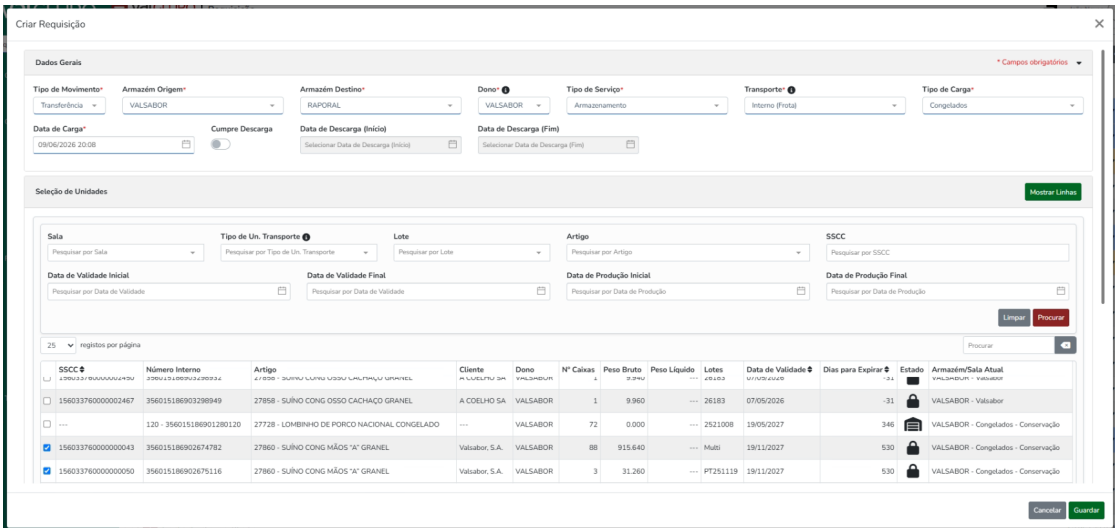

Figura 6.6: *Exemplo de criação de um pedido de transferência entre armazéns*

O sistema utiliza uma lógica baseada em estados para condicionar as operações permitidas ao longo do ciclo de vida dos pedidos. O estado inicial e a sua evolução dependem do tipo de movimento, da necessidade de planeamento de transporte e da situação das unidades logísticas associadas. Por exemplo, pedidos de entrada podem envolver fases de planeamento e detalhe, enquanto transferências e saídas utilizam estados específicos de aprovação. Durante a execução, os pedidos podem assumir estados como em expedição ou expedido, sendo concluídos após a realização das operações necessárias. O estado desativado representa uma situação alternativa associada à desativação do pedido, não correspondendo necessariamente a uma etapa posterior à sua execução.

A Figura 6.7 organiza os principais estados dos pedidos por fases gerais. Esta representação não corresponde a uma sequência rígida, uma vez que os estados aplicáveis e a sua evolução dependem do tipo de movimento, da necessidade de planeamento e das condições da operação.

6.4.2 Associação de Unidades Logísticas

Após a criação do pedido, as unidades de transporte (paletes ou grades) são associadas ao mesmo, estabelecendo a ligação entre o processo físico e o processo documental.

Cada associação gera um registo de movimento, permitindo rastrear a relação entre a unidade logística e o pedido. Este mecanismo assegura a consistência entre os dados

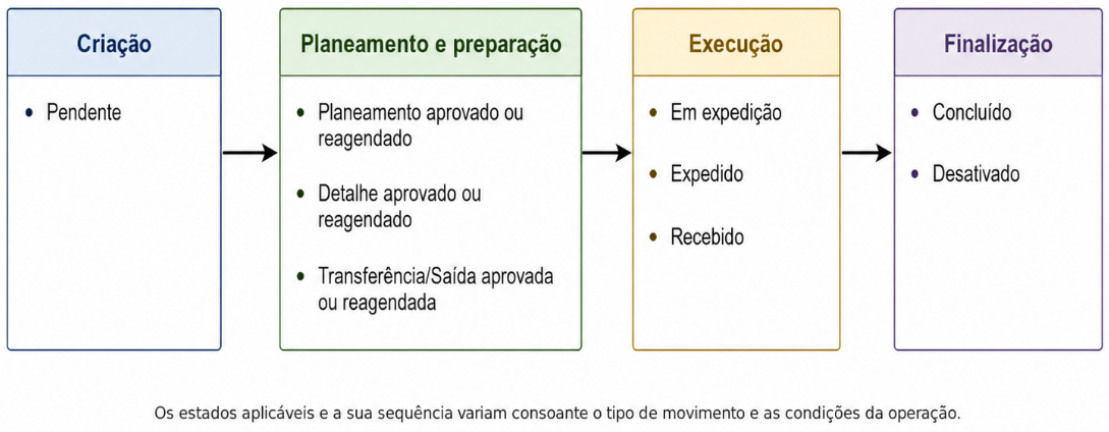

Figura 6.7: *Organização simplificada dos principais estados dos pedidos*

operacionais e a representação documental do processo.

Durante esta fase, o sistema valida:

- Existência das unidades de transporte;
- Coerência entre o tipo de operação e as unidades associadas;
- Compatibilidade entre armazéns, donos e estados das unidades;
- Integridade dos dados associados às paletes.

Estas validações garantem que apenas unidades válidas e coerentes são incluídas nos pedidos, evitando inconsistências nas fases seguintes.

6.4.3 Processos Logísticos: Expedição e Receção

O sistema suporta operações logísticas associadas à expedição e receção de unidades de transporte. Durante a expedição, são registados movimentos de saída, atualizados os estados das unidades e, quando aplicável, associados elementos como planeamentos de transporte.

Na receção, o sistema valida a entrada das unidades no armazém de destino, atualiza a sua localização e trata eventuais discrepâncias, como unidades em falta ou não rececionadas.

Estas operações garantem que o sistema reflete com precisão o estado físico das unidades logísticas, sendo um pré-requisito fundamental para a integração com sistemas externos.

6.4.4 Validações de Pré-Integração

Antes de qualquer integração com o sistema SAP, o sistema executa um conjunto rigoroso de validações que asseguram a consistência e completude dos dados.

Entre as principais validações destacam-se:

- Verificação do estado do pedido (apenas estados válidos permitem integração);
- Confirmação do tipo de operação (transferência ou saída);

- Validação da existência de unidades de transporte associadas;
- Verificação da configuração de integração para a empresa/dono;
- Garantia de que não existem integrações duplicadas;
- Validação do planeamento de transporte associado.

Estas validações são fundamentais para evitar o envio de informação incompleta ou incorreta para o sistema ERP, reduzindo o risco de inconsistências entre sistemas.

Para além das validações automáticas no *backend*, o sistema disponibiliza ao utilizador uma interface de confirmação, onde são apresentados os dados agregados a integrar, permitindo selecionar o planeamento de transporte e proceder à validação da consistência da informação antes do envio.

A Figura 6.8 apresenta o modal de pré-visualização da integração, onde são exibidos os principais dados consolidados por artigo e lote.

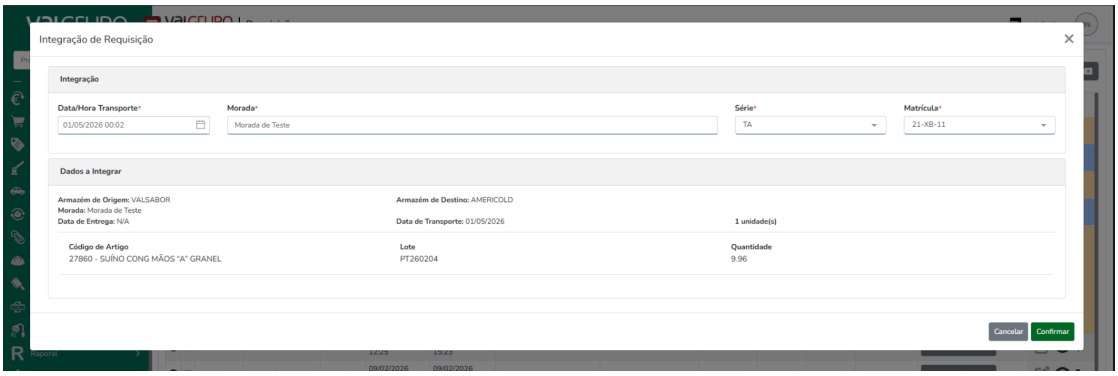

Figura 6.8: *Modal de pré-visualização da integração com o sistema SAP.*

Esta interface reforça o controlo do processo de integração, permitindo ao utilizador validar a informação antes do envio e reduzindo o risco de introdução de dados incorretos. Adicionalmente, contribui para a transparência do sistema, evidenciando a forma como os dados são agregados e estruturados antes de serem enviados para o sistema externo.

6.4.5 Arquitetura de Integração

A integração com o sistema SAP é implementada através de serviços dedicados, responsáveis por encapsular a lógica de validação, construção de *payload* e comunicação com os *endpoints* externos.

O sistema suporta múltiplos contextos de integração, nomeadamente:

- Integração para o sistema Valsabor, utilizando serviços ActonIT;
- Integração para o sistema Maporal, utilizando serviços FutureView.

Cada serviço segue uma arquitetura comum composta por três componentes principais:

- **Validator** – responsável por validar os dados antes da integração;

- **Builder** – responsável por construir o *payload* no formato exigido;
- **Service/Client** – responsável pela comunicação com o sistema externo.

Esta separação de responsabilidades permite maior modularidade e facilita a adaptação a diferentes integrações, sem impactar a lógica central do sistema.

O Código 4, apresentado no Apêndice C, demonstra a sequência utilizada durante uma integração: obtenção da autenticação, validação dos dados, construção do *payload* e envio para o serviço externo. Esta sequência garante que o *payload* apenas é construído e enviado após a conclusão das validações. Caso ocorra um erro, este é registado através de mecanismos de Logging e a operação é interrompida.

6.4.6 Construção e Envio de *payloads*

A informação enviada para o sistema SAP não corresponde diretamente à granularidade das caixas individuais. Antes da integração, o sistema agrega os dados das unidades de transporte, consolidando a informação por artigo e lote.

Este processo de agregação permite:

- Reduzir a complexidade dos dados enviados;
- Garantir alinhamento com o modelo de dados do ERP;
- Evitar redundâncias e inconsistências.

Os *payloads* incluem informação relevante como:

- Identificação do artigo;
- Lote;
- Quantidade total;
- Armazém de origem e destino;
- Dados de transporte e cliente (quando aplicável).

O Código 5, apresentado no Apêndice C, exemplifica a agregação das linhas por artigo, lote e armazém de origem antes da construção do *payload*. Esta transformação converte os registos internos numa estrutura adequada ao serviço externo, reduzindo redundâncias e assegurando a consistência das quantidades enviadas.

Após a construção, os dados são enviados para os serviços externos de integração, sendo posteriormente registado o resultado da operação.

6.4.7 Integração EWM (Caso Maporal)

No contexto da integração com Maporal, o sistema suporta uma etapa adicional de integração com o módulo Extended Warehouse Management (EWM) do SAP.

Esta integração ocorre antes da integração documental principal e consiste na sincronização das paletes com base nos seus identificadores (SSCC), garantindo que o movimento de *stock* é corretamente refletido entre os diferentes módulos do sistema ERP.

O sistema valida:

- Existência de SSCC em todas as unidades;
- Tipo de operação compatível;
- Associação correta das unidades ao pedido.

A integração EWM funciona como um mecanismo de consistência adicional, assegurando que o estado físico das paletes está alinhado com o sistema antes da criação dos documentos logísticos finais.

6.4.8 Importação e Integração Bidirecional

Para além da exportação de dados, o sistema suporta a importação automática de informação proveniente de serviços externos associados ao SAP. Esta funcionalidade permite sincronizar unidades de transporte, caixas, lotes e outros dados relevantes com os registos locais do Portal.

Durante a importação, os dados externos são inicialmente agrupados através do identificador da unidade de transporte. O sistema verifica se a unidade já existe localmente e distingue entre a criação de novos registos e a atualização de registos existentes. As unidades que já não se encontram em montagem não são reconstruidas, evitando alterações à sua composição após a conclusão do processo.

Antes da criação ou atualização das caixas, é verificada a existência local dos artigos associados. Os lotes são criados ou atualizados com base nos dados recebidos e o identificador SSCC de cada caixa é utilizado para verificar se esta já existe no sistema. Quando uma caixa existente surge associada a outra unidade de transporte, a respetiva associação é atualizada e os pesos da unidade anterior são corrigidos.

As operações de criação e atualização de cada unidade de transporte são executadas no âmbito de uma transação de base de dados. Desta forma, caso ocorra uma exceção durante o processamento, as alterações realizadas nessa transação podem ser revertidas. Os erros são também registados através de mecanismos de Logging e comunicados através de notificações internas.

A importação pode, contudo, originar inconsistências quando os dados externos estão incompletos, incorretos ou não possuem correspondência local. Por exemplo, a inexistência de um artigo no Portal impede a importação da respetiva caixa, podendo resultar numa unidade parcialmente reconstruida. Podem também ocorrer diferenças relacionadas com pesos, lotes, datas ou associações entre caixas e unidades de transporte. Adicionalmente, como a sincronização atualiza e remove registos locais com base na informação externa recebida, uma origem externa incompleta ou desatualizada poderá afetar a representação local da unidade.

Este processo permite aproximar a informação existente nos diferentes sistemas, mas exige monitorização dos erros registados e análise das situações em que os dados recebidos não podem ser importados integralmente.

6.4.9 Segurança na Integração e Orquestração

O módulo de gestão de pedidos e integração com SAP evidencia a importância da implementação de mecanismos de segurança em sistemas empresariais distribuídos.

A existência de validações rigorosas antes da integração, aliada à centralização da lógica no *backend*, reduz significativamente o risco de envio de dados inconsistentes ou manipulados. Adicionalmente, o controlo de permissões garante que apenas utilizadores autorizados podem executar operações críticas.

A separação entre validação, construção de *payload* e comunicação externa contribui também para uma maior robustez do sistema, permitindo isolar erros e facilitar a manutenção.

Desta forma, o sistema assegura não só a correta execução dos processos logísticos, como também a integridade, consistência e segurança da informação ao longo de todo o ciclo de integração.

Para além disso, a abordagem adotada demonstra uma integração eficaz entre requisitos de segurança e necessidades operacionais, evidenciando que a segurança não foi tratada como um elemento adicional, mas sim como parte integrante do desenho do sistema.

6.5 Síntese de Segurança do Sistema

Os mecanismos de segurança implementados procuraram responder principalmente a riscos associados à manipulação de pedidos, execução de operações não autorizadas, introdução de dados incoerentes, repetição de integrações e divergências entre as operações físicas e os registos digitais.

A validação no *backend* demonstrou ser particularmente relevante durante os testes funcionais realizados. Pedidos com campos obrigatórios ausentes, referências inexistentes, pesos fora dos limites definidos ou estados incompatíveis foram rejeitados antes da persistência dos dados. Esta abordagem permite mitigar tentativas de contornar as validações existentes no *frontend* e reduz a propagação de informação incorreta ao longo do processo (OWASP Foundation, 2025; Souppaya et al., 2022).

O controlo de permissões restringe o acesso dos utilizadores às entidades e funcionalidades para as quais possuem autorização, mitigando riscos de acesso indevido e execução de operações fora das respetivas responsabilidades. Contudo, não foi realizada uma auditoria formal e exaustiva às permissões existentes, pelo que não é possível afirmar que todos os cenários de autorização foram avaliados.

Nas integrações externas, a validação prévia dos dados, o controlo dos estados e a verificação de integrações anteriores permitem reduzir o risco de envio de dados incompletos, execução de operações fora de sequência e criação de documentos duplicados. O registo das respostas e dos erros através de mecanismos de Logging facilita ainda o diagnóstico de falhas. Apesar destes mecanismos, permanecem dependências relativamente à disponibilidade e ao comportamento dos serviços externos.

A correspondência entre os registos digitais e as operações físicas foi reforçada através da leitura de códigos de barras, do controlo de peso e da manutenção do histórico das unidades logísticas. Estes mecanismos contribuem para detetar determinadas discrepâncias operacionais, embora não eliminem completamente erros provenientes dos equipamentos, dos dados externos ou da utilização incorreta do sistema.

A eficácia dos mecanismos implementados foi avaliada principalmente através de testes funcionais, utilização em ambiente de *staging* e validação dos fluxos operacionais pela equipa. Não foram realizados testes formais de penetração, uma avaliação quantitativa da redução de vulnerabilidades ou uma verificação completa de conformidade com o OWASP ASVS. Assim, os resultados permitem confirmar o funcionamento esperado dos controlos nos cenários testados, mas não demonstram a ausência de vulnerabilidades.

A Tabela D.1, incluída no Apêndice D, estabelece uma correspondência indicativa entre os principais mecanismos implementados no sistema e as áreas de controlo do OWASP ASVS na versão 5.0.0 (OWASP Foundation, 2025). Este mapeamento foi realizado retrospetivamente, com base nas funcionalidades desenvolvidas, não representando uma avaliação formal de conformidade com todos os requisitos do referencial.

O mapeamento apresentado evidencia que os mecanismos implementados abrangem diferentes áreas relevantes do OWASP ASVS, nomeadamente validação, autenticação, autorização, lógica de negócio, tratamento de erros e arquitetura segura. Contudo, esta correspondência é parcial e retrospetiva, não constituindo uma certificação ou avaliação formal de conformidade.

Como limitações, destacam-se a ausência de testes automatizados abrangentes, de testes formais de segurança, de mecanismos centralizados de monitorização e alerta sobre as integrações e de uma avaliação sistemática das permissões. Embora existam mecanismos de Logging para apoiar o diagnóstico de falhas, estes não correspondem a uma solução centralizada de monitorização e emissão automática de alertas. Futuramente, estes aspetos poderão ser aprofundados através de testes unitários e de integração, revisão periódica dos acessos, análise de dependências, testes de segurança aplicacional e avaliação formal dos controlos segundo o OWASP ASVS.

Em síntese, os mecanismos implementados demonstraram capacidade para prevenir diferentes inconsistências e operações indevidas nos cenários funcionais avaliados. No entanto, a segurança do sistema deve ser entendida como um processo contínuo, cuja eficácia depende da validação recorrente dos controlos, da monitorização das integrações e da adaptação a novos riscos (Souppaya et al., 2022).

6.6 Discussão das Decisões Técnicas

As decisões técnicas adotadas foram condicionadas pela arquitetura já existente no Portal, pelas tecnologias utilizadas pela equipa e pela necessidade de disponibilizar rapidamente uma primeira versão funcional do projeto ValSabor. Por esse motivo, não foi realizada uma comparação formal entre diferentes tecnologias ou arquiteturas antes

do desenvolvimento. Laravel, Vue.js e os mecanismos internos de autenticação, permissões e integração foram mantidos por assegurarem compatibilidade com o sistema existente e facilitarem a manutenção futura pela equipa.

Uma das principais decisões consistiu em centralizar no *backend* as validações críticas e as regras de negócio. Esta abordagem encontra-se alinhada com as recomendações do OWASP ASVS na versão 5.0.0 e do NIST SSDF, uma vez que impede que a segurança e a integridade das operações dependam exclusivamente das validações existentes no *frontend* (OWASP Foundation, 2025; Souppaya et al., 2022). Durante os testes funcionais, esta decisão permitiu rejeitar pedidos com dados inválidos, estados incompatíveis, referências inexistentes ou utilizadores sem acesso às entidades envolvidas.

Contudo, esta centralização introduziu complexidade adicional no *backend*. Algumas operações exigem numerosas verificações dependentes do contexto, originando componentes de validação extensos e aumentando a dificuldade de compreender, testar e manter determinadas regras de negócio. A distribuição das validações entre *Form Requests*, validadores específicos e serviços permitiu reduzir parcialmente este problema, embora algumas responsabilidades pudessem ter sido separadas de forma mais granular.

Nas integrações externas, optou-se por validar os dados antes da construção do *payload*, separar as responsabilidades de validação, construção e comunicação, e registar as respostas recebidas. Esta organização permitiu reduzir o risco de envio de informação incompleta e facilitar o diagnóstico de falhas. O controlo de integrações anteriores permitiu também prevenir determinadas operações duplicadas, em concordância com os princípios de consistência e Idempotência discutidos no Estado da Arte (Fielding et al., 2022; OWASP Foundation, 2025).

Não foram consideradas arquiteturas ou tecnologias alternativas, uma vez que o desenvolvimento ocorreu sobre a arquitetura e os mecanismos já utilizados no Portal. Contudo, foi considerada a possibilidade de permitir que vários utilizadores trabalhassem simultaneamente sobre o mesmo processo, tanto durante a criação de caixas em agrupamentos como na montagem de paletes. Esta possibilidade não foi implementada, uma vez que não foi possível definir, dentro do prazo disponível, um mecanismo suficientemente fiável para controlar a concorrência, validar transações simultâneas e impedir que diferentes utilizadores realizassem operações incompatíveis sobre as mesmas entidades. Optou-se, assim, por não avançar com o suporte explícito à utilização simultânea do mesmo processo por vários utilizadores, privilegiando a consistência dos dados e a prevenção de conflitos em detrimento de uma maior capacidade de utilização concorrente.

A utilização de leitores de códigos de barras, códigos normalizados e controlo de peso permitiu complementar as validações digitais com informação proveniente das operações físicas. Esta decisão reforçou a correspondência entre o estado registado no Portal e o estado observado no processo, contribuindo para a integridade e rastreabilidade da informação (International Organization for Standardization, 2007; McCarthy

et al., 2022). No entanto, estes mecanismos não eliminam completamente a possibilidade de erro, uma vez que continuam dependentes da configuração dos equipamentos, da qualidade dos dados recebidos e da utilização correta do sistema.

O desenvolvimento inicial do projeto decorreu com um prazo reduzido para entrada em produção. Este compromisso permitiu disponibilizar rapidamente as funcionalidades essenciais, mas limitou o tempo disponível para formalizar requisitos, implementar testes automatizados e antecipar todas as exceções operacionais. Consequentemente, algumas regras de negócio foram refinadas posteriormente, à medida que eram identificadas novas necessidades e situações não previstas inicialmente. A posteriori, uma fase inicial mais extensa de levantamento e validação dos requisitos poderia ter reduzido parte deste retrabalho.

Não foram recolhidas métricas quantitativas que permitam determinar o número total de falhas prevenidas ou comparar formalmente o sistema antes e depois das alterações. O grau de sucesso foi avaliado principalmente através dos testes funcionais, da utilização em ambiente de *staging*, da validação pela equipa e da entrada em funcionamento dos fluxos desenvolvidos. Embora estes resultados indiquem que os mecanismos implementados funcionaram nos cenários avaliados, não permitem demonstrar a ausência de vulnerabilidades ou inconsistências noutros cenários.

De forma global, as decisões adotadas permitiram responder aos principais requisitos operacionais e aplicar, no contexto do caso de estudo, práticas relacionadas com validação de dados, controlo de acessos, integração entre sistemas e rastreabilidade discutidas no Capítulo 4 (OWASP Foundation, 2025; Souppaya et al., 2022; International Organization for Standardization, 2007). Contudo, estas decisões evidenciaram também compromissos entre rapidez de disponibilização, complexidade da implementação e profundidade da validação realizada.

Como principais limitações permanecem a ausência de testes automatizados, a ausência de uma avaliação formal de segurança e a dependência dos serviços externos.

7 Implementações Técnicas

Neste capítulo são apresentadas as principais implementações técnicas realizadas durante o estágio, com especial enfoque na arquitetura da aplicação, na organização do *backend* e frontend, bem como nos mecanismos de autenticação, autorização e segurança adotados. Para além da descrição das soluções implementadas, é dada ênfase às decisões técnicas subjacentes, enquadrando-as no contexto de sistemas empresariais reais e nas boas práticas de desenvolvimento seguro.

7.1 Arquitetura da Aplicação

O sistema desenvolvido segue uma arquitetura do tipo *client-server*, baseada na separação entre *frontend* e *backend*, comunicando através de APIs REST. Esta abordagem permite uma clara separação de responsabilidades, facilitando a escalabilidade, manutenção e evolução do sistema.

O *backend*, desenvolvido em Laravel, é responsável pela implementação da lógica de negócio, validação de dados, controlo de acessos e integração com sistemas externos. O *frontend*, desenvolvido em Vue.js, é responsável pela apresentação da informação e interação com o utilizador.

A Figura 6.1, apresentada no Capítulo 6, ilustra esta organização no contexto do módulo de gestão de unidades logísticas, incluindo a separação entre *frontend* e *backend* e a comunicação com dispositivos e serviços externos.

No sistema desenvolvido, a separação entre *frontend* e *backend* revelou-se essencial para garantir o isolamento da lógica de negócio, permitindo que todas as regras críticas fossem centralizadas no *backend* e não dependessem da integridade da interface do utilizador.

A comunicação entre os diferentes componentes é realizada através de pedidos HTTP autenticados, sendo todas as operações críticas sujeitas a validação no *backend*. Esta abordagem encontra-se alinhada com boas práticas de verificação de segurança em aplicações *web*, nomeadamente no que diz respeito à validação de entradas, autenticação e controlo de acessos (OWASP Foundation, 2025).

Adicionalmente, a arquitetura foi concebida de forma modular, permitindo a separação de componentes responsáveis por validação, processamento e integração. Esta decisão facilita a manutenção do sistema e permite a sua adaptação a diferentes contextos operacionais e integrações externas.

7.2 Implementação do *Backend*

O *backend* foi desenvolvido recorrendo ao *framework* Laravel, adotando uma estrutura organizada em camadas que promove a separação entre controladores, serviços e modelos de dados.

Os controladores são responsáveis por receber os pedidos provenientes do *frontend*, encaminhando-os para a lógica de negócio apropriada. No entanto, a lógica principal não é implementada diretamente nos controladores, sendo delegada em serviços específicos, permitindo uma maior reutilização de código e melhor organização da aplicação.

A lógica de negócio encontra-se centralizada em serviços, onde são implementadas regras críticas como validações, processamento de dados e integração com sistemas externos. Esta abordagem permite isolar a complexidade da aplicação e facilitar a sua manutenção.

No contexto do sistema desenvolvido, estes serviços assumem um papel central em módulos críticos, como a criação de caixas, validação de unidades logísticas e integração com o SAP. Por exemplo, no processo de integração, a lógica encontra-se distribuída por componentes específicos responsáveis pela validação dos dados, construção do *payload* e comunicação com o sistema externo, garantindo uma separação clara de responsabilidades e maior controlo sobre o fluxo de execução.

As validações são implementadas através de mecanismos como *Form Requests*, garantindo que todos os dados são verificados antes de serem processados. Esta validação inclui não só a verificação de formatos, mas também regras de negócio complexas, como coerência de estados, permissões de utilizador e integridade dos dados.

Importa destacar que todas as validações críticas são realizadas no *backend*, uma vez que o *frontend* pode ser manipulado e não deve ser considerado uma fonte confiável. Esta decisão permite reduzir o risco associado ao envio de pedidos adulterados e encontra-se alinhada com os requisitos de validação de entradas definidos pelo OWASP ASVS na versão 5.0.0 (OWASP Foundation, 2025).

A integração com sistemas externos, como o SAP, é realizada através de serviços dedicados, responsáveis pela validação, construção de *payloads* e comunicação com os *endpoints* externos. Esta separação de responsabilidades permite maior controlo sobre o processo de integração e facilita a deteção de erros.

A aplicação destas práticas encontra-se exemplificada nos excertos de código apresentados no Apêndice C. Em particular, o Código 2 demonstra a validação estrutural dos pedidos, enquanto os Códigos 4 e 5 exemplificam a separação entre a coordenação da integração e a construção dos dados enviados.

7.3 Implementação do *Frontend*

O *frontend* foi desenvolvido com recurso ao *framework* Vue.js, adotando uma arquitetura baseada em componentes reutilizáveis, permitindo a construção de interfaces modulares e de fácil manutenção.

Cada funcionalidade do sistema é representada por um conjunto de componentes que encapsulam a lógica de apresentação e interação com o utilizador. Esta abordagem permite reduzir a duplicação de código e facilita a evolução da interface ao longo do tempo.

No sistema desenvolvido, esta abordagem revelou-se particularmente relevante em módulos operacionais intensivos, como a leitura de códigos de barras e montagem de unidades logísticas, onde a interface necessita de responder em tempo real às ações do utilizador, garantindo fluidez e redução de erros durante a operação.

A comunicação com o *backend* é realizada através de serviços dedicados, responsáveis por efetuar pedidos HTTP às APIs do sistema. Esta separação permite centralizar a lógica de comunicação e tratar de forma consistente aspetos como autenticação, tratamento de erros e gestão de respostas.

Embora o *frontend* desempenhe um papel importante na experiência do utilizador, em sistemas reais não deve ser considerado um elemento de segurança. Por esse motivo, todas as validações críticas são novamente verificadas no *backend*, reduzindo o risco de contornar regras de negócio através da manipulação da interface.

Adicionalmente, a interface foi desenhada de forma a minimizar erros do utilizador, através da apresentação de *feedback* imediato, validações preliminares e restrições na interação. Estas medidas contribuem para a redução de erros operacionais, embora não substituam as validações realizadas no *backend*.

7.4 Autenticação e Autorização

O sistema implementa mecanismos de autenticação e autorização que garantem que apenas utilizadores autenticados e devidamente autorizados podem aceder às funcionalidades disponíveis.

A posição transversal destes mecanismos na comunicação entre o *frontend* e o *backend* encontra-se representada na Figura 6.1.

A autenticação é realizada com recurso ao Laravel Sanctum em modo Single-Page Application (SPA), recorrendo a sessões autenticadas, *cookies* HTTP e proteção Cross-Site Request Forgery (CSRF). Antes da autenticação, o *frontend* obtém o *cookie* CSRF necessário e, após o login, os pedidos subsequentes ao *backend* incluem as credenciais de sessão através dos cookies enviados pelo browser. As rotas protegidas são validadas pelo middleware `auth:sanctum`, garantindo que apenas utilizadores autenticados conseguem aceder aos recursos da aplicação.

Neste modelo, a autenticação entre o *frontend* e o *backend* não depende do armazenamento manual de *tokens* no cliente, como em abordagens baseadas em *Bearer tokens*.

Em vez disso, a sessão autenticada é mantida através dos *cookies* geridos pelo browser, enquanto a proteção CSRF contribui para mitigar pedidos não autorizados realizados a partir de origens externas.

Para além da autenticação, foram implementados mecanismos adicionais de verificação de permissões ao nível dos serviços e dos *endpoints*, garantindo que mesmo utilizadores autenticados apenas conseguem executar operações para as quais possuem autorização explícita.

No que diz respeito à autorização, o sistema implementa um modelo baseado em permissões e perfis de utilizador. Cada operação crítica é sujeita a verificações que garantem que o utilizador possui os privilégios necessários para a executar.

Um exemplo da verificação das permissões do utilizador relativamente às entidades envolvidas numa operação encontra-se no Código 3, apresentado no Apêndice C.

Este controlo é aplicado nos *endpoints* críticos analisados e desenvolvidos, procurando prevenir acessos indevidos e reduzir o risco de exploração de funcionalidades sensíveis. A correta implementação de mecanismos de controlo de acessos é amplamente reconhecida como um dos fatores mais críticos na mitigação de vulnerabilidades em aplicações *web* modernas (Xu et al., 2025; OWASP Foundation, 2021; OWASP Foundation, 2025).

7.5 Medidas de Segurança

As principais medidas de segurança adotadas incluem a validação sistemática no *backend*, o controlo de acessos e o controlo dos estados das entidades. Estes mecanismos procuram mitigar riscos relacionados com pedidos manipulados, acessos indevidos e execução de operações fora da sequência prevista, encontrando-se alinhados com áreas de verificação do OWASP ASVS na versão 5.0.0 (OWASP Foundation, 2025).

Nas integrações externas, são realizadas validações antes do envio dos dados e verificações sobre integrações anteriormente registadas. Estes mecanismos procuram reduzir o risco de propagação de informação incompleta ou inconsistente e de criação duplicada de determinados documentos, relacionando-se com os problemas associados à repetição de pedidos discutidos no Estado da Arte (Fielding et al., 2022).

A validação no *backend* permite rejeitar pedidos com campos inválidos, referências inexistentes, pesos fora dos limites definidos ou estados incompatíveis. Sem estas verificações, um pedido enviado diretamente para a API poderia contornar as restrições existentes no *frontend* e introduzir dados incoerentes no sistema.

O controlo de acessos procura impedir que utilizadores executem operações sobre entidades ou funcionalidades para as quais não possuem autorização. A ausência deste controlo poderia permitir a consulta ou alteração indevida de informação e comprometer a integridade dos processos. Por sua vez, o controlo de estados reduz o risco de execução de operações fora da sequência prevista, como modificar unidades já concluídas ou integrar pedidos que ainda não se encontram preparados.

A eficácia destes controlos foi observada através de testes funcionais, incluindo testes realizados com utilizadores sem as permissões necessárias, nos quais pedidos inválidos, operações não autorizadas e estados incompatíveis foram rejeitados antes da persistência ou integração. Contudo, não foram recolhidas métricas quantitativas sobre o número de operações bloqueadas, nem foi realizada uma avaliação formal de segurança. Assim, os resultados confirmam o comportamento esperado nos cenários testados, mas não demonstram a ausência de vulnerabilidades.

A Tabela 7.1 apresenta uma análise qualitativa e retrospetiva dos principais riscos residuais. Os níveis apresentados foram atribuídos considerando qualitativamente a possibilidade de ocorrência e o potencial impacto de cada risco, não correspondendo a uma avaliação formal de risco.

Tabela 7.1: *Análise qualitativa dos principais riscos residuais*

Risco residual	**Justificação**	**Nível**
Falhas em regras de autorização	Não foi realizada uma auditoria formal e exaustiva às permissões existentes.	Médio
Conflitos entre operações simultâneas	Não foi implementado suporte explícito para vários utilizadores trabalharem simultaneamente sobre o mesmo processo.	Médio
Falhas nos serviços externos	A disponibilidade e o comportamento dos serviços de integração não são controlados pelo Portal.	Médio
Regressões após alterações	A inexistência de testes automatizados aumenta a dependência de testes funcionais manuais.	Médio
Dados incorretos provenientes de equipamentos	As validações reduzem o risco, mas não eliminam falhas de configuração, comunicação ou utilização dos equipamentos.	Baixo a médio

As práticas adotadas encontram-se alinhadas com princípios do NIST SSDF e com áreas de verificação do OWASP ASVS na versão 5.0.0, embora não tenha sido realizada uma avaliação formal de conformidade com estes referenciais (Souppaya et al., 2022; OWASP Foundation, 2025). Apesar dos controlos implementados, permanecem riscos associados às permissões, concorrência, dependências externas e ausência de testes automatizados, reforçando a necessidade de avaliação e melhoria contínuas.

8

Análise Crítica e Aprendizagens

O presente capítulo apresenta uma análise crítica do trabalho desenvolvido ao longo do estágio, com especial enfoque no projeto ValSabor. Para além da descrição das dificuldades encontradas, são analisadas as soluções adotadas, a situação de segurança identificada à luz do OWASP Top 10 e as principais competências técnicas e pessoais desenvolvidas.

A análise crítica assume particular importância por permitir refletir sobre o trabalho para além da implementação técnica das funcionalidades. O estágio decorreu num contexto empresarial real, onde os requisitos evoluíram ao longo do desenvolvimento, as soluções tiveram de ser ajustadas às necessidades operacionais e a integração com sistemas externos introduziu dependências que nem sempre podiam ser resolvidas apenas ao nível da aplicação interna.

8.1 Dificuldades Encontradas

Uma das primeiras dificuldades encontradas esteve relacionada com a adaptação ao ambiente tecnológico e organizacional da empresa. O estágio decorreu numa equipa de desenvolvimento já estabelecida, com sistemas internos em funcionamento, padrões de implementação próprios e uma base de código em evolução. Esta realidade exigiu uma fase inicial de aprendizagem, tanto ao nível das tecnologias utilizadas como da organização do código, dos fluxos de trabalho e das regras internas seguidas pela equipa.

No caso específico do projeto ValSabor, a principal dificuldade esteve associada à complexidade das regras de negócio. O sistema não se limitava à criação de páginas ou formulários, sendo necessário representar processos industriais e logísticos com múltiplas etapas, estados, validações e exceções. Operações como a criação de caixas, o agrupamento de caixas, o fecho de paletes intermédias, a montagem de unidades logísticas e a integração com serviços externos exigiam a verificação de várias condições

antes da sua execução. Esta complexidade tornou essencial compreender corretamente o processo antes de avançar para a implementação.

Outra dificuldade relevante esteve relacionada com a natureza evolutiva dos requisitos. Como referido anteriormente, os requisitos não foram levantados através de um processo formal completo de engenharia de requisitos, tendo sido transmitidos principalmente pela equipa técnica e pelo responsável da área de informática. À medida que as funcionalidades eram desenvolvidas e testadas, surgiam novas regras, exceções e ajustamentos necessários ao funcionamento real da operação. Esta evolução obrigou a rever decisões técnicas, adaptar validações e manter uma comunicação frequente com a equipa.

A integração com serviços externos associados ao SAP constituiu também uma limitação prática significativa. Como a aplicação não comunicava diretamente com o SAP, mas sim através de serviços intermédios, determinados problemas dependiam de validações e comportamentos definidos fora da aplicação interna. Em alguns casos, foi necessário solicitar alterações ou ajustes nesses serviços para que os dados enviados fossem aceites e processados corretamente. Esta dependência introduziu complexidade adicional, sobretudo na definição dos dados a enviar, na adaptação dos *payloads* e na validação conjunta dos resultados.

As integrações com equipamentos industriais também apresentaram desafios. A utilização de leitores de códigos de barras, códigos GS1, balanças industriais e etiquetas em ZPL exigiu atenção à correspondência entre eventos físicos e registos digitais. Pequenas inconsistências na leitura de códigos, no peso registado ou no estado das entidades podiam afetar etapas posteriores do processo. Esta realidade reforçou a necessidade de validações robustas e de um controlo rigoroso da sequência das operações.

Por fim, a existência de funcionalidades com impacto operacional aumentou a responsabilidade associada ao desenvolvimento. Alterações incorretas poderiam comprometer processos utilizados em ambiente real, pelo que foi necessário valorizar os testes locais, a validação em ambiente de *staging*, a revisão pela equipa técnica e a análise cuidada dos erros identificados antes da passagem para produção.

8.2 Enquadramento da Situação Identificada no OWASP Top 10

Durante o estágio foi identificada, de forma observacional e não intrusiva, uma situação relacionada com a exposição indevida de informação técnica num serviço *web* associado ao grupo empresarial. A situação foi detetada durante uma navegação funcional normal, sem exploração ativa de vulnerabilidades, sem tentativa de autenticação indevida e sem alteração de qualquer sistema.

A situação observada consistia na apresentação pública de mensagens de erro detalhadas, incluindo informação técnica sobre a aplicação e o respetivo funcionamento interno. A exposição de um Stack trace e de detalhes técnicos num serviço acessível

publicamente pode facilitar a compreensão da arquitetura da aplicação por terceiros e aumentar a superfície de ataque. Este tipo de problema é frequentemente associado a configurações inadequadas de ambiente, tratamento insuficiente de erros e exposição de informação técnica que deveria permanecer restrita.

À luz do OWASP Top 10:2021, a situação enquadra-se sobretudo na categoria associada a Security Misconfiguration, uma vez que envolve a exposição de informação técnica e a provável utilização de uma configuração inadequada para um serviço acessível publicamente (OWASP Foundation, 2021). Esta categoria inclui problemas como configurações inseguras, mensagens de erro excessivamente detalhadas, ambientes incorretamente configurados e ausência de mecanismos adequados de proteção.

A situação também pode ser relacionada, de forma complementar, com preocupações de controlo de acessos e proteção da informação. Embora não tenha sido realizada qualquer ação de exploração, a disponibilidade pública de informação técnica pode contribuir para ataques posteriores, caso seja combinada com outras fragilidades. Por esse motivo, a exposição de detalhes internos deve ser evitada e tratada através de configurações adequadas, mensagens de erro controladas e registo interno da informação necessária para diagnóstico.

O OWASP ASVS na versão 5.0.0 reforça a importância de mecanismos adequados de tratamento de erros, registo de eventos, proteção de informação sensível e controlo de acessos em aplicações *web* (OWASP Foundation, 2025). A literatura recente sobre testes de segurança em aplicações *web* também evidencia que a deteção deste tipo de fragilidades depende da combinação entre ferramentas, análise manual e validação contextual do comportamento das aplicações (Qadir et al., 2025).

Esta situação permitiu compreender, em contexto real, a importância da segurança por configuração e da separação adequada entre ambientes de desenvolvimento, teste e produção. Demonstrou também que a segurança de uma aplicação não depende apenas do código desenvolvido, mas também da forma como os serviços são configurados, disponibilizados e monitorizados.

8.3 Soluções Adotadas

Perante as dificuldades encontradas, foram adotadas várias estratégias ao longo do estágio. A primeira consistiu na clarificação contínua dos requisitos com a equipa técnica. Sempre que surgiam dúvidas sobre o comportamento esperado de uma funcionalidade, eram discutidas as regras de negócio, os estados permitidos, as exceções aplicáveis e os impactos das alterações. Esta comunicação foi essencial para reduzir ambiguidades e alinhar a implementação com as necessidades operacionais.

No desenvolvimento do projeto ValSabor, uma das principais soluções adotadas foi a centralização da lógica crítica no *backend*. As validações mais relevantes, nomeadamente as relacionadas com artigos, lotes, pesos, estados, permissões e integrações, foram tratadas do lado do servidor. Esta abordagem permitiu reduzir a dependência de validações realizadas apenas no *frontend* e reforçar a Integridade dos dados. A

adoção desta estratégia encontra-se alinhada com boas práticas de desenvolvimento seguro, que recomendam a validação de dados em componentes controlados pelo servidor (Souppaya et al., 2022; OWASP Foundation, 2025).

Outra solução importante foi a separação de responsabilidades no código. Sempre que possível, a lógica de negócio, a validação, a construção de dados e a comunicação com serviços externos foram organizadas em componentes próprios. Esta organização facilitou a manutenção, reduziu a duplicação de código e tornou mais simples compreender o papel de cada parte da aplicação.

No caso das integrações com serviços externos associados ao SAP, a solução passou por reforçar as validações antes do envio dos dados e ajustar os *payloads* de acordo com os formatos esperados pelos serviços intermédios. Sempre que os dados não eram aceites ou surgiam incoerências, era necessário analisar o problema, confirmar o comportamento esperado e articular alterações com as entidades responsáveis pelos serviços de integração. Esta experiência evidenciou a importância de contratos de integração claros e de documentação técnica detalhada.

Relativamente aos equipamentos industriais, a validação foi reforçada através da combinação entre verificações lógicas e validações físicas. A leitura de códigos e a utilização de balanças industriais permitiram reduzir a introdução manual de dados e confirmar determinados aspetos da operação real. Esta articulação contribuiu para aproximar o estado digital registado no sistema do estado físico observado no processo operacional.

Foram também utilizados ambientes controlados para validação das funcionalidades antes da sua disponibilização em produção. As funcionalidades foram inicialmente testadas em ambiente local e, posteriormente, validado em ambiente de *staging*. Esta prática permitiu identificar problemas antes de afetarem o ambiente produtivo e contribuiu para aumentar a confiança nas alterações realizadas.

No âmbito da situação de segurança identificada, a abordagem adotada foi prudente e não intrusiva. A análise foi limitada à observação do comportamento do serviço e à recolha de informação suficiente para enquadrar o problema. Não foram realizadas ações de exploração, autenticação indevida ou alteração de sistemas. Esta decisão refletiu a importância de respeitar limites éticos em atividades de cibersegurança, especialmente em contexto empresarial.

8.4 Competências Técnicas

O estágio permitiu desenvolver e consolidar diversas competências técnicas. Ao nível do *backend*, houve uma evolução significativa na utilização de Laravel, nomeadamente na criação de controladores, modelos, pedidos de validação, serviços, regras de negócio e *endpoints* de API. A necessidade de implementar validações complexas e regras específicas contribuiu para uma compreensão mais aprofundada da organização de aplicações *web* empresariais.

No *frontend*, o trabalho com Vue.js permitiu desenvolver competências na criação

de componentes, formulários, tabelas, filtros e interfaces dinâmicas. Esta componente foi particularmente relevante em páginas com fluxos operacionais complexos, onde era necessário apresentar informação de forma clara, reagir a ações do utilizador e reduzir a probabilidade de erros durante a utilização.

A integração entre *frontend* e *backend* através de APIs internas permitiu compreender melhor a separação entre interface, lógica de negócio e persistência de dados. A utilização do Postman apoiou a validação de *endpoints*, a análise de respostas, a identificação de erros e a confirmação do comportamento esperado das operações antes da sua utilização através da interface.

O projeto permitiu também desenvolver competências em integração com sistemas e serviços externos. A preparação de dados, a construção de *payloads*, a análise de respostas e a adaptação a regras de validação externas foram aspetos importantes do trabalho desenvolvido. Esta experiência foi particularmente relevante por demonstrar que a integração entre sistemas exige não apenas programação, mas também coordenação, validação e entendimento dos processos envolvidos.

A utilização de tecnologias associadas ao ambiente industrial constituiu outra aprendizagem relevante. A leitura de códigos GS1, a utilização de leitores de códigos de barras, a integração com balanças industriais e a geração de etiquetas em ZPL permitiram compreender a ligação entre aplicações *web* e processos físicos. Esta experiência reforçou a importância da rastreabilidade e da correspondência entre dados digitais e operações reais.

No domínio da segurança, foram consolidados conhecimentos relacionados com validação de dados, controlo de acessos, proteção de *endpoints*, tratamento de erros e análise de configurações inseguras. A identificação observacional da situação enquadrada no OWASP Top 10 permitiu aplicar conceitos de segurança *web* num contexto real, reforçando a importância de boas práticas como redução da exposição de informação técnica, gestão adequada de erros e validação de ambientes antes da sua disponibilização pública.

8.5 Competências Pessoais

Para além das competências técnicas, o estágio contribuiu para o desenvolvimento de competências pessoais e profissionais. Uma das mais importantes foi a capacidade de comunicação. O trabalho exigiu contacto frequente com elementos da equipa técnica, esclarecimento de dúvidas, discussão de requisitos e apresentação de dificuldades encontradas durante o desenvolvimento.

A capacidade de adaptação foi igualmente relevante. O contexto empresarial implicou lidar com requisitos em evolução, prioridades variáveis, problemas inesperados e dependências externas. Esta realidade exigiu flexibilidade na abordagem às tarefas e capacidade de ajustar soluções à medida que surgiam novas informações ou restrições.

O estágio contribuiu também para o desenvolvimento da autonomia. Ao longo do tempo, foi necessário assumir maior responsabilidade na análise de problemas, pro-

cura de soluções, validação de funcionalidades e identificação de riscos associados às alterações realizadas. Esta autonomia foi acompanhada pela necessidade de saber pedir esclarecimentos quando os requisitos não eram suficientemente claros ou quando determinada decisão poderia ter impacto operacional.

A resolução de problemas foi uma competência constantemente exercitada. Muitas dificuldades não tinham uma solução imediata, exigindo análise do comportamento da aplicação, leitura de código existente, teste de diferentes cenários, consulta à equipa e validação progressiva das alterações. Esta experiência reforçou a importância de uma abordagem metódica e cuidadosa ao desenvolvimento.

A gestão do tempo e das prioridades também foi uma aprendizagem importante. Algumas tarefas exigiam resposta rápida, sobretudo quando estavam relacionadas com correções ou funcionalidades necessárias à operação, enquanto outras exigiam análise mais demorada e validação cuidadosa. Esta distinção permitiu compreender melhor a importância de equilibrar rapidez, qualidade e segurança.

Por fim, o estágio reforçou a consciência ética e profissional associada à área da cibersegurança. A situação de segurança identificada demonstrou a importância de atuar com prudência, respeitar limites de atuação e comunicar riscos de forma responsável. Esta aprendizagem é particularmente relevante no contexto do Mestrado em Cibersegurança e Informática Forense, onde a capacidade técnica deve ser sempre acompanhada por responsabilidade e sentido ético.

8.6 Síntese do Capítulo

A análise crítica do estágio permitiu identificar dificuldades, soluções e aprendizagens relevantes para a compreensão do trabalho desenvolvido. As principais dificuldades estiveram relacionadas com a adaptação ao contexto empresarial, a complexidade das regras de negócio, a evolução dos requisitos, a integração com serviços externos e a necessidade de garantir coerência entre processos físicos e registos digitais.

As soluções adotadas passaram pela clarificação contínua dos requisitos, centralização das validações no *backend*, separação de responsabilidades, validação em ambiente de *staging*, reforço da comunicação com a equipa técnica e adoção de uma abordagem prudente nas atividades relacionadas com segurança. Estas decisões contribuíram para melhorar a fiabilidade das funcionalidades implementadas e reduzir o risco de inconsistências.

De forma global, o estágio permitiu consolidar competências técnicas em desenvolvimento *web*, integração de sistemas, validação de dados, equipamentos industriais e segurança aplicacional. Simultaneamente, reforçou competências pessoais como comunicação, autonomia, adaptação, resolução de problemas, gestão de prioridades e responsabilidade ética. Estas aprendizagens contribuíram para uma compreensão mais completa da relação entre desenvolvimento de *software*, operação industrial e cibersegurança em contexto empresarial.

9 Conclusão

O relatório apresenta o trabalho desenvolvido no âmbito do estágio curricular realizado na ValFinance, empresa integrada no grupo ValGrupo, com especial enfoque no projeto ValSabor. Embora o estágio tenha incluído diferentes atividades de desenvolvimento e tarefas complementares relacionadas com segurança da informação, o relatório centrou-se no sistema de gestão de unidades logísticas, por este representar o projeto com maior complexidade técnica, impacto operacional e relação direta com os temas do Mestrado em Cibersegurança e Informática Forense.

Ao longo do trabalho foi possível analisar um contexto real de desenvolvimento de *software* empresarial, no qual a aplicação desenvolvida não se limitava à disponibilização de interfaces, mas suportava processos industriais e logísticos diretamente associados a operações físicas. Neste contexto, a leitura de códigos GS1, a utilização de leitores de códigos de barras, a integração com balanças industriais, a geração de etiquetas em ZPL, a validação de dados, o controlo de acessos e a comunicação com serviços externos associados ao SAP assumiram um papel essencial na fiabilidade global do sistema.

9.1 Síntese do Trabalho Desenvolvido

O trabalho desenvolvido incidiu principalmente sobre a implementação e evolução de funcionalidades *full stack*, recorrendo a Laravel no *backend* e Vue.js no *frontend*. No *backend* foram implementadas regras de negócio, validações, serviços, controladores e mecanismos de integração. No *frontend* foram desenvolvidas interfaces orientadas aos fluxos operacionais, permitindo aos utilizadores executar tarefas como criação de caixas, agrupamento de caixas, montagem de unidades logísticas, leitura de códigos e validação de operações.

O projeto ValSabor destacou-se pela necessidade de articular desenvolvimento aplicacional com processos industriais concretos. As funcionalidades implementadas tinham de assegurar que os dados registados no sistema correspondiam às operações realizadas no terreno, nomeadamente através da associação entre artigos, lotes, cai-

xas, agrupamentos, paletes e unidades logísticas. Esta ligação entre o mundo físico e o registo digital exigiu especial atenção à validação, à Integridade dos dados e à rastreabilidade das operações.

Para além do desenvolvimento funcional, foram considerados aspetos de segurança aplicacional ao longo do processo. Entre estes destacam-se a centralização das validações críticas no *backend*, o controlo de acessos com base em permissões, a proteção de operações sensíveis, a prevenção de duplicações e o tratamento adequado de erros. Estas decisões permitiram reduzir a dependência de validações realizadas apenas no *frontend* e reforçar a coerência das operações executadas.

O estágio incluiu ainda atividades complementares no domínio da cibersegurança, nomeadamente a identificação observacional de uma situação de exposição indevida de informação técnica num serviço *web* e a realização de uma campanha controlada de *phishing*. Estas atividades contribuíram para relacionar o desenvolvimento de *software* com preocupações mais amplas de segurança da informação em contexto empresarial.

9.2 Avaliação do Cumprimento dos Objetivos

De forma geral, considera-se que os objetivos definidos para o trabalho foram cumpridos. O estágio permitiu a integração em ambiente profissional, a participação em tarefas reais de desenvolvimento e o contacto com sistemas internos utilizados em contexto empresarial. Em particular, o projeto ValSabor permitiu aplicar conhecimentos de desenvolvimento *web*, integração entre sistemas, validação de dados, controlo de acessos e segurança aplicacional.

A correspondência entre os objetivos definidos na Secção 1.2, as principais evidências apresentadas no relatório e o respetivo grau de cumprimento encontra-se sistematizada na Tabela E.1, apresentada no Apêndice E.

O objetivo de compreender e desenvolver funcionalidades associadas a processos logísticos e industriais foi atingido através da participação no desenvolvimento de módulos relacionados com criação de caixas, agrupamento, montagem de unidades logísticas, controlo de peso, geração de etiquetas e integração com serviços externos associados ao SAP. Estas funcionalidades exigiram a tradução de regras operacionais em lógica aplicacional, reforçando a importância da análise de requisitos e da validação contínua.

O objetivo de integrar preocupações de segurança no desenvolvimento foi igualmente alcançado. Embora o projeto não tenha correspondido a um processo formal completo de DevSecOps, foram aplicadas práticas compatíveis com uma abordagem de desenvolvimento seguro, nomeadamente validação no *backend*, controlo de permissões, proteção de *endpoints* e tratamento controlado de erros. Estas práticas demonstraram que a segurança deve ser considerada como parte integrante do desenvolvimento e não apenas como uma verificação posterior.

O objetivo de analisar criticamente o trabalho desenvolvido também foi cumprido, uma vez que o relatório não se limita a descrever funcionalidades implementadas. Pelo

contrário, procura enquadrar as decisões técnicas adotadas, relacionando-as com problemas concretos de consistência, integração, rastreabilidade e segurança em aplicações empresariais.

9.3 Contributos do Trabalho

O principal contributo do trabalho consistiu na participação no desenvolvimento de um sistema interno com impacto direto nos processos operacionais da ValSabor. O sistema contribuiu para a digitalização e estruturação de operações logísticas e industriais, permitindo maior controlo sobre a criação, associação, validação e movimentação de unidades logísticas.

Um contributo relevante foi a implementação de mecanismos de validação e controlo que ajudam a reduzir erros operacionais. A validação de códigos, pesos, estados, permissões e regras de negócio permitiu reforçar a coerência entre os dados registados e as operações realizadas. Esta dimensão foi particularmente importante num contexto em que os dados processados pelo sistema podiam posteriormente ser comunicados a sistemas empresariais externos.

Outro contributo prende-se com a rastreabilidade. O registo das operações e a associação entre diferentes entidades, como artigos, lotes, caixas, agrupamentos e unidades logísticas, permitiram criar uma base para reconstruir o percurso das operações e apoiar processos de análise, auditoria ou correção de anomalias.

Do ponto de vista da segurança, o trabalho contribuiu para evidenciar a relação entre desenvolvimento de *software* e proteção da informação. A centralização da lógica crítica no *backend*, o controlo de acessos, a validação dos dados e o tratamento adequado de erros demonstram que decisões de implementação têm impacto direto na segurança e fiabilidade de aplicações empresariais.

Por fim, o estágio contribuiu para o desenvolvimento de competências técnicas e profissionais. A nível técnico, permitiu aprofundar conhecimentos em Laravel, Vue.js, APIs, integração com serviços externos, validação de dados e utilização de equipamentos em contexto industrial. A nível profissional, reforçou competências de comunicação, adaptação, análise de problemas, organização do trabalho e colaboração em equipa.

9.4 Limitações

Apesar dos resultados alcançados, o trabalho desenvolvido apresenta algumas limitações. A primeira prende-se com o processo de análise de requisitos. Os requisitos foram transmitidos maioritariamente pela equipa técnica e pelo responsável da área de informática, não tendo existido contacto direto sistemático com todos os utilizadores finais. Esta abordagem foi adequada ao contexto do estágio, mas limitou a recolha direta de perceções, dificuldades e necessidades específicas dos operadores.

Outra limitação está relacionada com a evolução progressiva dos requisitos. Muitas regras de negócio, exceções e validações foram identificadas ao longo do desenvolvimento, à medida que as funcionalidades eram testadas e confrontadas com situações reais. Embora esta abordagem tenha permitido adaptar o sistema às necessidades da operação, também aumentou a complexidade da implementação e exigiu revisões sucessivas de algumas decisões técnicas.

Uma limitação relevante esteve ainda associada às integrações com serviços externos. Como a comunicação com o SAP não era realizada diretamente pela aplicação, mas sim através de serviços intermédios, foi necessário articular sucessivamente alterações e ajustes para garantir que os dados enviados eram aceites e processados corretamente. Esta dependência introduziu complexidade adicional no desenvolvimento, uma vez que determinados problemas não podiam ser resolvidos apenas ao nível da aplicação interna, exigindo validação conjunta, adaptação dos *payloads* e comunicação contínua com as entidades responsáveis pelos serviços de integração. Esta situação evidenciou a importância de uma definição clara dos contratos de integração, dos formatos esperados e das regras de validação aplicadas por cada sistema envolvido.

A validação das funcionalidades foi realizada sobretudo através de testes funcionais, testes em ambiente local, validação em *staging* e revisão pela equipa técnica. Apesar de esta abordagem ter permitido detetar e corrigir vários problemas, não corresponde a uma estratégia completa de testes automatizados, testes de carga ou validação formal de segurança aplicacional.

Importa ainda referir que o trabalho desenvolvido incidiu sobre a aplicação interna e a sua comunicação com serviços externos associados ao SAP, não abrangendo a configuração interna desse sistema. Esta delimitação é relevante, uma vez que o foco do projeto esteve na preparação, validação e envio de dados a partir da aplicação operacional. Assim, algumas dificuldades de integração dependeram também do comportamento e das validações aplicadas pelos serviços externos, o que exigiu articulação com as entidades responsáveis por esses serviços.

9.5 Trabalho Futuro

Como trabalho futuro, uma das principais linhas de evolução consiste no reforço dos mecanismos de teste. A introdução gradual de testes automatizados, nomeadamente testes unitários, testes de integração e testes funcionais sobre fluxos críticos, permitiria aumentar a confiança nas alterações realizadas e reduzir o risco de regressões em funcionalidades sensíveis.

Outra possibilidade de evolução passa pelo reforço da monitorização e análise de erros. A estruturação de *logs*, alertas e mecanismos de acompanhamento das integrações permitiria detetar mais rapidamente falhas de comunicação, inconsistências ou operações não concluídas, facilitando o diagnóstico e a correção de problemas.

No domínio das integrações, seria igualmente relevante formalizar de forma mais detalhada os contratos de comunicação com os serviços externos associados ao SAP,

incluindo estrutura dos *payloads*, regras de validação, códigos de erro, exemplos de pedidos válidos e procedimentos de diagnóstico. Esta documentação permitiria reduzir ambiguidades, acelerar a resolução de problemas e facilitar futuras evoluções da integração.

Ainda no domínio da integração com sistemas externos, uma evolução relevante consistiria na consulta automática do *stock* disponível em SAP para os artigos e lotes utilizados nos processos de produção. Esta funcionalidade permitiria apresentar no *frontend* a quantidade disponível em *stock*, nomeadamente em quilogramas, e validar no *backend* se existe *stock* suficiente antes da criação de novas caixas associadas a determinado lote. Desta forma, seria possível impedir a execução de operações sobre lotes sem disponibilidade, reduzindo o risco de inconsistências entre o sistema operacional e o sistema empresarial externo. Esta melhoria reforçaria a integração entre sistemas, a Integridade dos dados e a fiabilidade dos processos de produção.

A nível funcional, o sistema poderá evoluir com novos mecanismos de apoio à auditoria e consulta histórica das operações. A criação de interfaces mais orientadas à análise de rastreabilidade, permitindo consultar o percurso de uma caixa, agrupamento, palete ou unidade logística, poderia reforçar a capacidade de controlo e análise por parte da organização.

No domínio da segurança, seria relevante aprofundar práticas de desenvolvimento seguro, incluindo revisões sistemáticas de permissões, testes de segurança aplicacional, análise de dependências e validação recorrente dos principais *endpoints*. Estas práticas poderiam aproximar progressivamente o processo de desenvolvimento de uma abordagem mais estruturada de segurança ao longo do ciclo de vida do *software*.

O caso de estudo ilustra que o desenvolvimento seguro de aplicações industriais em contexto empresarial pode ser reforçado através da integração progressiva de controlos nas próprias funcionalidades do sistema, mesmo sem a adoção de um processo formal completo de DevSecOps. A validação no *backend*, o controlo de acessos, o tratamento de erros e a separação de responsabilidades constituem práticas aplicáveis a diferentes sistemas empresariais, encontrando-se alinhadas com áreas de verificação do OWASP ASVS na versão 5.0.0 e com os princípios de desenvolvimento seguro apresentados pelo NIST SSDF (OWASP Foundation, 2025; Souppaya et al., 2022).

Alguns dos padrões de implementação observados podem ser generalizados para outras aplicações que interajam com processos físicos e serviços externos. Entre estes destacam-se a validação dos dados antes da persistência ou integração, o controlo dos estados das entidades, o registo das respostas externas e a associação entre identificadores físicos e registos digitais. Estes mecanismos podem contribuir para reduzir inconsistências e reforçar a capacidade de reconstrução do histórico das operações, aspeto particularmente relevante em sistemas que exigem rastreabilidade (International Organization for Standardization, 2007; Zhou et al., 2022).

Por fim, o trabalho futuro poderá incluir uma maior formalização do processo de requisitos, com recolha direta junto de utilizadores finais, validação documentada de fluxos operacionais e sistematização das regras de negócio. Esta evolução permitiria

reduzir ambiguidades, melhorar a documentação funcional e facilitar a manutenção futura do sistema.

Página intencionalmente deixada em branco.

Bibliography

Alqahtani, Sultan S. (jun. de 2025). «Beyond the code: analyzing OSS developers security awareness and practices». Em: *International Journal of Information Security* 24.3, p. 109. DOI: 10.1007/s10207-025-01023-1. URL: https://link.springer.com/10.1007/s10207-025-01023-1.

Banks, Andrew et al. (mar. de 2019). *MQTT Version 5.0*. OASIS Standard. OASIS Open. URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

Butarbutar, Zenfrison Tuah et al. (2023). «Systematic literature review of Critical success factors on enterprise resource planning post implementation». Em: *Cogent Business & Management* 10.3, p. 2264001. DOI: 10.1080/23311975.2023.2264001. URL: https://doi.org/10.1080/23311975.2023.2264001.

European Union Agency for Cybersecurity (2021). *Raising awareness of cybersecurity: a key element of national cybersecurity strategies*. en. LU: Publications Office. DOI: 10.2824/363629. URL: https://data.europa.eu/doi/10.2824/363629.

Fielding, Roy T., Mark Nottingham e Julian Reschke (jun. de 2022). *HTTP Semantics*. Rel. téc. RFC 9110. RFC Editor. DOI: 10.17487/RFC9110. URL: https://www.rfc-editor.org/rfc/rfc9110.html.

Gordon, William J. et al. (mar. de 2019). «Assessment of Employee Susceptibility to Phishing Attacks at US Health Care Institutions». en. Em: *JAMA Network Open* 2.3, e190393. ISSN: 2574-3805. DOI: 10.1001/jamanetworkopen.2019.0393. URL: http://jamanetworkopen.jamanetwork.com/article.aspx?doi=10.1001/jamanetworkopen.2019.0393.

GS1 (2019). *GS1 Logistic Label Guideline*. Release 1.3, ratified Jul 2019. URL: https://www.gs1.org/standards/gs1-logistic-label-guideline/current-standard.

GS1 (2025). *GS1 General Specifications*. Versão 25.0. URL: https://ref.gs1.org/standards/genspecs/.

Hansche, Susan et al. (set. de 2024). *Building a cybersecurity and privacy learning program*. Rel. téc. NIST SP 800-50r1. National Institute of Standards e Technology (U.S.), NIST SP 800–50r1. DOI: 10.6028/NIST.SP.800-50r1. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-50r1.pdf.

International Organization for Standardization (2007). *ISO 22005:2007 – Traceability in the feed and food chain – General principles and basic requirements for system design and implementation*. Reviewed and confirmed in 2022. URL: https://www.iso.org/standard/36297.html.

McCarthy, Jim et al. (fev. de 2022). *Securing the industrial internet of things : cybersecurity for distributed energy resources*. Rel. téc. NIST SP 1800-32. National Institute of Standards e Technology (U.S.), NIST SP 1800–32. DOI: 10.6028/NIST.SP.1800-32. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-32.pdf (acedido em 2026-06-06).

Mohammed, Khwaja, Bharanidharan Shanmugam e Jamal El-Den (25 de nov. de 2025). «Evolution of DevSecOps and Its Influence on Application Security: A Systematic Literature Review». Em: *Technologies* 13.12, p. 548. DOI: 10.3390/technologies13120548. URL: https://www.mdpi.com/2227-7080/13/12/548.

National Institute of Standards and Technology (fev. de 2024). *The NIST Cybersecurity Framework (CSF) 2.0*. Rel. téc. NIST CSWP 29. Gaithersburg, MD: National Institute of Standards e Technology, NIST CSWP 29. DOI: 10.6028/NIST.CSWP.29. URL: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf.

OWASP Foundation (2021). *OWASP Top 10:2021 – The Ten Most Critical Web Application Security Risks*. URL: https://owasp.org/www-project-top-ten/.

OWASP Foundation (2023). *OWASP API Security Top 10 2023*. URL: https://owasp.org/API-Security/editions/2023/en/0x00-header/.

OWASP Foundation (2025). *OWASP Application Security Verification Standard 5.0.0*. URL: https://owasp.org/www-project-application-security-verification-standard/.

Qadir, Sana et al. (30 de abr. de 2025). «Comparative evaluation of approaches & tools for effective security testing of Web applications». Em: *PeerJ Computer Science* 11, e2821. DOI: 10.7717/peerj-cs.2821. URL: https://peerj.com/articles/cs-2821.

Rajapakse, Roshan N. et al. (jan. de 2022). «Challenges and solutions when adopting DevSecOps: A systematic review». Em: *Information and Software Technology* 141, p. 106700. DOI: 10.1016/j.infsof.2021.106700. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950584921001543.

Sirawongphatsara, Patsita et al. (2024). *Comparative Simulation of Phishing Attacks on a Critical Information Infrastructure Organization: An Empirical Study*. URL: https://arxiv.org/abs/2410.20728.

Souppaya, Murugiah, Karen Scarfone e Donna Dodson (fev. de 2022). *Secure Software Development Framework (SSDF) version 1.1 : recommendations for mitigating the risk of software vulnerabilities*. Rel. téc. NIST SP 800-218. Gaithersburg, MD: National Institute of Standards e Technology (U.S.), NIST SP 800–218. DOI: 10.6028/NIST.SP.800-218.

URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf.

Wen, Shao-Fang e Basel Katt (dez. de 2023). «A quantitative security evaluation and analysis model for web applications based on OWASP application security verification standard». Em: *Computers & Security* 135, p. 103532. DOI: 10.1016/j.cose.2023.103532. URL: https://linkinghub.elsevier.com/retrieve/pii/S016740482300442X.

Xu, Ke et al. (ago. de 2025). «DRacv: Detecting and auto-repairing vulnerabilities in role-based access control in web application». Em: *Journal of Network and Computer Applications* 240, p. 104191. DOI: 10.1016/j.jnca.2025.104191. URL: https://linkinghub.elsevier.com/retrieve/pii/S1084804525000888.

Zhou, Xiongyong e Zhiduan Xu (2022). «Traceability in food supply chains: a systematic literature review and future research directions». Em: *International Food and Agribusiness Management Review* 25.2. DOI: 10.22004/ag.econ.320212. URL: https://ideas.repec.org/a/ags/ifaamr/320212.html.

Página intencionalmente deixada em branco.

Apêndices

Página intencionalmente deixada em branco.

A

Tecnologias e Ferramentas Utilizadas

Este apêndice apresenta a descrição detalhada das principais tecnologias e ferramentas utilizadas no projeto ValSabor, complementando a síntese apresentada na Secção 3.5. A informação encontra-se organizada por tecnologia ou ferramenta, respetiva utilização no projeto e relevância para o desenvolvimento da solução.

Tabela A.1: *Descrição detalhada das tecnologias e ferramentas utilizadas no projeto ValSabor*

Tecnologia/Ferramenta	**Utilização no projeto**	**Relevância**
Laravel	Desenvolvimento do *backend* da aplicação, incluindo controladores, modelos, pedidos de validação, serviços, regras de negócio e *endpoints* de API.	Permitiu centralizar a lógica crítica no servidor, aplicar validações consistentes e controlar operações sensíveis antes da persistência ou integração dos dados.
Vue.js	Desenvolvimento do *frontend*, incluindo páginas, componentes, formulários, filtros, tabelas e interfaces dinâmicas de interação com os utilizadores.	Permitiu construir interfaces adaptadas aos fluxos operacionais, facilitando a utilização da aplicação em contexto industrial.
APIs internas	Comunicação entre *frontend* e *backend*, bem como suporte a operações de consulta, criação, validação, atualização e integração de dados.	Asseguraram a separação entre interface e lógica de negócio, permitindo que as validações críticas fossem executadas no *backend*.

Continua na página seguinte

Tabela continuada da página anterior

Tecnologia/Ferramenta	Utilização no projeto	Relevância
Serviços externos associados ao SAP	Integração de informação operacional com sistemas empresariais externos, nomeadamente em operações logísticas e de movimentação de unidades.	Permitiram refletir operações realizadas no sistema interno em plataformas externas de gestão, exigindo validação prévia e controlo da consistência dos dados enviados.
GS1	Leitura e interpretação de códigos de barras associados a artigos, lotes, caixas e unidades logísticas.	Contribuiu para a identificação normalizada de entidades logísticas e para a correspondência entre objetos físicos e registos digitais.
ZPL	Criação de etiquetas destinadas a impressoras Zebra, contendo informação associada a caixas, agrupamentos, paletes ou unidades logísticas.	Permitiu automatizar a emissão de etiquetas, reduzindo introdução manual de dados e facilitando a identificação física das unidades.
Scanners de códigos de barras	Dispositivos utilizados para leitura automática de códigos associados a produtos, caixas, agrupamentos ou unidades logísticas.	Reduziram erros de introdução manual e aceleraram a execução dos processos operacionais no ambiente produtivo e logístico.
Balanças industriais	Equipamentos utilizados para obter informação de peso associada a operações e unidades logísticas.	Permitiram complementar as validações lógicas com validação física, reforçando a coerência entre o estado registado no sistema e a operação real.
Git / controlo de versões	Gestão do código-fonte, organização das alterações, criação de ramificações por tarefa e acompanhamento da evolução do desenvolvimento.	Facilitou a colaboração entre elementos da equipa, a rastreabilidade das alterações e a gestão de diferentes versões do código.
Ambiente local de desenvolvimento	Implementação e validação inicial das funcionalidades em contexto isolado.	Permitiu testar alterações sem impacto direto nos ambientes partilhados ou produtivos.
staging / ambiente de testes	Ambiente utilizado para validação de funcionalidades antes da sua passagem para produção.	Permitiu testar regras de negócio, integrações e fluxos operacionais em condições próximas do ambiente real, reduzindo o risco de erros em produção.

Continua na página seguinte

Tabela continuada da página anterior

Tecnologia/Ferramenta	**Utilização no projeto**	**Relevância**
Postman	Teste de pedidos a *endpoints* de API, verificação de respostas, validações, erros e comportamento das operações durante o desenvolvimento.	Apoiou a deteção de erros, a validação das integrações e a confirmação do comportamento esperado das operações antes da sua utilização através da interface.

B

Requisitos Detalhados do Projeto ValSabor

Este apêndice apresenta, na Tabela B.1, uma sistematização retrospetiva dos principais requisitos identificados a partir das funcionalidades desenvolvidas, das regras transmitidas pela equipa técnica e das necessidades operacionais do projeto ValSabor. A tabela não corresponde a uma especificação formal utilizada durante o desenvolvimento, mas procura organizar os requisitos mais relevantes para a compreensão do sistema e do caso de estudo.

Tabela B.1: *Requisitos detalhados do projeto ValSabor*

ID	Tipo	Requisito	Finalidade
R01	Funcional	O sistema deve permitir a leitura de códigos associados à norma GS1.	Identificar artigos, lotes, caixas ou unidades logísticas de forma automática.
R02	Funcional	O sistema deve permitir a criação de caixas a partir dos dados lidos e das regras de negócio definidas.	Registar unidades de produto e iniciar a rastreabilidade do processo.
R03	Funcional	O sistema deve permitir o agrupamento de caixas compatíveis.	Organizar unidades intermédias e preparar fases posteriores do processo logístico.
R04	Funcional	O sistema deve permitir o fecho de paletes intermédias apenas quando as condições necessárias forem cumpridas.	Garantir que uma palete intermédia só é finalizada quando os dados associados estão coerentes.

Continua na página seguinte

Tabela continuada da página anterior

ID	Tipo	Requisito	Finalidade
R05	Funcional	O sistema deve permitir a montagem de unidades logísticas, como paletes ou grades, a partir de caixas ou agrupamentos.	Consolidar unidades físicas de transporte e associá-las ao respetivo registo digital.
R06	Operacional	O sistema deve suportar a utilização de leitores de códigos de barras em ambiente industrial.	Acelerar a leitura de códigos e reduzir erros associados à introdução manual de dados.
R07	Operacional	O sistema deve integrar dados provenientes de balanças industriais.	Permitir validação física do peso e reforçar a coerência entre operação real e informação registada.
R08	Funcional	O sistema deve gerar etiquetas em ZPL para impressão em equipamentos Zebra.	Identificar fisicamente caixas, agrupamentos, paletes ou unidades logísticas.
R09	Funcional / Integração	O sistema deve preparar e enviar dados para integração com serviços externos associados a SAP.	Refletir operações logísticas ou produtivas em sistemas empresariais externos.
R10	Integridade	O sistema deve validar os dados no *backend* antes da sua persistência ou integração.	Evitar que dados inválidos, incompletos ou incoerentes sejam processados.
R11	Integridade	O sistema deve impedir a duplicação de operações críticas, como integrações repetidas ou associações indevidas de unidades.	Reduzir o risco de registos duplicados e inconsistências entre sistemas.
R12	Segurança	O sistema deve aplicar controlo de acessos com base em perfis e permissões.	Garantir que apenas utilizadores autorizados executam operações críticas.
R13	Segurança	O sistema deve proteger *endpoints* e operações sensíveis através de validações de autenticação e autorização.	Reduzir o risco de acesso indevido a funcionalidades ou dados restritos.
R14	Integridade	O sistema deve manter registos que permitam acompanhar o histórico das operações realizadas.	Assegurar rastreabilidade, apoio à análise de anomalias e capacidade de auditoria.

Continua na página seguinte

Tabela continuada da página anterior

ID	Tipo	Requisito	Finalidade
R15	Operacional	O sistema deve ser utilizável em contexto industrial, considerando rapidez de operação, leitura por scanner e interação simples.	Adequar a aplicação ao ambiente real de utilização e reduzir obstáculos à operação.
R16	Não funcional	As funcionalidades devem ser validadas em ambiente de *staging* antes da disponibilização em produção.	Reduzir o risco de erros em ambiente produtivo e validar alterações em contexto controlado.
R17	Não funcional	O sistema deve apresentar mensagens de erro compreensíveis para o utilizador e registar informação técnica relevante para análise interna.	Apoiar a resolução de problemas sem expor informação técnica desnecessária ao utilizador final.
R18	Integridade	O sistema deve controlar estados das entidades, impedindo operações fora da sequência prevista.	Garantir coerência no ciclo de vida de caixas, agrupamentos, paletes e pedidos operacionais.

C

Excertos de Código do Caso de Estudo

Este apêndice apresenta excertos anonimizados de código que complementam os mecanismos de validação, controlo de acessos e integração com serviços externos descritos no Capítulo 6. Os excertos foram simplificados e tiveram identificadores internos alterados, de forma a demonstrar as práticas adotadas sem expor informação sensível da organização.

C.1 Validação Estrutural dos Dados

O Código 2 apresenta algumas das regras de validação aplicadas aos dados recebidos antes da criação de uma caixa. Estas regras permitem verificar campos obrigatórios, formatos, referências a entidades existentes e restrições básicas de integridade.

```
1  public function rules()
2  {
3      return [
4          'item' => 'required|exists:items,id',
5          'sscc' => 'nullable|string|size:18|unique:transport_unit_lines,sscc',
6          'batchCode' => 'required|string',
7          'slaughterDate' => 'required|date',
8          'productionDate' => 'required|date|after_or_equal:slaughterDate',
9          'grossWeight' => 'required|numeric|gt:0',
10         'netWeight' => 'required|numeric|gt:0',
11         'owner' => 'required|integer|exists:owners,id',
12         'warehouseRoom' => 'required|integer|exists:warehouse_rooms,id',
13     ];
14 }
```

Listagem 2: *Excerto das regras de validação aplicadas à criação de caixas*

Neste excerto, o Laravel rejeita automaticamente pedidos com campos obrigatórios

ausentes, referências inexistentes, formatos inválidos ou valores que não respeitem as restrições definidas, impedindo que estes dados avancem para a lógica de negócio.

C.2 Controlo de Permissões e Regras de Negócio

O Código 3 exemplifica algumas das verificações realizadas após a validação estrutural, incluindo o controlo de acesso às entidades selecionadas e a validação do peso segundo a ficha técnica do artigo.

```
protected function passedValidation()
{
    $user = Auth::user();

    if (!PermissionTrait::filterPermissionByEntities(
        Owner::find($this->owner),
        Owner::class,
        User::class,
        $user->id,
        false
    )) {
        Utils::throwHttpResponseException([
            'error' => 'Entidade inválida.'
        ]);
    }

    $specification = PalletItemSpecification::firstWhere(
        'id_item',
        $this->item
    );

    if (isset($specification->minimumWeight)
        && $this->netWeight < $specification->minimumWeight) {
        Utils::throwHttpResponseException([
            'error' => 'Peso inferior ao mínimo.'
        ]);
    }

    if (isset($specification->maximumWeight)
        && $this->netWeight > $specification->maximumWeight) {
        Utils::throwHttpResponseException([
            'error' => 'Peso superior ao máximo.'
        ]);
    }
}
```

Listagem 3: *Excerto do controlo de permissões e validação de regras de negócio*

Caso o utilizador não possua acesso à entidade selecionada ou o peso não respeite os limites definidos, a operação é interrompida antes da persistência dos dados.

C.3 Orquestração da Integração Externa

O Código 4 demonstra a sequência utilizada durante uma integração externa: obtenção da autenticação, validação dos dados, construção do *payload*, envio para o serviço externo e registo de eventuais erros.

```
public function transferBetweenWarehouses($order, $request)
{
    $log = '[Transfer] - ' . $order->id . ' -';
    Log::channel(self::LOG_CHANNEL)->info("$log Integration Started");
    try {
        $accessToken = $this->client->getValidAuth(self::LOG_CHANNEL);

        $arguments = [
            'transportDate' => $request->transportDate,
            'planning' => $request->planning,
        ];

        $this->validator->transfer(
            $order,
            self::LOG_CHANNEL,
            needsMeanOfDispatch: true,
            extraArgs: $arguments
        );

        $payload = $this->builder->transferBetweenWarehouses(
            $accessToken,
            $order,
            $arguments
        );

        return $this->client->post(
            '/external-endpoint',
            $payload,
            self::LOG_TITLE,
            self::LOG_CHANNEL
        );
    } catch (Exception $exception) {
        Log::channel(self::LOG_CHANNEL)->error(
            $log . ' Integration failed - '
            . $exception->getMessage()
        );

        throw $exception;
    }
}
```

Listagem 4: *Excerto da orquestração de uma integração externa*

A sequência apresentada garante que o *payload* apenas é construído e enviado após a conclusão das validações. A separação entre componentes permite ainda isolar responsabilidades e registar eventuais falhas para posterior diagnóstico.

C.4 Construção e Agregação do *Payload*

O Código 5 apresenta um excerto anonimizado da agregação das linhas por artigo, lote e armazém de origem antes da construção do *payload* enviado ao serviço externo.

```
$lines = $transportUnits->flatMap(function ($transportUnit) {
        return $transportUnit->itemLines->map(
            function ($box) {
                return [
                    'itemCode' => $box->item->code,
                    'batch' => $box->batch->code,
                    'quantity' => $box->fixedWeight
                        ?? $box->netWeight,
                    'originWarehouse' =>
                        $box->originWarehouse->externalCode,
                ];
            }
        );
    })->groupBy(function ($line) {
        return $line['itemCode']
            . '|' . $line['batch']
            . '|' . $line['originWarehouse'];
    })->map(function ($lines) {
        return [
            'itemCode' => $lines->first()['itemCode'],
            'batch' => $lines->first()['batch'],
            'originWarehouse' =>
                $lines->first()['originWarehouse'],
            'quantity' => $lines->sum('quantity'),
        ];
    })->values()->toArray();
```

Listagem 5: *Excerto da agregação dos dados enviados ao serviço externo*

Esta transformação permite converter os registos internos, organizados ao nível das caixas e unidades de transporte, numa estrutura agregada adequada ao serviço externo, reduzindo redundâncias e assegurando a consistência das quantidades enviadas.

C.5 Validação e Integração de Produções

O Código 6 exemplifica algumas das verificações executadas antes da construção dos *payloads*, incluindo a existência da entidade, do identificador, da empresa associada, da

configuração do armazém e dos dados de produção necessários.

```
public function production($data, $channel)
{
    $this->handleError(
        !isset($data['entity']),
        $channel,
        'Entity is null',
        'Sem entidade.'
    );

    $this->handleError(
        !isset($data['identifier']),
        $channel,
        'Identifier not found',
        'Identificador não encontrado.'
    );

    $entity = $data['entity'];
    $header = $this->resolveHeader($entity);

    $this->handleError(
        !isset($header->owner?->company),
        $channel,
        'Owner or company not found',
        'Dono ou empresa não encontrado.'
    );

    $this->handleError(
        $header->warehouseRoom->warehouse
            ->integratorSettingLines->isEmpty(),
        $channel,
        'Warehouse ERP code not found',
        'Código do armazém não encontrado.'
    );

    $this->validateProductionLines($entity, $channel);
}
```

Listagem 6: *Excerto anonimizado das validações anteriores à integração da produção*

Estas verificações interrompem o processo quando faltam dados essenciais à integração. Apenas após a sua conclusão são construídos e enviados os respetivos *payloads*.

O Código 7 apresenta um excerto anonimizado do fluxo utilizado na integração das produções. A validação é executada antes da construção dos *payloads*, impedindo que dados incompletos ou inconsistentes sejam enviados para o serviço externo.

Este fluxo garante que a informação de produção apenas é estruturada e enviada ao serviço externo após a validação dos dados necessários. A separação entre validação, construção dos *payloads* e comunicação externa permite reduzir o risco de integração de operações incompletas ou inconsistentes.

```php
public function process($isFinished, $entity)
{
    $identifier = $this->getIdentifier($entity)
    $log = '[Process Production Order] - ' . $identifier . ' -';
    Log::channel(self::LOG_CHANNEL)->info("$log Integration started");
    $accessToken = $this->client->getValidAuth(self::LOG_CHANNEL);

    try {
        $data = [
            'isFinished' => $isFinished,
            'entity' => $entity,
            'identifier' => $identifier,
        ];

        $this->validator->production(
            $data,
            self::LOG_CHANNEL
        );

        $payloads = $this->builder->production(
            $data,
            $this->client->getValidAuth(self::LOG_CHANNEL)
        );

        $response = [];

        foreach ($payloads as $payload) {
            $response[] = $this->client->post(
                '/external-endpoint',
                $payload,
                self::LOG_TITLE,
                self::LOG_CHANNEL
            );
        }

        return $response;
    } catch (Exception $e) {
        Log::channel(self::LOG_CHANNEL)->error("$log Integration failed - " .
        ↪  $e->getMessage());
        throw $e;
    }
}
```

Listagem 7: *Excerto da validação e integração de produções*

C.6 Validação da Associação de Caixas

O Código 8 apresenta um excerto anonimizado das verificações executadas antes da associação de caixas a uma unidade logística.

```
1
2  $code = $this->code;
3  [$isTransportUnit, $isSSCC, $isGroup] = $this->resolveCodeType($code);
4
5  $query = TransportUnitLine::query();
6  if ($isSSCC)
7      $query->where('sscc', $code);
8
9  if ($isGroup)
10     $query->whereNull('id_transport_unit')
11         ->whereHas('group', fn($query) => $query->where('code', $code));
12
13 if ((clone $query)->whereHas('transportUnit')->exists())
14     Utils::throwHttpResponseException(['error' => 'Caixa já utilizada.']);
15
16 $lines = $isSSCC ? collect([$query->first()])->filter() : $query->get();
17 if ($lines->isEmpty())
18     Utils::throwHttpResponseException(['error' => 'Caixa não encontrada.']);
19
20 $header = $lines->first()->group->transportUnitLineGroupHeader;
21 if ($header->status != Header::FINISHED)
22     Utils::throwHttpResponseException(['error' => 'Caixa em preparação.']);
23
24 if ($header->idOwner != $this->owner)
25     Utils::throwHttpResponseException(['error' => 'Dono inválido.']);
26
27 if ($transportUnit->status != TransportUnit::ASSEMBLING)
28     Utils::throwHttpResponseException(['error' => 'Palete já concluída.']);
29
30 $totalBoxes = $transportUnit->numberBoxes + $lines->count();
31 $this->checkBoxLimit($totalBoxes, $specification->numberBoxes);
32 $this->checkItemConsistency($transportUnit, $lines);
```

Listagem 8: *Excerto das validações aplicadas à associação de caixas*

O excerto demonstra que, antes da associação, o sistema verifica se a unidade logística permanece em montagem, se o número máximo de caixas é respeitado e se os artigos são compatíveis. Apenas após estas validações são atualizados o peso e a associação das caixas.

D

Correspondência Indicativa com o OWASP ASVS

A Tabela D.1 apresenta uma correspondência indicativa entre os mecanismos implementados e as áreas de controlo do OWASP ASVS na versão 5.0.0 (OWASP Foundation, 2025). O mapeamento foi realizado retrospetivamente e não representa uma avaliação formal de conformidade.

Tabela D.1: *Correspondência indicativa entre os mecanismos implementados e o OWASP ASVS*

Mecanismo implementado	**Aplicação no sistema**	**Área do OWASP ASVS**
Validação no *backend*	Validação dos dados recebidos através de *Form Requests* e aplicação de regras de negócio antes da persistência.	Validação e lógica de negócio
Controlo de permissões	Verificação das permissões e das entidades às quais cada utilizador possui acesso antes da execução de operações.	Autorização
Sessões autenticadas	Verificação da sessão autenticada antes de permitir o acesso aos recursos e operações disponibilizados pelas APIs internas.	Autenticação
Controlo de estados e duplicações	Validação dos estados das entidades e verificação de integrações anteriores antes da execução de operações críticas.	Validação e lógica de negócio
Validação prévia às integrações	Verificação da consistência e completude dos dados antes da construção do *payload* e envio para os serviços externos.	Validação e lógica de negócio
Tratamento de erros e Logging	Apresentação de mensagens controladas ao utilizador e registo interno de informação técnica para diagnóstico.	*Logging* de segurança e tratamento de erros
Separação de responsabilidades	Separação entre os componentes responsáveis pela validação, construção do *payload* e comunicação externa.	Arquitetura segura

E

Avaliação do Cumprimento dos Objetivos

Este apêndice apresenta a correspondência entre os objetivos definidos na Secção 1.2, as principais evidências identificadas e o respetivo grau de cumprimento.

Tabela E.1: *Avaliação do cumprimento dos objetivos definidos*

Objetivo	Evidências de cumprimento	Avaliação
Analisar o contexto tecnológico e operacional	Caracterização do Portal, das tecnologias utilizadas e dos processos industriais e logísticos suportados pelo projeto ValSabor.	Cumprido
Identificar os principais requisitos do sistema	Sistematização retrospetiva dos requisitos funcionais, operacionais, de segurança e integridade.	Cumprido
Desenvolver e adaptar funcionalidades *backend* e *frontend*	Implementação dos fluxos de criação e agrupamento de caixas, montagem e movimentação de unidades logísticas.	Cumprido
Implementar validações e regras de negócio no *backend*	Utilização de *Form Requests*, validadores e serviços para verificar dados, estados e regras antes da persistência.	Cumprido
Assegurar o controlo de acessos e permissões	Aplicação de autenticação e verificações de autorização sobre entidades e operações críticas.	Cumprido
Apoiar a integração com serviços externos associados ao SAP	Validação prévia, construção de *payloads*, controlo de integrações anteriores e registo das respostas.	Cumprido
Contribuir para a rastreabilidade das operações	Associação entre artigos, lotes, caixas, agrupamentos, unidades logísticas e movimentos realizados.	Cumprido
Refletir criticamente sobre as decisões técnicas	Discussão das vantagens, limitações, compromissos e riscos residuais das decisões adotadas.	Cumprido

F

Comunicação com Serviços Externos

A Figura F.1 apresenta o fluxo geral de comunicação entre o Portal e os serviços externos de integração. A comunicação posterior entre estes serviços e os sistemas empresariais externos é apresentada de forma abstrata, uma vez que não foi analisada.

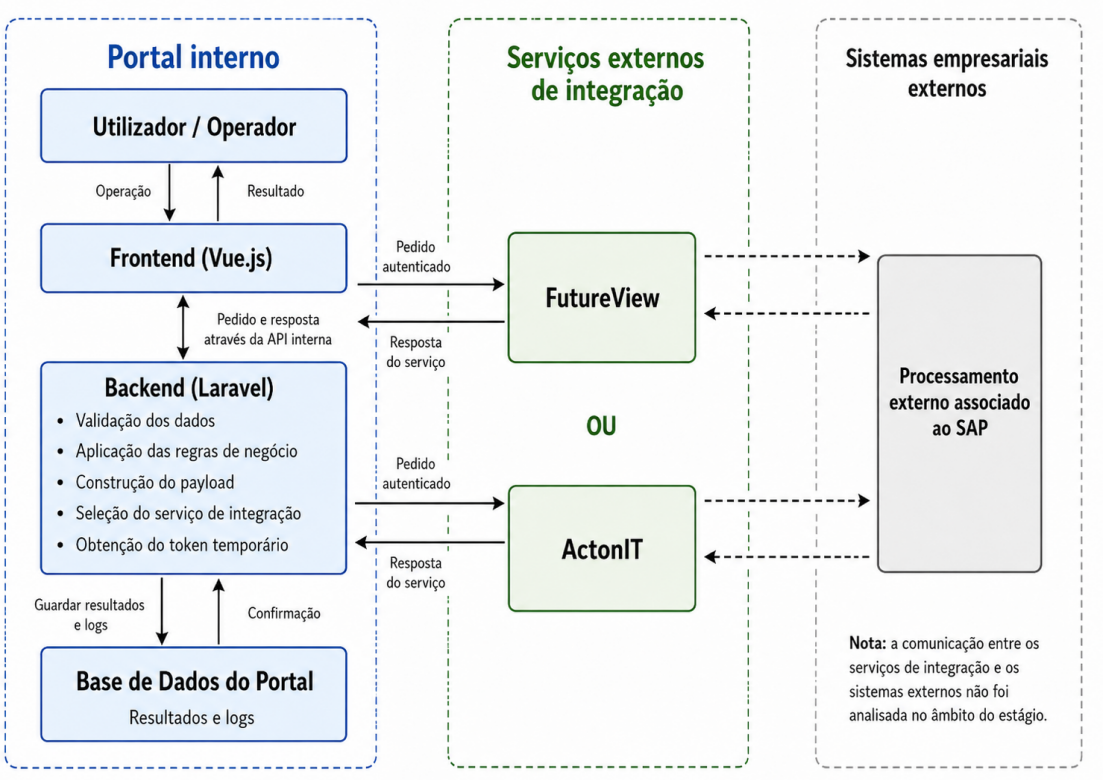

Figura F.1: *Fluxo de comunicação entre o Portal e os serviços externos de integração*

G

Evidências da Campanha Simulada de *Phishing*

Este apêndice apresenta exemplos da mensagem e da página de autenticação utilizadas na campanha simulada de *phishing*. As imagens foram ajustadas de forma a evitar a exposição de informação sensível da organização.

Figura G.1: *Email de phishing utilizado na campanha simulada*

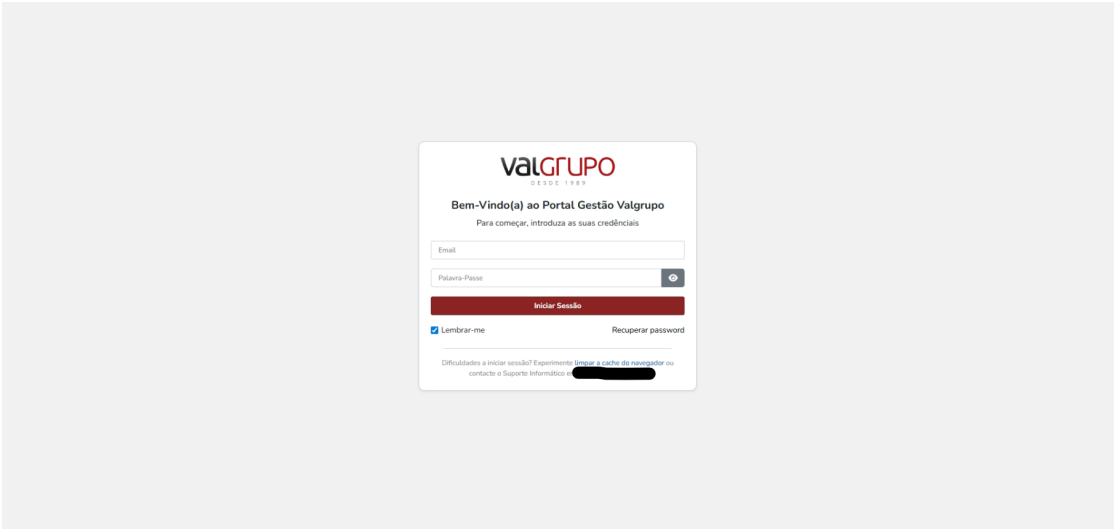

Figura G.2: *Página de autenticação simulada utilizada na campanha*

H

Interfaces do Sistema

Este apêndice apresenta exemplos de interfaces do sistema cuja dimensão não permite a sua inclusão adequada no corpo principal do relatório.

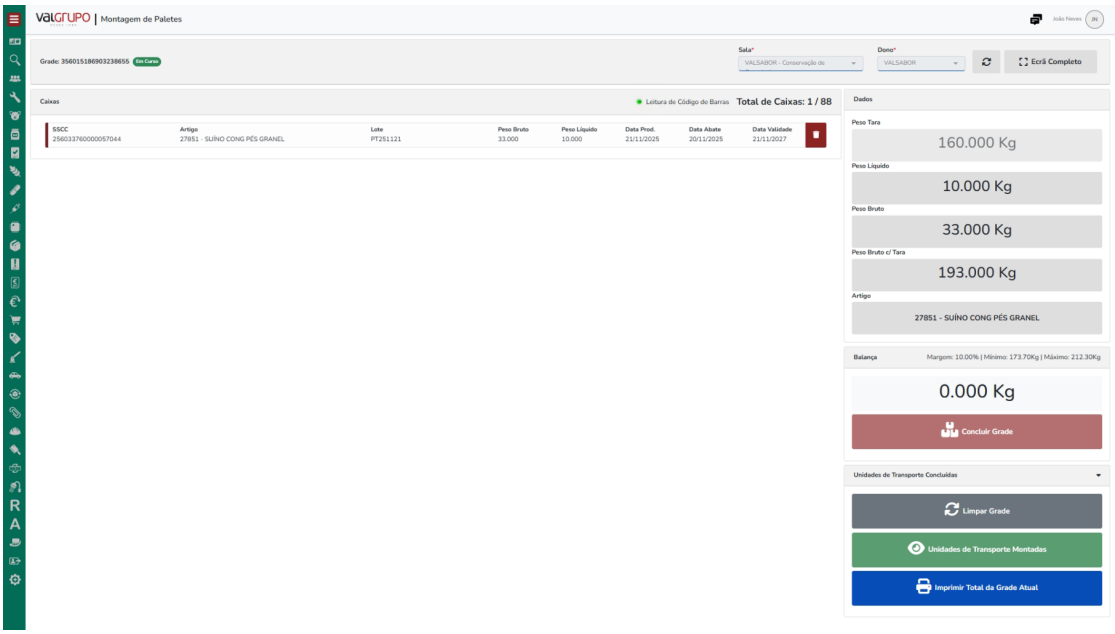

Figura H.1: *Interface de montagem de unidades logísticas*

Página intencionalmente deixada em branco.