

**Do Legado à Inovação: Superando a Obsolescência Tecnológica na Malha
Ferroviária com o RailBridge**

**Autor(a): Kateriny Bispo de Deus
Eggo Henrique Pinheiro
Adriano Francisco Rodrigues da Silva
Lindemberg De Lima Brito**

Resumo

A Estrada de Ferro Carajás (EFC), importante corredor logístico brasileiro, enfrentava sérios riscos operacionais devido à obsolescência do BRG (Bit Regenerator), equipamento crítico na comunicação entre o Sistema de Gestão Ferroviária (SGF) e os dispositivos de campo. O BRG apresentava limitações como hardware proprietário, sistema operacional descontinuado e baixa flexibilidade, dificultando manutenções e impedindo atualizações. Diante desse cenário crítico, foi concebida uma nova solução tecnológica para modernizar a infraestrutura de comunicação da EFC: o RailBridge, uma proposta baseada em arquitetura aberta, independente de hardware e sistema operacional, que utiliza tecnologias modernas como Node.js, RabbitMQ e JSON. O RailBridge promove comunicação bidirecional entre o SGF e os dispositivos de sinalização, compatível com protocolos legados (serial STD) e modernos (Ethernet), permitindo escalabilidade e integração futura com novas tecnologias. Com suporte a hardware redundante e mecanismos de contingência automáticos (keepalive), a solução assegura alta disponibilidade e resiliência operacional. Além disso, a coleta estruturada de logs possibilita a adoção de manutenção preditiva, ampliando a eficiência e a segurança da operação. Os resultados incluem redução no tempo de processamento de comandos, menor dependência tecnológica, queda nos custos de manutenção e maior capacidade de resposta a falhas. O projeto demonstra como a inovação, aliada à interoperabilidade e resiliência digital, pode modernizar infraestruturas críticas e manter a competitividade no setor ferroviário. O RailBridge estabelece uma base sólida para a evolução contínua da EFC frente às crescentes demandas logísticas e tecnológicas.

Palavras-chave: Transformação digital, Ferrovia, Comunicação Inteligente, automação, Modernização.

Abstract

The Carajás Railway (EFC), a major Brazilian logistics corridor, faced serious operational risks due to the obsolescence of its BRG (Bit Regenerator), a critical piece of equipment for communication between the Railway Management System (SGF) and field devices. The BRG had limitations such as proprietary hardware, a discontinued operating system, and low flexibility, hindering maintenance and preventing upgrades. Faced with this critical situation, a new technological solution was designed to modernize the EFC's communication infrastructure: RailBridge, a proposal based on open architecture, independent of hardware and operating system, which uses modern technologies such as Node.js, RabbitMQ, and JSON. RailBridge promotes bidirectional communication between the SGF and signaling devices, compatible with legacy (serial STD) and modern (Ethernet) protocols, enabling scalability and future integration with new technologies. With support for redundant hardware and automatic contingency mechanisms (keepalive), the solution ensures high availability and operational resilience. Furthermore, structured log collection enables the adoption of predictive maintenance, increasing operational efficiency and safety. The results include reduced command processing time, reduced technological dependence, lower maintenance costs, and greater failure response capabilities. The project demonstrates how innovation, combined with interoperability and digital resilience, can modernize critical infrastructure and maintain competitiveness in the rail sector. RailBridge establishes a solid foundation for the continued evolution of EFC in the face of growing logistical and technological demands.

Keywords: Digital transformation, Railway, Intelligent Communication, Automation, Modernization.

SUMÁRIO

1. Introdução.....	6
2. Metodologia	7
2.1. Diagnóstico do Problema	7
2.2. Levantamento de Requisitos do RailBridge	8
2.2.1. Independência de Hardware e Sistema Operacional	8
2.2.2. Comunicação Bidirecional com Suporte à Múltiplos Protocolos:.....	9
2.2.3. Suporte a Hardware Dualizado e Expansível:.....	9
2.3. Escolha de Tecnologias	10
2.3.1. Nodejs:	10
2.3.2. RabbitMQ:.....	10
2.3.3. Json:.....	11
3. Arquitetura da Solução	11
3.1. Implementação Técnica:	13
3.1.1 RailBridge	13
3.1.2. Broker RabbitMQ.....	14
3.1.2. Monitoramento (KeepAlive).....	15
4.Resultados.....	15
4.1. Eficiência Operacional	15
4.2. Suporte a Manutenção Preditiva:	15
4.3. Eficiência Operacional	16
5. Conclusão	18
6. Referências	19

1. INTRODUÇÃO

De acordo com informações do Programa de Parcerias de Investimentos (2025), a Estrada de Ferro Carajás, com 892 km de extensão e conectada ao Porto de Itaqui e à Ferrovia Norte-Sul, destaca-se como uma infraestrutura de alta capacidade voltada principalmente ao transporte de minério de ferro. Sua estrutura quase totalmente duplicada e os investimentos previstos com a renovação da concessão indicam sua relevância estratégica para a logística nacional. Nesse contexto, por ser uma das mais importantes ferrovias de carga do Brasil, a EFC tem investido continuamente em inovação para garantir a eficiência, segurança e sustentabilidade de suas operações. No entanto, esse compromisso com a modernização também envolve superar desafios técnicos significativos. Um dos principais obstáculos enfrentados nos últimos anos foi a obsolescência do **BRG (Bit Regenerator)**, equipamento responsável por intermediar a lógica de comunicação entre os dispositivos de automação e sinalização instalados ao longo da ferrovia e o **SGF (Sistema de Gestão Ferroviária)**, que atua como interface entre o centro de controle operacional e a infraestrutura ferroviária. O BitRegenerator era responsável por garantir o envio e recebimento de informações entre o centro de controle e os equipamentos de via permanente, permitindo a operação ferroviária de forma segura e eficiente.

Apesar de sua importância, o BRG apresentava uma série de limitações. Seu hardware era proprietário, o sistema operacional era exclusivo da fornecedora original e o código de comunicação era compatível apenas com essa plataforma. Com a descontinuação do equipamento e a escassez de peças sobressalentes, tornou-se impossível realizar manutenções, atualizações ou implementar melhorias. Essa dependência tecnológica impedia a evolução do sistema e representava um risco crescente à continuidade e à eficiência da operação ferroviária.

A substituição de sistemas legados é um dos maiores desafios enfrentados por organizações que buscam inovação e eficiência operacional. Segundo Gartner (2025), embora esses sistemas ainda sejam funcionais e familiares, eles impõem sérias limitações, como altos custos de manutenção, baixa escalabilidade, riscos de segurança, dificuldades de integração com tecnologias modernas e restrições à inovação.

Diante desse cenário, foi concebido o projeto **RailBridge**, uma solução de software baseada em arquitetura aberta, desenvolvida para substituir o BRG e modernizar a comunicação entre os dispositivos de sinalização e o SGF. A nova solução utiliza tecnologias modernas como **JSON** para estruturação de comandos e **RabbitMQ** como middleware de mensagens, promovendo uma comunicação mais eficiente, segura e escalável. Além disso, o RailBridge é

projetado para rodar em qualquer hardware e sistema operacional, eliminando a dependência de fornecedores específicos e aumentando a resiliência da operação.

A proposta vai além da simples substituição de hardware e software: trata-se de uma transformação digital que habilita a comunicação via Ethernet, permite contingência automática em caso de falhas, amplia a capacidade de processamento e facilita a integração com futuras funcionalidades.

Este artigo apresenta os desafios enfrentados, as decisões tecnológicas adotadas, os benefícios alcançados e as perspectivas futuras da solução RailBridge, demonstrando como a inovação pode ser um vetor estratégico para a modernização do setor ferroviário brasileiro.

2. METODOLOGIA

A metodologia adotada para o desenvolvimento da nova solução teve como base a análise técnica do sistema legado e a definição de critérios funcionais e operacionais que orientaram todas as etapas do projeto. Esses critérios foram estabelecidos a partir das limitações identificadas no BRG e das necessidades futuras da operação ferroviária, servindo como norteadores metodológicos para o desenho da nova arquitetura e solução em software.

A seguir, são descritos os princípios técnicos e funcionais que fundamentaram o desenvolvimento da solução:

2.1. Diagnóstico do Problema

A primeira etapa da metodologia consistiu na análise técnica e operacional do sistema legado BRG, com foco na identificação de suas limitações e riscos associados à continuidade operacional. Como já foi citado, o Bit regenerator passou a representar um ponto crítico de vulnerabilidade. Fabricado por uma empresa que descontinuou tanto o hardware quanto o sistema operacional proprietário, o equipamento tornou-se insustentável do ponto de vista de manutenção e evolução. A impossibilidade de realizar reparos, aliada à escassez de peças de reposição, colocava em risco a continuidade operacional da ferrovia. Além disso, sua arquitetura fechada impedia qualquer tipo de customização, integração com novas tecnologias ou implementação de melhorias funcionais.

A seguir, apresenta-se uma tabela com os principais pontos negativos identificados e seus respectivos impactos operacionais:

Tabela 1 - Limitações vs impactos operacionais

Limitação Identificada	Impacto Operacional
Hardware proprietário e exclusivo	Dependência total do fornecedor; impossibilidade de substituição por equipamentos disponíveis no mercado
Sistema operacional descontinuado	Falta de suporte técnico e atualizações de segurança
Escassez de peças de reposição	Riscos de paralisação total em caso de falhas
Arquitetura fechada	Impossibilidade de customização ou evolução do sistema
Comunicação baseada em protocolo serial (STD)	Baixa velocidade e flexibilidade na troca de dados
Conversão complexa entre objetos Java e C++ (via protocolo OMS)	Processo lento e sujeito a falhas na comunicação com o centro de controle
Incompatibilidade com tecnologias modernas (ex: Ethernet)	Dificuldade de integração com novos sistemas e redes
Ausência de contingência automatizada	Falta de resposta rápida em caso de falhas de comunicação com campo

Fonte: Autores (2025).

2.2. Levantamento de Requisitos do RailBridge

Com base no diagnóstico técnico da arquitetura obsoleta do BRG, foram definidos os requisitos fundamentais para o desenvolvimento de uma nova solução que garantisse a continuidade operacional da ferrovia, ao mesmo tempo em que promovesse modernização, flexibilidade e escalabilidade tecnológica.

A seguir, são apresentados os principais requisitos identificados para que o desenvolvimento da solução:

2.2.1. Independência de Hardware e Sistema Operacional:

A nova solução deveria ser projetada para operar em qualquer plataforma computacional, seja ela baseada em sistemas Windows, Linux ou outros ambientes compatíveis. Essa independência elimina a dependência de hardware proprietário, que no caso do BRG era exclusivo da fornecedora original e já não estava mais disponível

no mercado. Além disso, a adoção de uma arquitetura multiplataforma permite maior flexibilidade na escolha de equipamentos, facilita a reposição em caso de falhas e reduz significativamente os custos de manutenção e aquisição. Essa abordagem também contribui para a longevidade da solução, permitindo sua adaptação a diferentes ambientes operacionais ao longo do tempo. Conforme discutido por diversos autores (HEITKÖTTER, H.; HANSCHKE, S.; MAJCHRZAK, T., 2013; EL-KASSAS et al., 2015), o desenvolvimento multiplataforma proporciona agilidade na entrega de software, reutilização de código, menor custo de manutenção e maior alcance de usuários ao permitir que uma única aplicação seja executada em diferentes sistemas operacionais.

2.2.2. Comunicação Bidirecional com Suporte à Múltiplos Protocolos:

A comunicação entre o centro de controle e os dispositivos de campo é essencial para a operação ferroviária. Por isso, a nova solução deveria manter a comunicação bidirecional, garantindo que comandos e indicações fluam de forma contínua e confiável nos dois sentidos. Para isso, era necessário suportar tanto o protocolo serial STD, uma vez que os equipamentos de campo atualmente em operação ainda utilizam esse tipo de comunicação, quanto a comunicação via Ethernet, cuja implementação foi pensada estrategicamente para viabilizar a transição futura, à medida que a modernização dos dispositivos de campo for sendo realizada. Essa abordagem garante compatibilidade com a infraestrutura existente e, ao mesmo tempo, prepara o sistema para evoluções tecnológicas sem necessidade de reestruturações profundas. A coexistência de diferentes tecnologias de comunicação em um mesmo sistema é apontada como uma estratégia eficaz para garantir compatibilidade com infraestruturas legadas, ao mesmo tempo em que se promove a adaptação gradual a soluções mais modernas. Essa abordagem facilita a integração entre dispositivos de diferentes gerações, assegura continuidade operacional durante processos de transição tecnológica e amplia a flexibilidade do sistema frente a futuras evoluções (TYROVOLAS; HAJNAL, 2021).

2.2.3. Suporte a Hardware Dualizado e Expansível:

A nova solução deveria ser compatível com arquiteturas de hardware dualizado, permitindo a operação simultânea em dois equipamentos físicos redundantes, com possibilidade de expansão para mais de dois dispositivos. Essa configuração visa garantir **alta disponibilidade e continuidade operacional**, especialmente em ambientes críticos como o ferroviário, onde interrupções podem gerar impactos

logísticos e financeiros significativos. Segundo Julitz, Tordeux e Löwer (2022), arquiteturas que utilizam processadores em modo lockstep e esquemas de redundância (como DMR ou TMR) oferecem maior tolerância a falhas e probabilidades elevadas de sobrevivência do sistema em cenários críticos, como os encontrados em sistemas automotivos autônomos. Essas abordagens permitem mascarar falhas de hardware e aumentar a confiabilidade operacional por meio de replicação e votação de resultados

A compatibilidade com esse tipo de arquitetura foi considerada um requisito estratégico, pois amplia a **escalabilidade horizontal** da solução e reforça sua capacidade de operar de forma confiável em ambientes dinâmicos e em constante evolução.

2.3. Escolha de Tecnologias

A escolha das tecnologias utilizadas no desenvolvimento da nova solução foi orientada pelos requisitos técnicos definidos na etapa anterior, com foco em flexibilidade, escalabilidade, facilidade de manutenção e compatibilidade com a infraestrutura existente. A seguir, são apresentadas as principais tecnologias adotadas:

2.3.1. Nodejs:

A lógica de comunicação da nova solução foi desenvolvida utilizando um framework do JavaScript, o nodejs, uma linguagem moderna, multiplataforma e amplamente utilizada no desenvolvimento de aplicações distribuídas. Sua sintaxe acessível, vasta comunidade de suporte e compatibilidade com diferentes ambientes de execução foram fatores determinantes para sua escolha. Além disso, o uso de JavaScript permitiu maior agilidade no desenvolvimento e facilidade de integração com outras tecnologias utilizadas no projeto. Conforme Ioffe (2023), JavaScript e Node.js são ideais para aplicações distribuídas modernas por conta da sua estrutura *event-driven* e I/O não bloqueante, o que favorece a escalabilidade, confiabilidade e integração entre microserviços.

2.3.2. RabbitMQ:

Para a troca de mensagens entre os módulos do sistema, foi adotado o **RabbitMQ**, um middleware – software intermediário entre diferentes sistemas - de mensageria robusto e confiável, baseado no protocolo AMQP (Advanced Message Queuing Protocol). O RabbitMQ permite a comunicação assíncrona entre os componentes, garantindo o desacoplamento entre os processos de envio e recebimento de dados. Isso contribui diretamente para a escalabilidade e resiliência da solução, além de permitir a criação de filas específicas para diferentes tipos de

mensagens, como comandos, indicações e logs. Segundo Čatović, Buzadžija e Lemeš (2023), o RabbitMQ se destaca como um middleware eficiente na arquitetura de microsserviços por atuar como intermediário entre serviços, permitindo a distribuição equilibrada de tarefas e contribuindo para a redução da latência e do uso de recursos.

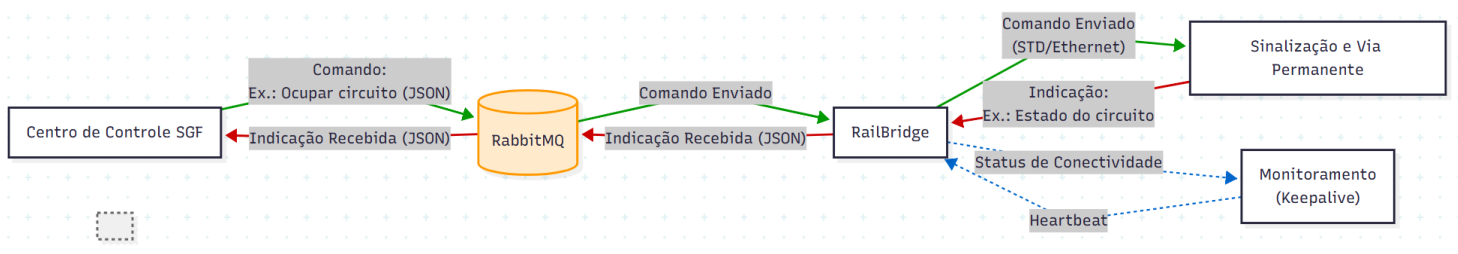
2.3.3. Json:

O formato JSON (JavaScript Object Notation) foi escolhido como padrão para a estruturação e troca de dados entre os módulos do sistema, devido à sua legibilidade, leveza e ampla aceitação em aplicações distribuídas. Sua compatibilidade com diversas linguagens de programação, incluindo JavaScript (usado no Node.js), facilita a integração e a serialização de comandos e indicações. Além disso, o JSON suporta a escalabilidade ao permitir a representação flexível de dados complexos, como hierarquias de mensagens, o que é essencial para a operação da EFC. Conforme os princípios FAIR (Encontrável, Acessível, Interoperável, Reutilizável) adotados pelo NCBI (Centro Nacional de Informação em Biotecnologia), o JSON é amplamente compreendido por humanos e máquinas, facilitando a interoperabilidade entre sistemas e a integração com APIs modernas no ecossistema REST (NCBI, 2023).

3. ARQUITETURA DA SOLUÇÃO

A implementação do RailBridge foi projetada para superar as limitações do BRG, promovendo a transformação digital da EFC através de uma solução flexível e resiliente. A Figura 1 ilustra a arquitetura geral, destacando a conexão entre o SGF, o RailBridge e os dispositivos de campo.

Figura 1 – Arquitetura de comunicação (Centro x Campo)



Fonte: Autores (2025).

A tabela a seguir detalha os principais módulos que compõem a arquitetura do RailBridge:

Tabela 2 – Módulos Acoplados ao RailBridge

Módulo	Funcionalidade	Ganho
Centro de Controle (SGF)	Interface responsável pelo Gerenciamento ferroviário, enviando comandos (ex.: ligar sinais) e recebendo indicações (ex.: estado de sensores) do campo.	Permite decisões rápidas e seguras no centro de controle, otimizando o tráfego ferroviário.
Broker RabbitMQ	Facilita a troca assíncrona de comandos e indicações entre o Centro de Controle e o RailBridge.	Garante comunicação confiável e em tempo real, reduzindo atrasos e falhas operacionais.
RailBridge	Conecta o Centro de Controle aos dispositivos de Sinalização e Via Permanente, suportando comunicação serial (STD) e moderna (Ethernet/multicast).	Moderniza a infraestrutura, eliminando a dependência do BRG promovendo flexibilidade e escalabilidade.
Sinalização e Via Permanente	Equipamentos de campo (sinais, sensores, atuadores) que controlam e monitoram a ferrovia.	---
Monitoramento (Keepalive)	Verifica a conectividade entre o RailBridge e o Centro de Controle, garantindo alta disponibilidade.	Minimiza interrupções, assegurando resiliência e continuidade operacional em caso de falhas.

--	--	--

Fonte: Autores (2025).

Diferentemente do BRG, o RailBridge conecta o Centro de Controle (SGF) aos dispositivos de Sinalização e Via Permanente com flexibilidade e resiliência, suportando tanto tecnologias legadas quanto modernas. A seguir, exploramos como essa arquitetura foi implementada para promover a inovação digital e a eficiência operacional da Estrada de Ferro Carajás (EFC).

3.1. Implementação Técnica:

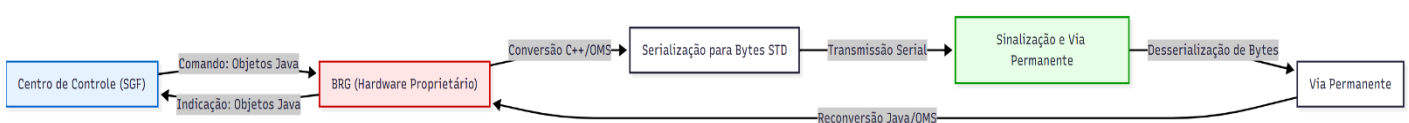
A implementação da solução RailBridge foi conduzida com foco em modularidade, interoperabilidade e robustez, utilizando tecnologias modernas e padronizadas que atendem aos requisitos definidos na metodologia. A seguir, são descritos os principais componentes técnicos e suas respectivas funções na arquitetura do sistema:

3.1.1 RailBridge

O RailBridge foi desenvolvido em Node.js, adotando uma estrutura modular que permite fácil manutenção e evolução. Sua principal função é realizar a conversão de comandos e indicações entre o SGF e os dispositivos de campo.

Anteriormente, com o BRG, a conversão de informações envolvia um processo mais complexo e rígido: os comandos originados no SGF eram estruturados como objetos Java, convertidos para objetos C++ por meio do protocolo OMS, e então transformados em bits para transmissão via protocolo serial STD até os equipamentos de campo. No sentido inverso, as indicações provenientes do campo eram recebidas em formato binário, convertidas para objetos C++ e, posteriormente, para objetos Java, possibilitando sua interpretação pelo SGF. Esse fluxo garantiu por anos a comunicação bidirecional entre o centro de controle e a via permanente, mas apresentava limitações de flexibilidade e escalabilidade.

Figura 2 – Arquitetura de comunicação antiga



Fonte: Autores (2025).

Com a nova implementação, os comandos originados no SGF são estruturados em JSON, convertidos em bytes e transmitidos ao campo via protocolo STD (serial) ou Ethernet, dependendo da infraestrutura disponível. A lógica de conversão foi implementada de forma flexível, permitindo que o sistema reconheça o tipo de protocolo utilizado e adapte o formato da mensagem conforme necessário. Além disso, o RailBridge é capaz de realizar o caminho inverso: receber dados do campo, decodificá-los e reestruturá-los em JSON para envio ao SGF. A compatibilidade com multicast Ethernet também foi incorporada, preparando o sistema para futuras expansões e modernizações dos equipamentos de campo.

3.1.2. Broker RabbitMQ

O RabbitMQ foi adotado como um middleware de mensageria robusto, desempenhando um papel central na garantia de uma troca assíncrona e altamente confiável de dados entre os módulos do sistema RailBridge na EFC. Essa solução, baseada no protocolo AMQP, ou seja, um modelo de comunicação que utiliza filas especializadas para gerenciar diferentes tipos de mensagens, permite que os módulos atuem como produtores (que enviam os dados) e consumidores (que recebem os dados). Essa arquitetura desacoplada, proporcionada pelo AMQP otimiza a comunicação entre os componentes do sistema, garantindo escalabilidade, confiabilidade e controle eficiente na logística ferroviária. As filas específicas incluem:

Fila de Comandos: Dedicada ao envio de instruções do Centro de Controle (SGF) para os dispositivos de Sinalização e Via Permanente, como ordens para ativar sinais ou ajustar atuadores, assegurando respostas rápidas ao tráfego ferroviário.

Fila de Indicações: Responsável por receber e organizar informações enviadas pelos equipamentos de campo, como o estado de sensores (ex.: detecção de trens) ou falhas em sinais, permitindo monitoramento em tempo real.

Fila de Logs e Eventos: Projetada para capturar registros operacionais, alertas de falhas e eventos críticos, facilitando auditorias e análises para a gestão do sistema e a futura implementação de manutenção preditiva.

Essa estrutura de filas promove o desacoplamento entre os módulos – como o SGF, o RailBridge e os dispositivos de campo –, permitindo que cada componente funcione de forma independente, sem depender da disponibilidade imediata dos outros. Por exemplo, se o RailBridge estiver temporariamente off-line, as mensagens ficam armazenadas nas filas do RabbitMQ, evitando perdas de dados e garantindo continuidade operacional. Esse design

aumenta significativamente a resiliência do sistema, reduzindo o impacto de falhas e protegendo a segurança da via permanente.

3.1.2. Monitoramento (KeepAlive)

Para assegurar a alta disponibilidade e a continuidade operacional da comunicação na Estrada de Ferro Carajás (EFC), foi implementado um mecanismo de keepalive, uma funcionalidade crítica integrada ao RailBridge. Esse sistema realiza verificações periódicas da conectividade entre o RailBridge e o Centro de Controle (SGF), utilizando pacotes de teste enviados a intervalos regulares (ex.: a cada 5 segundos) para detectar falhas em tempo real, como quedas de rede ou interrupções nos dispositivos de Sinalização e Via Permanente. Em caso de falha detectada, o keepalive aciona automaticamente rotinas de contingência: reenvia mensagens não entregues, alterna para um canal de comunicação secundário (ex.: de serial STD para Ethernet, se disponível), ou ativa uma instância redundante em hardware dualizado, minimizando o tempo de inatividade.

Esse monitoramento contínuo é fundamental para evitar interrupções no controle do tráfego ferroviário, como atrasos na ativação de sinais ou falhas na detecção de trens, garantindo respostas rápidas a incidentes. A resiliência proporcionada pelo keepalive protege a segurança do processo ferroviário e reduz riscos logísticos. Além disso, os dados coletados pelo keepalive – como histórico de falhas e tempos de resposta – podem ser utilizados para análises preditivas, ajudando a prever manutenções e otimizar a infraestrutura da EFC.

4.RESULTADOS

4.1. Eficiência Operacional

A implementação do RailBridge trouxe ganhos significativos na eficiência operacional da EFC. A substituição do BRG por uma solução baseada em comunicação moderna, como Ethernet e JSON, reduziu os tempos de processamento de comandos, como a ativação de sinais ao longo da via permanente, permitindo uma resposta mais rápida ao tráfego ferroviário. A flexibilidade para operar em múltiplos protocolos (serial STD e Ethernet) facilitou a integração com dispositivos legados e novos, otimizando a coordenação entre o Centro de Controle (SGF) e os equipamentos de campo. Essa melhoria reflete uma transformação digital que agiliza a logística de transporte de minério, alinhando-se às metas de eficiência do setor.

4.2. Suporte a Manutenção Preditiva:

A introdução da Fila de Logs e Eventos no RabbitMQ abriu caminho para a manutenção preditiva. Registros detalhados de falhas, como panes em atuadores ou variações nos sensores, são capturados e organizados, permitindo análises que identificam padrões de desgaste na infraestrutura ferroviária. Essa capacidade prepara a EFC para intervenções preventivas, reduzindo custos de reparos emergenciais e aumentando a durabilidade dos equipamentos de campo. A flexibilidade para integrar protocolos futuros, como OPC UA, reforça o potencial de inovação a longo prazo.

4.3. Eficiência Operacional

A independência de hardware e sistema operacional do RailBridge eliminou a dependência do fornecedor do BRG, que havia se tornado um gargalo devido à obsolescência e escassez de peças. Agora, a EFC pode utilizar equipamentos comerciais (Windows ou Linux), facilitando reposições e reduzindo custos de manutenção. Essa mudança também diminuiu os gastos com suporte técnico especializado, liberando recursos para investimentos em inovação, como a expansão da rede Ethernet ao longo da ferrovia.

Tabela 3 – Ganhos Operacionais

Indicador	Descrição do Resultado	Impacto na Operacional
Eficiência Operacional	Redução no tempo de processamento de comandos (ex.: ativação de sinais) com uso de Ethernet e JSON.	Agiliza o tráfego ferroviário e a logística de minério.
Resiliência da Comunicação	Alta disponibilidade com keepalive e alternância automática para instâncias redundantes.	Minimiza interrupções, garantindo segurança na via permanente.
Manutenção Preditiva	Coleta de logs e eventos para análise de falhas (ex.: sensores, atuadores).	Permite intervenções preventivas, reduzindo custos de reparo.

Redução de Custos	Eliminação da dependência de hardware proprietário e suporte a equipamentos comerciais.	Diminui gastos com manutenção e reposição de peças.

Fonte: Autores (2025).

Figura 3 - Comparativo técnico entre o módulo legado (hack obsoleto) e o novo módulo (hack atualizado), evidenciando a evolução da arquitetura, protocolos de comunicação e compatibilidade com infraestrutura moderna.



Fonte: Autores (2025).

5. CONCLUSÃO

A implementação do RailBridge representa um avanço transformador na modernização da Estrada de Ferro Carajás (EFC), superando os desafios impostos pela obsolescência do BRG e sua dependência de hardware proprietário. Este projeto exemplifica como a transformação digital, sustentada por tecnologias atuais, pode redefinir a eficiência e a segurança do processo ferroviário, atendendo às demandas logísticas do transporte de minério. Os resultados obtidos – como a redução de tempos de processamento, a resiliência assegurada pelo keepalive e o suporte à manutenção preditiva via logs – consolidam a capacidade da EFC de operar com automação e controle inteligente, minimizando interrupções e custos operacionais.

Além dos ganhos imediatos, o RailBridge abre perspectivas para inovações futuras, como a possibilidade de integração com outros ativos de automação, interface com outros sistemas e análises preditivas mais robustas, ampliando a interoperabilidade e a flexibilidade da operação. Em um contexto de renovação de concessões e investimentos em infraestrutura, esta solução demonstra como a resiliência digital e a adoção de novas tecnologias podem impulsionar a competitividade e a sustentabilidade do transporte de carga. À medida que os desafios evoluem, o RailBridge estabelece uma base sólida para continuar avançando, assegurando que a EFC permaneça na vanguarda da inovação ferroviária.

6. REFERÊNCIAS

BRASIL. Programa de Parcerias de Investimentos. *Estrada de Ferro Carajás (EFC)*. 2025. Disponível em: <https://ppi.gov.br/projetos/estrada-de-ferro-carajas-efc>.

ĆATOVIĆ, A.; BUZADIJA, N.; LEMEŠ, S. *Microservice development using RabbitMQ message broker*. Sarajevo: Isatis Software Solutions / University of Zenica, 2023. Disponível em:

https://www.researchgate.net/publication/375540532_Microservice_development_using_RabbitMQ_message_broker.

EL-KASSAS, W. S. et al. *Cross-platform mobile applications development: a review*. *International Journal of Computer Applications*, v. 118, n. 6, p. 26-32, 2015.

HEITKÖTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. *Evaluating Cross-Platform Development Approaches for Mobile Applications*. In: INTERNATIONAL CONFERENCE ON WEB INFORMATION SYSTEMS AND TECHNOLOGIES, 8., 2013, [S. l.]. Anais [...]. [S. l.]: SciTePress, 2013.

IOFFE, A. *Node.js and Microservices: The Perfect Combination for Scalable Web Applications*. *MultiQoS*, 2023. Disponível em: <https://multiqos.com/blogs/nodejs-and-microservices/>.

JULITZ, T. M.; TORDEUX, A.; LÖWER, M. *Reliability of fault-tolerant system architectures for automated driving*. arXiv, arXiv:2210.04040, 2022. Disponível em: <https://arxiv.org/abs/2210.04040>.

NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION (NCBI). *Why JSONL?*. NCBI Datasets, (2023). Disponível em: <https://www.ncbi.nlm.nih.gov/datasets/docs/v2/reference-docs/file-formats/metadata-files/why-jsonl/>.

SINGIRIKONDA, E. K. *Modernizing legacy systems in the new digital age*. *InnoTech Today*, 2025. Disponível em: <https://innotechtoday.com/modernizing-legacy-systems-in-the-new-digital-age>.

TYROVOLAS, M.; HAJNAL, T. *Inter-communication between programmable logic controllers using IoT technologies: a Modbus RTU/MQTT approach*. Preprint, abr. 2021. Disponível em: <https://arxiv.org/abs/2102.05988>.