

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL FERNANDO SEFFRIN

**EXPEDISOFT - SISTEMA PARA A GESTÃO E RASTREABILIDADE NO
CARREGAMENTO DE MERCADORIA**

GUARAPUAVA

2026

GABRIEL FERNANDO SEFFRIN

**EXPEDISOFT - SISTEMA PARA A GESTÃO E RASTREABILIDADE NO
CARREGAMENTO DE MERCADORIA**

ExpediSoft - System for Cargo Loading Management and Traceability

Proposta de Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Dr. Emerson André Fedechen

GUARAPUAVA

2026



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

AGRADECIMENTOS

Quando entrei na faculdade, me formar parecia um cenário tão distante, e hoje percebo como esse período passou rapidamente. Recordo com carinho a ansiedade e a empolgação de saber que iria estudar na UTFPR, conhecer novos colegas e construir novas amizades. Certamente foi uma experiência única que, se pudesse, viveria novamente.

Hoje, escrevendo estas linhas, percebo que estou próximo de concluir essa etapa e que definitivamente não sou mais o mesmo. A universidade transforma a vida das pessoas, proporcionando crescimento pessoal e profissional. Agradeço a Deus por todo discernimento, sabedoria e paciência durante essa caminhada. Também gostaria que minha madrinha Adriana estivesse aqui para poder acompanhar o profissional que me tornei.

Quero agradecer à minha família, meu pai Sandro, minha mãe Maura e minha irmã Camila, pelo apoio constante durante toda a minha trajetória acadêmica. Obrigado por todo incentivo, paciência e ajuda nesses anos de estudo. Sem vocês, essa caminhada teria sido muito mais difícil.

Aos meus avós Orlando e Noeli, obrigado por serem parte fundamental da minha vida. Tenho muito orgulho de ser o primeiro neto a concluir a graduação.

À Chelsea Brito, que a vida me trouxe e que se tornou cada dia mais importante na minha caminhada, obrigado por fazer parte dessa trajetória. Seu incentivo, compreensão e apoio foram essenciais durante esses anos.

Aos meus colegas e amigos de faculdade, muitos dos quais levarei para a vida, meu muito obrigado. Vocês foram parte importante dessa etapa, e certamente aprendi muito com cada um de vocês.

Também agradeço ao meu orientador, Prof. Dr. Emerson André Fedechen, por ser meu guia e conduzir este trabalho pelo melhor caminho. Sua orientação, conhecimento e apoio foram essenciais para a construção deste projeto.

Agradeço também a todos os professores da UTFPR. Sem dúvida, vocês tiveram papel fundamental na minha formação, e muitos levarei como referência profissional para minha vida. Estendo meus agradecimentos a todos os colaboradores da universidade, pois cada um contribui para que esse ambiente exista e funcione.

Estas linhas são poucas para mencionar todos que fizeram parte dessa caminhada, mas deixo aqui meu sincero agradecimento a todos que cruzaram meu caminho durante esse período.

RESUMO

O processo de expedição de mercadorias é uma etapa crítica na cadeia logística, na qual falhas operacionais, como o carregamento incorreto e a falta de rastreabilidade, geram prejuízos financeiros significativos e comprometem a credibilidade das empresas corporativas. Diante desse cenário, este trabalho apresenta o desenvolvimento do Expedisoft, um sistema integrado composto por uma plataforma administrativa *web* de gestão e um aplicativo móvel voltado ao controle operacional e à auditoria do carregamento de cargas. A solução proposta atua de forma integrada aos sistemas centrais da empresa, permitindo a importação automatizada de pedidos e o agendamento estratégico de cargas associadas a veículos, motoristas e operadores específicos. Na ponta de execução, o operador utiliza o dispositivo móvel para realizar um *checklist* digital baseado na leitura de códigos de identificação por volume, validando os itens em tempo real e bloqueando falhas humanas por duplicidade ou embarque cruzado de mercadorias. Em caso de inconformidades na carga, a ferramenta exige o preenchimento de justificativas e gerencia a captura de evidências fotográficas, que são enviadas de forma assíncrona para armazenamento remoto em nuvem para garantir a integridade dos dados e a fluidez do trabalho em campo. Como resultados, o ecossistema foi integralmente construído e validado por meio de testes automatizados de funcionalidades. O sistema demonstrou-se apto a proporcionar maior confiabilidade operacional, eliminação do uso de planilhas e relatórios em papel, aumento na velocidade de liberação de veículos e suporte abrangente à tomada de decisões gerenciais através de indicadores visuais e telemetria em tempo real do desempenho do armazém.

Palavras-chave: expedição de mercadorias; rastreabilidade logística; integração de sistemas; conferência digital; sistema web e mobile.

ABSTRACT

The goods dispatch process is a critical stage in the logistics chain, where operational failures such as incorrect loading and a lack of traceability generate significant financial losses and compromise corporate credibility. Against this backdrop, this work presents the development of Expedisoft, an integrated system comprising a web-based administrative management platform and a mobile application dedicated to operational control and cargo loading auditing. The proposed solution operates seamlessly with the company's core systems, enabling automated order importing and the strategic scheduling of loads associated with specific vehicles, drivers, and operators. At the execution end, the operator utilizes a mobile device to perform a digital checklist based on scanning unique identification codes per volume, validating items in real time and instantly blocking human errors caused by duplication or mismatched cargo loading. In the event of cargo non-conformities, the tool enforces the submission of justifications and manages the capture of photographic evidence, which is asynchronously transmitted to remote cloud storage to ensure data integrity and seamless field workflow continuity. As a result, the ecosystem was fully built and validated through automated feature testing. The system proved capable of providing greater operational reliability, eliminating the use of spreadsheets and paper reports, accelerating vehicle dispatch speed, and offering comprehensive support for managerial decision-making through visual indicators and real-time warehouse performance telemetry.

Keywords: cargo expedition; logistics management; traceability system; barcode scanning; system integration.

LISTA DE FIGURAS

Figura 1 – Exemplo do fluxo do sistema.	20
Figura 2 – Web: Tela de Login.	32
Figura 3 – Web: Página Principal.	33
Figura 4 – Web: Ordens de Carregamento.	34
Figura 5 – Web: Modal de Agendamento.	34
Figura 6 – Web: Selecionar Ordem para Detalhes.	35
Figura 7 – Web: Detalhes da Ordem.	36
Figura 8 – Web: Detalhes da Ordem 2.	36
Figura 9 – Web: Detalhes da Foto da Carga.	37
Figura 10 – Web: Ordem Apresentando Divergência.	38
Figura 11 – Web: Ordem em Andamento.	39
Figura 12 – Web: Ordem em Andamento 2.	39
Figura 13 – Mobile: Tela de Autenticação.	40
Figura 14 – Mobile: Tela Inicial.	40
Figura 15 – Mobile: Lista de Cargas Atribuídas.	41
Figura 16 – Mobile: Detalhamento da Ordem Logística.	42
Figura 17 – Mobile: Validação com Sucesso.	43
Figura 18 – Mobile: Alerta de Inconsistência.	43
Figura 19 – Mobile: Registro de Justificativa.	44
Figura 20 – Mobile: Captura de Evidências.	44
Figura 21 – Recorte de Tabela - 01	54
Figura 22 – Recorte de Tabela - 02	54
Figura 23 – Recorte de Tabela - 03	55
Figura 24 – Recorte de Tabela - 04	56

LISTA DE TABELAS

Tabela 1 – Funcionalidades <i>Must Have</i>	15
Tabela 2 – Funcionalidades <i>Should Have</i>	16
Tabela 3 – Funcionalidades <i>Could Have</i>	16
Tabela 4 – Funcionalidades <i>Won't Have</i>	16
Tabela 5 – Peso para cada Story Point	22
Tabela 6 – Cronograma de Sprints e Entrega de Story Points (SP)	22
Tabela 7 – Product Backlog: Histórias de Usuário e Pontuação	24

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Modelo JSON de Usuários	51
Listagem 2 – Modelo JSON de Doca e Localização	51
Listagem 3 – Modelo JSON das Ordens de Carregamento	52

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	<i>Application Programming Interface</i>
CI	<i>Continuous Integration</i>
DDD	<i>Domain-Driven Design</i>
ERP	<i>Enterprise Resource Planning</i>
HMR	<i>Hot Module Replacement</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model-View-Controller</i>
MVP	<i>Minimum Viable Product</i>
ORM	<i>Object-Relational Mapper</i>
QR Code	<i>Quick Response Code</i>
REST	<i>Representational State Transfer</i>
SP	<i>Story Points</i>
SQL	<i>Structured Query Language</i>
TDD	<i>Test-Driven Development</i>
UI	<i>User Interface</i>
UUID	<i>Universally Unique Identifier</i>
UX	<i>User Experience</i>

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	10
2	MATERIAIS E MÉTODOS	12
2.1	Materiais	12
2.2	Métodos	14
3	EXPEDISOFT	19
3.1	Metodologia de Desenvolvimento	19
3.2	Arquitetura do Sistema	24
3.3	Modelagem do Banco de Dados	26
3.4	Integração com Sistemas Externos <i>Enterprise Resource Planning</i> (ERP)	27
3.5	Processamento e Armazenamento de Fotos	29
3.6	Estratégia de Testes e Qualidade	30
3.7	Interfaces e Funcionamento do Sistema	32
4	CONCLUSÃO	45
5	CONSIDERAÇÕES FINAIS	46
	REFERÊNCIAS	48
	APÊNDICES	49
	APÊNDICE A – CONTRATO DE DADOS	51
	APÊNDICE B – DIAGRAMA ENTIDADE RELACIONAMENTO COMPLETO	54

1 INTRODUÇÃO

Longe de ser apenas a etapa final de entrega, o processo de expedição é um pilar estratégico para o sucesso e a escalabilidade de um negócio. Para empresas de todos os portes e mercados, de operações locais a grandes exportadoras, a eficiência e a confiabilidade deste processo são determinantes para a competitividade e a satisfação do cliente. Conforme a visão de BALLOU (2006), o nível de serviço prestado ao cliente é o resultado direto de um bom desempenho logístico, funcionando como um dos principais diferenciais no mercado.

Operacionalmente, o processo de expedição compreende uma sequência de atividades críticas que finalizam o ciclo do pedido dentro do armazém, como a conferência de itens, a embalagem (packing), a geração da documentação fiscal e de transporte, e o correto carregamento do veículo. Essa etapa é de alta responsabilidade, pois, como defendem BOWERSOX, CLOSS e COOPER (2013), é o último ponto de controle para assegurar os atributos do “pedido perfeito”: a entrega completa, no prazo, sem avarias e com a documentação correta. Desta forma, a expedição funciona como a garantia final da qualidade percebida pelo cliente.

Contudo, a criticidade dessa etapa a torna um ponto de vulnerabilidade para a operação. Falhas nesta etapa podem gerar diversos problemas, como atrasos nas entregas, inconsistências nas documentações dos pedidos, problemas em empacotamentos, avarias no carregamento dos produtos e até mesmo itens errados no pedido do cliente. Essas falhas impactam diretamente o desempenho logístico, gerando retrabalho, aumento de custos operacionais com devoluções e reenvios, e, de forma mais danosa, comprometem a satisfação e a confiança do cliente, pondo em risco a reputação da empresa no mercado. Essa cadeia de prejuízos vai na contramão do que BOWERSOX, CLOSS e COOPER (2013) defendem como um dos objetivos da logística integrada: a busca pelo menor custo total para atender às necessidades do cliente.

A concepção deste projeto foi inspirada na observação de desafios operacionais em uma empresa de grande porte do setor de exportação. Embora o problema tenha sido identificado em um cenário específico, ele pode ocorrer em qualquer organização, de modo que a solução proposta aqui visa a adaptabilidade a diferentes contextos empresariais.

Todos estes problemas são acentuados em um cenário observado como objeto de estudo: operações de expedição de grande porte, especialmente no setor de exportação de manufaturados, cujo modelo de negócio é majoritariamente voltado para mercados de alta exigência, como Estados Unidos e Europa. Em operações dessa natureza, o processo de expedição frequentemente ocorre como a última etapa antes do faturamento, tornando qualquer alteração posterior extremamente complexa e custosa.

Nesse tipo de operação é comum o desafio crítico e recorrente de falhas na etapa de carregamento: ocorre com frequência o esquecimento de caixas pertencentes ao pedido e/ou carregamento de pacotes que não pertencem ao pedido, fazendo com que a remessa seja enviada inconsistente ao cliente. Esta situação específica mobiliza diversas áreas para sua correção, impactando prazos, metas e, principalmente, a credibilidade junto aos clientes internacionais.

Nesse cenário, torna-se essencial investir em ferramentas que ofereçam maior controle e rastreabilidade para esta fase crítica da operação, auxiliando no carregamento e mitigando erros operacionais. Desta forma, a presente proposta visa o desenvolvimento de uma solução, por meio da análise, planejamento e desenvolvimento de um sistema web e mobile integrados.

Entre os principais desafios do projeto, destaca-se a definição de uma arquitetura flexível, capaz de se comunicar com sistemas legados. Um ponto crítico é o tratamento das imagens, cuja integração deve visar a estabilidade da aplicação. Neste trabalho, um sistema robusto é caracterizado por sua confiabilidade (operar de forma consistente), tolerância a erros (lidar com inconsistências de dados) e manutenibilidade (facilidade para evoluir e corrigir). Nesse contexto, a decisão de utilizar armazenamento por referência é fundamental, pois, além de garantir a escalabilidade, contribui diretamente para esses pilares da estabilidade do sistema.

Em síntese, este projeto visa o desenvolvimento de uma ferramenta voltada à confiabilidade operacional e à escalabilidade para apoiar os colaboradores no setor de expedição, mas podendo ser adaptado a outras empresas.

1.1 Objetivos

Nesta seção serão apresentados o objetivo geral e os objetivos específicos do presente trabalho.

1.1.1 Objetivo Geral

Desenvolver um sistema para o controle e rastreabilidade do processo de expedição de mercadorias.

1.1.2 Objetivos Específicos

Os objetivos específicos foram divididos em duas subseções, interface Web e interface Mobile.

Interface Mobile

- Desenvolver um módulo para identificação das mercadorias, auxiliando na verificação da caixa correta para o carregamento;
- Implementar um *checklist* de carregamento, possibilitando marcar as caixas já carregadas, auxiliando que nenhum item seja esquecido ou adicionado indevidamente;
- Adicionar a funcionalidade de inclusão de observações, possibilitando o registro de comentários e justificativas após o carregamento;
- Desenvolver a funcionalidade de *upload* de fotos via aplicativo para conferência.

Interface Web

- Desenvolver um sistema para agendamento de carregamentos e alocação do funcionário responsável, facilitando a organização das operações logísticas;
- Criar um recurso para registro de responsabilidade, associando cada carregamento ao colaborador responsável, promovendo rastreabilidade e responsabilização;
- Criar a funcionalidade de conferência de carregamentos anteriores, com acesso às justificativas, fotos e observações previamente registradas.

2 MATERIAIS E MÉTODOS

A seguir, um descritivo de como o trabalho foi conduzido e quais tecnologias foram utilizadas.

2.1 Materiais

Para o desenvolvimento do projeto, foram utilizadas as seguintes tecnologias, ferramentas e ambientes, cujas escolhas são justificadas a seguir.

- **Backend: PHP 8+ e Laravel 10+** (LARAVEL, 2025):

O Laravel foi escolhido por ser um *framework*¹ *PHP* robusto e com um ecossistema que acelera o desenvolvimento de *Application Programming Interface* (API)s. Sua arquitetura *Model-View-Controller* (MVC), o *Object-Relational Mapper* (ORM) *Eloquent* e as ferramentas integradas permitem a construção de um *backend*² seguro e de manutenção descomplicada.

- **Frontend (Web): React 18+ e Vite** (META, 2025) (VITE, 2025):

O *React* é consistentemente classificado como a biblioteca *JavaScript* mais utilizada pela comunidade de desenvolvedores para a criação de interfaces de usuário. Sua abordagem baseada em componentes permite a criação de *User Interface* (UI)s reativas e escaláveis. A adoção do *TypeScript* é justificada por sua capacidade de adicionar tipagem estática ao *JavaScript*, auxiliando no desenvolvimento de um código mais seguro. O *Vite* é utilizado como ferramenta de *build*³, pois sua arquitetura serve módulos ES nativos durante o desenvolvimento, resultando em um *Hot Module Replacement* (HMR) e inicialização de servidor mais rápidos que ferramentas como o *Webpack*.

- **Mobile: React Native e Expo** (REACT, 2025):

Para manter a consistência tecnológica e otimizar o tempo de desenvolvimento, o *React Native* foi selecionado. O uso do *Expo* simplifica o processo de compilação, distribuição e acesso a recursos nativos, mantendo a consistência entre as plataformas Android e iOS.

- **Componentes de UI (Web): shadcn/ui** (SHADCN, 2025):

¹ *Framework*: Estrutura de suporte definida na qual um outro projeto de software pode ser organizado e desenvolvido.

² *Backend*: Camada de acesso a dados e regras de negócio da aplicação, não visível diretamente ao usuário final.

³ *Build*: Processo de converter código-fonte em artefatos de software executáveis ou otimizados para produção.

Para a construção da UI, optou-se pelo *shadcn/ui*. Diferente das bibliotecas de UI tradicionais, esta é uma coleção de componentes não-estilizados (baseados em *Radix UI*) que são copiados para a base de código do projeto. Conforme sua documentação oficial, essa abordagem possibilita total propriedade e controle sobre o código do componente, facilitando a personalização e a acessibilidade sem adicionar dependências externas ao projeto.

- **Banco de Dados: *PostgreSQL 15+*** (GROUP, 2025):

O *PostgreSQL* é um sistema de gerenciamento de banco de dados relacional de código aberto (*open source*⁴), reconhecido por sua conformidade com os padrões *Structured Query Language* (SQL).

- **Ambiente de Contêineres: *Docker*** (DOCKER, 2025):

O *Docker* foi utilizado para a criação de ambientes padronizados e isolados através de contêineres. Conforme a literatura de *DevOps*, a containerização resolve o problema de inconsistência entre ambientes de desenvolvimento e produção. Ela garante que a aplicação e todas as suas dependências (*PHP*, *PostgreSQL* etc.) sejam empacotadas juntas, possibilitando a portabilidade e simplificando o processo de implantação (*deploy*⁵) (VASYUKOV A. PETROV, 2018).

- **Prototipação: *Figma*** (FIGMA, 2025):

O *Figma* é uma ferramenta de *design* de interfaces UI e experiência do usuário *User Experience* (UX), utilizada para a criação de protótipos interativos nos testes de usabilidade antes do início da codificação.

- **Controle de Versão: *Git*** (GIT, 2025):

Justificativa: Ferramenta para controle e versionamento de código, fundamental para o gerenciamento do projeto, combinada com o *GitHub*.

- **GitHub Copilot: *Copilot*** (GitHub Inc., 2025):

Para auxílio no desenvolvimento de código, utilizou-se a ferramenta de assistência por inteligência artificial *GitHub Copilot* para revisão de código.

- **IDE: *WebStorm*, *PHPStorm* e *DataGrip*:**

Por conta da familiaridade e recursos avançados, serão utilizadas as IDEs da *JetBrains* como ferramentas de desenvolvimento.

⁴ *Open source*: Software cujo código-fonte é disponibilizado publicamente para uso, modificação e distribuição.

⁵ *Deploy*: Ato de implantar ou disponibilizar uma aplicação para uso em um ambiente específico, como produção ou testes.

- **Documentação API: Swagger:**

A escolha pelo *Swagger* é importante, pois o projeto possui um *backend* que serve a dois *frontends* distintos (*Web e Mobile*). O *Swagger* gera uma documentação interativa que permite a qualquer desenvolvedor visualizar e testar os *endpoints*⁶ diretamente pelo navegador.

2.2 Métodos

O desenvolvimento do sistema seguiu uma abordagem aplicada, fundamentada em princípios da Engenharia de Software e metodologias ágeis, com foco na entrega incremental de valor e na validação contínua das funcionalidades implementadas. As etapas metodológicas são descritas a seguir.

- **Levantamento de Requisitos**

O levantamento de requisitos do sistema Expedisoft foi fundamentado na experiência prévia do autor no setor de expedição de uma empresa exportadora de grande porte, complementada por entrevistas com usuários-chave e observação direta das rotinas operacionais. Essa abordagem é comumente utilizada em projetos aplicados, conforme destaca SOMMERVILLE (2011), possibilitando que o pesquisador derive requisitos a partir da análise de sua própria vivência e conhecimento técnico do domínio. A observação das práticas cotidianas permitiu identificar as principais dificuldades enfrentadas no processo de carregamento, consolidando a necessidade de uma solução que fosse, ao mesmo tempo, descentralizada e flexível a diferentes cenários. Destacam-se como pontos críticos:

- A fragilidade da conferência manual dos volumes destinados ao carregamento;
- A dificuldade na coleta, organização e vinculação de evidências (registros fotográficos e justificativas de divergências);
- A ausência de um histórico estruturado e de fácil consulta sobre carregamentos realizados anteriormente.

A partir desse diagnóstico, os requisitos foram documentados, categorizados e priorizados de acordo com seu impacto no processo de expedição. Os requisitos funcionais descrevem as funcionalidades necessárias para apoiar a operação, como a leitura de códigos, validação de volumes e registro de divergências.

⁶ *Endpoint*: Um ponto final de comunicação ou URL específica em uma API onde um serviço pode ser acessado.

Já os requisitos não funcionais estabelecem os critérios de qualidade e restrições técnicas, incluindo segurança da informação, controle de acesso, rastreabilidade, desempenho e usabilidade em dispositivos móveis. A padronização da comunicação por meio de uma API foi considerada de forma transversal, de modo que a arquitetura do sistema suporte os requisitos de integração e escalabilidade necessários para o ambiente corporativo.

• Priorização dos Requisitos

Após definidos, os requisitos foram organizados segundo a técnica *MoSCoW*, que classifica as funcionalidades em “*Must Have*”, “*Should Have*”, “*Could Have*” e “*Won’t Have*” CLEGG D.; BARKER (1994). Essa técnica permitiu a construção de um *Minimum Viable Product* (MVP), concentrando esforços nas funcionalidades essenciais para o controle e rastreabilidade do carregamento.

***Must Have*:** Requisitos críticos e indispensáveis para o funcionamento do sistema, detalhados na Tabela 1.

***Should Have*:** Requisitos importantes que agregam valor significativo ao produto, detalhados na Tabela 2.

***Could Have*:** Requisitos desejáveis, detalhados na Tabela 3.

***Won’t Have*:** Requisitos que estão fora do escopo do sistema, detalhados na Tabela 4.

ID	Categoria	Requisito	Descrição
R1	Geral	API de Integração (<i>Backend</i>)	Desenvolvimento de <i>endpoints Representational State Transfer</i> (REST) (método <i>POST</i>) para receber dados de ordens de carregamento.
R2	Geral	Autenticação e Autorização	Sistema seguro de <i>login</i> para diferenciar perfis de acesso (Gestor <i>Web</i> e Operador).
R3	<i>App Mobile</i>	Leitura de	Funcionalidade para leitura de <i>Quick Response Code</i> (QR Code) utilizando a câmera do dispositivo <i>mobile</i> .
R4	<i>App Mobile</i>	Validação em Tempo Real	Lógica no aplicativo que verifica se o volume lido corresponde à ordem de carregamento ativa, mitigando erros.
R5	Plataforma <i>Web</i>	Gestão de Ordens	Interface para visualizar as ordens de carregamento importadas, acompanhar seu status e realizar o agendamento da ordem.

Tabela 1 – Funcionalidades *Must Have*

ID	Categoria	Requisito	Descrição
R6	App <i>Mobile</i>	Documentação Visual	Funcionalidade para capturar e/ou enviar fotos da carga como comprovante visual.
R7	App <i>Mobile</i>	Registro de Divergências	Fluxo que obriga o operador a inserir uma justificativa em texto ao finalizar uma carga com divergências.
R8	Plataforma <i>Web</i>	Detalhamento de Carga	Visualização detalhada do histórico de conferência de uma ordem, incluindo quem realizou a conferência, horário, fotos e justificativa.

Tabela 2 – Funcionalidades *Should Have*

ID	Categoria	Requisito	Descrição
R9	Plataforma <i>Web</i>	<i>Dashboard</i>	Painel inicial com <i>cards</i> e indicadores/gráficos básicos de desempenho (ex: quantidade de carregamentos programados, tempo médio de carregamento).
R10	Geral	Notificações	Alertas por <i>e-mail</i> ou <i>push notification</i> para gestores quando uma carga com divergência for finalizada.
R11	Plataforma <i>Web</i>	Integração	Página dedicada ao monitoramento de integração dos dados no sistema.

Tabela 3 – Funcionalidades *Could Have*

ID	Categoria	Requisito	Descrição
R12	Geral	Integração Ativa	Sistema buscando dados no ERP.
R13	Geral	Funcionalidade WMS	Controle de estoque e endereçamento.
R14	Geral	Documentos	Emissão de documentos fiscais (NF-e).

Tabela 4 – Funcionalidades *Won't Have*

• Prototipação de Interfaces

As interfaces do sistema foram prototipadas no *Figma*, de modo a avaliar o fluxo de navegação e a experiência do usuário antes da codificação. Essa etapa permitiu antecipar problemas de usabilidade e ajustar a interface conforme necessidade.

• Metodologia de Desenvolvimento

Para o acompanhamento e planejamento do desenvolvimento do sistema, foi adotada uma abordagem incremental, inspirada na metodologia *Scrum*, mesmo tratando-se de um projeto desenvolvido por um único autor. As funcionalidades foram organizadas em pequenas entregas, favorecendo a evolução contínua e validação progressiva do sistema.

Como técnica de estimativa e controle do esforço, foram utilizados *Story Points* (SP), que representam uma medida relativa de complexidade, esforço e risco de cada funcionalidade a ser implementada. As estimativas foram realizadas com base em comparação entre tarefas, considerando critérios como volume de código, regras de negócio envolvidas e dependências técnicas.

O registro das estimativas e do progresso foi feito por meio de planilhas, possibilitando o acompanhamento da quantidade de SP planejados e concluídos em cada período de trabalho. Essa prática permitiu analisar a evolução do desenvolvimento ao longo do tempo, fornecendo dados objetivos para avaliação do processo e apoio à reflexão sobre a execução do projeto.

- **Modelagem e Arquitetura**

A modelagem do sistema foi orientada pelos princípios do *Domain-Driven Design* (DDD), buscando representar o domínio da expedição e manter o código coeso e de fácil manutenção. Essa abordagem favoreceu a separação de responsabilidades e a comunicação clara entre as camadas do sistema (EVANS, 2004).

- **Testes e Garantia de Qualidade**

Foram aplicadas técnicas de *Test-Driven Development* (TDD) nos módulos principais do sistema, contribuindo para que cada funcionalidade atenda aos critérios definidos e reduzindo o risco de regressões Beck (2010).

- **Gerenciamento do Projeto**

A implantação foi realizada por meio de contêineres *Docker*, que favorecem a padronização do ambiente de execução e a portabilidade do sistema entre desenvolvimento e produção. O controle de versão foi feito com *Git* e *GitHub*, utilizando o fluxo *Git Flow* e *Conventional Commits* aliado a práticas de *Continuous Integration* (CI), em conformidade com os princípios de *DevOps* KIM Gene; HUMBLE (2016).

- **Integração entre Sistema e ERP Cliente**

Para que a integração entre o sistema desenvolvido e os diferentes sistemas ERP dos clientes ocorra de maneira flexível e padronizada, foram elaborados três modelos (ordens para carregamento, usuários e docas) de integração baseados em *JavaScript Object Notation* (JSON), permitindo que o sistema receba dados estruturados independente da tecnologia utilizada pelo cliente.

Essa abordagem visa garantir compatibilidade e simplicidade na comunicação entre sistemas, utilizando o protocolo e o método *POST* em *endpoints* REST disponibilizados pela API do *backend*. Dessa forma, os dados referentes às ordens e aos usuários podem ser transmitidos diretamente pelo ERP do cliente para o sistema, possibilitando a integridade e atualização das informações em tempo real.

O formato JSON foi escolhido por ser amplamente aceito e de fácil manipulação, além de permitir uma estrutura clara para a transmissão de dados.

Importante salientar que todas as integrações entre os sistemas devem ter uma camada de segurança via *token* de autenticação, com o objetivo de estabelecer uma comunicação segura.

- **Banco de Dados**

A modelagem do banco de dados foi estruturada com o objetivo de representar, de forma organizada e fiel, todos os elementos envolvidos no processo de expedição.

O banco foi projetado em *PostgreSQL*, utilizando identificadores do tipo *Universally Unique Identifier* (UUID). As entidades principais foram agrupadas em quatro categorias funcionais: estrutura organizacional, entidades externas, entidades operacionais e entidades de auditoria e rastreabilidade.

3 EXPEDISOFT

O Expedisoft foi idealizado para funcionar como um sistema integrado que digitaliza e controla o processo de carregamento de pedidos, substituindo a conferência manual por um fluxo de trabalho digital e rastreável. A arquitetura do sistema será estruturada em duas interfaces interdependentes: uma interface web para gestão e uma interface mobile para execução operacional.

O fluxo de trabalho se inicia na interface web, onde um gestor logístico realiza o agendamento de um novo carregamento. Nesta etapa, ele associa um pedido (previamente importado do sistema ERP do cliente) a um operador específico e define detalhes como a doca de carregamento e veículo. A Figura 1 exemplifica o fluxo do sistema.

Com a tarefa atribuída, o operador, utilizando o aplicativo mobile, visualiza o carregamento em sua lista de tarefas. Ao iniciar o processo, o aplicativo exibe um *checklist* digital de todos os volumes que devem ser carregados. Para cada caixa, o operador pode ou utilizar a câmera do dispositivo (celular ou coletor android) para realizar a leitura de um QR Code ou realizar manualmente a conferência no dispositivo. O sistema valida a informação em tempo real, atualizando o *status* do item no *checklist* e emitindo *feedbacks* sonoros e visuais em caso de leitura de um item incorreto ou duplicado. O QR Code nesse caso deve ser gerado pelo ERP do cliente e adicionado a caixa para identificação.

Em caso de finalização com divergências (caixas faltando), o aplicativo torna obrigatório a justificativa, fazendo com que o operador insira os motivos dessa divergência identificada pelo sistema, do contrário a justificativa/observação é opcional. Também é permitido que sejam capturadas e anexadas fotos do carregamento/carga, que servirão como evidência visual da integridade do carregamento.

Após a finalização, todas as informações são sincronizadas e ficam disponíveis no painel de histórico da interface web. O gestor pode consultar o *status* de qualquer carregamento, auditar os itens conferidos, visualizar as fotos e ler as observações do operador, tendo uma visão completa e centralizada do processo.

Poderá também visualizar em uma central de monitoramento, todos os *status* dos carregamentos atuais, informações como a quantidade de carregamentos realizados VS programados e o desempenho dos operadores.

Estará disponível para consulta, em uma página de relatórios, informações gerais de desempenho, gráficos exibindo a eficiência da operação e tempo de carregamento, podendo ser aplicados filtros temporais, operador e se existente doca.

3.1 Metodologia de Desenvolvimento

O desenvolvimento do sistema fundamentou-se na metodologia ágil *Scrum*, adaptada para a realidade de um projeto individual. A gestão do escopo e do tempo foi gerenciada atra-

Processo de Carregamento

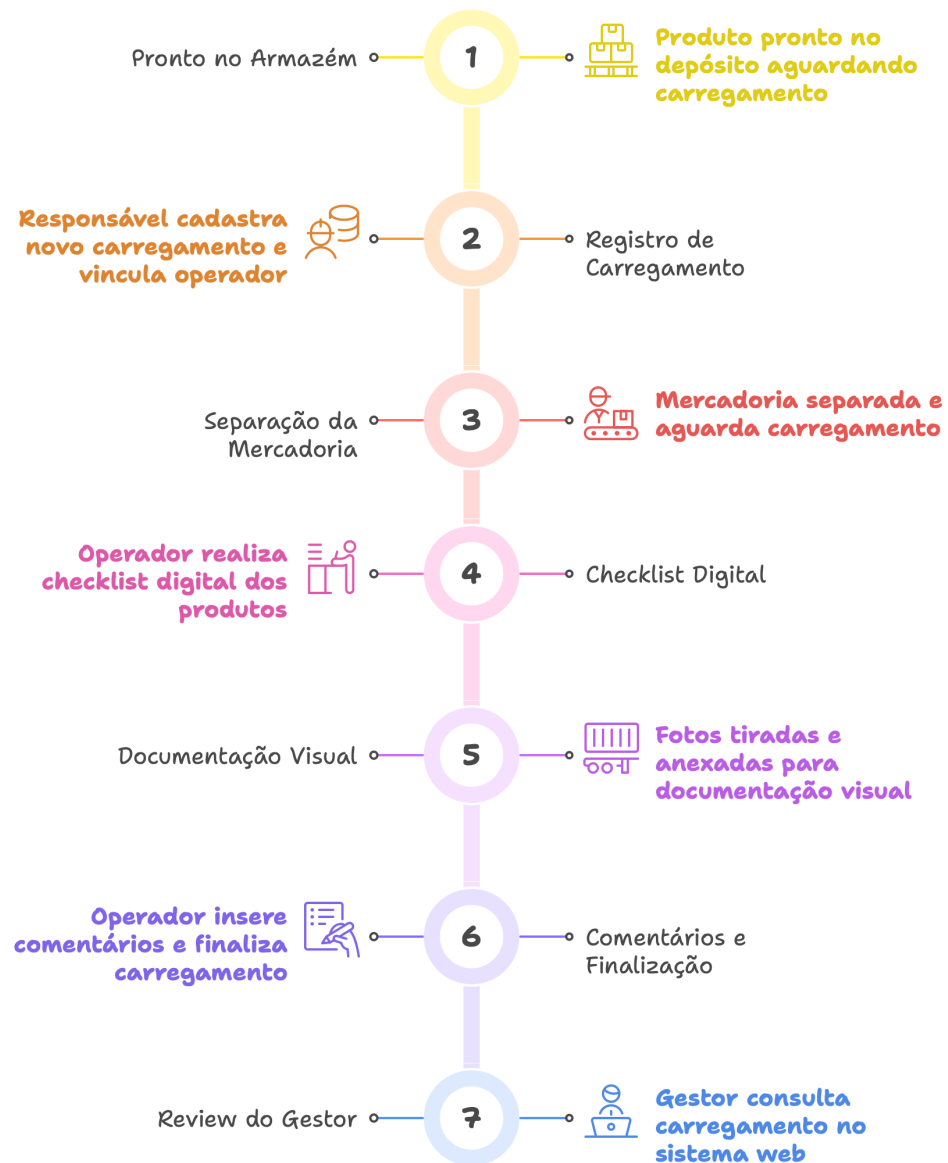


Figura 1 – Exemplo do fluxo do sistema.

vés do controle de esforço via SP, o que permitiu o mapeamento sistemático das funcionalidades, a mitigação de riscos técnicos e a organização de iterações com níveis de complexidade sustentáveis.

3.1.1 Estimativa de Esforço e *User Stories*

Por ser um projeto individual, que diverge do padrão de equipes encontrado no *Scrum*, os pesos de esforço passaram por uma revisão, conforme detalhado na Tabela 5. Para a definição e refinamento dos requisitos, utilizou-se a técnica de Histórias de Usuário (*User Stories*), estruturadas sob o padrão:

- **Como:** [papel do usuário]
- **Quero:** [ação ou funcionalidade]
- **Para:** [valor de negócio ou objetivo]

O conjunto dessas definições compões o *Product Backlog* consolidado na Tabela 7, que totaliza a pontuação de esforço entregue ao longo do projeto.

3.1.2 Ciclos de Entrega e Velocidade

O cronograma de execução foi segmentado em 9 *sprints*, com periodicidade quinzenal (duas semanas por ciclo). A dedicação diária ao desenvolvimento foi estabelecida em aproximadamente duas horas. A quantidade de pontos por *sprint* concentrou-se entorno de 12 a 15 pontos.

Considerando que o tempo cronológico em projeto de *software* apresenta variáveis de incerteza, a métrica principal de desempenho adotada foi a velocidade. Esta que é calculada a partir da soma dos pontos entregues por *sprint*, podendo ser conferida na tabela Tabela 6, na coluna “Entrega”.

Estruturar todo o desenvolvimento do sistema em *sprints* foi fundamental para reforçar a percepção de progresso e mitigar a ansiedade. Esse modelo evitou a sensação de atraso e eliminou a incerteza sobre as etapas seguintes, fatores importantes dado o nível de complexidade do sistema, cujo fluxo de desenvolvimento poderia se tornar confuso sem o devido direcionamento.

Story Point	Significado Prático
1	Muito simples, trivial.
2	Simple.
3	Médio.
5	Complexo.
8	Muito complexo / incerto.

Tabela 5 – Peso para cada Story Point

Sprint	Início	Fim	Planj.	Entreg.	Dif.	Observações
Sprint 1	05/01/26	13/01/26	13	8	-5	Fim de ano e festas atrasaram o desenvolvimento do projeto.
Sprint 2	26/01/26	13/02/26	11	6	-5	Alta demanda profissional comprometeu a continuidade do desenvolvimento.
Sprint 3	16/02/26	25/02/26	10	10	0	Realizado atividades conforme planejado.
Sprint 4	02/03/26	13/03/26	11	8	-3	Dificuldades na refatoração do código de integração com ERP.
Sprint 5	17/03/26	25/03/26	13	10	-3	Atividades extracurriculares comprometeram os últimos dias.
Sprint 6	30/03/26	04/04/26	14	14	0	Maior tempo disponível devido ao feriado de Páscoa.
Sprint 7	04/04/26	11/04/26	18	18	0	Foram desenvolvidas as interfaces <i>web</i> e <i>mobile</i> , além da implementação da funcionalidade de início e finalização do carregamento e da API de conferência.
Sprint 8	16/04/26	20/04/26	15	15	0	O desempenho da <i>sprint</i> foi favorecido pela reutilização de estruturas desenvolvidas em ciclos anteriores.
Sprint 9	22/04/26	01/05/26	13	10	-3	Foi implementada a funcionalidade de envio de fotos e configurada a página inicial da plataforma.

Tabela 6 – Cronograma de Sprints e Entrega de Story Points (SP)

ID	Épico	User Story	SP	Sprint
US01	Infra	Como gestor, quero autenticar-me no sistema web para acessar funcionalidades restritas.	3	1
US13	Infra	Como sistema, quero configurar o ambiente inicial (backend, frontend e mobile).	2	1
US06	Integração	Como sistema, quero armazenar ordens e itens no banco para persistência dos dados.	2	1
US11	Infra	Como gestor, quero encerrar sessão com segurança (Logout).	1	1

ID	Épico	User Story (cont.)	SP	Sprint
US03	Integração	Como sistema, quero expor uma API REST para receber dados do ERP.	3	2
US04	Integração	Como sistema, quero validar os dados recebidos via API para garantir integridade.	2	2
US08	API	Como sistema, quero padronizar respostas da API para facilitar consumo.	1	2
US09	API	Como desenvolvedor, quero documentar endpoints da API (Swagger/OpenAPI).	3	3
US10	Obs.	Como sistema, quero registrar logs básicos para rastrear erros.	2	3
US15	Integração	Como sistema, quero disponibilizar API para integração de usuários.	2	3
US16	API	Como sistema, quero disponibilizar endpoints para consulta de ordens.	3	3
US05	Web	Como gestor, quero visualizar uma lista de ordens importadas.	2	4
US14	Integração	Como sistema, quero autenticar integrações via TOKEN.	3	4
US21	Integração	Como sistema, quero gerenciar integrações em fila (Processamento assíncrono).	3	4
US17	Web	Como gestor, quero agendar ordens de carregamento.	3	5
US18	API	Como sistema, quero disponibilizar endpoints para agendamento de ordens.	3	5
US19	Integração	Como sistema, quero disponibilizar API para integração de docas.	2	5
US20	Web	Como gestor, quero visualizar dados resumidos no painel inicial (Dashboard).	2	5
US07	Mobile	Como operador, quero autenticar-me no aplicativo mobile para acessar minhas tarefas.	3	6
US12	Mobile	Como operador, quero manter sessão ativa no aplicativo.	2	6
US22	Mobile	Como operador, quero visualizar as ordens atribuídas a mim.	3	6
US30	API	Como sistema, quero consultar os itens de uma ordem.	2	6
US31	API	Como sistema, quero consultar os packages vinculados a um item.	2	6
US32	Operação	Como operador, quero visualizar o status do carregamento.	2	6

ID	Épico	User Story (cont.)	SP	Sprint
US02	Acesso	Como sistema, quero gerenciar perfis de usuário (Gestor vs Operador).	2	7
US23	Mobile	Como operador, quero abrir uma ordem para visualizar detalhes.	3	7
US24	Mobile	Como operador, quero visualizar os itens da ordem.	2	7
US29	Web	Como gestor, quero visualizar o detalhamento completo da ordem no sistema web.	3	7
US33	Operação	Como sistema, quero registrar o início do carregamento (Timestamp).	2	7
US34	Operação	Como sistema, quero registrar a finalização do carregamento.	2	7
US40	API	Como sistema, quero registrar o início/fim para o processo de conferência.	2	7
US25	Mobile	Como operador, quero escanear o QR Code de um package.	5	8
US26	Mobile	Como operador, quero confirmar a conferência do item.	3	8
US36	Operação	Como sistema, quero validar se o QR Code pertence àquela ordem.	3	8
US37	Operação	Como operador, quero visualizar progresso da conferência.	2	8
US42	Mobile	Como sistema, quero exigir uma justificativa para carregamentos incompletos.	2	8
US27	Mobile	Como operador, quero registrar fotos do carregamento.	3	9
US28	API	Como sistema, quero realizar upload das fotos para o backend.	3	9
US38	Obs.	Como sistema, quero registrar logs detalhados de operação.	2	9
US39	Web	Como gestor, quero visualizar métricas de carregamento no Dashboard.	2	9
US35	Operação	Como gestor, quero visualizar o histórico de status da ordem.	3	9

Tabela 7 – Product Backlog: Histórias de Usuário e Pontuação

3.2 Arquitetura do Sistema

A arquitetura do Expedisoft foi projetada sob o modelo de arquitetura desacoplada (*client-server*), onde o *backend* funciona como um provedor central de serviços (API) para múltiplas interfaces de consumo. Essa escolha permite que a lógica de negócio seja centralizada,

enquanto as interfaces *Web* e *Mobile* evoluem de forma independente, comunicando-se através de contratos via protocolo *Hypertext Transfer Protocol* (HTTP) e formato JSON.

3.2.1 Camada de Serviços (API REST)

O ponto central do sistema reside no *backend* desenvolvido em Laravel, que atua como uma API. Conforme observado na estrutura de rotas e controladores, o sistema utiliza o padrão de Camada de Serviços (*Service Layer Pattern*) para isolar a lógica de negócio complexa dos controladores HTTP.

A comunicação entre os componentes ocorre da seguinte forma:

- **Contrato de Dados:** As informações trafegam exclusivamente no formato JSON, favorecendo a conectividade com sistemas ERP externos com respostas padronizadas para as interfaces *Web* e *Mobile*.
- **Processamento Assíncrono:** Para operações pesadas, como o recebimento de grandes volume de dados de integração ou o *upload* de imagens para a nuvem, o sistema utiliza Filas e *Jobs* (Processamento em Segundo Plano), evitando o bloqueio da interface do usuário e contribuindo para maior estabilidade operacional da aplicação.

3.2.2 Persistência e Integridade dos Dados

A modelagem do banco de dados PostgreSQL reflete uma preocupação com a normalização e segurança. Uma decisão arquitetural crítica foi a implementação de UUIDs como chaves primárias. Esta prática reduz a previsibilidade dos identificadores utilizados pelo sistema e facilita a futura escalabilidade horizontal do banco de dados.

As tabelas principais foram estruturadas de uma forma que permite a rastreabilidade completa do carregamento:

- **Loading Orders:** Tabela central do sistema, contém os principais relacionamentos.
- **Order Items:** Relacionamento entre produtos e ordens de carregamento específicas.
- **Packages:** Identificação individual de cada volume físico através de códigos únicos, de forma a permitir a conferência item a item via dispositivo mobile.

3.2.3 Segurança e Autenticação

Para proteger o acesso às informações do sistema, a arquitetura implementa o Laravel Sanctum, que provê autenticação baseada em *tokens* para as APIs. Nas interfaces, o estado de autenticação é gerenciado de forma reativa:

- No *Frontend Web*, utiliza-se a *Context API* do *React* para manter a sessão do gestor e proteger rotas administrativas.
- No Aplicativo *Mobile*, a persistência do *token* permite que o operador mantenha a sessão ativa durante o turno de trabalho, garantindo agilidade no acesso às ordens atribuídas.

3.3 Modelagem do Banco de Dados

O banco de dados do Expedisoft foi projetado com foco em confiabilidade na integridade das informações e viabilizar processos rigorosos de auditoria. A modelagem final divergiu do planejamento inicial para acomodar a necessidade de maior rastreabilidade operacional, resultando na inclusão de tabelas dedicadas a *logs* de operação e colunas que identificam a origem e a integridade de cada registro.

O banco conta com 25 tabelas, organizadas em quatro categorias funcionais: estrutura organizacional, entidades externas, entidades operacionais e entidades de auditoria e rastreabilidade. Dentre elas, destacam-se:

- ***loading_order***: Tabela central que consolida o status do carregamento e vincula os operadores às docas.
- ***packages***: Responsável pelo vínculo entre o item e os pacotes da ordem de carregamento, armazenando o código único do pacote para leitura.
- ***scan_logs***: Contém todas as leituras realizadas pelo operador, independente de sucesso ou falha, armazenando o código da caixa, conteúdo da requisição, status e mensagem de erro.
- ***photos***: A tabela armazena as informações das fotos coletadas pelo operador durante o carregamento, tendo colunas como o *store_path*, ID do *Google Drive* e tipo do arquivo.
- ***integration_logs***: Contém as informações dos dados recebidos do ERP do cliente como, *endpoint* requisitado, *payload* e *status* da integração.
- ***order_status_history***: Tabela que registra todas as mudanças de *status* das ordens de carregamento, contendo o *status* novo e antigo e o usuário que realizou a alteração.

O diagrama completo do banco de dados pode ser conferido no Apêndice B.

3.4 Integração com Sistemas Externos ERP

Com o objetivo de favorecer a flexibilidade e a integração do Expedissoft, a comunicação com sistemas corporativos externos (ERP) baseia-se em contratos de dados padronizados via API REST. O *backend* disponibiliza rotas específicas de integração baseadas em JSON, estruturados de maneira a favorecer a integridade e a validação dos dados recebidos em tempo de integração.

O ecossistema de integração é composto por três fluxos distintos: ordens de carregamento, usuários e docas. Cada fluxo possui o seu respectivo modelo de *payload* padrão. Essa abordagem promove o desacoplamento de responsabilidades, exigindo que o sistema parceiro se adeque ao formato esperado pelo Expedissoft reduzindo a necessidade de adaptações específicas para diferentes formatos de integração oriundos de sistemas legados.

Ao receber uma requisição de integração, a camada de controle valida as propriedades e delega o processamento para uma fila de execução assíncrona baseada em rotinas assíncronas de processamento. Durante a persistência, o serviço adota a estratégia de busca e criação automática de registros: caso entidades de infraestrutura como cliente, destino, transportadora ou produto não sejam identificadas na base de dados, o sistema efetua o cadastro automático ou a atualização dos registros de forma automatizada.

Os fluxos e suas especificações técnicas são detalhados a seguir:

- **Ordem de Carregamento:** Representa o fluxo principal de dados do sistema, consolidando o cabeçalho do pedido, dados cadastrais de transporte, itens comerciais e os agrupamentos de volumes físicos para conferência. O modelo estrutural é apresentado na Listagem 3.
- **Usuário:** Fluxo responsável pela sincronização de colaboradores da empresa cliente, populando a base interna com perfis de acesso restritos e credenciais iniciais, conforme exemplificado na Listagem 1.
- **Doca:** Módulo encarregado do mapeamento físico das áreas de carregamento da empresa, alimentando o cadastro de docas para agendamento logístico na interface administrativa, cujo modelo consta na Listagem 2.

3.4.1 Modelo de Dados: Ordens de Carregamento

O padrão definido para a integração de ordens de carregamento encapsula toda a complexidade operacional da expedição logística. A estrutura organiza hierarquicamente a rastreabilidade do processo, vinculando uma ordem comercial aos seus respectivos volumes físicos mapeados em campo.

Detalhes do Modelo:

- **external_id:** Atua como chave estrangeira lógica, preservando o vínculo direto da entidade com o identificador nativo do ERP de origem.
- **unique_package_code:** Código atribuído a cada volume físico, consumido diretamente via scanner de QR Code pelo aplicativo móvel para a validação do *checklist*.
- **Processamento de Itens:** A tabela de itens (*order_items*) é vinculada a um catálogo normalizado de produtos (*products*), minimizando a redundância de dados cadastrais.
- **Interface de Acesso:** Disponibilizado sob o método HTTP *POST* no caminho relativo */api/integration/order*.

3.4.2 Modelo de Dados: Usuários

A integração de usuários centraliza e otimiza a gestão de acessos e a governança de credenciais no sistema. O modelo mapeia os identificadores necessários para o controle interno de privilégios, possibilitando que a sincronização popule a base de dados mantendo os vínculos corporativos intactos.

Detalhes do Modelo:

- **external_id:** Identificador único do funcionário no banco de dados do sistema central do cliente.
- **email:** Atributo obrigatório utilizado como credencial exclusiva e persistente para a realização da autenticação (*login*).
- **Segurança:** Os usuários importados são associados a perfis padrão de permissões (ex: Operador ou Gestor), com senhas geradas por algoritmos de criptografia *hash* ativadas no primeiro acesso.
- **Interface de Acesso:** Disponibilizado sob o método HTTP *POST* no caminho relativo */api/integration/user*.

3.4.3 Modelo de Dados: Docas

A infraestrutura logística do armazém é alimentada por meio do contrato de docas, estruturando os locais físicos disponíveis para a alocação de carregamentos no painel administrativo do gestor logístico.

Detalhes do Modelo:

- **external_id:** Código de identificação exclusivo da doca originário do ERP.

- **dock_code:** Código abreviado ou sigla utilizada nas listagens visuais do painel administrativo.
- **location:** Delimitação de espaço (unidade, filial ou pátio) onde a estrutura operacional está instalada.
- **Interface de Acesso:** Disponibilizado sob o método HTTP *POST* no caminho relativo */api/integration/dock*.

3.5 Processamento e Armazenamento de Fotos

Conforme planejado nos requisitos operacionais do sistema, o Expedisoft contempla a captura e o armazenamento de registros fotográficos durante o encerramento do carregamento logístico para fins de auditoria e comprovação de integridade da carga. Alinhado às boas práticas de modelagem de dados, optou-se por não persistir os arquivos de imagem em formato bruto (*Binary Large Objects* - BLOB) diretamente no banco de dados, evitando o crescimento desordenado da base e a degradação da performance das consultas.

Para solucionar este desafio, foi arquitetada uma solução de armazenamento distribuído por referência integrada à API do *Google Drive*. Visando a alta disponibilidade do sistema, o fluxo operacional foi estruturado por meio de um desacoplamento síncrono e assíncrono composto pelas seguintes etapas:

- **Interface Móvel e Envio:** A interface *mobile* gerencia a captura ou seleção da imagem na galeria do dispositivo. Ao finalizar a conferência, os arquivos são transmitidos para o *backend* por meio de uma requisição HTTP do tipo *POST* utilizando o formato *multipart/form-data*.
- **Envio de Dados e Resposta Ágil:** O processamento síncrono é intermediado pela classe *OrderPhotoController*. O sistema valida a requisição, salva temporariamente os arquivos binários no disco do servidor local através do *OrderPhotoService* e cria um registro na tabela *photos* com o estado inicial definido como pendente (*STATUS_PENDING*). Imediatamente, a API retorna uma resposta JSON com o código de estado HTTP *202 Accepted*, liberando a interface do operador móvel para novos carregamentos sem a necessidade de aguardar o término da transferência de mídia.
- **Fila Assíncrona e Upload Remoto:** O serviço dispara o *Job* encarregado pelo processamento em segundo plano, denominado *UploadPhotoToDriveJob*, que é gerenciado de forma assíncrona pelas filas do Laravel. Este componente lê o binário do disco local, estabelece conexão segura com o *Google Drive* por meio da abstração de discos de armazenamento (*Flysystem Driver*) e realiza a persistência na pasta de destino correspondente ao identificador externo da ordem logística.

- **Atualização Cadastral e Coleta de Metadados:** Concluída a transferência com sucesso, o algoritmo extrai os metadados diretamente do adaptador de rede da nuvem externa, capturando o identificador global único do arquivo (*drive_id*). O registro mapeado no banco de dados é atualizado para o estado de enviado (*STATUS_UPLOADED*), salvando o caminho lógico para renderizações futuras no painel *web* administrativo. Por fim, o arquivo binário temporário é deletado do servidor, otimizando o espaço em disco do ecossistema.

3.6 Estratégia de Testes e Qualidade

Para assegurar a confiabilidade, a tolerância a falhas e a manutenibilidade do Expedi-soft, o processo de codificação dos módulos principais do *backend* fundamentou-se na técnica de TDD. Esta abordagem consiste em um ciclo iterativo no qual os cenários de teste são desenhados antes mesmo da implementação da funcionalidade de produção (*Red*), observando-se sua falha controlada; em seguida, desenvolve-se o código mínimo necessário para a aprovação do teste (*Green*) e, por fim, executa-se a melhoria do código sob uma rede de segurança estável (*Refactor*). A adoção do desenvolvimento guiado por testes mostrou-se um elemento estratégico mitigador de riscos, especialmente no contexto de um projeto de autoria individual, onde a ausência de uma equipe de garantia de qualidade de software (*QA*) exige mecanismos automatizados de verificação de regressões.

A suíte de testes foi estruturada utilizando a ferramenta nativa de testes do Laravel (PHPUnit), concentrando esforços em testes de integração de funcionalidades (*Feature Tests*). Esses testes simulam requisições HTTP reais contra os *endpoints* da API, avaliando desde o comportamento das regras de negócio até a persistência correta no banco de dados.

3.6.1 Impacto do TDD no Ciclo de Desenvolvimento

O impacto prático da aplicação do TDD no ExpediSoft manifestou-se em quatro vertentes principais:

- **Design de Código Orientado ao Domínio:** Escrever o teste antes do código de produção forçou o isolamento de responsabilidades em camadas de serviço puras (*Service Layer*). Classes complexas, como o validador de bipagem de pacotes (*CheckListEntryService*), foram criadas de forma modular para permitir o isolamento de cenários de sucesso e falha nas leituras operacionais.
- **Segurança em Ciclos de Refatoração:** Conforme registro na Tabela 6, a *Sprint 4* apresentou dificuldades técnicas na refatoração do código de integração com o ERP. A existência de uma base sólida de testes automatizados permitiu reestruturar com-

pletamente os algoritmos de processamento de carga sem o risco de introduzir efeitos colaterais indesejados nas APIs já consumidas pela interface *web* e *mobile*.

- **Consistência e Concorrência:** Os testes guiaram a implementação de mecanismos sólidos para lidar com cenários críticos de produção, como tentativas de duplicação de leitura de pacotes e requisições concorrentes idênticas enviadas pelo ERP, favorecendo respostas HTTP consistentes e estabilidade transacional.
- **Compreensão do Domínio e Refinamento de Escopo:** A antecipação lógica exigida pelo método propiciou um entendimento analítico profundo acerca do comportamento esperado para cada funcionalidade. Ao mapear o que o sistema deveria ou não permitir, materializaram-se cenários práticos que reduziram o nível de abstração decorrente da modelagem conceitual estática. Esse exercício reflexivo revelou-se oportuno para identificar inconsistências no planejamento, o que motivou ajustes estruturais em tabelas, otimização de relacionamentos e, inclusive, a especificação de novas Histórias de Usuário.

3.6.2 Mapeamento dos Testes de Funcionalidades (*Feature Tests*)

A cobertura de testes automatizados concentrou-se nos fluxos críticos do ecossistema, mapeando os seguintes componentes de produção:

- **AuthApiTest:** Valida os mecanismos de autenticação e proteção de rotas via *Laravel Sanctum*, contribuindo para que requisições desprovidas de *token* válido sejam rejeitadas com o código HTTP 401.
- **OrderIntegrationTest e DockIntegrationTest:** Testam o comportamento dos *end-points* de ingestão de dados, simulando o envio de estruturas *JSON* válidas e inválidas para garantir o correto funcionamento dos validadores de requisição (*FormRequests*) e o disparo das filas assíncronas (*Jobs*).
- **CheckListEntryTest:** Blinda a regra de negócio mais sensível do aplicativo operacional, verificando se o sistema valida corretamente os identificadores (*unique_package_code*), incrementa o progresso do *checklist* e rejeita volumes pertencentes a outras ordens de carregamento.
- **PhotoUploadTest e UploadPhotoToDriveJobTest:** Asseguram a integridade do fluxo de documentação visual, avaliando o recebimento do arquivo de imagem binária pela API e o encadeamento correto do *Job* assíncrono responsável por persistir o arquivo no serviço de armazenamento remoto.

3.7 Interfaces e Funcionamento do Sistema

Esta seção apresenta as interfaces desenvolvidas no ecossistema Expedisoft, relacionando-as às respectivas regras de negócio, fluxos operacionais e componentes de código implementados nas arquiteturas *web* e *mobile*.

3.7.1 Interface Web (Módulo Administrativo)

A interface *web* concentra as funcionalidades de gestão, agendamento de cargas, auditoria e acompanhamento logístico por parte dos gestores.

- **Tela de Autenticação:** Apresentada na Figura 2, a tela de autenticação restringe o acesso às funcionalidades administrativas do sistema. Após a validação das credenciais do usuário, a sessão é mantida de forma segura, permitindo a navegação entre os módulos da plataforma sem a necessidade de novas autenticações durante o período de uso.

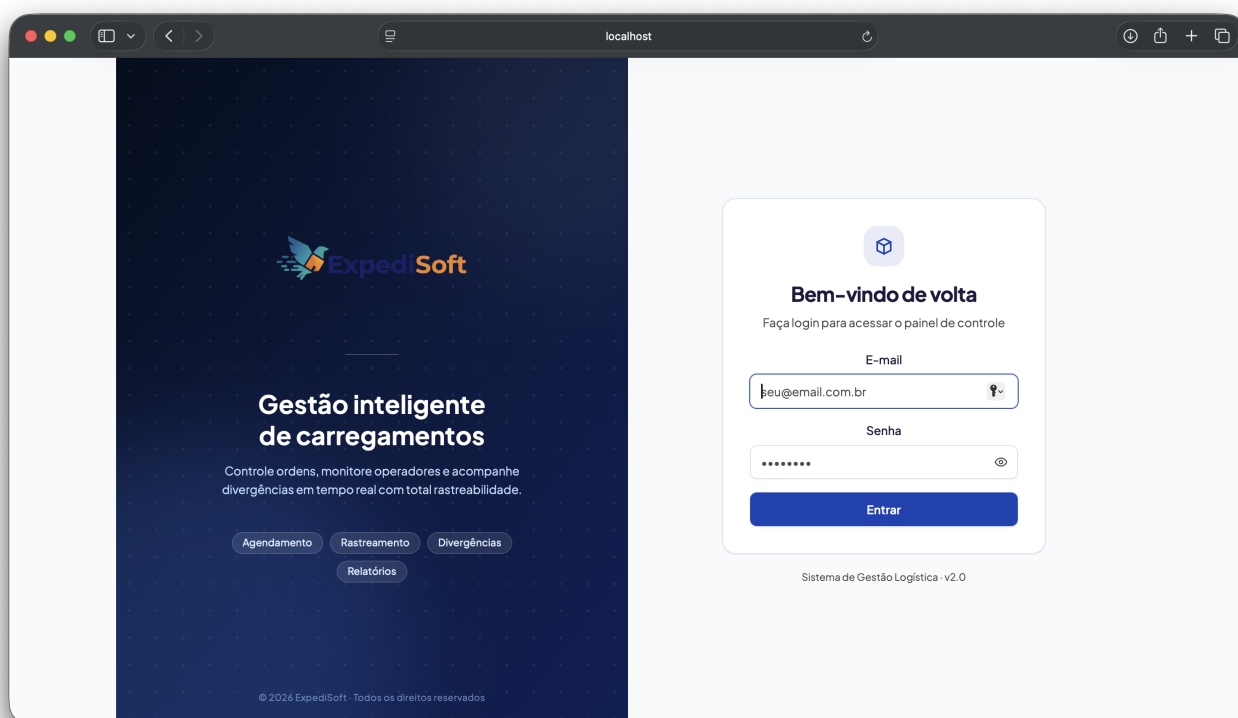


Figura 2 – Web: Tela de Login.

- **Monitoramento (*Dashboard*):** Conforme ilustrado na Figura 3, a página inicial disponibiliza indicadores operacionais que permitem acompanhar o desempenho da expedição em tempo real. O painel apresenta a distribuição das ordens de carregamento por status, contemplando carregamentos agendados, em andamento, concluídos e fi-

nalizados com divergência. Além disso, são exibidos gráficos de acompanhamento das operações realizadas nos últimos quatorze dias e uma tabela contendo as ordens mais recentemente atualizadas no sistema.

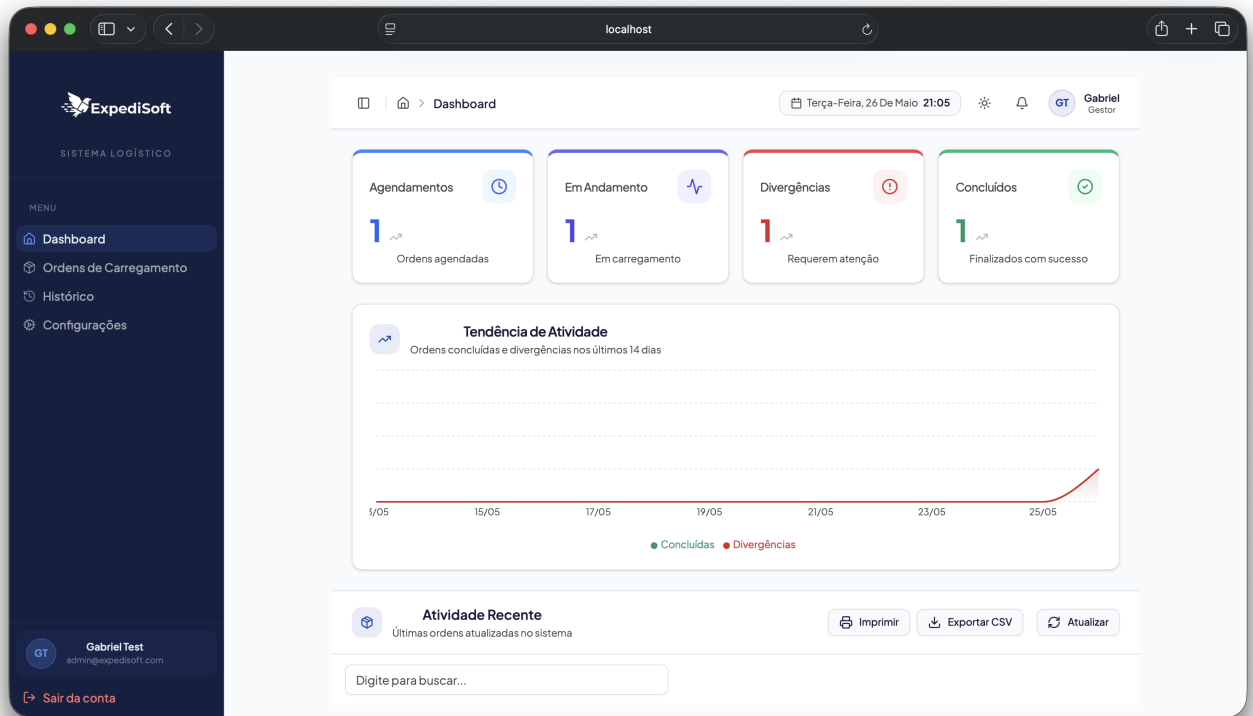


Figura 3 – Web: Página Principal.

- **Listagem e Agendamento de Ordens:** A tela de gerenciamento logístico, apresentada na Figura 4, reúne todas as ordens de carregamento integradas ao sistema. Para programar uma operação, o gestor deve selecionar a opção “Agendar Carregamento” na coluna de ações correspondente à ordem desejada. Em seguida, é exibido um formulário modal para o preenchimento das informações necessárias ao planejamento da atividade, incluindo data, horário, operador responsável e doca de carregamento, conforme demonstrado na Figura 5.

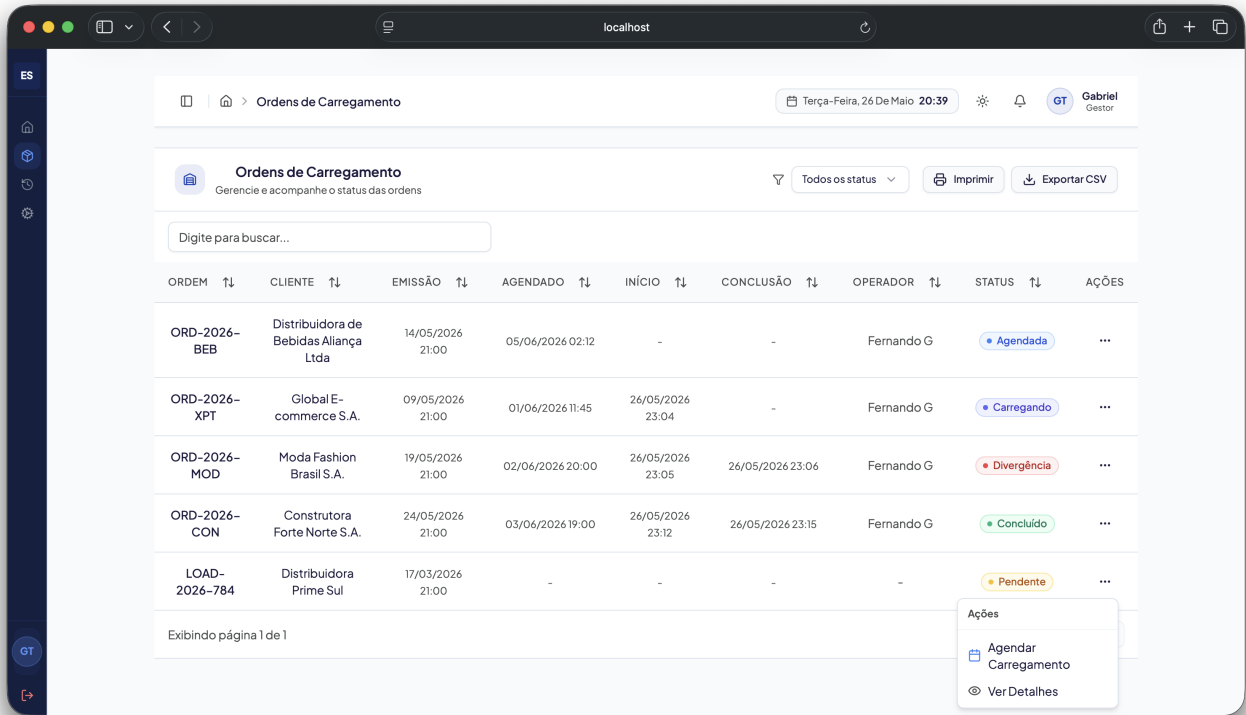


Figura 4 – Web: Ordens de Carregamento.

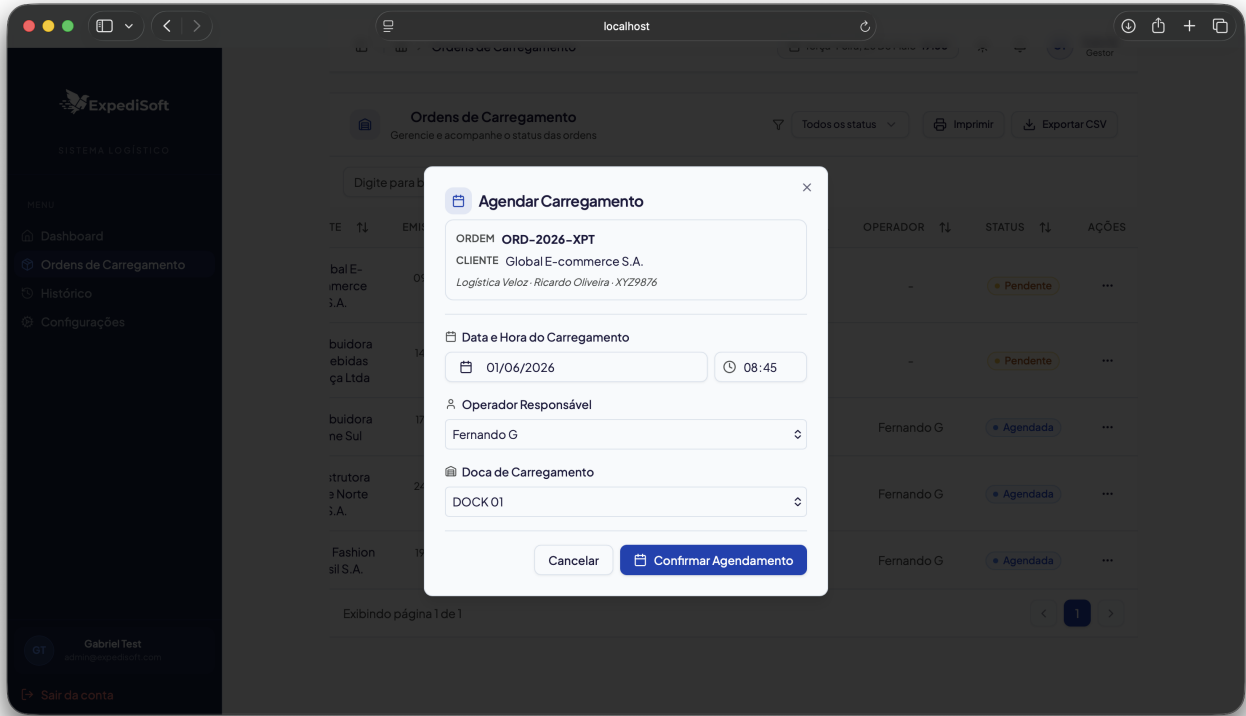


Figura 5 – Web: Modal de Agendamento.

- **Consulta e Auditoria de Ordens:** Para acompanhar ou auditar uma operação, o gestor pode selecionar a opção “Ver Detalhes” na coluna de ações da ordem desejada, conforme ilustrado na Figura 6. A partir desse acesso, o sistema exibe informações detalhadas sobre a operação logística, incluindo dados da carga, cliente, transportadora, operador responsável, cronologia de execução e situação atual do carregamento.

As telas apresentadas nas Figura 7 e Figura 8 demonstram a visualização completa da ordem, permitindo a consulta dos itens vinculados ao carregamento e seus respectivos status de conferência. Também são disponibilizadas as evidências fotográficas registradas pelo operador durante a execução da atividade.

Ao selecionar uma imagem anexada, o sistema apresenta uma visualização ampliada da fotografia, conforme exemplificado na Figura 9, favorecendo a análise detalhada das evidências coletadas durante o carregamento.

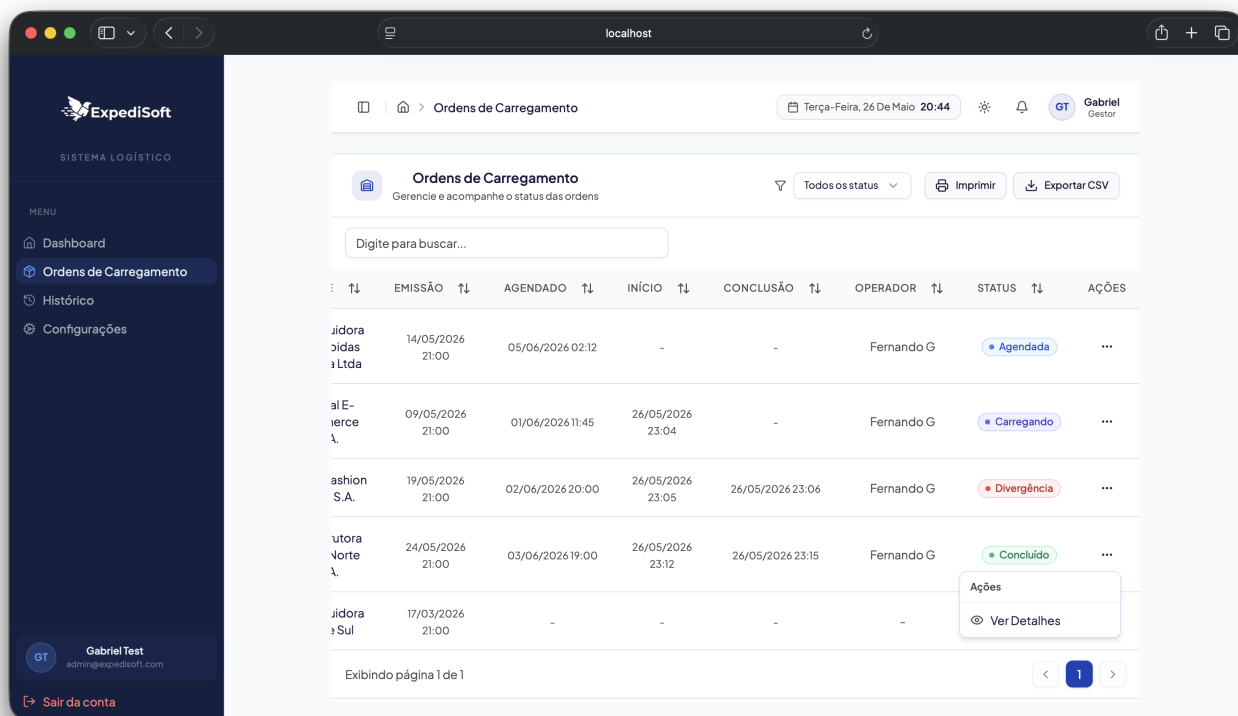


Figura 6 – Web: Selecionar Ordem para Detalhes.

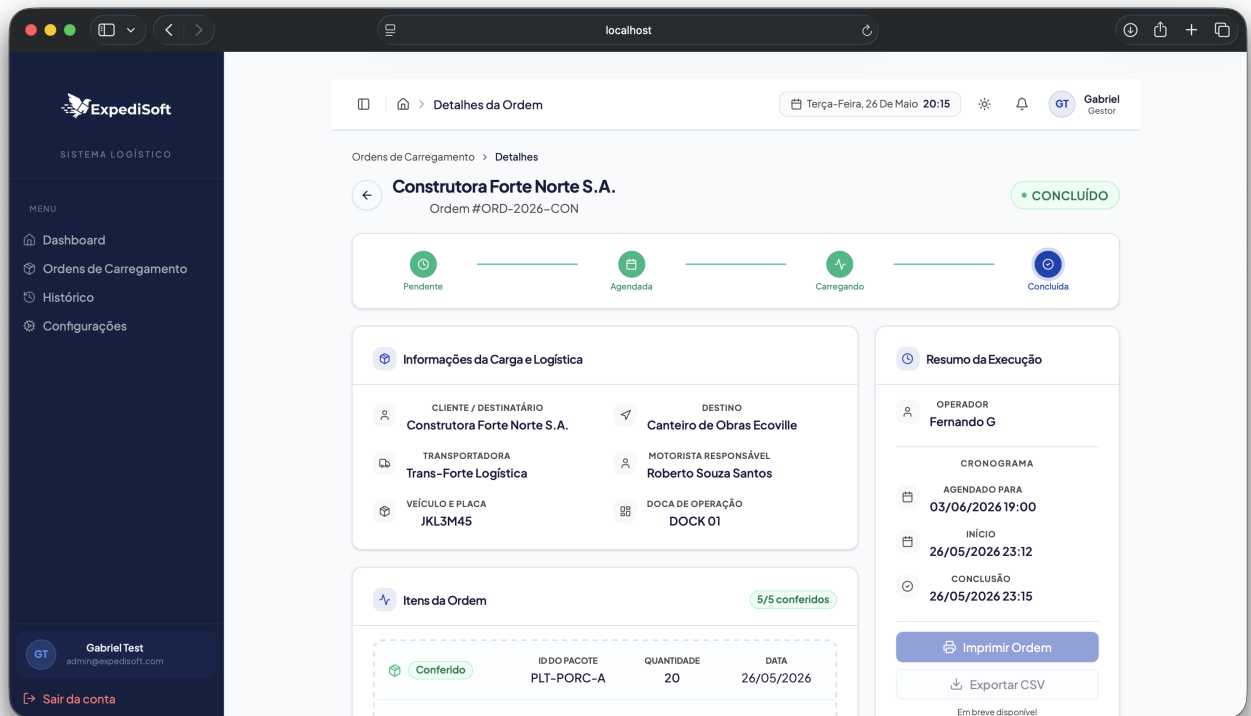


Figura 7 – Web: Detalhes da Ordem.

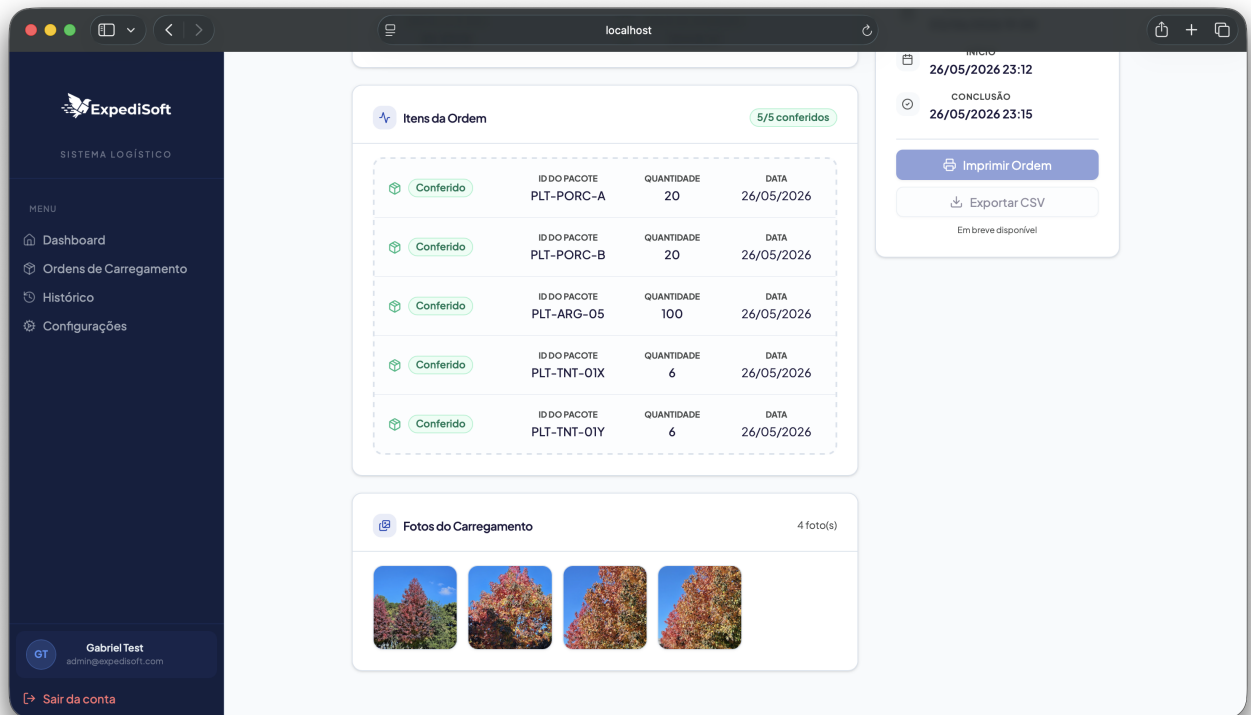


Figura 8 – Web: Detalhes da Ordem 2.

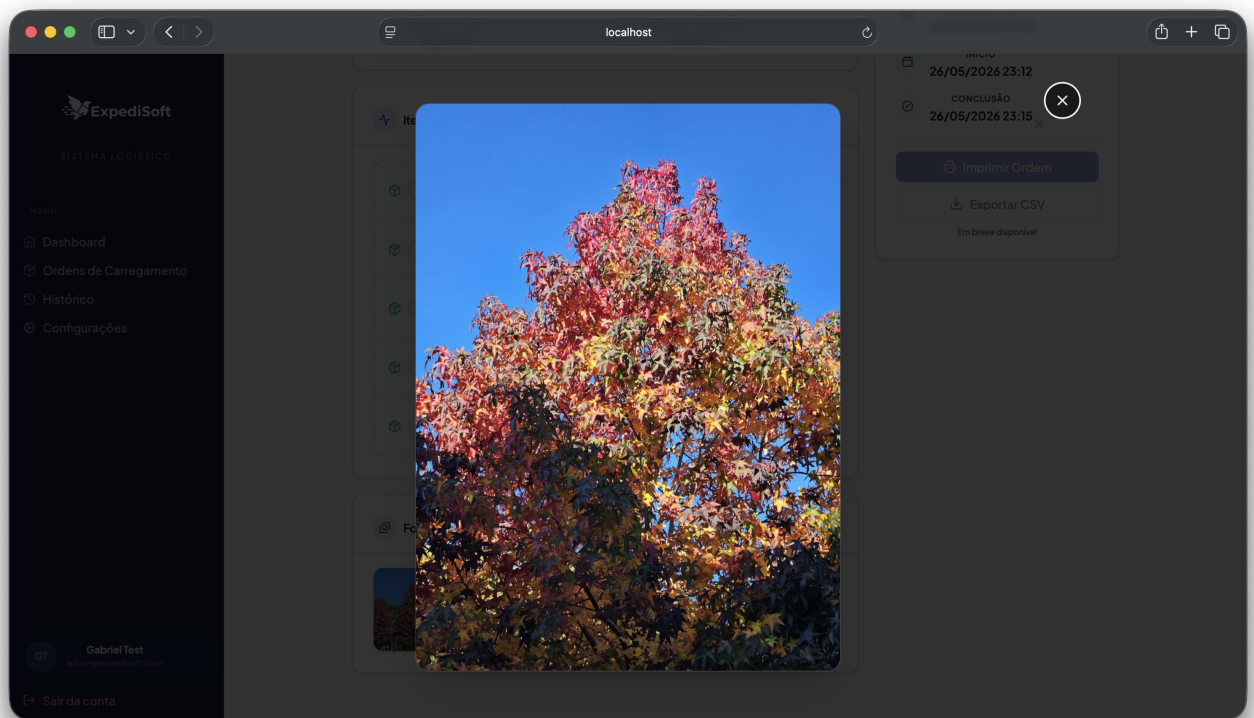


Figura 9 – Web: Detalhes da Foto da Carga.

- **Ordens com Divergência:** Quando uma ordem é finalizada com divergência, a interface exibe a justificativa informada pelo operador durante a execução da atividade. Essa informação permite aos gestores compreender os motivos da inconsistência registrada, conforme apresentado na Figura 10.

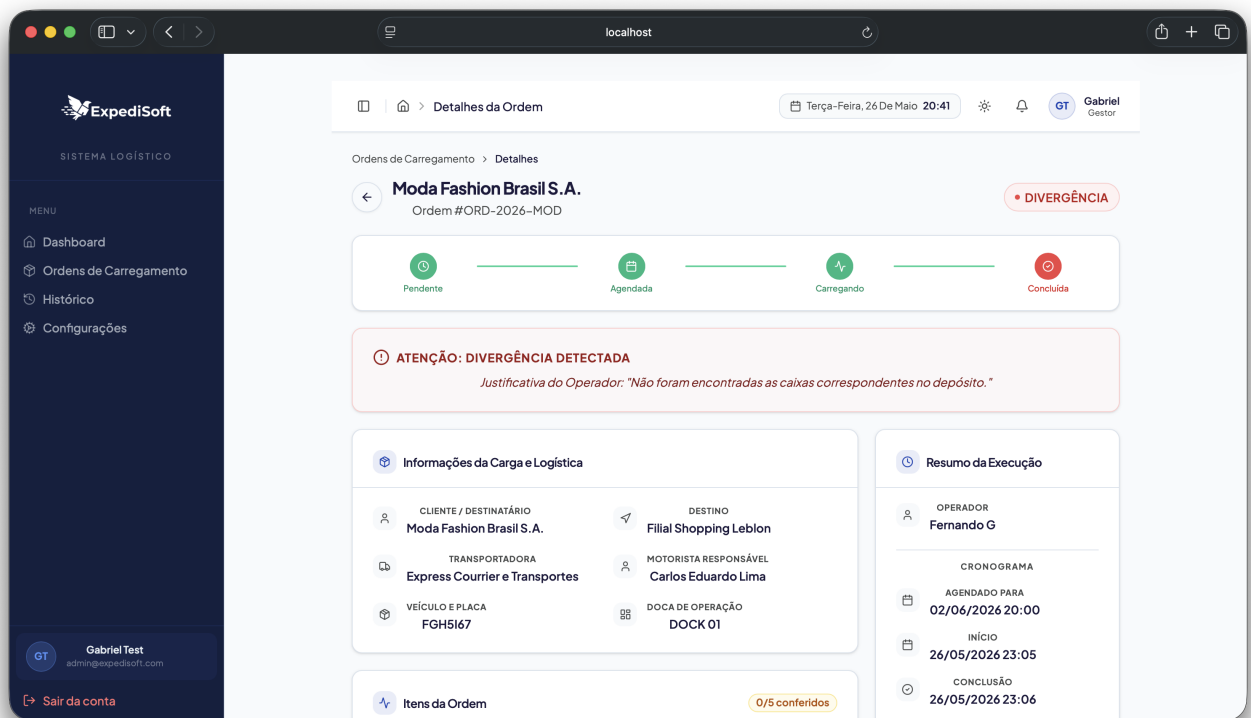


Figura 10 – Web: Ordem Apresentando Divergência.

- **Conferindo uma Ordem em Andamento:** Ao acessar uma ordem que se encontra em execução, o gestor pode acompanhar seu progresso por meio da tela de detalhes. Conforme exemplificado na Figura 11, é apresentado o status atual da operação, permitindo identificar a etapa em que o carregamento se encontra.

Complementarmente, a Figura 12 apresenta os itens já conferidos e aqueles que ainda permanecem pendentes, possibilitando o acompanhamento da execução da atividade em tempo real.

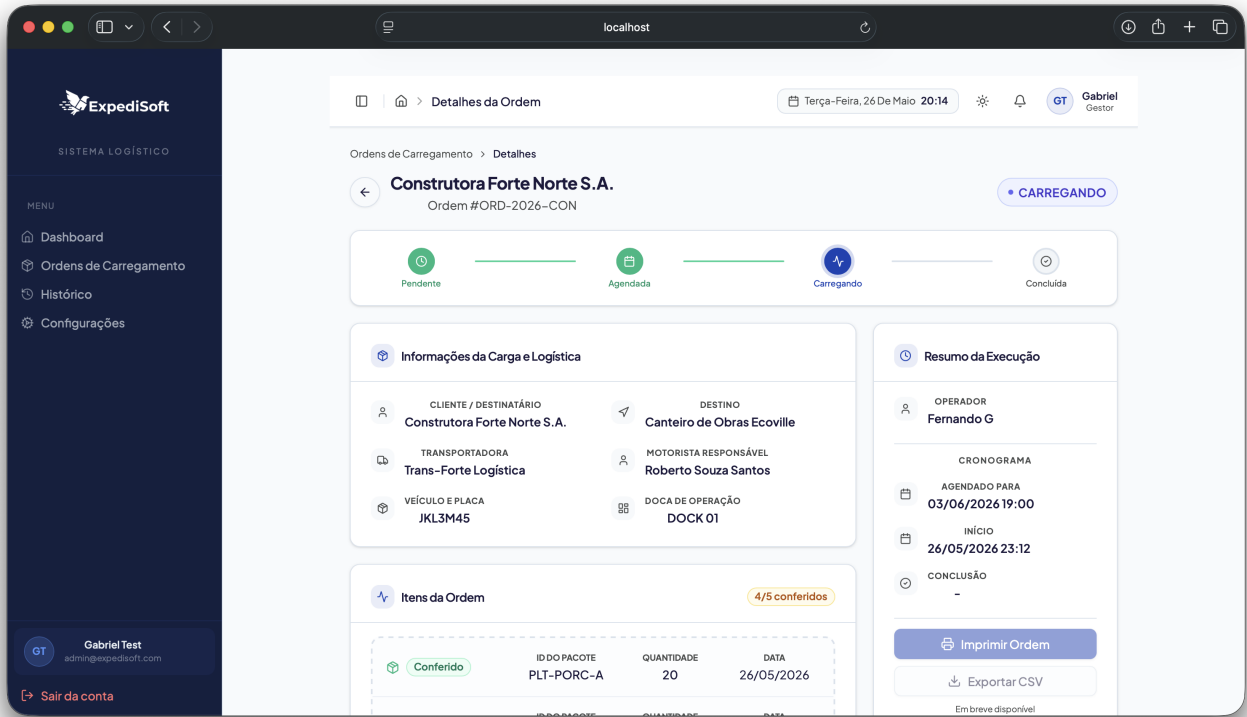


Figura 11 – Web: Ordem em Andamento.

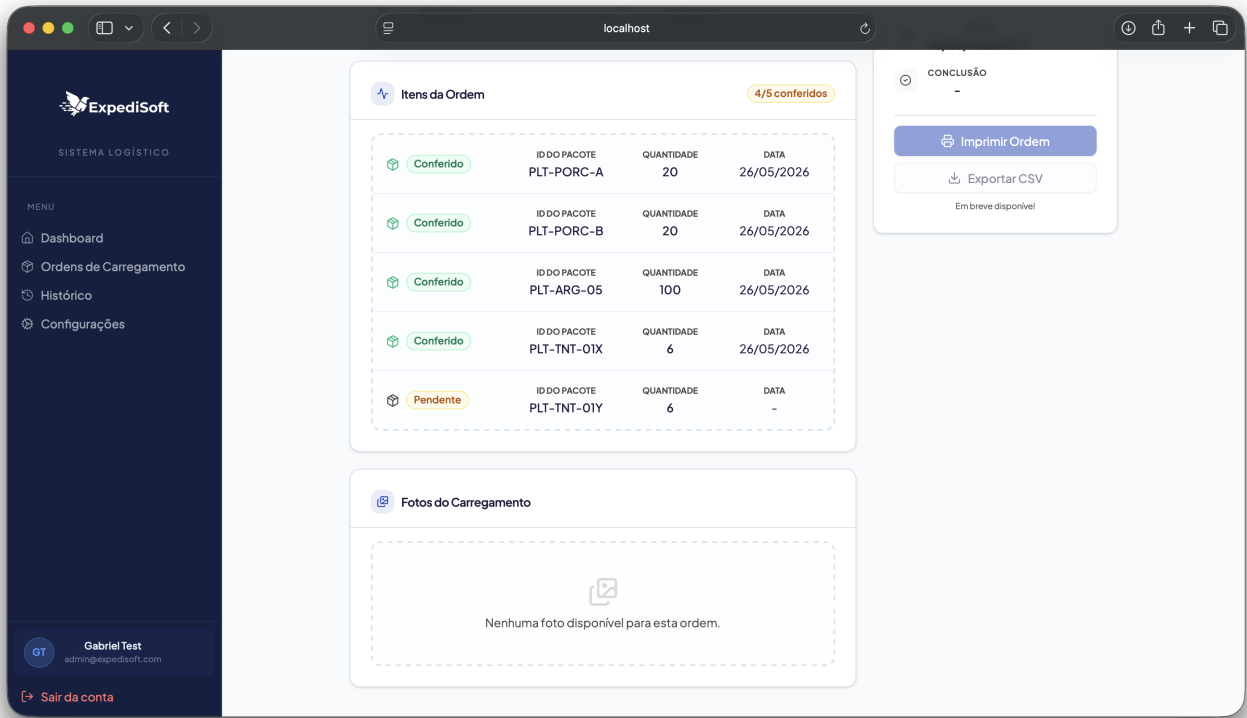


Figura 12 – Web: Ordem em Andamento 2.

3.7.2 Aplicativo Móvel do Operador

A interface *mobile* contém as funcionalidades de detalhamento dos carregamentos, listagem, captura de fotos e o processo de conferência que o operador irá realizar.

- **Autenticação do Aplicativo:** A tela de acesso (*login*), apresentada na Figura 13, restringe o uso do aplicativo a funcionários autorizados. Após a validação das credenciais, o operador é direcionado para a tela inicial do sistema (Figura 14), que funciona como um menu principal para a seleção das atividades. A sessão do usuário permanece ativa no dispositivo de forma contínua, evitando a necessidade de novas solicitações de acesso a cada abertura do aplicativo.



Figura 13 – Mobile: Tela de Autenticação.

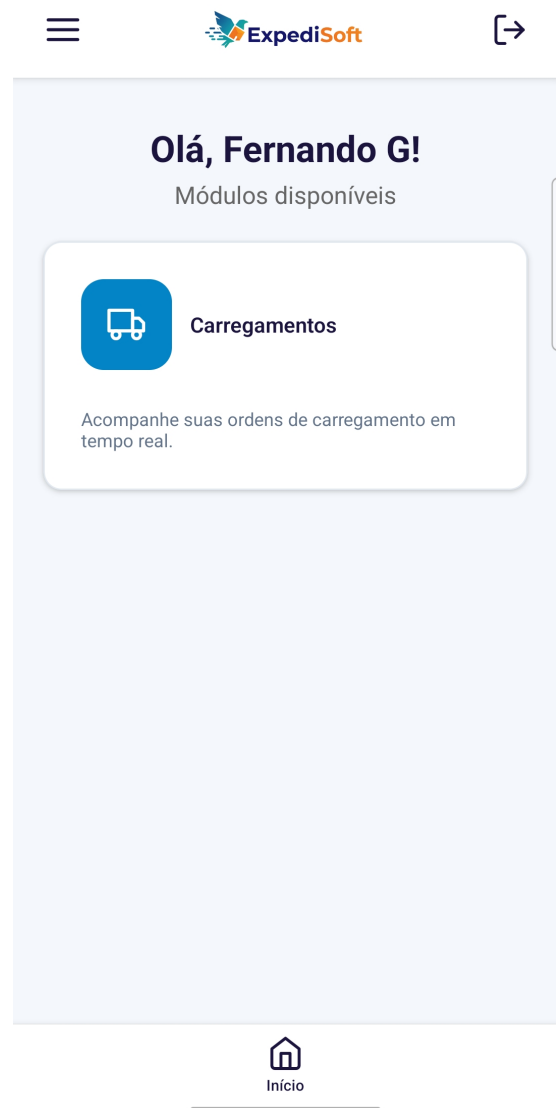


Figura 14 – Mobile: Tela Inicial.

- **Lista de Carregamentos:** Nesta página são exibidas de maneira organizada as ordens de carregamento disponibilizadas para o funcionário que está operando o dispositivo, conforme ilustrado na Figura 15. Por meio dela, o operador consegue conferir rapidamente a situação atual de cada tarefa, o número de identificação da carga e a data e hora programada para a expedição.



Figura 15 – Mobile: Lista de Cargas Atribuídas.

- **Detalhamento e Início do Checklist:** Ao selecionar uma atividade na lista, a tela de detalhes (Figura 16) apresenta as informações operacionais do carregamento, incluindo o número do pedido de origem, o cliente de destino, a placa do veículo transportador e a doca de alocação física. Abaixo desses dados, o aplicativo lista a porcentagem da conferência e detalha as caixas que devem ser embarcadas, exibindo o código do produto, nome comercial, quantidade total e a unidade de medida, além de disponibilizar botões intuitivos para iniciar o fluxo, acionar a câmera ou anexar imagens.

← Detalhes da Carga

Carregamento: JKL3M45
#ORD-2026-CON

Cliente	Destino
Construtora Forte Norte S.A.	Canteiro de Obras Ecoville
Total de Caixas	Transportadora
5	Trans-Forte Logística

Status da Conferência 100%
5 de 5 conferidos.

Caixa 1 PLT-PORC-A

PRODUTO (SKU: MAT-045)
Porcelanato Retificado 60x60 Extra

QUANTIDADE	UN. DE MEDIDA
20	un

Escanear Fotos (4) Concluir

Figura 16 – Mobile: Detalhamento da Ordem Logística.

- **Conferência Automatizada via QR Code:** Durante o processo de conferência física, o operador aciona a câmera nativa do aparelho para realizar a leitura contínua das etiquetas coladas nos volumes. O sistema valida as informações em tempo real; caso o pacote pertença à carga correta, a lista é atualizada e exibida a mensagem de sucesso (Figura 17). Se houver uma tentativa de leitura duplicada, um volume pertencente a outro pedido, erro e/ou inconsistência, a interface dispara uma mensagem de erro (Figura 18), prevenindo falhas humanas.

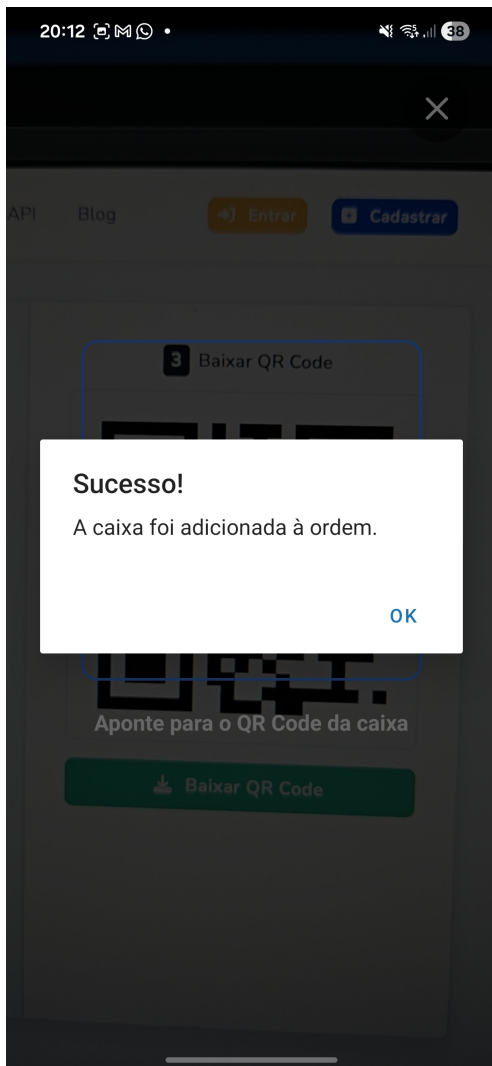


Figura 17 – Mobile: Validação com Sucesso.

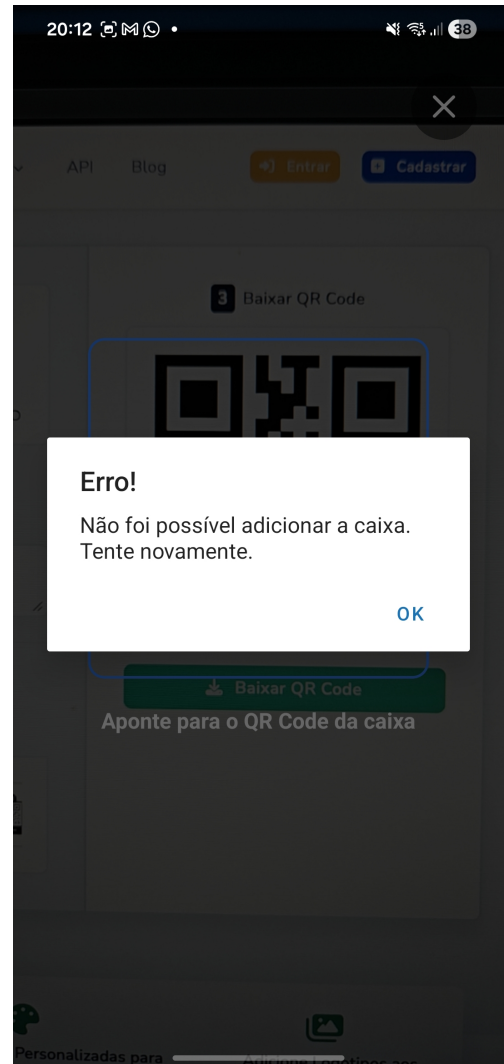


Figura 18 – Mobile: Alerta de Inconsistência.

- **Tratamento de Divergências e Coleta de Fotos:** Se a conferência for finalizada com alguma inconformidade (como a falta de caixas), o aplicativo bloqueia a conclusão direta. O operador torna-se obrigado a relatar uma justificativa textual detalhando o motivo da finalização com divergência (Figura 19). Também é disponibilizado a funcionalidade de capturar fotos comprobatórias da carga (Figura 20). Essas imagens e relatos são enviados após a finalização para a central administrativa para conferência em tempo real pela gerência.



Figura 19 – Mobile: Registro de Justificativa.



Figura 20 – Mobile: Captura de Evidências.

4 CONCLUSÃO

O presente trabalho teve como objetivo desenvolver uma solução para auxiliar o processo de expedição de mercadorias, buscando solucionar limitações identificadas em processos realizados de forma manual, especialmente relacionadas ao acompanhamento das operações, registro de evidências e rastreabilidade das atividades executadas durante o carregamento.

A partir da análise do processo operacional, foi identificada a necessidade de uma ferramenta que permitisse integrar as etapas de planejamento, execução e acompanhamento das ordens de carregamento. Nesse contexto, o sistema Expedisoft foi desenvolvido contemplando uma plataforma *web* destinada à gestão administrativa e um aplicativo móvel direcionado aos operadores responsáveis pela conferência em campo.

A solução desenvolvida atende ao problema proposto ao possibilitar que os gestores acompanhem o andamento das operações, consultem informações detalhadas dos carregamentos e tenham acesso aos registros realizados durante a execução das atividades. Além disso, o aplicativo móvel permite que os operadores realizem a conferência dos volumes por meio da leitura dos códigos identificadores, registrem divergências encontradas e anexem evidências fotográficas, proporcionando maior confiabilidade às informações geradas durante o processo.

A integração entre as interfaces desenvolvidas possibilita centralizar os dados relacionados à expedição, permitindo maior visibilidade sobre o fluxo operacional e facilitando a identificação de inconsistências. Dessa forma, o sistema contribui para reduzir a dependência de controles manuais, melhorar a organização das informações e fornecer suporte à tomada de decisão por parte dos responsáveis pela operação logística.

Durante o desenvolvimento, foram aplicados conceitos relacionados à engenharia de software, arquitetura de sistemas, desenvolvimento de aplicações *web* e móveis, integração entre sistemas e boas práticas de desenvolvimento. A utilização dessas abordagens possibilitou a criação de uma solução estruturada, com separação de responsabilidades e preparada para futuras evoluções conforme as necessidades do processo operacional.

Dessa forma, o Expedisoft estabelece uma base tecnológica para apoiar o processo de expedição de mercadorias, atendendo aos requisitos levantados durante a análise do problema e oferecendo uma alternativa integrada para o acompanhamento, execução e rastreabilidade das operações logísticas.

5 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento do Expedisoft foram identificadas necessidades complementares ao escopo inicialmente definido, relacionadas principalmente à governança da plataforma, segurança das integrações e ampliação das funcionalidades de apoio à operação logística. Devido às limitações de tempo e ao escopo estabelecido para este trabalho, alguns módulos não foram implementados, porém foram considerados como oportunidades de continuidade e evolução da solução desenvolvida.

Entre as funcionalidades identificadas, destaca-se inicialmente a criação de um módulo de auditoria, responsável por disponibilizar aos gestores um histórico das alterações realizadas no sistema. Esse recurso permitiria consultar modificações relacionadas às principais entidades da aplicação, como ordens de carregamento, processos executados em segundo plano e demais registros operacionais. A implementação desse módulo ampliaria a rastreabilidade das ações realizadas pelos usuários e facilitaria processos de análise e conferência das informações.

Também foi identificado como evolução necessária o desenvolvimento de um módulo de configurações administrativas, permitindo centralizar o gerenciamento de usuários, perfis de acesso, credenciais e parâmetros gerais da plataforma. Além disso, esse módulo possibilitaria a configuração das integrações externas, tornando o sistema mais flexível para diferentes ambientes de utilização e reduzindo a necessidade de alterações diretamente na aplicação.

Em relação às integrações com sistemas externos, atualmente a comunicação utiliza um mecanismo de autenticação baseado em *token* fixo. Como evolução, pretende-se implementar um processo de renovação automática das credenciais de integração, incluindo controle de expiração e atualização dos *tokens* utilizados na comunicação. Essa abordagem permitirá aumentar a segurança do processo e criar uma estrutura mais adequada para futuras integrações com diferentes clientes.

Considerando a possibilidade de evolução da solução para um produto utilizado por diferentes organizações, uma melhoria arquitetural relevante consiste na adaptação do sistema para uma abordagem *multi-tenant*. Esse modelo permitiria que diferentes empresas utilizassem a mesma plataforma mantendo o isolamento lógico das informações, configurações específicas e controle individualizado de acesso, tornando a solução mais preparada para cenários de expansão.

Além das melhorias estruturais, novos módulos relacionados ao processo logístico podem ser incorporados. Um exemplo consiste na implementação de um módulo de rastreamento de cargas, utilizando informações provenientes de dispositivos de rastreamento instalados nos veículos de transporte. Essa funcionalidade permitiria acompanhar a movimentação das cargas durante o deslocamento, disponibilizando informações relacionadas à localização e ao status do transporte.

Para viabilizar essa integração de forma escalável, uma possível abordagem seria a utilização do Apache Kafka como plataforma de processamento de eventos. Essa tecnologia permitiria receber, distribuir e processar dados provenientes de diferentes fontes externas, como sistemas de rastreamento, reduzindo o acoplamento entre os componentes e permitindo que novas integrações sejam adicionadas futuramente sem comprometer a estrutura existente.

Outro módulo de expansão consiste no planejamento inteligente de carregamentos. Essa funcionalidade teria como objetivo auxiliar na organização dos volumes dentro do veículo ou contêiner, considerando as dimensões disponíveis e as características dos itens transportados. A partir dessas informações, o sistema poderia gerar uma simulação do carregamento, indicando o posicionamento recomendado dos volumes para melhor aproveitamento do espaço físico, além de disponibilizar uma representação tridimensional da distribuição dos itens.

Soluções existentes no mercado, como plataformas voltadas ao planejamento e otimização logística, demonstram a aplicabilidade desse tipo de recurso ao utilizar informações relacionadas às dimensões dos volumes e dos espaços de transporte para gerar simulações de carregamento. Como evolução do Expedisoft, uma funcionalidade semelhante poderia ser integrada ao processo de expedição, utilizando os dados já disponíveis nas ordens de carregamento para auxiliar o planejamento operacional.

Como possibilidade adicional de evolução, destaca-se a integração do sistema com agentes baseados em modelos de linguagem de grande escala (*Large Language Models - LLMs*). Essa integração poderia auxiliar na consulta de informações operacionais, geração de relatórios, análise de ocorrências e apoio à tomada de decisão, utilizando os dados registrados pela própria plataforma.

Dessa forma, o Expedisoft estabelece uma base tecnológica para apoiar o processo de expedição de mercadorias e possibilitar sua evolução conforme novas demandas sejam identificadas. As funcionalidades não implementadas neste trabalho representam oportunidades de expansão da plataforma, permitindo aprimorar aspectos relacionados à governança, escalabilidade, integração e inteligência aplicada ao processo logístico.

REFERÊNCIAS

- BALLOU, R. H. **Gerenciamento da cadeia de suprimentos/logística empresarial. 5. ed. Porto Alegre: Bookman.** 2006. Disponível em: https://www.google.com.br/books/edition/Gerenciamento_da_Cadeia_de_Suprimentos_5/QAHrq0r6E7cC?hl=pt-BR&gbpv=1&pg=PA3&printsec=frontcover. Acesso em: 13 ago. 2025.
- BECK, K. **TDD desenvolvimento guiado por testes.** [S.l.]: Bookman, 2010.
- BOWERSOX, D. J.; CLOSS, D. J.; COOPER, M. B. **Gestão da cadeia de suprimentos e logística. 4. ed. Rio de Janeiro: Elsevier.** 2013. Disponível em: https://www.google.com.br/books/edition/Gerenciamento_da_Cadeia_de_Suprimentos_5/QAHrq0r6E7cC.
- CLEGG D.; BARKER, R. **CASE Method Fast-track: A RAD Approach.** [S.l.]: Addison-Wesley Publishing Company, 1994.
- DOCKER. **Docker Documentation.** [S.l.]: Docker, Inc. 2025. Disponível em: <https://docs.docker.com/>.
- EVANS, E. **Domain-Driven Design: Tackling Complexity in the Heart of Software.** [S.l.]: Boston: Addison-Wesley, 2004.
- FIGMA. **Figma: The collaborative interface design tool.** [S.l.]: Figma, Inc. 2025. Disponível em: <https://www.figma.com/>.
- GIT. **Git Reference.** [S.l.]: Git. 2025. Disponível em: <https://git-scm.com/doc>.
- GitHub Inc. **GitHub Copilot: Your AI pair programmer.** 2025. Acesso em: 03 nov. 2025. Disponível em: <https://github.com/features/copilot>.
- GROUP, P. G. D. **PostgreSQL: Documentation (Version 15).** [S.l.]: The PostgreSQL Global Development Group. 2025. Disponível em: <https://www.postgresql.org/docs/15/>.
- KIM GENE; HUMBLE, J. D. P. W. J. **The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.** [S.l.]: IT Revolution Press, 2016.
- LARAVEL. **Laravel: The PHP Framework for Web Artisans.** 2025. Disponível em: <https://laravel.com/docs/10.x>.
- META. **React: The library for web and native user interfaces.** [S.l.]: Meta Platforms, Inc. 2025. Disponível em: <https://react.dev/>.
- REACT. **React Native: Learn once, write anywhere.** [S.l.]: Meta Platforms, Inc. 2025. Disponível em: <https://reactnative.dev/>.
- SHADCN. **shadcn/ui.** [S.l.]: shadcn. 2025. Disponível em: <https://ui.shadcn.com/>.
- SOMMERVILLE, I. **Software Engineering. 9ª ed.** [S.l.]: Pearson, 2011.
- VASYUKOV A. PETROV, I. **Using Computing Containers and Continuous Integration to Improve Numerical Research Reproducibility.** 2018. Disponível em: <https://ijcjournal.org/InternationalJournalOfComputer/article/view/1249/503>.
- VITE. **Vite: Next Generation Frontend Tooling.** [S.l.]: Vite. 2025. Disponível em: <https://vitejs.dev/>.

APÊNDICES

APÊNDICE A – Contrato de Dados

Este apêndice apresenta os *payloads* (contratos) da API de integração do sistema, sendo eles: Ordem de Carregamento, Usuários e Docas.

Listagem 1 – Modelo JSON de Usuários

```
1 {  
2   "source_system": "SISTEMA_A",  
3   "user": {  
4     "external_id": "123463",  
5     "name": "teste user",  
6     "email": "teste@email.com"  
7   }  
8 }
```

Fonte: Autoria própria (2026).

Listagem 2 – Modelo JSON de Doca e Localização

```
1 {  
2   "source_system": "SAP",  
3   "dock": {  
4     "external_id": "ERP-DOCK-123",  
5     "dock_code": "DOCK 01",  
6     "description": "Pátio A",  
7     "location": "Matriz"  
8   }  
9 }
```

Fonte: Autoria própria (2026).

Listagem 3 – Modelo JSON das Ordens de Carregamento

```

1  {
2    "source_system": "SAP",
3    "loadingOrder": {
4      "external_id": "TESTE-2025",
5      "issue_date": "2025-11-05",
6      "status": "pending",
7      "customer": {
8        "external_id": "999888777",
9        "name": "Tech Store Brasil",
10       "document": "9998887770001",
11       "email": "logistica@techstore.com.br",
12       "phone": "1133224455"
13     },
14     "destination": {
15       "external_id": "WH-CTBA-01",
16       "name": "Centro de Distribuição Curitiba",
17       "address": "Rua das Indústrias, 500",
18       "postal_code": "80010000",
19       "city": "Curitiba",
20       "state": "PR"
21     },
22     "carrier": {
23       "external_id": "202",
24       "name": "Rapidão Trans",
25       "document": "202202"
26     },
27     "vehicle": {
28       "external_id": "ABC1234",
29       "vehiclePlate": "ABC1234",
30       "model": "Mercedes-Benz Accelo"
31     },
32     "driver": {
33       "external_id": "455",
34       "name": "João Silva",
35       "document": "455",
36       "phone": "41999887766"
37     },
38     "items": [
39       {
40         "product_sku": "EL-001",
41         "product_description": "Smart TV 55 pol",
42         "quantity": 20,
43         "unit": "un",
44         "packages": [
45           { "unique_package_code": "PKG-TV-101-A",
46             "quantity_in_package": 10 },
47           { "unique_package_code": "PKG-TV-101-B",
48             "quantity_in_package": 10 }
49         ]
50       }
51     ]
52   }

```

APÊNDICE B – Diagrama Entidade Relacionamento Completo

Abaixo estão os diagramas do banco de dados, divididos entre tabelas de cadastro, tabelas de operação, tabelas do sistema e o diagrama completo.

Visite a imagem do Diagrama do Banco Completo para mais informações.

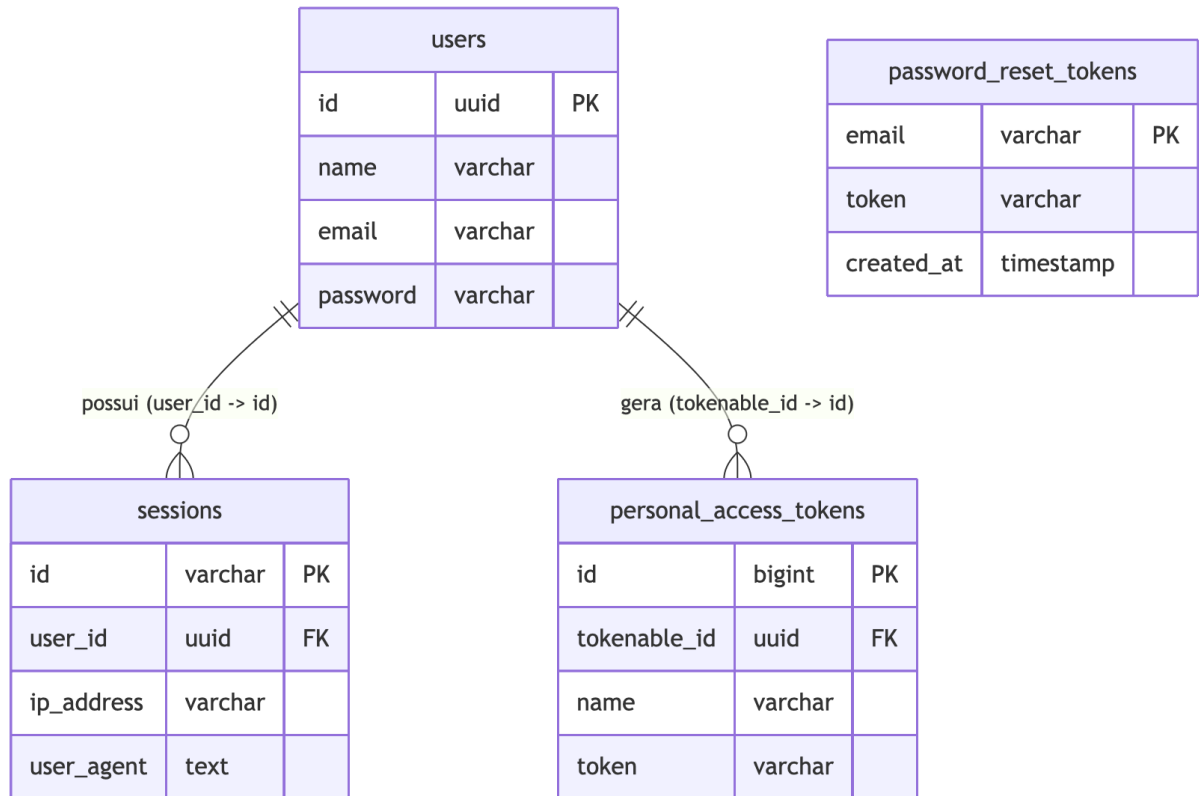


Figura 21 – Recorte de Tabela - 01

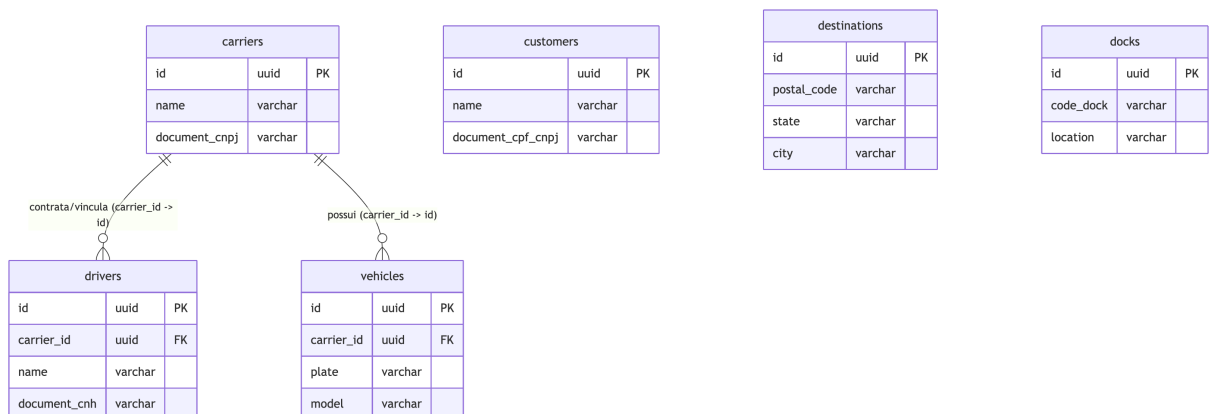


Figura 22 – Recorte de Tabela - 02

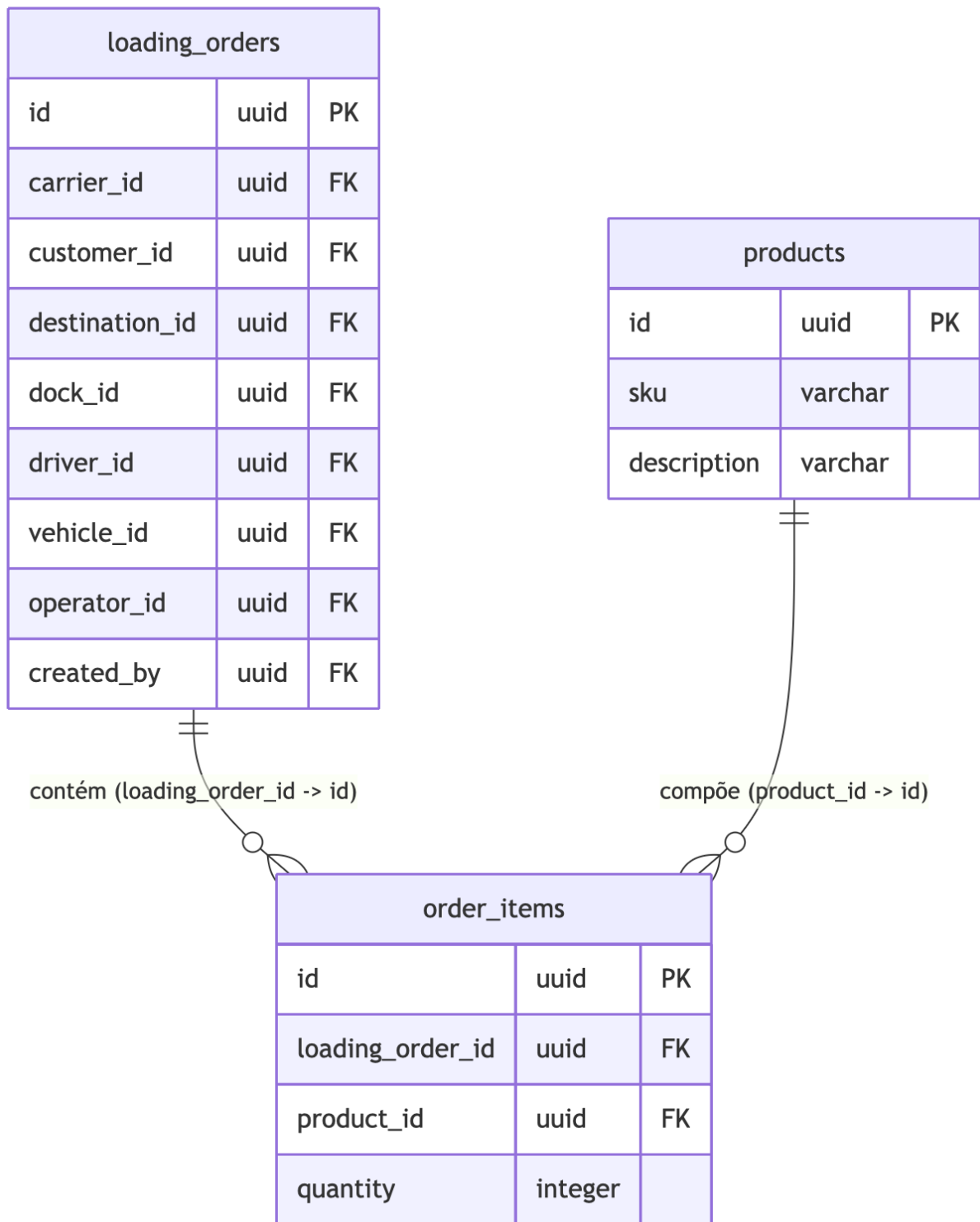


Figura 23 – Recorte de Tabela - 03

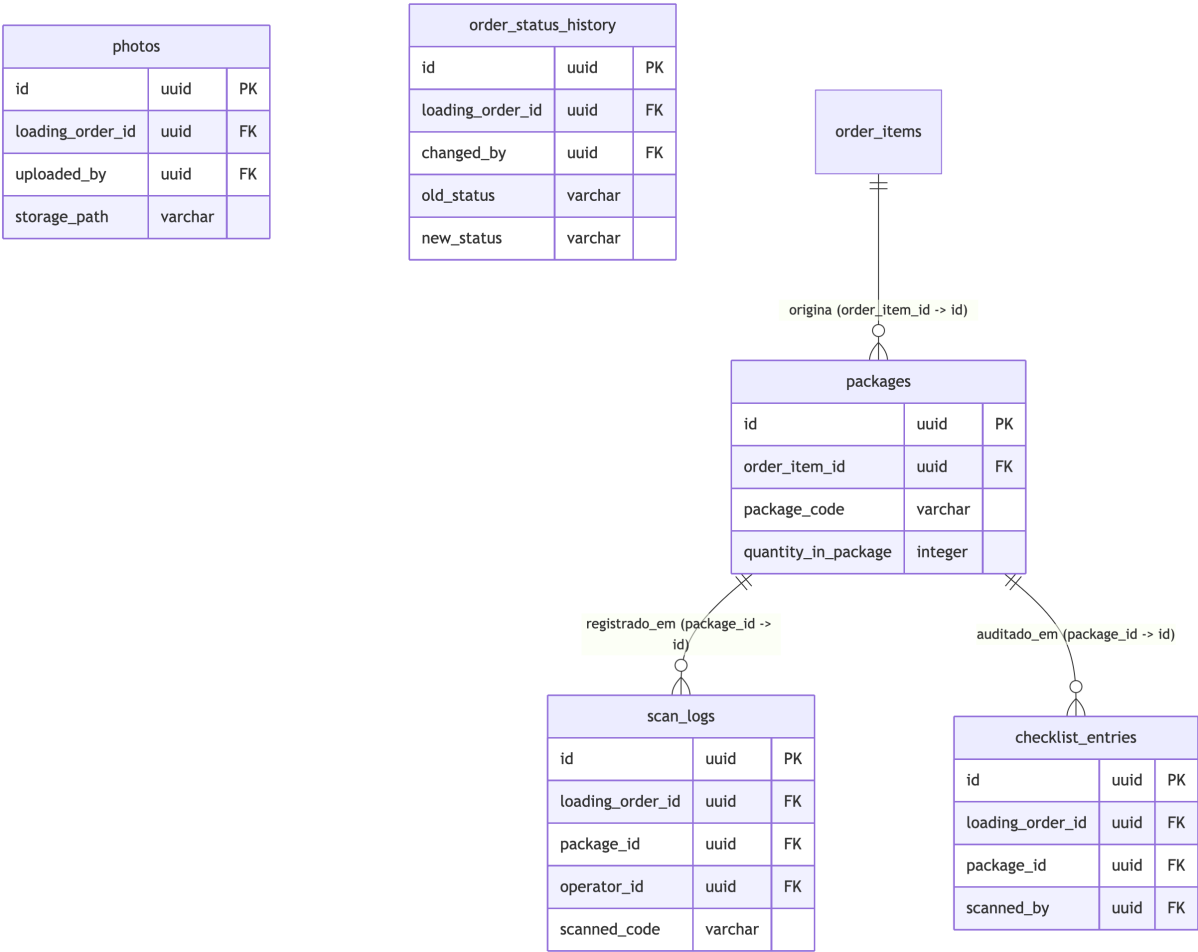


Figura 24 – Recorte de Tabela - 04