

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL FERNANDO SEFFRIN

**EXPEDISOFT - SISTEMA PARA A GESTÃO E RASTREABILIDADE DO
CARREGAMENTO DE MERCADORIA**

GUARAPUAVA

2025

GABRIEL FERNANDO SEFFRIN

**EXPEDISOFT - SISTEMA PARA A GESTÃO E RASTREABILIDADE DO
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

2025



[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.

RESUMO

O processo de expedição de mercadorias é uma etapa crítica na cadeia logística, onde falhas operacionais, como o carregamento incorreto, podem gerar prejuízos significativos e comprometer a credibilidade da empresa. Diante desse cenário, este trabalho propõe o desenvolvimento do Expedisoft, um sistema integrado *web* e *mobile* voltado ao controle e rastreabilidade do carregamento de mercadorias. O projeto busca mitigar falhas recorrentes, como volumes esquecidos ou inserção indevida de caixas, por meio de conferência digital, leitura de códigos, registro fotográfico e histórico detalhado das operações. O objetivo geral consiste em criar uma ferramenta capaz de auxiliar gestores e operadores na execução segura e auditável do processo de expedição. A metodologia utilizada inclui levantamento de requisitos baseado em observação prática, priorização pelo método MoSCoW, prototipação no Figma, definição da arquitetura utilizando práticas de engenharia de software e modelagem do banco de dados em PostgreSQL com identificadores UUID. A solução será implementada com Laravel no backend, React na plataforma *web* e React Native no aplicativo *mobile*. Até o momento, foram concluídos o mapeamento dos requisitos essenciais, a definição dos modelos de integração com o ERP via API REST, o diagrama completo do banco de dados e os protótipos de interface que representam os fluxos operacionais críticos. Esses elementos fornecem a base estrutural para o desenvolvimento da solução final e garantem que o sistema seja escalável, seguro e preparado para integração com sistemas corporativos. Espera-se que o sistema resulte em maior confiabilidade do processo de expedição, redução de falhas humanas, aumento da rastreabilidade e suporte à tomada de decisão baseada em dados. O Expedisoft possui potencial para melhorar significativamente a qualidade do carregamento e a eficiência logística, podendo ser adaptado futuramente para atender múltiplas empresas e cenários operacionais distintos.

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

ABSTRACT

The cargo expedition process is a critical stage in the logistics chain, where operational failures, such as incorrect loading, can generate significant losses and compromise a company's credibility. In this context, this work proposes the development of Expedisoft, an integrated web and mobile system designed to support the control and traceability of cargo loading activities. The project aims to reduce recurring issues, such as forgotten volumes or the incorrect inclusion of boxes, through digital verification, code scanning, photographic documentation, and detailed operational history. The main objective is to provide a tool capable of assisting managers and operators in executing the expedition process with greater accuracy, security, and auditability. The methodology includes requirement gathering based on practical observation, requirement prioritization using the MoSCoW method, interface prototyping in Figma, architectural definition based on software engineering practices, and database modeling in PostgreSQL using UUID identifiers. The solution will be implemented using Laravel for the backend, React for the web platform, and React Native for the mobile application. To date, the essential requirements have been mapped, the integration models with the ERP via REST API have been defined, the complete database diagram has been developed, and interface prototypes representing the system's critical operational flows have been created. These elements provide the structural foundation for the remaining stages of development, ensuring that the system is scalable, secure, and prepared for integration with corporate environments. The expected outcomes include increased reliability in the expedition process, reduction of human errors, improved traceability, and enhanced support for data-driven decision-making. Expedisoft demonstrates strong potential to significantly improve loading quality and logistical efficiency and may be adapted in the future to support multiple companies and diverse operational scenarios.

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

LISTA DE FIGURAS

Figura 1 – Exemplo do fluxo do sistema.	12
Figura 2 – Mobile: Tela de conferência de itens 01.	22
Figura 3 – Mobile: Tela de conferência de itens 02.	23
Figura 4 – Mobile: Tela com carregamento finalizado.	24
Figura 5 – Web: Tela para o detalhamento do carregamento.	25
Figura 6 – Recorte da entidade de carregamento.	26
Figura 7 – Mobile: Tela de login.	33
Figura 8 – Mobile: Tela home page.	34
Figura 9 – Mobile: Tela lista de carregamentos.	35
Figura 10 – Mobile: Tela de upload de fotos.	36
Figura 11 – Mobile: Tela para informar justificativas e observações.	37
Figura 12 – Mobile: Tela de aviso de inconsistência no carregamento.	38
Figura 13 – Web: Tela de login.	39
Figura 14 – Web: Tela home page / Dashboard.	39
Figura 15 – Web: Tela para agendar carregamento.	40
Figura 16 – Web: Modal para agendar carregamento.	40
Figura 17 – Web: Tela de histórico de carregamentos.	41
Figura 18 – Diagrama geral do banco de dados	42

LISTA DE TABELAS

Tabela 1 – Funcionalidades <i>Must Have</i>	18
Tabela 2 – Funcionalidades <i>Should Have</i>	18
Tabela 3 – Funcionalidades <i>Could Have</i>	19
Tabela 4 – Funcionalidades <i>Won't Have</i>	19

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Modelo JSON Ordens de Carregamento	27
Listagem 2 – Modelo JSON Usuários	28
Listagem 3 – Modelo JSON Doca/Localização	28

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>
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>
WMS	<i>Warehouse Management System</i>

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivos	9
1.1.1	Objetivo Geral	9
1.1.2	Objetivos Específicos	9
1.1.2.1	Plataforma Mobile	9
1.1.2.2	Plataforma Web	10
2	DETALHAMENTO DA SOLUÇÃO	11
3	MATERIAIS E MÉTODOS	13
3.1	Materiais	13
3.2	Métodos	15
4	RESULTADOS PARCIAIS	17
4.1	Levantamento de Requisitos	17
4.2	Priorização dos Requisitos	17
4.3	Integração Sistema e <i>Enterprise Resource Planning</i> (ERP) Cliente	19
4.3.1	Modelo de Dados: Ordens de Carregamento	19
4.3.2	Modelo de Dados: Usuários	20
4.3.3	Modelo de Dados: Docas	20
4.4	Protótipos de Interface	21
4.4.1	Telas <i>Mobile</i>	21
4.4.2	Telas <i>Web</i>	24
4.5	Banco de Dados	25
4.5.1	Estrutura das Tabelas	26
5	CONSIDERAÇÕES FINAIS	29
	REFERÊNCIAS	30
	APÊNDICES	31
	APÊNDICE A – TELAS COMPLEMENTARES DO SISTEMA E BANCO DE DADOS	33

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 garantir a robustez 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 robustez do sistema..

Em síntese, este projeto visa o desenvolvimento de uma ferramenta confiável e escalável 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, plataforma Web e plataforma Mobile.

1.1.2.1 Plataforma Mobile

- Desenvolver um módulo para identificação das mercadorias, auxiliando na verificação da caixa correta para o carregamento;
- Implementar um checklist de carregamento, permitindo 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.

1.1.2.2 Plataforma 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 DETALHAMENTO DA SOLUÇÃO

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

O fluxo de trabalho se inicia na plataforma 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 da aplicação.

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.

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 plataforma 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 na plataforma, 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.

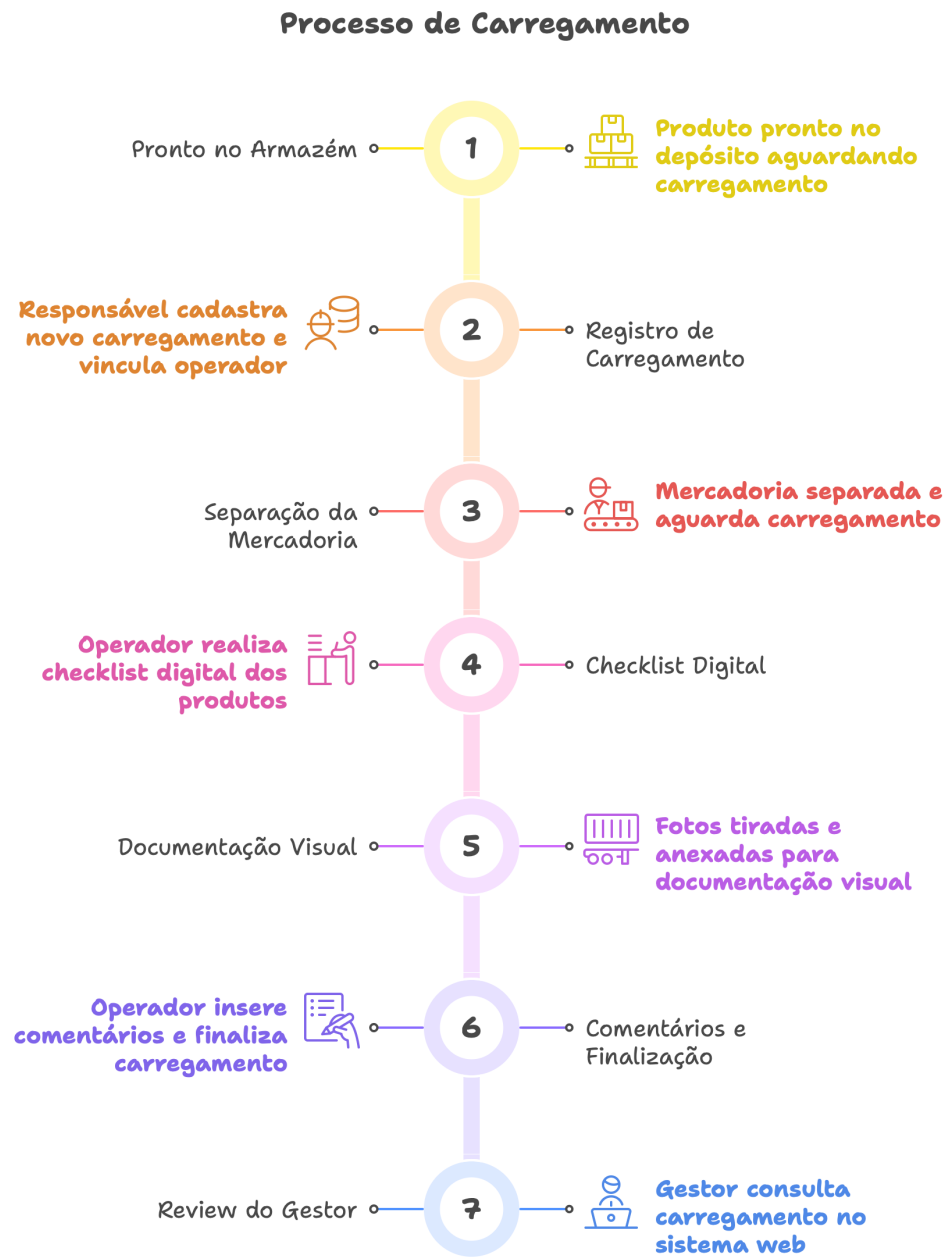


Figura 1 – Exemplo do fluxo do sistema.

3 MATERIAIS E MÉTODOS

A seguir, um descritivo de como o trabalho será conduzido e quais tecnologias serão utilizadas.

3.1 Materiais

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

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

Justificativa: 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):

Justificativa: 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):

Justificativa: 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.

Justificativa: 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):

Justificativa: 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):

Justificativa: O *Docker* será 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, garantindo a portabilidade e simplificando o processo de implantação (*deploy*⁵) (VASYUKOV A. PETROV, 2018).

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

Justificativa: 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*.

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

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

- **Documentação API: Swagger:**

Justificativa: 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 docu-

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.

mentação interativa que permite a qualquer desenvolvedor visualizar e testar os *end-points*⁶ diretamente pelo navegador.

3.2 Métodos

O desenvolvimento do sistema seguirá 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 será elaborado com base na experiência prévia do autor no setor em uma empresa exportadora de grande porte. Por meio da observação prática das rotinas operacionais, foram identificadas as principais dificuldades do setor de expedição e as necessidades do processo, adaptando as dificuldades em uma solução descentralizada e flexível a diferentes empresas e cenários. Essa abordagem é comumente utilizada em projetos aplicados, conforme destaca SOMMERVILLE (2011), quando o pesquisador tem acesso direto ao ambiente de negócio e pode derivar os requisitos a partir da análise de sua própria vivência e conhecimento técnico do domínio. Os requisitos funcionais e não funcionais serão documentados, categorizados e priorizados de acordo com o impacto no processo de expedição.

- **Priorização dos Requisitos**

Após definidos, os requisitos serão 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.

- **Prototipação de Interfaces**

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

- **Metodologia de Desenvolvimento**

Para a condução do projeto, optou-se por uma abordagem de desenvolvimento iterativo e incremental. Essa metodologia permite que o sistema seja construído em ciclos curtos de implementação, teste e validação (Pressman (2011)).

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

- **Modelagem e Arquitetura**

A modelagem da aplicação será 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 favorece a separação de responsabilidades e a comunicação clara entre as camadas do sistema Evans (2004).

- **Testes e Garantia de Qualidade**

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

- **Gerenciamento do Projeto**

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

4 RESULTADOS PARCIAIS

Este capítulo contém os resultados parciais do projeto obtidos até o momento, abrangendo o levantamento de requisitos, a definição da arquitetura de dados e a prototipação das interfaces.

4.1 Levantamento de Requisitos

O Expedisoft é um sistema integrado composto por uma plataforma *web* e um aplicativo *mobile*, projetado para gerenciar e rastrear o processo de expedição de mercadorias em operações logísticas. Seu escopo principal abrange desde o agendamento dos carregamentos até a conferência final dos itens no veículo. A plataforma *web* servirá como o centro de controle para os gestores, permitindo o cadastro de usuários, a criação de ordens de carregamento e o monitoramento em tempo real do status de cada operação. Já o aplicativo *mobile* será a ferramenta de trabalho dos operadores na doca, facilitando a execução das tarefas com foco na redução de erros manuais.

Tecnicamente, o sistema deve auxiliar na integridade do processo de conferência. O aplicativo *mobile* deverá utilizar a câmera do dispositivo para a leitura de códigos *Quick Response Code* (QR Code) de cada volume (caixa, pacote, *box*, etc.), validando se a caixa pertence ao pedido em questão. Para assegurar a qualidade da operação, o sistema impedirá a finalização de carregamentos com divergências sem que o operador registre uma justificativa obrigatória. Além disso, o aplicativo permitirá o *upload* de fotos da carga finalizada, que serão armazenadas e vinculadas ao registro do carregamento na plataforma *web*, servindo como evidência digital auditável.

Para a integração de dados, o sistema disponibilizará uma API *Representational State Transfer* (REST) com *endpoints* específicos (método *POST*) para que o ERP do cliente possa enviar, de forma automatizada, as cargas de dados necessárias, como as ordens de carregamentos e seus respectivos itens, seguindo um layout *JavaScript Object Notation* (JSON) pré-definido pelo Expedisoft.

Em relação às funcionalidades fora do escopo, o sistema não realizará o gerenciamento de estoque (*Warehouse Management System* (WMS)) nem a emissão de documentos fiscais, focando exclusivamente na validação e rastreabilidade do processo de expedição.

4.2 Priorização dos Requisitos

Para a priorização do escopo do projeto, conforme detalhado na seção 3.2, foi utilizada a técnica *MoSCoW*, que caracteriza os requisitos em quatro níveis de prioridade.

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
RF1	Geral	API de Integração (<i>Backend</i>)	Desenvolvimento de <i>endpoints</i> REST (método <i>POST</i>) para receber dados de ordens de carregamento.
RF2	Geral	Autenticação e Autorização	Sistema seguro de <i>login</i> para diferenciar perfis de acesso (Gestor <i>Web</i> e Operador).
RF3	App <i>Mobile</i>	Leitura de Códigos	Funcionalidade para leitura de QR Code utilizando a câmera do dispositivo <i>mobile</i> .
RF4	App <i>Mobile</i>	Validação em Tempo Real	Lógica no aplicativo que verifica se o volume lido corresponde à ordem de carregamento ativa, mitigando erros.
RF5	Plataforma <i>Web</i>	Gestão de Ordens	Interface para visualizar as ordens de carregamento importadas e acompanhar seu status.

Tabela 1 – Funcionalidades *Must Have*

ID	Categoria	Requisito	Descrição
RF6	App <i>Mobile</i>	Documentação Visual	Funcionalidade para capturar e/ou enviar fotos da carga como comprovante visual.
RF7	App <i>Mobile</i>	Registro de Divergências	Fluxo que obriga o operador a inserir uma justificativa em texto ao finalizar uma carga com divergências.
RF8	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
RF9	Plataforma Web	<i>Dashboard</i>	Painel inicial com <i>cards</i> e indicadores/gráficos básicos de desempenho (ex: quantidade de carregamentos programados, tempo médio de carregamento).
RF10	Geral	Notificações	Alertas por <i>e-mail</i> ou <i>push notification</i> para gestores quando uma carga com divergência for finalizada.
RF11	Plataforma Web	Integração	Página dedicada ao monitoramento de integração dos dados no sistema.

Tabela 3 – Funcionalidades *Could Have*

ID	Categoria	Requisito	Descrição
RF12	Geral	Integração Ativa	Sistema buscando dados no ERP.
RF13	Geral	Funcionalidade WMS	Controle de estoque e endereçamento.
RF14	Geral	Documentos	Emissão de documentos fiscais (NF-e).

Tabela 4 – Funcionalidades *Won't Have*

4.3 Integração 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 dois modelos (ordens para carregamento e usuários) de integração baseados em JSON, permitindo que a plataforma receba dados estruturados independente da tecnologia utilizada pelo cliente.

Essa abordagem visa garantir compatibilidade e simplicidade na comunicação entre sistemas, utilizando o protocolo *Hypertext Transfer Protocol* (HTTP) e o método *POST* em *end-points* 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 a plataforma, assegurando 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, visando uma comunicação segura.

4.3.1 Modelo de Dados: Ordens de Carregamento

O padrão definido para os dados de ordens de carregamento contém campos essenciais para a integração, como o identificador da ordem, código do cliente, as informações do transporte, os itens e o status da expedição. A Listagem 1 representa o modelo.

Observações:

- *externalId*: mantém o vínculo da ordem com o identificador do ERP;
- *uniquePackageCode*: utilizado para a leitura via QR Code no *app mobile*;
- Formato flexível: permite adicionar campos sem quebrar a estrutura;
- Exemplo de *endpoint*: `POST .../api/integracao/ordens-carregamento`.

4.3.2 Modelo de Dados: Usuários

Para a integração de usuários na plataforma, foi elaborado um modelo de dados representado na Listagem 2, que detalha as principais informações incluídas no processo de sincronização. Destacam-se o identificador único do usuário, nome, *e-mail* e cargo, elementos essenciais para controle de permissões e autenticação. Dessa forma, o gerenciamento de usuários que possuem acesso à plataforma torna-se mais centralizado e eficiente, permitindo que todos os colaboradores sejam cadastrados, ainda que apenas parte deles tenha acesso efetivo ao sistema.

Observações:

- *userId*: mantém o vínculo do usuário com o identificador do ERP;
- *name*: utilizado para obter o nome do usuário no sistema;
- *email*: necessário para realização de *login*;
- Exemplo de *endpoint*: `POST .../api/integracao/usuarios`.

4.3.3 Modelo de Dados: Docas

Utilizado como cadastro na plataforma *web* para agendamento do carregamento, tendo informações como identificador da doca, descrição e local, conforme exemplificado na Listagem 3.

Observações:

- *dockId*: identificador único da doca;
- *description*: descrição da localização da doca;
- *location*: localização da doca;
- Exemplo de *endpoint*: `POST .../api/integracao/docas`.

4.4 Protótipos de Interface


Nesta etapa, foram desenvolvidos protótipos de interface navegáveis utilizando a ferramenta *Figma*, com o objetivo de validar o fluxo de navegação e a experiência do usuário antes da implementação do código.

A seguir, são apresentadas as telas principais que representam o fluxo crítico de operação (conferência e finalização). As demais telas complementares, como *login*, listagens iniciais e configurações de agendamento, encontram-se disponíveis no Apêndice A.

4.4.1 Telas *Mobile*

O aplicativo *mobile* foca na operação de campo e na agilidade do processo. As Figura 2, Figura 3 e Figura 4 demonstram, respectivamente, o início do processo de conferência dos itens, a marcação de progresso no *checklist* e a tela de sucesso na finalização do carregamento.

Tela 3

 Detalhes do Carregamento

Carregamento: #010/25

PP0005789

Cliente
Zé Mercados

Destino
Guarapuava - Centro

Total de Caixas
3

Unidade de Medida
30 [kg]


Itens a conferir

Progresso: 2/3

Nº Caixa: 1

235458965 - Descrição do item


30 und X X X

 Conferido

Nº Caixa: 2

235458965 - Descrição do item

30 und X X X


 Conferido


Nº Caixa: 3

235458965 - Descrição do item

30 und X X X

Conferir

 Escanear

 Fotos

Concluir

Figura 2 – Mobile: Tela de conferência de itens 01.

Tela 7

← Detalhes do Carregamento

Carregamento: #010/25 PP0005789

Cliente Zé Mercados	Destino Guarapuava - Centro
Total de Caixas 3	Unidade de Medida 30 [kg]

Itens a conferir Progresso: 3/3

Nº Caixa: 1
235458965 - Descrição do item

30 und	X	X	X
--------	---	---	---

☒ Conferido

Nº Caixa: 2
235458965 - Descrição do item


30 und	X	X	X
--------	---	---	---


☒ Conferido

Nº Caixa: 3
235458965 - Descrição do item

30 und	X	X	X
--------	---	---	---

☒ Conferido

 Escanear

 Fotos

Concluir

Figura 3 – Mobile: Tela de conferência de itens 02.

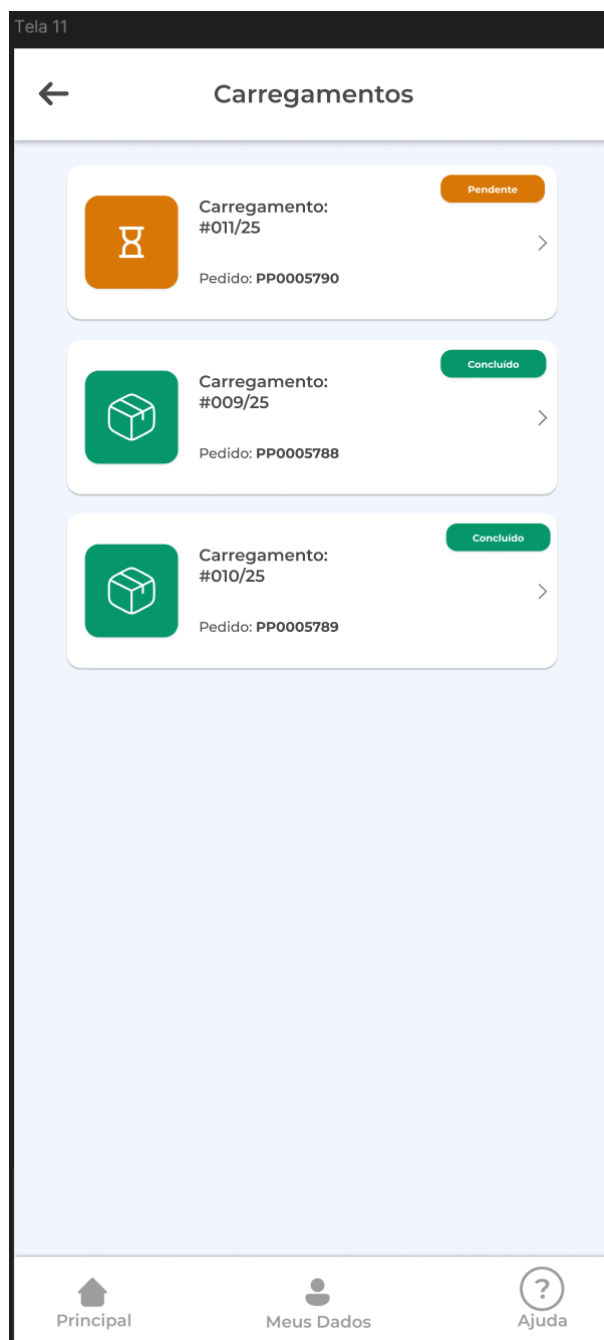


Figura 4 – Mobile: Tela com carregamento finalizado.

4.4.2 Telas Web

A interface *web* é voltada para a gestão e auditoria. A Figura 5 apresenta a tela de detalhamento de um carregamento, onde o gestor pode visualizar o histórico, as evidências fotográficas e as justificativas inseridas pelo operador. Outras telas do fluxo administrativo, como o agendamento de cargas e o *dashboard*, podem ser consultadas no Apêndice A.

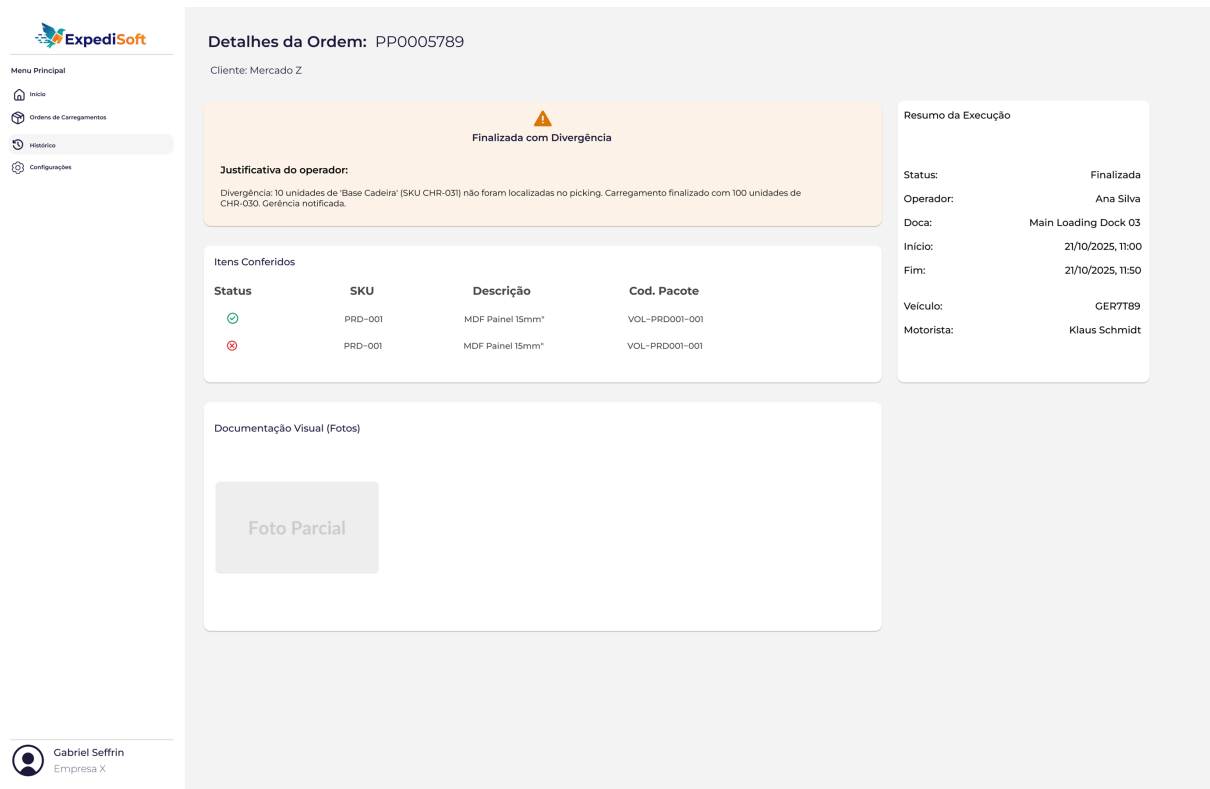


Figura 5 – Web: Tela para o detalhamento do carregamento.

4.5 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.

A Figura 6 apresenta um recorte das principais relações operacionais do sistema, destacando a interação entre as tabelas de carregamento (*loadings*) e pacotes (*packages*). O diagrama completo de entidade-relacionamento (DER), contendo todas as tabelas, atributos e chaves, pode ser visualizado no Apêndice A.

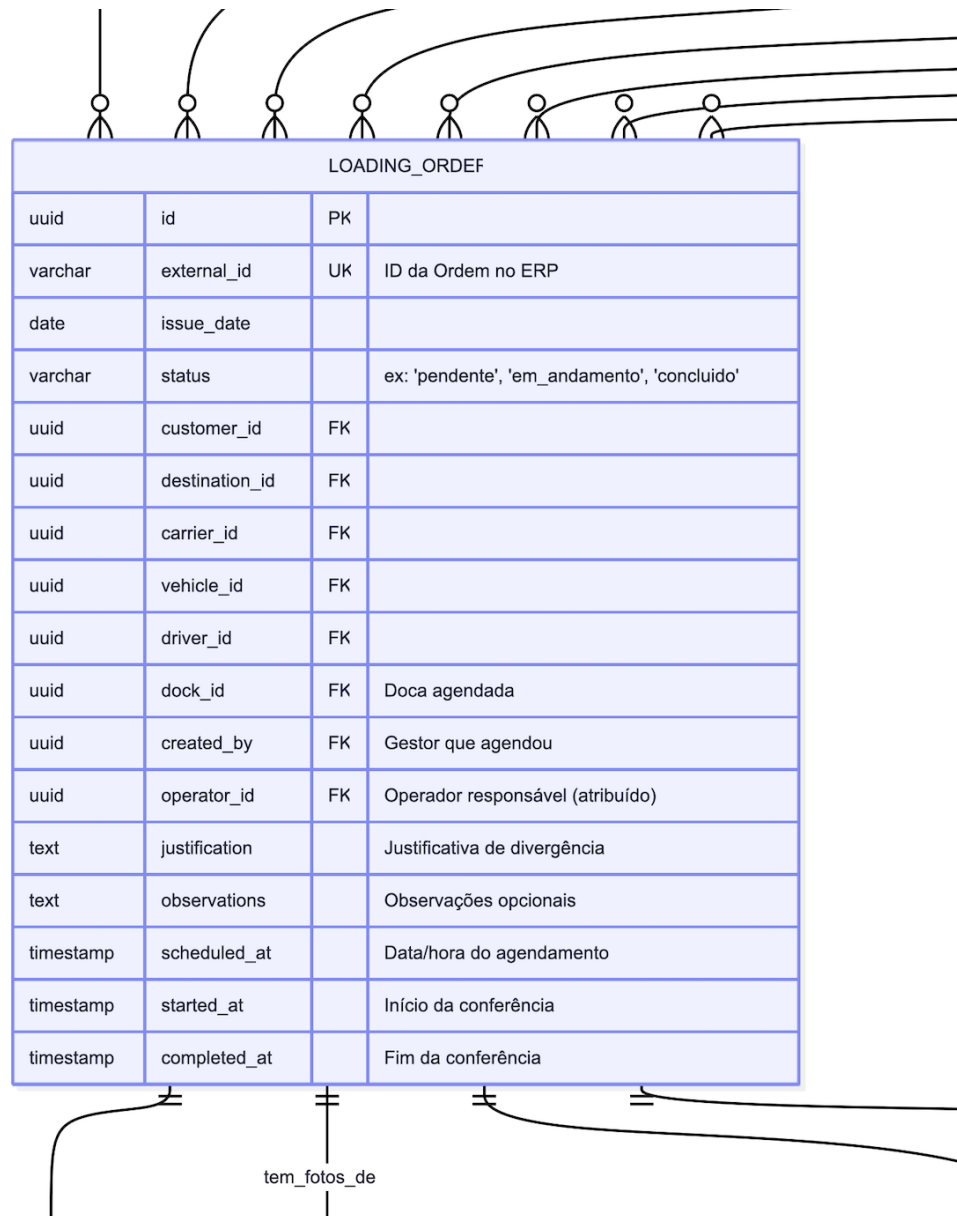


Figura 6 – Recorte da entidade de carregamento.

4.5.1 Estrutura das Tabelas

- **Estrutura Organizacional:** *USERS* (usuários) e *DOCKS* (docas).
- **Entidades Externas:** *CUSTOMERS*, *CARRIERS*, *VEHICLES*, *DRIVERS* e *DESTINATIONS*.
- **Entidades Operacionais:**
 - **LOADING_ORDERS:** ordens de carregamento recebidas do ERP.
 - **ORDER_ITEMS:** itens pertencentes a cada ordem.
 - **PACKAGES:** volumes físicos com código único para leitura via QR Code.

- **Entidades de Auditoria:** *CHECKLIST_ENTRIES*, *PHOTOS*, *ORDER_STATUS_HISTORY* e *INTEGRATION_LOGS*.

A modelagem reflete uma arquitetura alinhada às necessidades reais do setor de expedição, permitindo integração confiável com o ERP e rastreabilidade individualizada.

Listagem 1 – Modelo JSON Ordens de Carregamento

```

1  {
2    "loadingOrder": {
3      "externalId": "PED-2025-00123",
4      "issueDate": "2025-10-20",
5      "status": "Waiting for Loading",
6      "customer": {
7        "name": "Alfa Wood Industry S.A.",
8        "taxId": "12.345.678/0001-99",
9        "address": "Araucárias Street, 1500, Guarapuava, PR"
10     },
11     "destination": {
12       "location": "Paranaguá Port",
13       "address": "Port Avenue, 2000, Paranaguá, PR"
14     },
15     "carrier": {
16       "name": "Fast Logistics Ltd.",
17       "vehiclePlate": "ABC1D23",
18       "driver": {
19         "name": "John Silva",
20         "taxId": "987.654.321-00",
21         "phone": "+55 42 99999-8888"
22       }
23     },
24     "items": [
25       {
26         "sku": "PRD-001",
27         "description": "MDF Panel 15mm",
28         "quantity": 120,
29         "uniquePackageCode": "VOL-PRD001-001"
30       },
31       {
32         "sku": "PRD-002",
33         "description": "Marine Plywood 18mm",
34         "quantity": 80,
35         "uniquePackageCode": "VOL-PRD002-001"
36       }
37     ]
38   }
39 }
```

Fonte: Autoria própria (2025).

Listagem 2 – Modelo JSON Usuários

```
1 {  
2   "user_id": "USR-045",  
3   "name": "Carlos Mendes",  
4   "email": "carlos.mendes@empresa.com",  
5   "position": "Gestor de Logística"  
6 }
```

Fonte: Autoria própria (2025).

Listagem 3 – Modelo JSON Doca/Localização

```
1 {  
2   "dockId": "DOCK-03",  
3   "description": "Main Loading Dock 03",  
4   "location": "Warehouse A – Sector 2"  
5 }
```

Fonte: Autoria própria (2025).

5 CONSIDERAÇÕES FINAIS

Este projeto visa mitigar erros operacionais nos setores de expedição, como carregamento incorreto de pedidos, a ausência de rastreabilidade e documentação das etapas e atores existentes no processo.

Para alcançar esse objetivo, a primeira etapa do trabalho concentrou-se no planejamento e na arquitetura da solução. Os resultados alcançados até o momento incluem um levantamento detalhado de requisitos, a priorização de funcionalidades pela técnica *MoSCoW* e a definição do contrato de integração via API, detalhando os modelos JSON para a comunicação com o ERP do cliente. Além disso, foram entregues os protótipos de tela no *Figma*, que validam o fluxo de usuário, e a arquitetura completa do banco de dados, projetada em *PostgreSQL* com UUIDs para garantir a escalabilidade e a integridade dos dados.

A principal dificuldade nesta etapa, foi abstrair a complexidade de um processo logístico, em um modelo de dados digital que fosse ao mesmo tempo, estável e flexível para se adaptar a diferentes cenários. A definição de integração via API, elaboração dos modelos de integração, e a modelagem do banco de dados foram decisões delicadas que exigiram um tempo de análise dedicada.

Como desafios previstos, pode-se destacar a sincronização de dados entre o aplicativo *mobile* e a plataforma *web*, a implementação de um sistema de autenticação e autorização sólido para diferentes perfis de usuário, o desenvolvimento da funcionalidade de leitura de e a captura, envio e o armazenamento de imagens.

Espera-se, ao final deste trabalho, entregar um sistema funcional podendo ser integrado com outros sistemas. O impacto esperado inclui a redução de falhas humanas, maior rastreabilidade dos processos, diminuição do retrabalho e suporte a auditorias e tomadas de decisão com base em dados registrados.

A relevância deste projeto justifica-se pelo impacto direto na eficiência operacional. O impacto esperado com a solução implementada inclui a drástica redução de falhas humanas no carregamento, o fornecimento de maior rastreabilidade dos processos, a diminuição do retrabalho e a criação de um repositório de dados que serve como suporte vital para auditorias e tomadas de decisão estratégicas.

A segunda etapa deste trabalho, será focada integralmente no desenvolvimento e implementação da solução. Os próximos passos incluem a codificação do *backend* (API REST em Laravel), o desenvolvimento da plataforma *web* (React) e do aplicativo *mobile* (React Native), e a aplicação das metodologias de teste TDD.

Espera-se, ao final deste trabalho, entregar um sistema funcional podendo ser integrado com outros sistemas. O impacto esperado inclui a redução de falhas humanas, maior rastreabilidade dos processos, diminuição do retrabalho e suporte a auditorias e tomadas de decisão com base em dados registrados. Demonstrando a aplicação efetiva das habilidades do curso de Sistemas para Internet em um desafio complexo e real do mercado.

REFERÊNCIAS

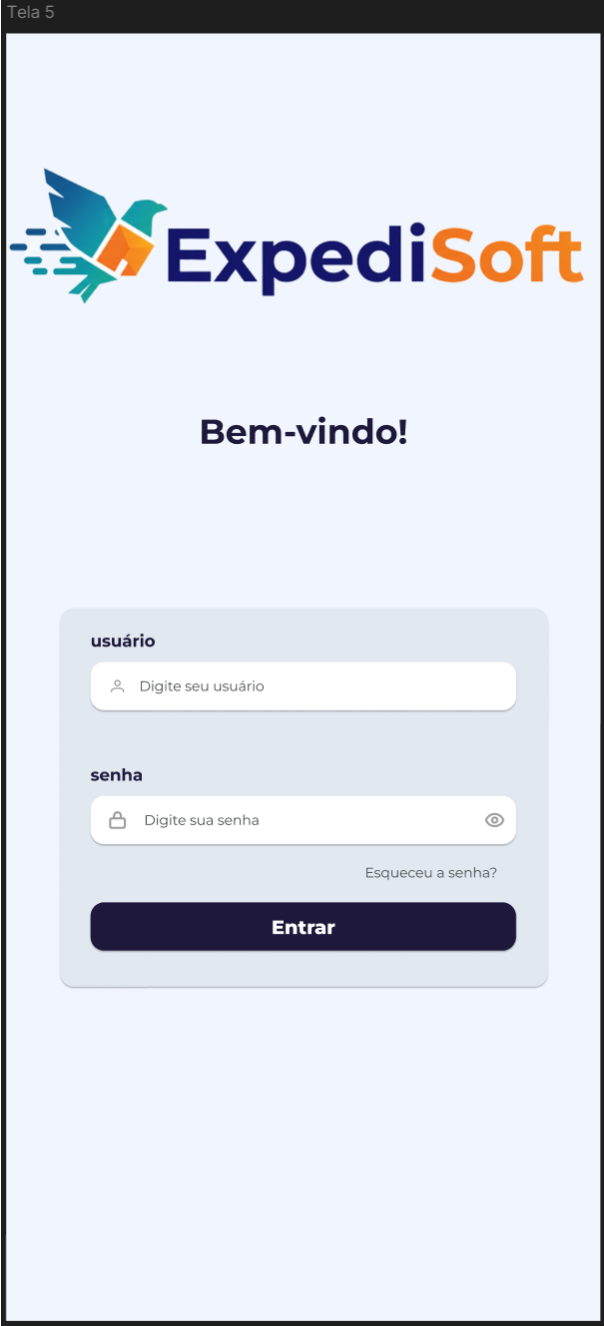
- BALLOU, R. H. **Gerenciamento da cadeia de suprimentos/logística empresarial. 5. ed. Porto Alegre: Bookman.** 2006. Disponível em: https://www.google.com.br/books/edition/Gerenciamento_da_Cadeia_de_Suprimentos_5/QAHrq0r6E7cC?hl=pt-BR&gbpv=1&pg=PA3&printsec=frontcover. Acesso em: 13 ago. 2025.
- BECK, 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>.
- 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/>.
- PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional. 7. ed.** [S.l.]: AMGH, 2011.
- 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 – Telas Complementares do Sistema e Banco de Dados

Este apêndice apresenta as telas e banco de dados complementares do sistema Expedisoft, abrangendo funcionalidades de acesso, navegação inicial e gestão administrativa na plataforma *web*.

Tela 5



The image shows a mobile application login screen. At the top, there is a header bar with the text 'Tela 5'. Below the header, the ExpediSoft logo is displayed, featuring a stylized bird icon in blue and orange. The text 'ExpediSoft' is in a bold, sans-serif font, with 'Expedi' in blue and 'Soft' in orange. Below the logo, the text 'Bem-vindo!' is centered. The main content area contains a light blue rounded rectangle with a white background. Inside this rectangle, there are two input fields. The first is labeled 'usuário' and has a placeholder text 'Digite seu usuário' with a user icon. The second is labeled 'senha' and has a placeholder text 'Digite sua senha' with a lock icon and a toggle eye icon. Below the password field, there is a link 'Esqueceu a senha?'. At the bottom of the rounded rectangle is a dark blue button with the text 'Entrar' in white.


ExpediSoft

Bem-vindo!

usuário

👤 Digite seu usuário

senha

🔒 Digite sua senha 

[Esqueceu a senha?](#)

Entrar

Figura 7 – Mobile: Tela de login.

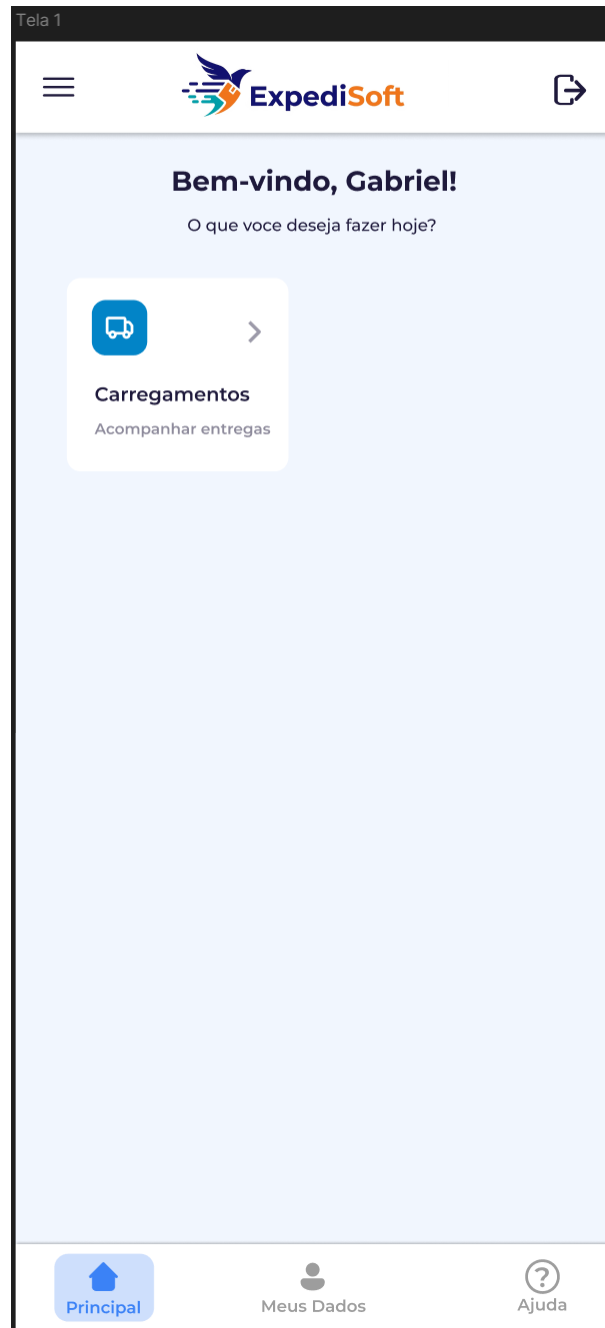


Figura 8 – Mobile: Tela home page.

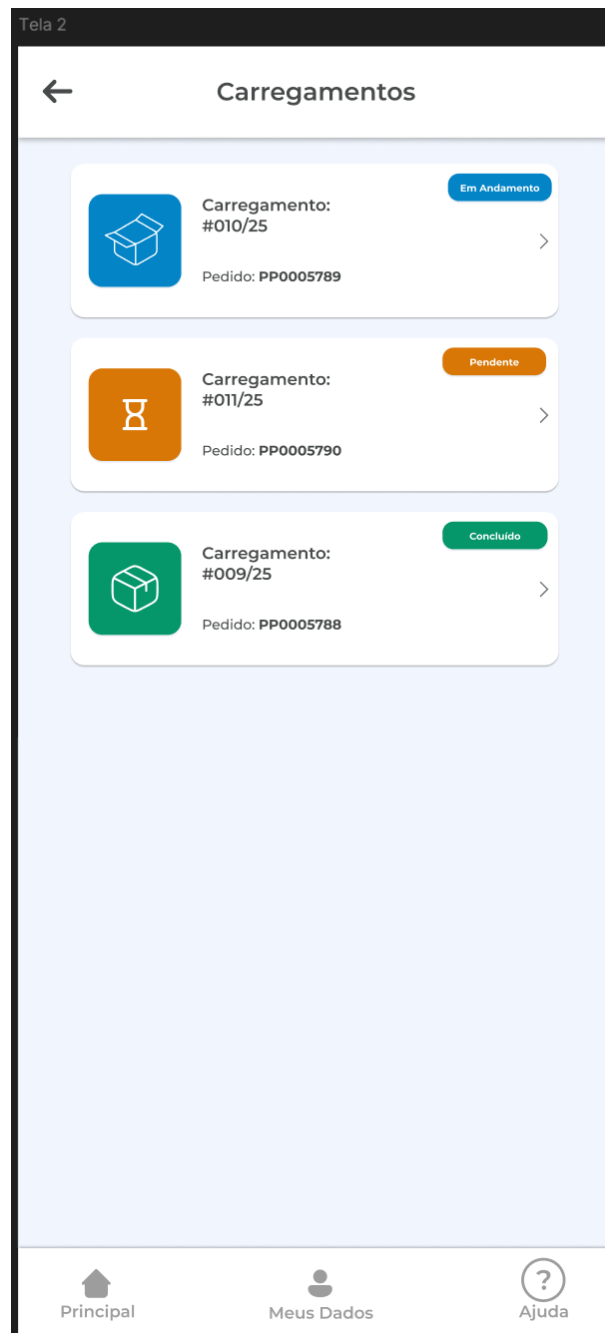


Figura 9 – Mobile: Tela lista de carregamentos.

Tela 8

← Detalhes do Carregamento

Carregamento: #010/25 PP0005789

Cliente **Zé Mercados** Destino **Guarapuava - Centro**

Total de Caixas **3** Unidade de Medida **30 [kg]**

Adicionar Fotos do Carregamento


Clique para selecionar fotos

  
Foto (1) Foto (2) Foto (3)

Cancelar

Salvar

☒ Conferido

 Escanear  Fotos **Concluir**

Figura 10 – Mobile: Tela de upload de fotos.

Tela 10

← Detalhes do Carregamento

Carregamento: #010/25 PP0005789

Cliente **Zé Mercados** Destino **Guarapuava - Centro**

Total de Caixas **3** Unidade de Medida **30 [kg]**

Itens a conferir Progresso: 3/3

Deseja inserir observações/justificativa?

Digite neste campo.

Cancelar

Concluir

Nº Caixa: 3

235458965 - Descrição do item

30 und X X X

✓ Conferido

Escanear Fotos Concluir

Figura 11 – Mobile: Tela para informar justificativas e observações.

Tela 9

← Detalhes do Carregamento

Carregamento: #010/25 PP0005789

Cliente **Zé Mercados** Destino **Guarapuava - Centro**

Total de Caixas **3** Unidade de Medida **30 [kg]**


ATENÇÃO!
Existem caixas que não foram conferidas. Deseja continuar?

Pacotes pendentes:
• N° Caixa: 3

Justificativa

É obrigatório justificar a finalização.

Cancelar

Finalizar

☒ Conferido

 Escanear

 Fotos

Concluir

Figura 12 – Mobile: Tela de aviso de inconsistência no carregamento.

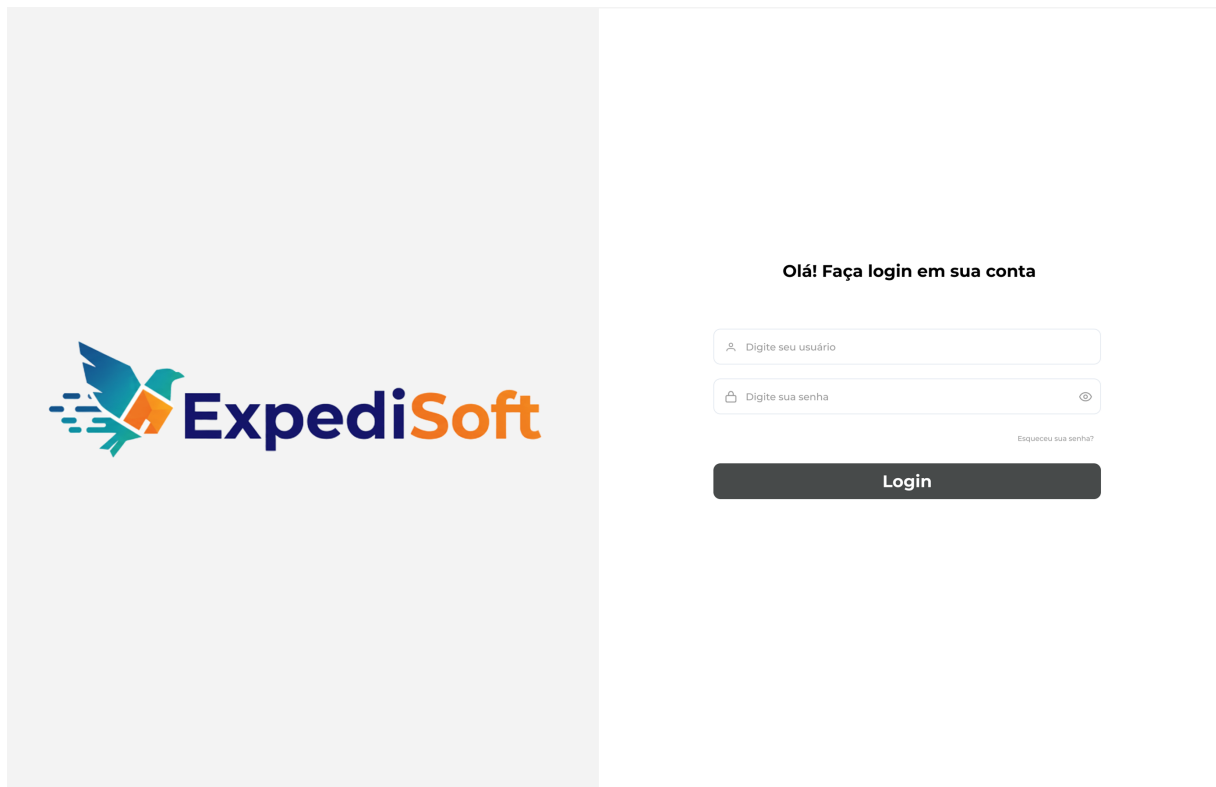


Figura 13 – Web: Tela de login.

Menu Principal

- Início
- Ordens de Carregamentos
- Histórico
- Configurações

Dashboard

Sábado, 18 de Outubro 2025

Carregamentos Hoje
3
 Total de ordens programadas

Em Andamento
15
 Sendo executados agora

Divergências
2
 Cargas finalizadas com divergência

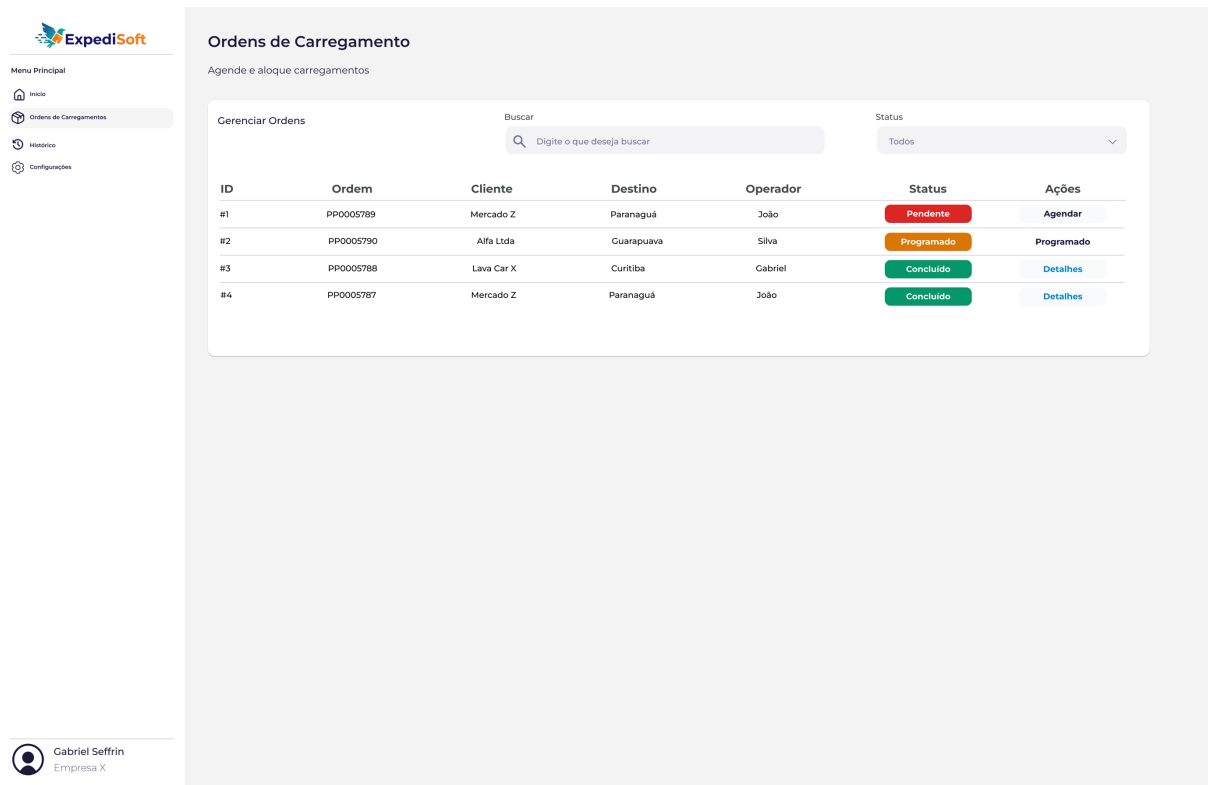
Em Andamento
4
 Sendo processados agora

Atividade Recente
Últimas ordens finalizadas ou agendadas

ID	Ordem	Cliente	Status	Operador	Atualizado
#1	PP0005789	Mercado Z	Concluído	João	10/10/2025 08:55
#2	PP0005790	Alfa Ltda	Programado	Silva	10/10/2025 07:50
#3	PP0005788	Lava Car X	Concluído	Gabriel	09/10/2025 23:00
#4	PP0005787	Mercado Z	Concluído	João	09/10/2025 15:45

Gabriel Seffrin
Empresa X

Figura 14 – Web: Tela home page / Dashboard.



Ordens de Carregamento
Agende e aloque carregamentos

Gerenciar Ordens

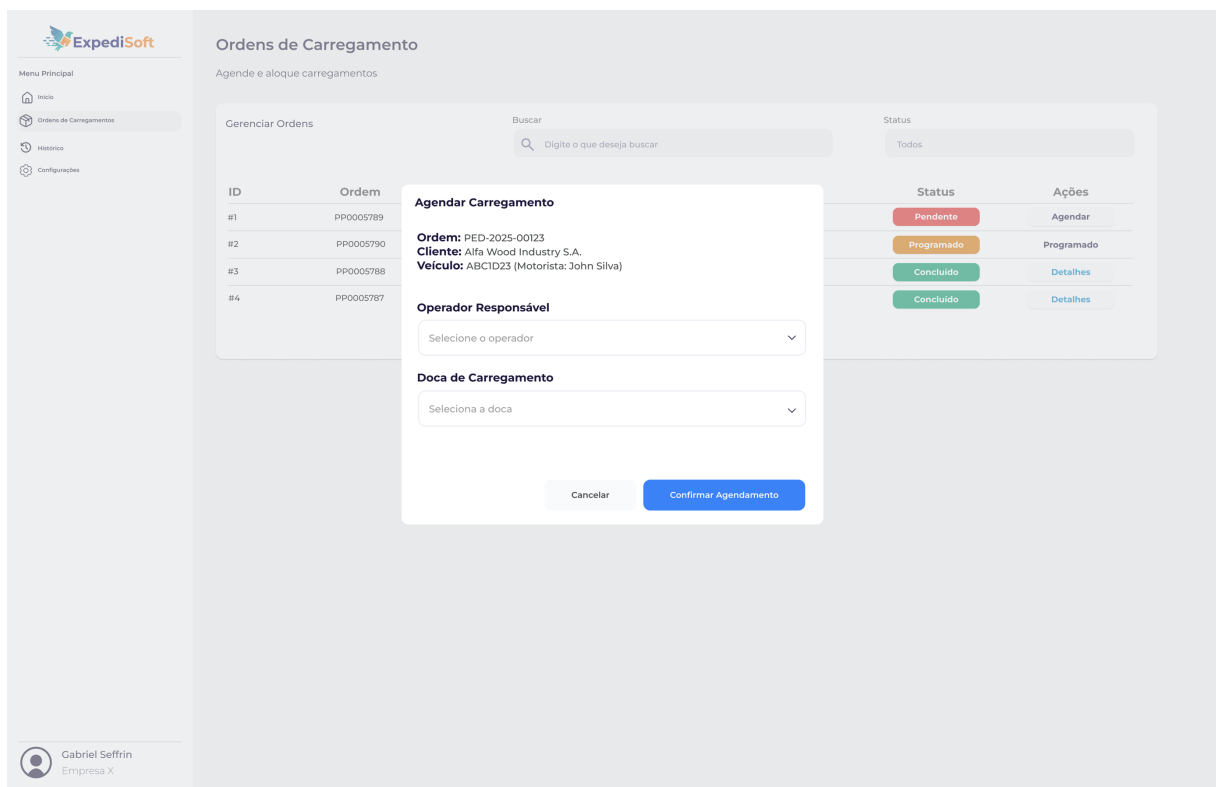
Buscar

Status Todos

ID	Ordem	Cliente	Destino	Operador	Status	Ações
#1	PP0005789	Mercado Z	Paranaguá	João	Pendente	Agendar
#2	PP0005790	Alfa Ltda	Guarapuava	Silva	Programado	Programado
#3	PP0005788	Lava Car X	Curitiba	Gabriel	Concluído	Detalhes
#4	PP0005787	Mercado Z	Paranaguá	João	Concluído	Detalhes

Gabriel Seffrin
Empresa X

Figura 15 – Web: Tela para agendar carregamento.



Ordens de Carregamento
Agende e aloque carregamentos

Gerenciar Ordens

Buscar

Status Todos

ID	Ordem	Status	Ações
#1	PP0005789	Pendente	Agendar
#2	PP0005790	Programado	Programado
#3	PP0005788	Concluído	Detalhes
#4	PP0005787	Concluído	Detalhes

Agendar Carregamento

Ordem: PED-2025-00123
Cliente: Alfa Wood Industry S.A.
Veículo: ABC1D23 (Motorista: John Silva)


Operador Responsável

Doca de Carregamento


Cancelar Confirmar Agendamento


Gabriel Seffrin
Empresa X


Figura 16 – Web: Modal para agendar carregamento.




Menu Principal

 Início

 Ordens de Carregamentos

 Histórico

 Configurações

Histórico de Carregamentos

Verifique todo o histórico dos carregamentos

Gerenciar Ordens

Buscar

ID	Ordem	Cliente	Operador	Data Finalização	Status	Ações
#1	PP0005789	Mercado Z	João	21/10/2025 11:50	Com Divergência	Detalhes
#2	PP0005790	Alfa Ltda	Silva	20/10/2025 13:45	Concluído	Detalhes


 Gabriel Seffrin
Empresa X

Figura 17 – Web: Tela de histórico de carregamentos.

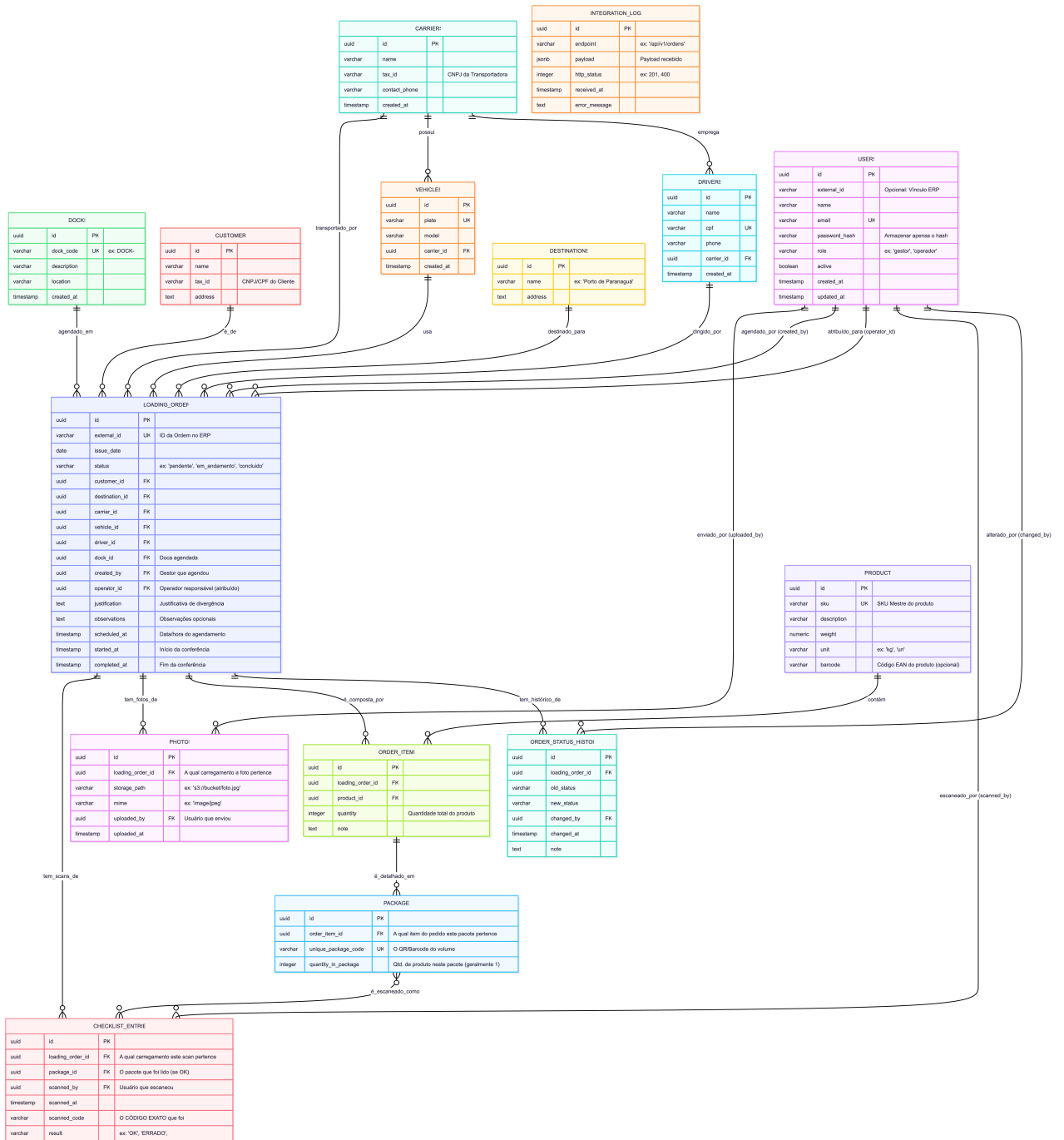


Figura 18 – Diagrama geral do banco de dados