RAILWISE: PROJETO DE DESENVOLVIMENTO DE UM SISTEMA PARA GERENCIAMENTO DE FERROVIAS CARGUEIRAS

Fabrício Tolotti[1]
Gabriel Périco[1]
João Paulo Gregolon Paludo[1]
Luiz Augusto Dal Bello[1]
Franciele Petry[2]
Roberson Junior Fernandes Alves[3]
Otília Donato Barbosa[4]

Resumo

Este artigo tem por objetivo principal demonstrar as atividades realizadas, assim como os resultados obtidos no processo de construção de um sistema para gerenciamento de linhas ferroviárias. Durante a construção do projeto, vários aspectos foram avaliados a fim de que o software caracterize um bom sistema de gerenciamento. Com a execução do projeto, acreditamos ser possível auxiliar os usuários a alcançarem a eficiência operacional, aumentando a produtividade e garantindo maior segurança das operações. A construção desse software utilizou-se de ferramentas como a linguagem de programação Java, banco de dados Postgresql, Notion, Kanban, Visual Paradigm e outras.

Palavras-chaves: Gerenciamento ferroviário; Ferrovias cargueiras; Java Spring Boot.

[1] Discentes do Curso de Ciência da Computação
Unoesc - Campus de São Miguel do Oeste
Rua Oiapoc, 211. São Miguel do Oeste-SC
fabriciotolotti1@gmail.com; gabrielperico2014@gmail.com; joaopgpaludo@gmail.com; luiz.bello@unoesc.edu.br.
[2] Mestre em Informática
Docente do Curso de Ciência da Computação
Unoesc - Campus de São Miguel do Oeste
Rua Oiapoc, 211. São Miguel do Oeste-SC.
franciele.petry@unoesc.edu.br
[3] Mestre em Computação Aplicada
Docente do Curso de Ciência da Computação
Unoesc - Campus de São Miguel do Oeste
Rua Oiapoc, 211. São Miguel do Oeste-SC.
roberson.alves@unoesc.edu.br
[4] Mestre em Informática
Docente do Curso de Ciência da Computação
Unoesc-Campus de São Miguel do Oeste
Rua Oiapoc, 211. São Miguel do Oeste-SC.
otilia.barbosa@unoesc.edu.br

1 INTRODUÇÃO

De acordo com a Associação Nacional de Transportadores Ferroviários, "Em mais de duas décadas de concessões, as associadas à ANTF apresentaram um crescimento de 98% na movimentação de cargas pelas ferrovias, em relação a 1997". A média de crescimento anual foi de 2,76%, sendo iniciado as atividades ainda em 1997, onde transportava-se 253 milhões de toneladas úteis, e no ano de 2022 foram transportadas pouco mais de 500 milhões de toneladas úteis. Atualmente as linhas ferroviárias atendem a matriz de transporte de cargas do Brasil em 21,5%, sendo a segunda forma de transporte de cargas predominante, perdendo apenas para a rodoviária com cerca de 67,6% dos transportes.

Levando em consideração que o transporte ferroviário “emite cerca de 96% de dióxido de carbono (CO_2)” a menos que o transporte rodoviário, e também que “em 2021, o modo ferroviário de cargas foi responsável por aproximadamente 2,9% das emissões nacionais oriundas do setor de transporte de carga”, a RailWise, uma empresa líder no fornecimento de soluções inovadoras e abrangentes para linhas de trens surgiu, trazendo um software especializado em oferecer uma solução completa que aborda todos os aspectos essenciais do gerenciamento ferroviário de cargas, desde o fretamento de cargas por meio de vagões e locomotivas até o controle de rotas ferroviárias, gerenciamento de manutenção dos equipamentos e controle de recursos humanos ligados às operações.

A RailWise busca fornecer aos seus usuários a experiência e expertise em gerenciamento ferroviário, sempre empenhados em ajudar nossos usuários a alcançarem a eficiência operacional, aumentando a produtividade, reduzindo os custos e garantindo maior segurança em suas operações. Na RailWise, estamos comprometidos em proporcionar uma experiência eficiente e livre de problemas, ajudando a otimizar as operações ferroviárias e melhorar a eficiência logística.

2 DESENVOLVIMENTO

O desenvolvimento apresenta informações referentes à evolução do mercado ferroviário brasileiro, assim como demonstra as principais tecnologias utilizadas na construção do nosso software, e também os resultados encontrados.

2.1 REFERENCIAL TEÓRICO

Mesmo que não sejam tão rápidas e abrangentes como as estradas, as ferrovias oferecem uma alternativa mais barata de transportar grandes quantidades de carga em longas distâncias. Elas são especialmente boas quando operam em rotas especiais, com trens grandes e carga parecida. São perfeitas para transportar cargas a granel, como grãos e minérios, e até mesmo contêineres, em rotas longas. Como o Brasil é um país de território extenso e produz muitos produtos agrícolas e minerais, faz sentido usar as ferrovias para transportar cargas com mais frequência (BARAT, 2009).

Segundo a CNN, que se inspira nos dados da Agência Nacional de Transportes Terrestres (ANTT), “A projeção de recursos privados para investimentos no setor ferroviário brasileiro ultrapassa R$ 170 bilhões, A previsão para este ano é de 12 mil quilômetros de novos trilhos, cruzando 19 unidades da Federação.” (CNN Brasil, 2023).

Por mais que o investimento esteja definido, “a perspectiva para novos investimentos públicos ainda é incerta”, (CNN Brasil, 2023). Todavia, ao contrário do modelo de concessão ferroviária, a autorização permite que o setor privado construa as novas ferrovias sem a necessidade de pagamentos de outorgas bilionários ao governo federal.

O secretário Nacional de Transporte Ferroviário, explica que as autorizações ferroviárias foram inspiradas nas “shortlines” dos Estados Unidos, que são linhas de menor distância para conectar pontos próximos, sendo um novo modelo de exploração econômica do setor, permitindo que a iniciativa privada invista em projetos de seu interesse (Leonardo Ribeiro, 2023).

Além disso, “o investimento no setor ferroviário possibilita uma queda nos custos dos produtos, uma vez que as linhas férreas geram empregos, diminuem custos com aquisição e manutenção de veículos e desoneram as rodovias.” (Ismael Trinks, 2023).

Na Bahia, a empresa VLI Multimodal S.A recebeu autorização para construir duas ferrovias, que somadas possuirão extensão de mais de 220 quilômetros. Segundo a empresa, o investimento estimado é de quase R$5 bilhões, os quais gerarão cerca de 33 mil novos empregos, diretos e indiretos (CNN Brasil, 2023).

O papel dado ao transporte ferroviário de cargas tem sido o de voltar a operar em seu potencial máximo há anos, desde o início da queda de sua importância, no

início do século XX. Nos anos 1990, a malha ferroviária então danificada e ineficiente deixou de ser preocupação do governo com a concessão das ferrovias à iniciativa privada. Os investimentos do setor privado aumentaram a capacidade das ferrovias cargueiras, fazendo a malha ferroviária brasileira alcançar o posto de 6ª maior rede do mundo, mas ainda com produtividade inferior aos principais sistemas ferroviários internacionais (Massa, 2021).

Existe expectativa de que o crescimento do setor ferroviário cargueiro seja impulsionado por mais investimentos privados, aumentando a malha ferroviária e integrando diferentes modais de transporte terrestre. Com potencial para crescer e expectativa de investimentos, evidencia-se a necessidade de um gerenciamento eficiente, eficaz e ágil para permitir a devida operacionalização da malha ferroviária, que na atual era da informação seria grandemente favorecido pelo uso de um sistema informatizado que abrangesse todas as funções relacionadas às operações do transporte cargueiro ferroviário.

2.2 MATERIAIS E MÉTODOS

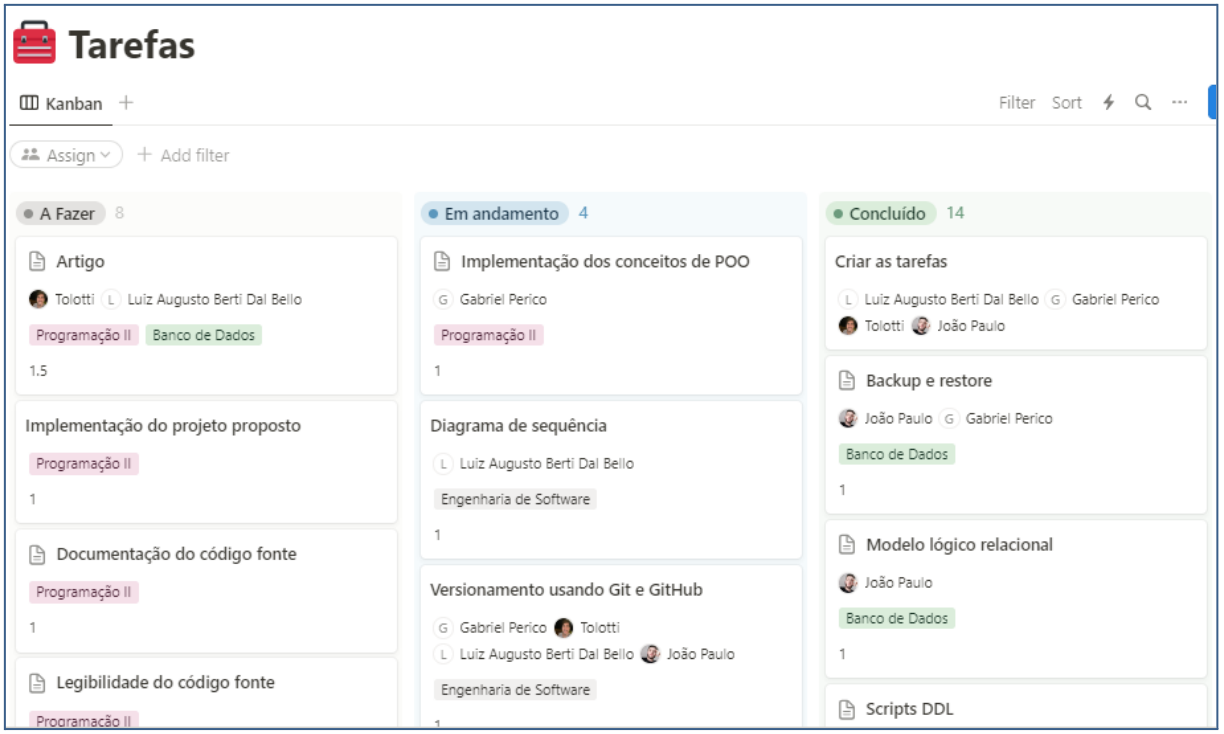
O presente tópico trata do levantamento de requisitos para o desenvolvimento das atividades propostas e do sistema de gerenciamento do transporte ferroviário de cargas.

Para a construção do software, foram utilizados os editores de código-fonte Eclipse e também o Visual Studio Code, onde a linguagem de programação escolhida foi o Java, levando em consideração que é uma linguagem multiplataforma, orientada a objetos e centrada em rede que pode ser usada como uma plataforma em si, fazendo uso do framework Spring Boot, que facilita a criação de aplicações autossuficientes que podem ser facilmente executadas.

Para a modelagem do banco de dados em Postgresql e também a criação dos diagramas, foi utilizado o Visual Paradigm, ferramenta que facilita as construções pois possui um conjunto abrangente de ferramentas DevOps, de Design e demais recursos que tornam o desenvolvimento mais ágil e eficiente.

A Ferramenta Notion foi utilizada para separar as atividades que cada membro da equipe deveria realizar, e também para estipular o prazo de realização. A separação foi feita em modelo Kanban, onde separamos as tarefas da seguinte maneira: “A fazer”, “Em andamento” e “Concluído”, onde as tarefas avançavam para as colunas seguintes conforme eram executadas, vide a imagem em sequência.

Imagem 1: Tarefas divididas no Notion.



Fonte: Os autores (2023).

2.2.1 Requisitos e modelagem

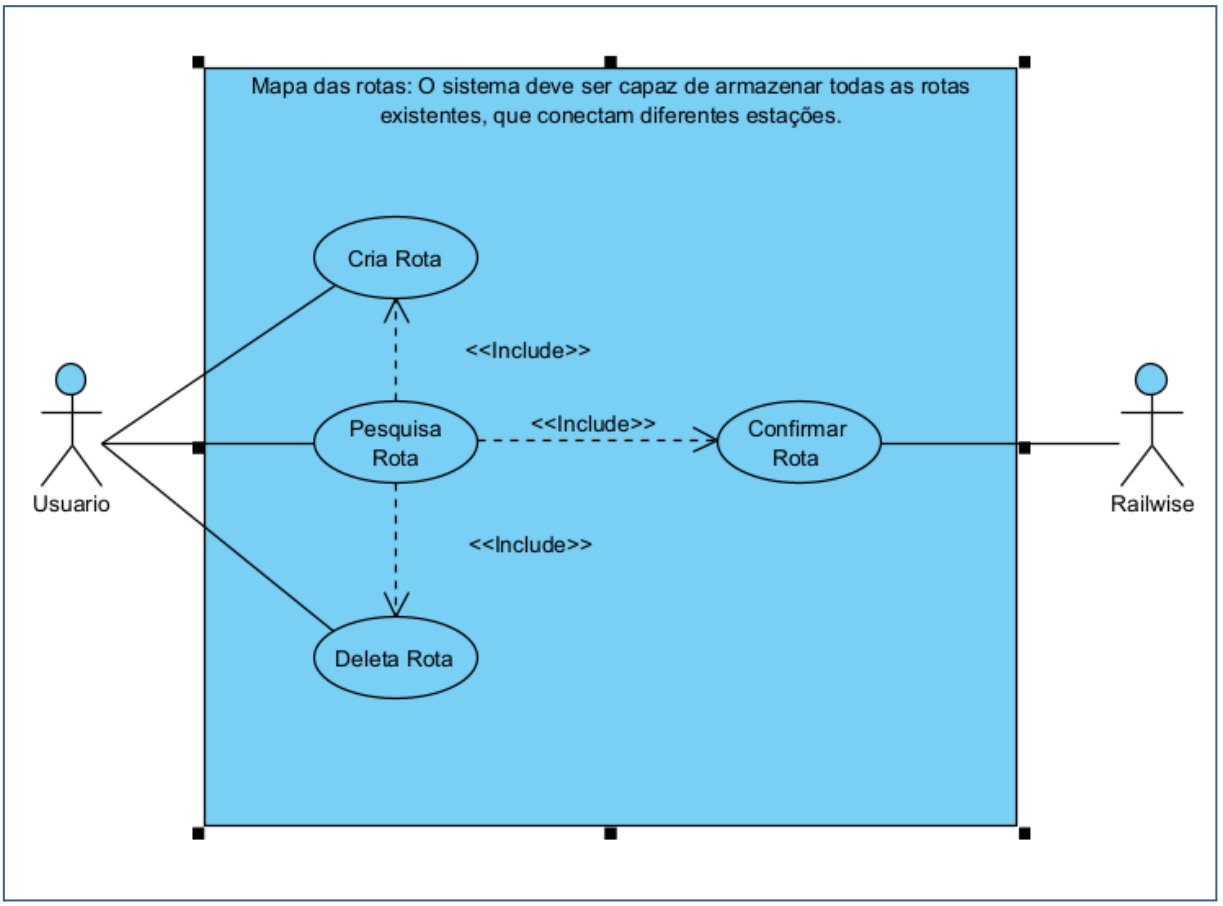
O primeiro passo para o desenvolvimento da aplicação se deu com o levantamento dos requisitos levando em consideração dados e informações disponíveis na internet e em trabalhos acadêmicos relacionados que exploraram a temática dos transportes ferroviários no Brasil.

Todos os aspectos e informações obtidos nas primeiras pesquisas, foram utilizados com ênfase na criação dos diagramas responsáveis pela explicação do funcionamento do sistema, limites dos atores contidos no mesmo, interferências no desenvolver da aplicação e reduzir ambiguidades que poderiam surgir durante o processo de desenvolvimento.

O objetivo da criação desses diagramas é passar uma mensagem de maneira padronizada, onde todos os receptores desses diagramas consigam compreender o que os criadores do sistema quiseram propor com o desenvolvimento do próprio sistema ou de suas aplicações.

O diagrama da figura 1 representa a comunicação do usuário com o software, onde o usuário tem a possibilidade de criar, pesquisar ou deletar uma rota, que seria o caminho percorrido pelos trens e seus vagões ligando duas estações.

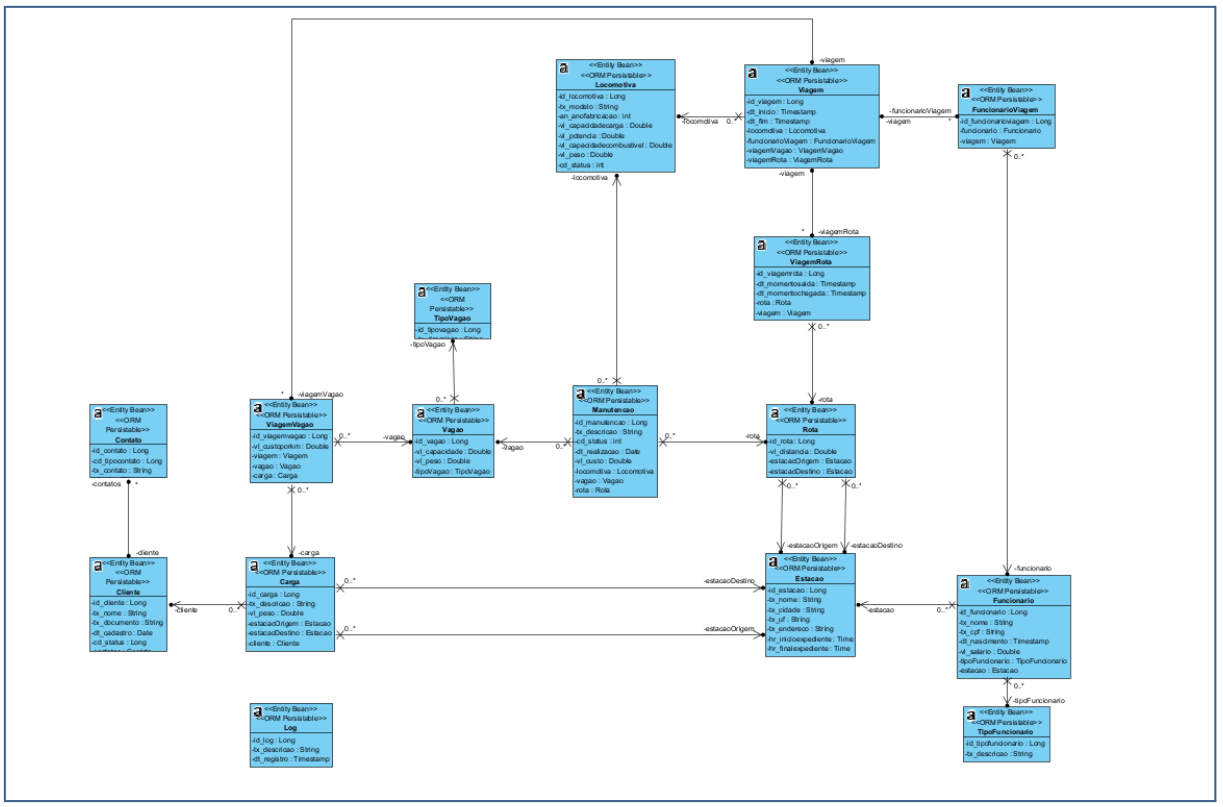
Figura 1: Diagrama de caso de uso.



Fonte: Os autores (2023).

Para se obter uma visualização da estrutura do sistema, e também demonstrar os relacionamentos entre as classes e objetos do banco de dados Postgresql, um diagrama de classes foi modelado utilizando o Visual Paradigm, onde podemos verificar todas as ligações entre as tabelas e outras informações importantes para se certificar de que o banco de dados seja coeso e sem falhas.

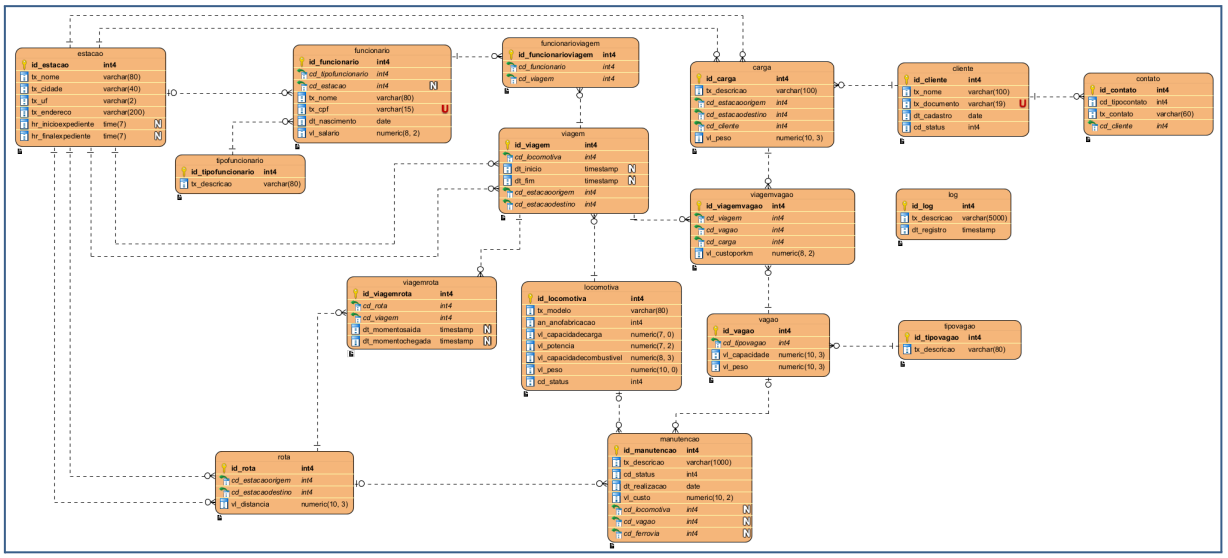
Figura 2: Diagrama de classes.



Fonte: Os autores (2023).

O modelo do banco de dados foi feito utilizando-se do Visual Paradigm, que também fornece, a partir do modelo construído, o script SQL responsável por gerar as estruturas construídas. Posteriormente, foi usado o aplicativo e ferramenta de edição SQL DBeaver. A modelagem final do banco de dados se encontra na figura abaixo.

Figura 3: Modelagem do banco de dados.



Fonte: Os autores (2023).

2.2.2 Funcionalidades do banco de dados

O desenvolvimento de funcionalidades no banco de dados foi feito por meio da ferramenta DBeaver, com o qual foram implementados os comandos SQL e códigos PL/pgSQL.

Os scripts de criação da base foram gerados a partir da modelagem e então revisados adicionando ao banco além das restrições(*constraints*) de chaves primárias e chaves estrangeiras, a restrição de unicidade para o documento do cliente e do funcionário, a restrição de valor padrão *current_timestamp* para a data de registro na tabela de logs de auditoria e restrições de checagem para garantir que colunas de código de status e código de tipo possuam apenas valores conforme os documentados. A partir disso, foram incluídos índices de pesquisa do tipo único para buscas mais eficientes no banco de dados na coluna tx_cpf da tabela de funcionários e na coluna tx_documento da tabela de clientes.

Para os relatórios exigidos em banco de dados, foram criadas *Views*, objetos de banco de dados que armazenam um comando de SELECT e o executam quando solicitado. Os comandos implementados nas *Views* fazem uso de junções entre tabelas para permitir a busca de dados que possuem relação mas encontram-se em tabelas separadas.

As políticas de acesso ao banco de dados do RailWise foram feitas por meio da criação de grupos no Postgresql, que permitem agrupar privilégios e permissões que podem ser mais facilmente manipulados e concedidos aos usuários. Foram criados três grupos de usuários, cada um com permissões condizentes com as funções que devem desempenhar, sendo eles: administrador, usuario_basico e gerente. Além dos grupos, foi criado um usuário de backup, com permissão de visualizar as informações de todas as tabelas, para acessar o banco e efetuar o backup do mesmo.

Foram incluídos no banco gatilhos de verificação. Foi desenvolvida uma função genérica de inserção para a tabela de logs de auditoria e gatilhos do tipo *before* que acionam a função quando ocorre uma inserção, atualização ou deleção de registro nas tabelas viagem, manutenção e carga. A função grava um registro de log com a data da alteração, a operação realizada, a tabela que acionou o gatilho, o usuário que fez a ação e as alterações feitas no registro.

Outro gatilho implementado é um gatilho de verificação de datas quando ocorre uma inserção ou alteração de registro na tabelas de viagens, onde só é permitido que os dados sejam persistidos na base caso a data de término da viagem seja posterior à data de início da viagem, e também só permite que a viagem tenha uma data de término se possuir uma data de início.

Para facilitar alguns processos como a busca das informações dos relatórios e a persistência em algumas tabelas, foram criadas funções e procedimentos no banco de dados. A função *distancia_minima_estacoes* busca nos registros de rotas e retorna a distância mínima entre duas estações. A função *rota_minima_estacoes* busca nas rotas e estações e retorna um vetor contendo todas as estações que compõem uma rota. Por fim, o procedimento *criar_rotas_viagem* insere os registros que relacionam as rotas percorridas em uma viagem.

A fim de assegurar que os dados do sistema não sejam perdidos ou avariados, pensou-se em uma política de backup do banco de dados, tendo tanto um backup físico dos dados quanto um backup lógico do banco, sendo o backup agendado para ser realizado diariamente às 01:00, utilizando o agendador de tarefas do Windows e um script armazenado em um arquivo em lote com a extensão *.bat* que armazena os comandos a serem executados para que o backup físico e lógico seja feito. Dessa forma, se algo acontecer ao banco ou aos dados, a informação não será perdida completamente, mas apenas 24 horas de alterações no banco, no máximo.

2.2.3 Desenvolvimento back-end

O desenvolvimento do back-end foi feito por meio das ferramentas IDEs Visual Studio Code e Eclipse para a escrita e implementação das funcionalidades utilizando a tecnologia Java.

O projeto também utiliza em sua base o framework Spring Data para a criação de pontos de acesso por meio de links e chamativas HTTP, para a comunicação com o banco de dados por possuir uma camada acima do JPA, facilitando a criação de métodos para o mesmo. O pacote lombok também é utilizado em conjunto com o Spring Data para a criação de getters e setters para as entidades.

Foram criados os arquivos entidades como classes utilizando os conceitos de POO e utilizando a anotação @Entity e a anotação @Table do Spring Data para a vinculação da entidade com a tabela do banco de dados.

Os arquivos controllers foram criados com base na anotação @RestController e @RequestMapping do Spring Data, assim fazendo com que a comunicação via HTTP possua uma rota para o método em suas requisições.

Já os arquivos repositories foram criados com base na extensão da classe JpaRepository, que gera alguns métodos padrões para a leitura, criação, atualização e deleção dos dados vinculados com as entidades.

A documentação do código foi criada por meio de comentários no código fonte e automaticamente criada por meio da funcionalidade da IDE Eclipse para a geração do JavaDoc.

2.2.4 Desenvolvimento front-end

O desenvolvimento do front-end foi feito por meio da ferramenta Eclipse e utilizando as tecnologias HTML5, CSS3 e JavaScript para a interação do usuário, alguns frameworks e bibliotecas utilizados para o auxílio na criação da interface foram o JQuery para o JavaScript e Bootstrap para o JavaScript e CSS.

Para o acesso ao front-end foi utilizado o arquivo index.html e uma rota padrão para servir documentos estáticos dentro do Spring Boot.

Foram criados páginas para a visualização dos relatórios e introdução ao projeto proposto.

Para a visualização dos relatórios foi utilizado o método de comunicação AJAX da biblioteca JQuery para que o front-end se comunique com o backend sem que seja necessário recarregar a página.

2.3 RESULTADOS E DISCUSSÕES

O desenvolvimento do projeto foi modularizado a fim de suprir as exigências propostas pelos requisitos determinados no início do projeto, tendo sido desenvolvidos recursos em banco, como *Views*, gatilhos e funções, no back-end, onde foi implementado o *CRUD* das principais entidades do sistema, e no front-end, com a implementação de algumas telas básicas de apresentação e exibição de dados.

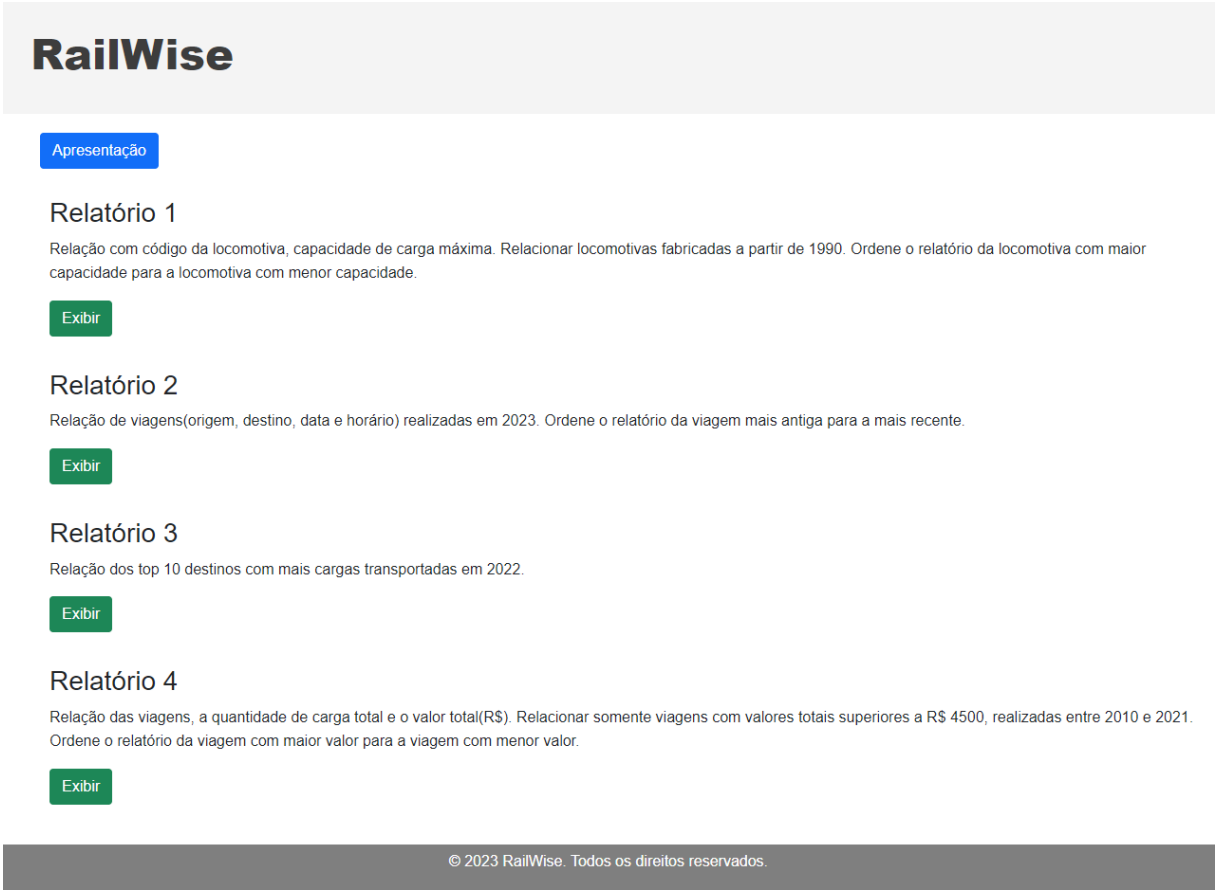
O front-end da aplicação é composto por dois módulos, ambos desenvolvidos nas linguagens JavaScript, CSS3 e HTML5, onde o primeiro dos módulos se trata de um painel de apresentação do sistema, constando as funcionalidades do mesmo e suas aplicabilidades.

Tela 1: Tela de apresentação do sistema.

Fonte: Os autores (2023).

O segundo módulo da aplicação front-end se trata dos relatórios implementados no banco de dados por meio de *Views*, uma vez que os mesmos já estão integrados ao sistema.

Tela 2: Tela de exibição dos relatórios



Fonte: Os autores (2023).

O back-end desenvolvido para a aplicação fornece endpoints para realizar operações de cadastro, atualização, visualização e exclusão de dados das entidades do sistema, permitindo interagir com a base de dados conforme as permissões do usuário, definidas em banco.

Em geral, a aplicabilidade do software não chegou a ser validada, pois em nossa região não possuímos linhas de trens, e também não temos familiaridade com tal assunto. Por se tratar de um projeto acadêmico de curto prazo, também não conseguimos contato com trabalhadores do ramo para discutir tais assuntos, mas acreditamos que o desenvolvimento do projeto seria de grande importância para o setor, assim como facilitaria os processos englobados pelo sistema.

3 CONCLUSÃO

Este artigo expõe as etapas na criação de um sistema para gerenciamento de ferrovias cargueiras, levando em consideração o aumento desse setor nas últimas duas décadas, e também os promissores futuros investimentos. A RailWise considera válidos os investimentos para desenvolvimento desse sistema, que visa melhorar e aumentar a produtividade do setor de fretamento cargueiro, levando nossos usuários a alcançarem a eficiência operacional e também garantindo maior segurança das operações.

Acreditamos que conseguimos concluir todos os objetivos propostos no início do projeto, uma vez que o banco de dados se demonstrou íntegro e consistente, assim como o desenvolvimento do back-end em Java. A equipe se demonstrou organizada e consistente, uma vez que todas as atividades separadas para cada membro da equipe, foi realizada pelo seu responsável, apenas debatendo com os colegas para se certificar de que a mesma estava seguindo no padrão desejado, assim como, para alinhamento, uma vez que acreditamos que todos os membros da equipe devem estar cientes de todos os desenvolvimentos das tarefas, por mais que não seja o responsável pela mesma.

Haveria, ainda, espaço e intenções de desenvolver mais funcionalidades para o sistema, principalmente em termos de front-end, pois foram feitas apenas telas básicas de apresentação, então a parte visual teria muito a evoluir.

RAILWISE: PROJECT TO DEVELOP A SYSTEM FOR MANAGING CARGO RAILWAYS

Abstract

This article's main objective is to demonstrate the activities carried out, as well as the results obtained in the process of building a system for managing railway lines. During the construction of the project, several aspects were evaluated in order for the software to be a good management system. By executing the project, we believe it will be possible to help users achieve operational efficiency, increasing productivity and ensuring greater operational safety. The construction of this software used tools such as the Java programming language, Postgresql database, Notion, Kanban, Visual Paradigm and others.

Keywords: Railway management; Freight railways; Java Spring Boot.

REFERÊNCIAS

Boletim de Logística - A Retomada dos Investimentos Ferroviários para Aumentar a Eficiência da Matriz de Transportes. 2023. Disponível em: <https://ontl.infrasa.gov.br/wp-content/uploads/2023/03/Setor-Ferroviario-Brasileiro.pdf>. Acesso em: 01 nov. 2023.

NASCIMENTO, Eunice Passaglia. **SISTEMA DE APOIO AO PLANEJAMENTO E GERENCIAMENTO DA OPERAÇÃO DO TRANSPORTE FERROVIÁRIO DE CARGA.** 1993. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/75876/93253.pdf;jsessionid=1079880E00EAB9C8972CFDE2098C8DA2?sequence=1>. Acesso em: 25 nov. 2023.

ANTF - Associação Nacional dos Transportadores Ferroviários. **Informações gerais.** 2023. Disponível em: <https://www.antf.org.br/informacoes-gerais/>. Acesso em 25 nov. 2023.

CNN Brasil. **Projeção de investimentos privados no setor ferroviário ultrapassa R$170 bi, mostram dados da ANTT.** 2023. Disponível em: <https://www.cnnbrasil.com.br/economia/projecao-de-investimentos-privados-no-setor-ferroviario-ultrapassa-r-170-bi-mostram-dados-da-antt/>. Acesso em 25 nov. 2023.

Amazon AWS. **O que é o Java?.** 2023. Disponível em: <https://aws.amazon.com/pt/what-is/java/#:~:text=Java%20%C3%A9%20uma%20linguagem%20multiplataforma,data%20e%20tecnologias%20do%20servidor>. Acesso em: 25 nov. 2023.

BARAT, Josef. **Transporte ferroviário de carga no Brasil.** 2009. Disponível em: <https://www.ipea.gov.br/desafios/index.php?option=com_content&view=article&id=1066:catid=28&Itemid=23> Acesso em: 26 nov. 2023.

PostgreSQL. **Documentation.** 2023. Disponível em: <https://www.postgresql.org/docs/> Acesso em: 26 nov. 2023

Spring. **Spring Boot Reference Documentation**. 2023. Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>. Acesso em: 26 nov. 2023

Massa Pesagem e Automação Industrial. **Transporte ferroviário de cargas: tudo sobre.** 2021. Disponível em: <https://massa.ind.br/transporte-ferroviario-de-cargas/> Acesso em: 26 nov. 2023.