

Infraestrutura Descentralizada para Registro Auditável de Emissões Veiculares de CO₂: Aplicação de *Blockchain* em Veículos Conectados

Miguel Amaral¹, Matheus Andrade¹, Hilton Machado¹,
Marianne Silva^{1,2}, Ivanovitch Silva¹

¹ Universidade Federal do Rio Grande do Norte (UFRN), Natal - Brasil

² Universidade Federal de Alagoas (UFAL), Penedo - Brasil

{miguel.amaral.111, matheus.diniz.122, hilton.machado.088}@ufrn.edu.br,
marianne.silva@penedo.ufal.br, ivanovitch.silva@ufrn.br

Abstract. *The digitalization of the automotive sector and advances in communication technologies have enabled the development of connected vehicles capable of monitoring and reporting CO₂ emissions. However, ensuring the reliability and integrity of this data remains a significant challenge, especially in the face of fraudulent practices such as greenwashing, which are facilitated by centralized systems vulnerable to tampering. This paper proposes an innovative blockchain-based solution using the Hyperledger Besu network with the QBFT consensus protocol and smart contracts in Solidity, aiming to securely, transparently, and immutably record vehicular emissions collected via OBD-II devices integrated into a mobile application. Tests conducted in controlled and real-world environments confirmed low latency, accurate logging, and operational efficiency. The results indicate that the proposed solution significantly strengthens the credibility of environmental initiatives and holds strong potential to positively impact carbon credit markets by promoting verifiable and auditable sustainable practices.*

Resumo. *A digitalização do setor automotivo e o avanço das tecnologias de comunicação possibilitaram o desenvolvimento de veículos conectados capazes de monitorar e reportar emissões de CO₂. Contudo, assegurar a confiabilidade e integridade desses dados ainda é um desafio significativo, especialmente diante de práticas fraudulentas, como o greenwashing, facilitadas por sistemas centralizados e vulneráveis a adulterações. Este artigo propõe uma solução baseada em blockchain, utilizando a rede Hyperledger Besu com protocolo QBFT e contratos inteligentes em Solidity, visando registrar de forma segura, transparente e imutável as emissões veiculares coletadas por dispositivos OBD-II integrados a uma aplicação móvel. Testes realizados em cenários controlados e reais confirmaram baixa latência, precisão dos registros e eficiência operacional. Os resultados indicam que a solução proposta fortalece significativamente a credibilidade das iniciativas ambientais e possui potencial para influenciar positivamente os mercados de créditos de carbono, incentivando práticas sustentáveis verificáveis e auditáveis.*

1. Introdução

A crescente digitalização do setor automotivo e o avanço das tecnologias de comunicação têm impulsionado o desenvolvimento de veículos conectados, capazes de coletar, processar e transmitir dados em tempo real [Banerjee et al. 2024]. Essa nova era da mobilidade inteligente oferece oportunidades significativas para otimizar o tráfego, melhorar a segurança viária e reduzir impactos ambientais [Lu et al. 2022, Biswas et al. 2023]. Contudo, garantir a confiabilidade dos dados veiculares, especialmente em aplicações críticas como monitoramento e compensação de emissões de carbono, permanece um desafio importante [Alalwany and Mahgoub 2024].

Atualmente, muitos sistemas para monitoramento de emissões dependem de bases de dados centralizadas, que são vulneráveis à adulteração e manipuláveis por práticas fraudulentas, como o *greenwashing*, caracterizado por alegações enganosas sobre sustentabilidade sem efetivas reduções de emissões [Ling and Wang 2024, Lu et al. 2022]. Nesse contexto, é essencial a existência de sistemas robustos e transparentes para garantir que créditos de carbono concedidos correspondam efetivamente a emissões evitadas [Li et al. 2023]. A tecnologia *blockchain* surge como solução ideal para mitigar esses problemas, devido à sua característica de imutabilidade e auditabilidade, garantindo que os dados, uma vez registrados, não possam ser alterados sem o consenso da rede [Banerjee et al. 2024, Yang et al. 2024]. Embora soluções baseadas em *blockchain* sejam promissoras, abordagens existentes frequentemente enfrentam desafios técnicos como escalabilidade limitada, custos elevados e desempenho inadequado para aplicações em tempo real [Banerjee et al. 2024].

Diante dessas limitações, este trabalho propõe uma solução inovadora utilizando a rede Hyperledger Besu com protocolo *Quorum Byzantine Fault Tolerance* (QBFT), contratos inteligentes desenvolvidos em Solidity e operação sem custos de transação (*free gas*). Essa abordagem permite registrar, de forma segura e transparente, dados de emissões de CO_2 coletados por dispositivos OBD-II instalados em veículos conectados. Além disso, a solução incorpora uma API intermediária que integra dispositivos OBD-II à blockchain, permitindo que dados sobre trajetos e emissões sejam automaticamente registrados.

Testes realizados em ambiente controlado e em cenário real confirmaram que o sistema registra corretamente os dados veiculares, permitindo recuperação eficiente e garantindo integridade. Dessa forma, a proposta fortalece a credibilidade das informações fornecidas aos órgãos reguladores, consumidores e mercados de créditos de carbono, promovendo práticas ambientais sustentáveis de forma justa e verificável [Ling and Wang 2024].

O restante deste artigo está estruturado da seguinte maneira: a Seção 2 apresenta trabalhos relacionados e discute a relevância do uso da *blockchain* para o rastreamento de emissões. A Seção 3 aborda os conceitos fundamentais da tecnologia empregada e descreve a plataforma escolhida. A Seção 4 detalha a metodologia utilizada, descrevendo a arquitetura e os componentes do sistema. Na Seção 5, são apresentados e discutidos os resultados dos testes realizados. Por fim, a Seção 6 conclui o trabalho e aponta possíveis caminhos para futuras pesquisas.

2. Trabalhos Relacionados

O uso da *blockchain* para garantir a confiabilidade e transparência de dados ambientais tem sido amplamente explorado. Em particular, o rastreamento das emissões de carbono requer soluções robustas para minimizar fraudes e manipulações, especialmente em contextos suscetíveis ao *greenwashing*. Esse cenário demanda não apenas mecanismos de registro imutáveis, mas também estruturas capazes de fornecer auditabilidade em tempo real e interoperabilidade com sistemas de coleta de dados heterogêneos. Tecnologias descentralizadas como a *blockchain* emergem como alternativa promissora ao oferecer transparência, integridade e rastreabilidade dos dados, sendo, portanto, alvo de diversas propostas acadêmicas e industriais nos últimos anos. Entretanto, soluções existentes frequentemente apresentam limitações importantes relacionadas à escalabilidade, altos custos operacionais ou dificuldades práticas na integração com sistemas veiculares, justificando a busca por abordagens mais eficazes.

Em [Gu et al. 2024], é apresentado um framework baseado em contratos inteligentes (*smart contracts*) para combater o *greenwashing* em divulgações ambientais corporativas. O estudo define categorias específicas de risco de manipulação, demonstrando a capacidade da *blockchain* em fornecer auditoria imutável das informações ambientais. Porém, seu escopo limita-se à auditoria corporativa, não contemplando diretamente os desafios técnicos da coleta em tempo real e a integração com sensores veiculares, aspectos fundamentais abordados pelo presente trabalho.

Já em [Liu et al. 2024], os autores propõem uma abordagem baseada em autenticação anônima e segurança avançada para o registro das emissões veiculares, empregando *blockchain*, hashes camaleão e assinaturas digitais (ECDSA). Embora este método assegure privacidade robusta e segurança das transações, a solução carece de uma arquitetura detalhada e prática para integração direta com sistemas embarcados de coleta de dados, além de não apresentar uma validação empírica extensiva em ambiente real, o que é um foco do presente estudo.

O estudo apresentado por [Lu et al. 2022], conhecido como STRICTs, propõe uma plataforma de auditoria e negociação de permissões de emissão veicular baseada na plataforma Hyperledger Fabric. A abordagem estabelece limites individuais de emissão e permite negociação de créditos de carbono entre usuários. Contudo, a complexidade operacional do sistema e sua ênfase na negociação de permissões individuais podem limitar sua escalabilidade prática e aumentar os custos de implementação, especialmente em larga escala. Em contraste, nosso trabalho propõe uma abordagem mais direta e eficiente, utilizando a rede Hyperledger Besu com o protocolo QBFT, que oferece uma operação sem taxas (*free gas*), integração direta com sensores OBD-II, e um modelo simplificado, escalável e econômico de rastreamento das emissões.

Portanto, embora os trabalhos analisados demonstrem claramente o potencial da *blockchain* em aprimorar a transparência e segurança no rastreamento de emissões, persistem limitações significativas em termos de escalabilidade, custo, integração prática e complexidade operacional. Diferentemente dessas abordagens, a solução proposta por este trabalho busca suprir essas lacunas por meio de uma plataforma eficiente baseada no Hyperledger Besu, integrando uma API intermediária para comunicação direta com dispositivos OBD-II e proporcionando um mecanismo robusto, acessível e auditável para compensação de emissões no setor automotivo.

3. Abordagem Proposta

A arquitetura proposta visa garantir a rastreabilidade e auditabilidade das emissões veiculares de CO_2 por meio de uma solução modular baseada em *blockchain*. A solução integra a coleta de dados embarcados, processamento em tempo real, estimativa de emissões e registro imutável em uma rede permissionada Hyperledger Besu. Para facilitar a compreensão estrutural, a arquitetura é apresentada em módulos com responsabilidades bem definidas, conforme ilustrado na Figura 1.

O módulo inicial da solução realiza a coleta de dados embarcados, sendo responsável pela interface com o veículo por meio de dispositivos OBD-II e pela aquisição, em tempo real, de informações operacionais como velocidade, rotação do motor, localização, entre outras. Essa coleta é mediada por um aplicativo móvel, que também serve como interface para o usuário iniciar e finalizar o monitoramento. Em seguida, o módulo de processamento e armazenamento recebe os dados brutos, realiza seu tratamento e os armazena temporariamente de forma eficiente, utilizando uma arquitetura orientada a eventos, que garante desacoplamento e escalabilidade. O terceiro módulo consiste em um serviço de *backend* com uma API REST, acionada ao final de cada trajeto. Este serviço é responsável por recuperar os dados temporais armazenados, aplicar o modelo de cálculo das emissões de CO_2 e consolidar as informações da viagem (como distância percorrida e volume estimado de emissão) em um formato padronizado. O quarto módulo interage com a *blockchain*, funcionando como intermediário entre a API e o contrato inteligente, sendo responsável por receber os dados consolidados, construir, assinar e submeter a transação para registro. Por fim, o quinto módulo corresponde à camada de persistência, composta pela rede *blockchain* permissionada e pelo contrato inteligente nela implantado, assegurando o registro imutável e auditável dos dados da viagem, com aplicação de regras de validação e controle de acesso.

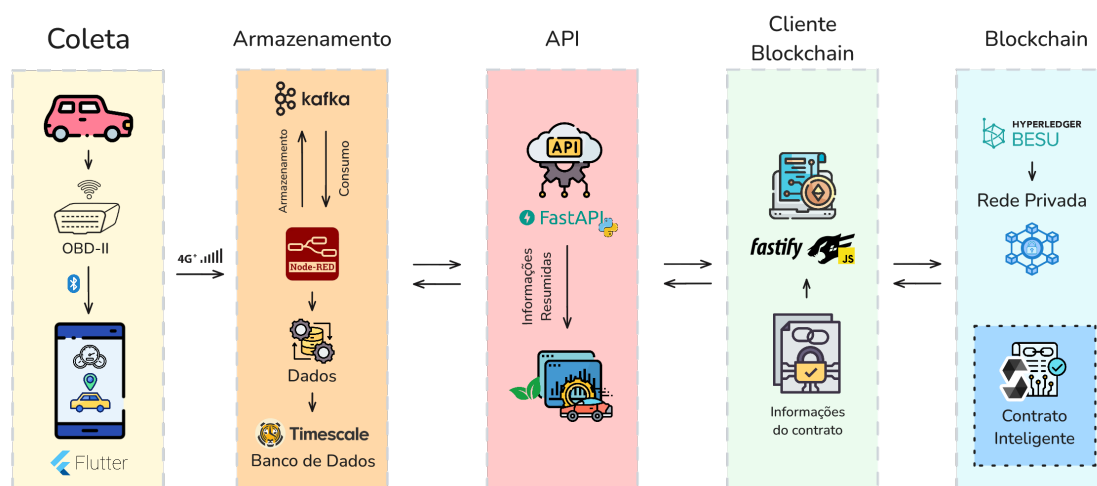


Figura 1. Componentes e fluxo da solução proposta.

A interação entre esses módulos ao longo de um ciclo completo de uso, desde a coleta até o registro na *blockchain*, é ilustrada no diagrama de sequência da Figura 2. O usuário inicia o trajeto no aplicativo, que envia os dados veiculares para a API. Esta processa os dados, calcula as emissões e aciona o cliente *blockchain*, responsável por

invocar o contrato inteligente. Após a execução da transação a confirmação é enviada ao aplicativo, completando o ciclo de auditoria da viagem.

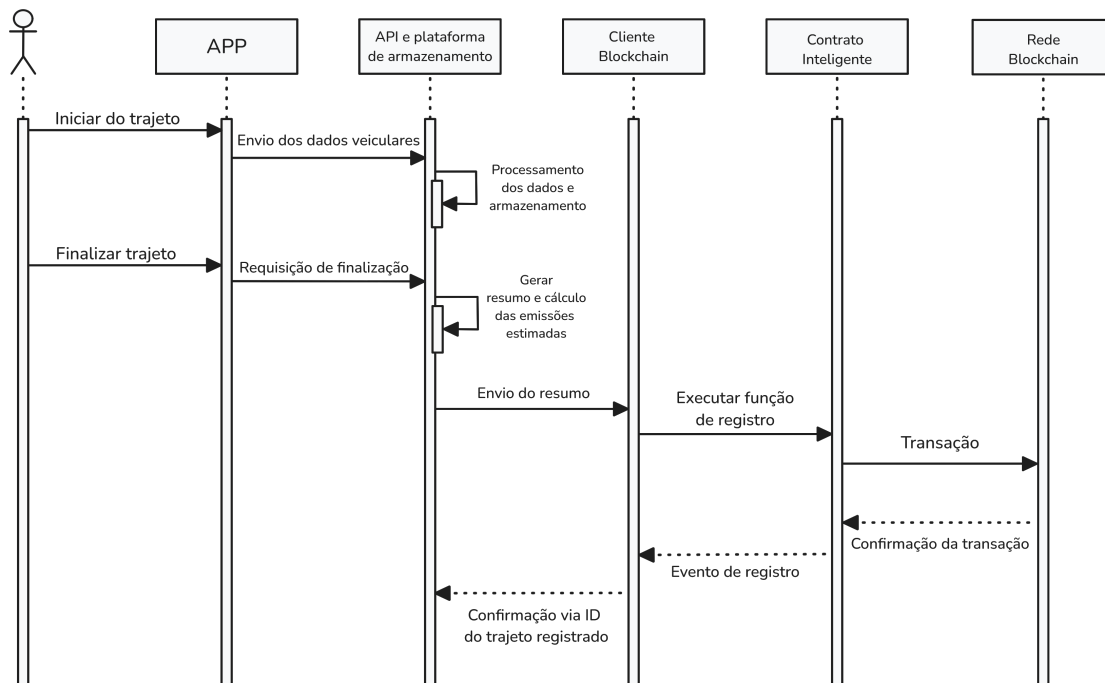


Figura 2. Componentes e fluxo da solução proposta.

As subseções seguintes detalham a implementação específica de cada módulo e da infraestrutura de *blockchain*.

3.1. Coleta de dados veiculares

A coleta de dados na solução proposta é realizada por meio da interface OBD-II (*On-Board Diagnostics*), um padrão internacional adotado em veículos modernos que permite acessar dados operacionais em tempo real diretamente da central eletrônica do automóvel (*ECU – Engine Control Unit*). Utilizando um leitor OBD-II com conectividade Bluetooth, conectado à porta de diagnóstico do veículo, é possível obter informações como velocidade, rotação do motor (RPM), temperatura, consumo de combustível, localização e outros parâmetros disponíveis.

Esses dados são transmitidos em tempo real para um aplicativo móvel desenvolvido em Flutter, que atua como a interface principal entre o usuário e o sistema. O motorista pode iniciar e finalizar a coleta de forma manual por meio do aplicativo, controlando a duração do monitoramento durante o trajeto. A precisão e o detalhamento das informações coletadas são fundamentais para a estimativa posterior das emissões de CO_2 e para análises relacionadas ao desempenho veicular.

3.2. Armazenamento e Processamento de Dados

Após a coleta, os dados são transmitidos para o serviço de backend que, por meio de sua API REST, os processa e os insere em um fluxo de tratamento baseado em eventos, projetado para garantir eficiência, modularidade e escalabilidade. Essa arquitetura segue

o modelo proposto em [Azevedo et al. 2023] e permite desacoplar a coleta de dados do processamento e do registro final em blockchain.

O fluxo inicia-se no Node-RED, configurado como produtor, que recebe os dados estruturados do aplicativo móvel, converte-os para um formato padronizado e os publica em um tópico do Apache Kafka. O Kafka foi escolhido por ser uma plataforma de *streaming* distribuída altamente escalável, capaz de lidar com grandes volumes de dados e garantir entrega assíncrona e durável de mensagens — características essenciais para aplicações com múltiplos veículos conectados em tempo real.

Em seguida, o Node-RED, agora atuando como consumidor, consome os dados do Kafka e os armazena no TimescaleDB, um banco de dados relacional otimizado para séries temporais. O TimescaleDB permite armazenamento eficiente e consultas de alto desempenho sobre grandes volumes de dados históricos, o que é fundamental para o cálculo posterior de emissões de CO_2 e análises veiculares.

Essa arquitetura modular e orientada a eventos permite a coleta, o processamento e o armazenamento contínuos de dados veiculares com alta confiabilidade. Além de facilitar o monitoramento do desempenho do veículo e a análise das emissões, ela fornece a base técnica para posterior integração com os contratos inteligentes na blockchain.

3.3. API e Estimativa de Emissão

Ao final de cada trajeto, o usuário aciona um botão no aplicativo móvel que indica a finalização da viagem. Esse evento dispara uma requisição do tipo POST para o serviço de estimativa e consolidação, cuja API é exposta utilizando o framework FastAPI, escolhido por sua leveza, suporte a operações assíncronas e compatibilidade com aplicações baseadas em microserviços.

O serviço realiza a recuperação dos dados brutos do trajeto diretamente no banco TimescaleDB, filtrando as entradas associadas ao identificador do usuário e ao intervalo temporal da viagem. Com base nesses dados, a API executa uma série de operações de pré-processamento, que incluem a ordenação temporal dos registros, limpeza de dados inconsistentes e cálculo da distância total percorrida, com base na diferença entre as coordenadas geográficas capturadas ao longo do trajeto.

Em seguida, é aplicado um modelo de cálculo de emissão de CO_2 , descrito na próxima subseção, que estima a emissão total com base nos dados coletados pelos sensores veiculares, como o fluxo de massa de ar (MAF), razão ar-combustível (AFR) e tipo de combustível utilizado.

Os resultados consolidados são então estruturados em formato JSON, contendo informações como distância, tipo de combustível, tempo da viagem e emissão estimada. Esse *payload* é posteriormente encaminhado via requisição POST para o cliente responsável por intermediar a comunicação com a rede *blockchain*, permitindo o registro imutável da viagem por meio do contrato inteligente.

A API também incorpora verificações de consistência dos dados antes do envio à blockchain, como validação de faixas aceitáveis para variáveis sensíveis (por exemplo, distâncias negativas ou tempos nulos), reforçando a confiabilidade do sistema.

3.3.1. Cálculo das Emissões

A estimativa de emissões de CO_2 adotada nesta abordagem baseia-se em parâmetros obtidos diretamente dos sensores veiculares disponíveis via interface OBD-II, eliminando a necessidade de equipamentos externos ou estimativas indiretas. O modelo utiliza como base a massa de ar admitida no motor (*Mass Air Flow* - MAF) e a razão ar-combustível (*Air-Fuel Ratio* - AFR), ambos fornecidos em tempo real pelo sistema eletrônico do veículo. Também são considerados a densidade do combustível e o fator de emissão específico de CO_2 para cada tipo de combustível.

Com base nesses parâmetros, o volume de combustível queimado é calculado pela seguinte equação:

$$\text{Volume de Combustível} = \frac{\text{MAF}}{\text{AFR} \times \text{Densidade do Combustível}}$$

A partir desse volume, estima-se a emissão de CO_2 em gramas por segundo (g/s) com a seguinte expressão:

$$\text{Emissão de } CO_2 = \text{Volume de Combustível} \times \text{Fator de Emissão de } CO_2$$

Os valores de AFR adotados foram 14,7 para gasolina e 9,1 para etanol, conforme [Gois et al. 2023]. Os fatores de emissão utilizados foram obtidos a partir de bases técnicas para combustíveis comerciais padrão. A soma das emissões ao longo do tempo fornece o total emitido no trajeto analisado.

Essa metodologia apresenta boa acurácia e é amplamente utilizada em estudos que estimam emissões com base em dados embarcados, especialmente em contextos de análise de frotas ou diagnóstico ambiental veicular. No entanto, sua precisão depende da qualidade dos sensores embarcados e da correta identificação do tipo de combustível, sendo possível adaptar o modelo para outros combustíveis desde que se conheçam os parâmetros físico-químicos correspondentes.

Em implementações futuras, pretende-se incorporar mecanismos de ajuste adaptativo com base em *benchmarks* de laboratório ou dados empíricos complementares, visando reduzir incertezas associadas à variação entre veículos e condições de uso.

3.4. Cliente *Blockchain*

Para viabilizar a comunicação entre o sistema de *backend* e a rede *blockchain*, foi desenvolvido um cliente especializado, implementado com o framework Fastify (Node.js), que atua como camada intermediária entre a API principal e o contrato inteligente implantado na rede Hyperledger Besu. A implementação e o *deploy* do contrato foram realizados utilizando a ferramenta Hardhat, amplamente utilizada no ecossistema Ethereum para testes, migrações e interações programáticas com contratos desenvolvidos em Solidity.

Esse cliente blockchain é responsável por construir, assinar e enviar transações para a rede, utilizando chamadas ao nó RPC (Remote Procedure Call) exposto pela infraestrutura Besu. Para isso, a biblioteca Ethers.js foi empregada, facilitando a serialização de dados, assinatura de transações e escuta de eventos emitidos pelo contrato inteligente.

Duas rotas principais foram implementadas nesse cliente: (i) uma rota do tipo **POST** denominada `/store-on-chain`, responsável por receber os dados processados de uma viagem e invocar a função `storeTrip` do contrato inteligente; e (ii) uma rota **GET** que permite recuperar uma viagem registrada com base em seu identificador único. Após o envio da transação, o cliente aguarda sua confirmação na rede e monitora a emissão do evento `TripStored`, utilizado como mecanismo de confirmação para garantir que os dados foram de fato armazenados na *blockchain*.

Além das validações executadas no próprio contrato, o cliente realiza verificações adicionais para prevenir o envio de dados inconsistentes ou duplicados, como registros com campos nulos, identificadores inválidos ou assinaturas não autorizadas.

A assinatura das transações é realizada a partir de uma carteira controlada pelo backend, gerada por meio de uma chave privada configurada no ambiente da aplicação. Essa carteira possui permissões para registrar viagens na rede, garantindo que apenas agentes autorizados realizem a gravação de dados sensíveis. A chave privada é armazenada de forma segura e sua exposição é minimizada por meio de práticas de segregação de responsabilidades.

Por fim, o cliente blockchain está preparado para lidar com falhas de comunicação ou erros de execução de transações, utilizando mecanismos de retentativa e tratamento de exceções para garantir a robustez da operação. Essa camada desempenha papel central na confiabilidade da solução, atuando como elo entre os dados veiculares coletados e a imutabilidade garantida pela rede *blockchain*.

3.5. Rede Hyperledger Besu e Contrato Inteligente

A infraestrutura de *blockchain* adotada na proposta foi implementada com base na plataforma Hyperledger Besu, uma solução empresarial compatível com o padrão Ethereum, projetada para redes permissionadas. A rede foi configurada com o protocolo de consenso QBFT (Quorum Byzantine Fault Tolerance), que oferece tolerância a falhas bizantinas e garante segurança e disponibilidade mesmo na presença de nós maliciosos ou inoperantes. A escolha do QBFT, em detrimento de outros consensos como IBFT ou Raft, se deu por seu equilíbrio entre desempenho e robustez em ambientes controlados, mantendo baixa latência e alto nível de confiança entre os participantes.

Além disso, a rede foi configurada para operar em um modelo sem taxas de transação (*free gas*), característica viabilizada por tratar-se de um ambiente permissionado e de uso institucional. Essa configuração é especialmente vantajosa em cenários nos quais a monetização via taxas não é necessária, como projetos pilotos, plataformas de monitoramento público ou ambientes acadêmicos.

A arquitetura da rede, ilustrada na Figura 3, é composta por quatro nós validadores, responsáveis pelo processo de consenso, um nó replicador para garantir redundância e integridade dos dados, e um nó RPC (Remote Procedure Call) dedicado à comunicação com aplicações externas. Para facilitar a visualização dos dados registrados, foi integrado um explorador de blocos, que permite acompanhar em tempo real as transações e os contratos inteligentes implantados, reforçando a transparência do sistema.

A rede foi estruturada com foco na simplicidade e no uso eficiente de recursos computacionais, permitindo sua execução em ambientes de teste com infraestrutura redu-

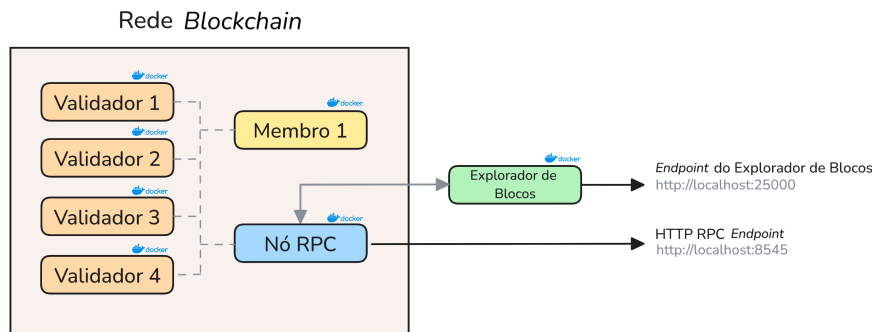


Figura 3. Arquitetura da rede Hyperledger Besu proposta.

zida. No entanto, devido à flexibilidade da plataforma Besu, a arquitetura é facilmente escalável para cenários de produção, podendo incorporar novos validadores, nós de leitura, ou organizações distintas em um consórcio.

Por fim, destaca-se que escolha da plataforma Hyperledger Besu, fundamenta-se na adequação aos requisitos iniciais do projeto, focados em um ambiente institucional controlado para validação da rastreabilidade e auditoria das emissões. Neste contexto, o controle de acesso granular sobre os participantes que podem validar transações e registrar dados, aliado à ausência de custos por transação, mostrou-se vantajoso para a implementação de um protótipo funcional e para testes de viabilidade. Apesar da adequação atual, reconhece-se que blockchains públicas, como Ethereum ou Polygon, podem oferecer vantagens relevantes para futuras extensões do sistema, especialmente no que diz respeito à transparência e integração com finanças descentralizadas.

3.5.1. Contrato Inteligente

O núcleo da rastreabilidade e da imutabilidade da solução reside em um contrato inteligente desenvolvido em Solidity, denominado `EmissionStorage`. Esse contrato é responsável por registrar, de forma verificável e inviolável, os dados de trajetos veiculares processados e enviados pelo backend autorizado.

A estrutura principal do contrato é a `Trip`, que armazena os seguintes dados: identificador único da viagem (gerado via *hash* dos campos principais), endereço do registrador (signatário da transação), identificador do motorista, coordenadas de origem e destino, número VIN (*Vehicle Identification Number*), tipo de combustível, distância percorrida, emissão estimada de CO_2 , horários de início e fim da viagem, e timestamp do registro na *blockchain*. Todos os campos são explicitamente validados quanto à consistência e presença.

A função `storeTrip` é responsável por inserir novos registros. Antes de persistir os dados, ela executa validações formais para evitar registros inválidos, como campos obrigatórios em branco, horários inconsistentes ou duplicidade de trajetos. A unicidade é assegurada por meio da geração de um identificador hash exclusivo para cada combinação de dados, que é utilizado como chave de armazenamento no *mapping* interno do contrato. Após a inserção bem-sucedida, o contrato emite um evento `TripStored`, contendo o

identificador da viagem, endereço do registrador e timestamp. Esses eventos são fundamentais para a auditoria externa e sincronização com sistemas off-chain, garantindo rastreabilidade com independência do estado interno do contrato.

A função `getTrip`, por sua vez, permite a consulta pública dos dados registrados, a partir do identificador da viagem. Essa função retorna todas as informações associadas àquela transação, permitindo reconstruir a jornada registrada de forma transparente. Em termos de controle de acesso, apenas endereços previamente autorizados — configurados no momento do `deploy` ou por meio de funções de controle de permissões (por exemplo, via modificador `onlyAuthorized`) — podem inserir registros. Essa abordagem evita que entidades externas interfiram no sistema, protegendo a integridade dos dados.

O contrato foi projetado com foco em simplicidade, segurança e baixo custo computacional, evitando operações dispendiosas e limitando o volume de dados armazenado on-chain. Em implementações futuras, pode-se incorporar estratégias complementares como *IPFS* ou *Merkle proofs* para armazenamento híbrido e verificação de dados em larga escala.

4. Resultados e Discussões

A validação da solução proposta foi conduzida por meio de experimentos projetados para avaliar a eficácia da arquitetura desenvolvida nas etapas de coleta, processamento, transmissão e registro de dados veiculares em uma rede *blockchain*. Os objetivos principais desta validação incluem: (i) verificar a correta comunicação e integração entre os diferentes módulos da arquitetura (coleta, processamento, armazenamento, cliente blockchain, rede blockchain); (ii) avaliar a integridade e a consistência dos dados registrados na blockchain em comparação com os dados originais coletados; (iii) analisar o desempenho da rede blockchain em termos de latência na confirmação das transações; e (iv) demonstrar a capacidade de auditoria das emissões armazenadas, consultando os dados diretamente na blockchain ou através do explorador de blocos. Espera-se que os resultados demonstrem que a arquitetura atende aos requisitos de confiabilidade, transparência e funcionalidade para o rastreamento de emissões veiculares. Para isso, os testes foram divididos em dois cenários complementares:

(i) Simulação controlada em ambiente computacional: nesse cenário, todos os módulos do sistema foram executados em um ambiente virtual, com dados de entrada gerados de forma sintética a partir de padrões conhecidos. O objetivo foi validar o funcionamento integrado dos componentes, incluindo a estimativa de emissões, a geração do hash de verificação, a escrita no contrato inteligente e a posterior recuperação dos dados. Métricas como tempo de resposta da API, tempo médio de transação na *blockchain*, e validação da integridade dos dados recuperados foram analisadas. O ambiente controlado permitiu aferir a consistência entre os dados processados localmente e os armazenados on-chain.

(ii) Execução prática com veículo real: este cenário envolveu a execução do sistema em um automóvel equipado com leitor OBD-II Bluetooth e o aplicativo móvel de coleta, em trajetos urbanos e rodoviários. O teste visou verificar o comportamento do sistema sob variabilidade real de dados — incluindo oscilações de rede, diferentes padrões de condução e ruído nos sensores. Foram avaliadas a robustez da comunicação

entre módulos (aplicativo-API-cliente blockchain), a precisão do modelo de emissão sob dados reais, e a capacidade da infraestrutura blockchain em registrar as informações sem perda ou inconsistência. O tempo médio de envio da requisição até a confirmação na blockchain foi monitorado, assim como a emissão dos eventos pelo contrato inteligente e a posterior consulta dos registros.

Em ambos os experimentos, os componentes do sistema foram conectados à mesma rede local (Wi-Fi), de modo a eliminar interferências externas relacionadas à conectividade e isolar as variáveis associadas à lógica do sistema. Essa abordagem permitiu concentrar a análise na validação funcional da arquitetura.

4.1. Validação em Ambiente Simulado

A primeira etapa de validação foi conduzida em um ambiente computacional controlado, utilizando o dispositivo Freematics OBD-II Emulator MK2. Esse equipamento simula com precisão uma interface OBD-II funcional, respondendo a requisições típicas de um veículo real. A simulação permitiu a coleta de parâmetros veiculares em tempo real pelo aplicativo móvel, como se este estivesse conectado a um automóvel em movimento.

Durante os testes, o aplicativo estabeleceu comunicação via *Bluetooth* com o scanner OBD-II, e os dados coletados foram encaminhados à API REST, que os processou, estimou as emissões e os enviou ao cliente *blockchain*. O cliente, por sua vez, transmitiu os dados ao contrato inteligente, que os registrou na rede Hyperledger Besu. A integridade dos registros foi verificada por meio da função `getTrip()`, que retornou corretamente os dados associados à viagem simulada, incluindo distância, combustível, coordenadas e estimativa de emissões. Além disso, após o armazenamento, o contrato emitiu o evento `TripStored`, contendo o identificador único da viagem e o endereço da carteira responsável, permitindo a auditoria completa do registro.

Durante esse experimento, o tempo médio entre o envio dos dados pela API e a confirmação do registro na *blockchain* foi de aproximadamente 4.35 segundos, e nenhuma perda de dados foi observada. As operações de escrita e leitura na rede ocorreram sem falhas, e os eventos emitidos foram consistentemente detectados pelo cliente, indicando que a arquitetura proposta é funcional, rastreável e auditável sob condições ideais de operação.

Esse cenário permitiu validar não apenas a comunicação entre os módulos, mas também a fidelidade dos dados armazenados na *blockchain* e a robustez da lógica de verificação implementada no contrato inteligente.

4.2. Validação em Cenário Real

Após a validação em ambiente simulado, a infraestrutura foi testada em um cenário prático, utilizando um veículo real equipado com um leitor OBD-II conectado via *Bluetooth* ao aplicativo móvel. Durante um trajeto urbano de curta duração (Figura 4), o sistema coletou continuamente dados dos sensores veiculares, como velocidade, rotação, temperatura e fluxo de massa de ar, permitindo a estimativa das emissões de CO_2 associadas à viagem.

Ao final do trajeto, o condutor acionou a finalização da viagem pelo aplicativo, disparando o envio dos dados para a API intermediária. Esta processou as informações,

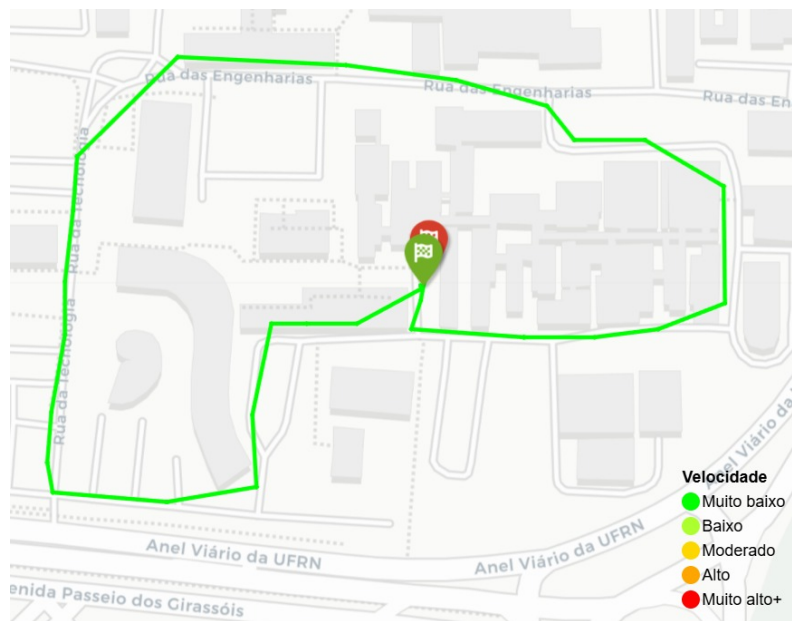


Figura 4. Visualização da rota utilizada para prova de conceito.

estimou as emissões e invocou a função de registro no contrato inteligente, por meio do cliente *blockchain*. A transação correspondente foi processada com sucesso pela rede Hyperledger Besu e registrada no bloco 1514, como ilustrado na Figura 5. O hash da transação, o endereço da carteira emissora e o consumo de *gas* foram devidamente armazenados, assegurando a rastreabilidade e a auditabilidade do evento.

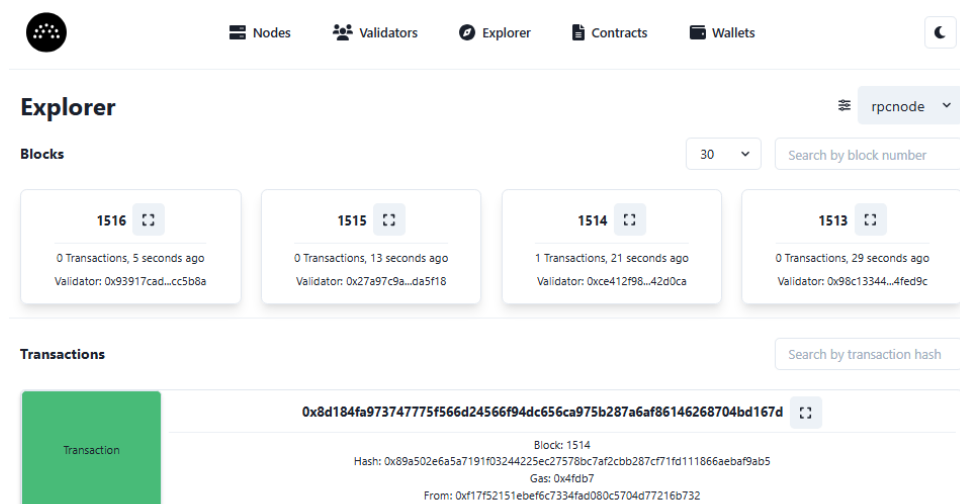


Figura 5. Transação registrada na rede Hyperledger Besu.

Posteriormente, os dados foram consultados por meio da função `getTrip()`, e os campos armazenados foram validados quanto à integridade e consistência. A Figura 6 mostra parte do payload recuperado, contendo informações como o identificador da viagem, distância percorrida, tipo de combustível, horários de início e fim, coordenadas geográficas, número VIN, e a estimativa de emissão, que foi calculada em 154g de CO_2 .

para o trajeto analisado.

```
{
  "tripOwner": "0xf17f52151EbEF6C7334FAD080c5704D77216b732",
  "driverId": "12",
  "startLocation": "(-5.8428215, -35.1970728)",
  "endLocation": "(-5.8427682, -35.1970496)",
  "carVin": "93...92",
  "fuelType": "Gasolina",
  "distance": "912",
  "emissions": "154",
  "startTime": "1739559618",
  "endTime": "1739559821",
  "timestamp": "1739559830"
}
```

Figura 6. Dados recuperados da transação na rede *blockchain*.

O tempo médio entre o envio da requisição pela API e a confirmação do registro na *blockchain* foi de aproximadamente 4.40 segundos, indicando boa adequação e responsividade da infraestrutura mesmo em ambiente móvel. O cliente *blockchain* detectou corretamente a emissão do evento *TripStored*, e os dados foram validados sem inconsistências.

Além da validação funcional, esse teste evidenciou aspectos operacionais importantes, como a necessidade de conectividade estável entre o aplicativo e a API, e a sensibilidade da transmissão Bluetooth em ambientes urbanos. A infraestrutura comportou-se de maneira robusta, e todos os componentes mantiveram comunicação contínua sem perdas durante o experimento. Esses resultados reforçam a viabilidade da solução para cenários reais, demonstrando que o sistema é capaz de registrar emissões de forma segura, verificável e auditável, com potencial de aplicação em programas de monitoramento ambiental veicular ou esquemas de compensação de carbono.

5. Conclusão

Este trabalho apresentou o desenvolvimento e validação de uma arquitetura modular baseada em *blockchain*, voltada à rastreabilidade e auditabilidade de emissões de CO_2 em veículos conectados. Através da integração entre sensores OBD-II, processamento em tempo real e um contrato inteligente implantado em rede permissionada Hyperledger Besu, foi possível registrar trajetos veiculares e estimativas de emissão de forma transparente, imutável e acessível para auditoria.

A abordagem demonstrou-se tecnicamente viável, tanto em ambiente simulado quanto em cenários reais, garantindo integridade na coleta, transmissão e armazenamento dos dados. A solução validou sua capacidade de operar com precisão e robustez em todas as etapas do fluxo, tornando-se uma alternativa promissora para sistemas de monitoramento ambiental e geração de créditos de carbono.

Como perspectivas futuras, propõe-se a avaliação da arquitetura em contextos multi-organizacionais, com múltiplas carteiras interagindo simultaneamente com o contrato inteligente. Pretende-se também investigar a integração com mecanismos de verificação externa e camadas de privacidade baseadas em *zk-SNARKs* ou *rollups*, visando ampliar a escalabilidade e garantir confidencialidade dos dados sensíveis. Outra

frente será a adaptação do sistema a plataformas públicas, como Ethereum Goerli, para estudos comparativos com ambientes não-permissionados.

Acredita-se que a proposta apresentada contribui significativamente para o avanço de soluções digitais sustentáveis, oferecendo uma infraestrutura confiável para o monitoramento descentralizado de emissões, alinhada aos desafios atuais da mobilidade inteligente e da governança ambiental.

Referências

- Alalwany, E. and Mahgoub, I. (2024). Security and trust management in the internet of vehicles (ioV): Challenges and machine learning solutions. *Sensors*, 24(2).
- Azevedo, M., Medeiros, T., Medeiros, M., Viegas, C. M., Silva, M., Silva, I., and Costa, D. G. (2023). A framework to support collection, processing and analysis of smart vehicles data. In *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE.
- Banerjee, S., Das, D., Chatterjee, P., Blakely, B., and Ghosh, U. (2024). A blockchain-enabled sustainable safety management framework for connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):5271–5281.
- Biswas, M., Das, D., Banerjee, S., Mukherjee, A., AL-Numay, W., Biswas, U., and Zhang, Y. (2023). Blockchain-enabled communication framework for secure and trustworthy internet of vehicles. *Sustainability*, 15(12).
- Gois, T., Andrade, M., Medeiros, T., Silva, M., and Silva, I. (2023). Metodologia baseada em tinymml para estimar as emissões de co2: Uma análise comparativa entre abastecimentos de veículo com etanol e gasolina. *Simpósio Brasileiro de Automação Inteligente - SBAI*, 1(2).
- Gu, Y., Jiang, L., and Dai, J. (2024). Using blockchain and smart contracts to combat greenwashing in environmental disclosures. *SSRN*.
- Li, Z., Xu, X., Bai, Q., Chen, C., Wang, H., and Xia, P. (2023). Implications of information sharing on blockchain adoption in reducing carbon emissions: A mean–variance analysis. *Transportation Research Part E: Logistics and Transportation Review*, 178:103254.
- Ling, X. and Wang, H. (2024). Green behavior strategies in the green credit market: Analysis of the impacts of enterprises’ greenwashing and blockchain technology. *Sustainability*, 16(11).
- Liu, X., Li, L., Li, X., and Qi, Y. (2024). Lcaa: Lightweight conditional anonymous authentication scheme for vehicular carbon emissions environment. *IEEE Internet of Things Journal*, 11(19):32025–32040.
- Lu, Y., Li, Y., Tang, X., Cai, B., Wang, H., Liu, L., Wan, S., and Yu, K. (2022). Stricts: A blockchain-enabled smart emission cap restrictive and carbon permit trading system. *Applied Energy*, 313:118787.
- Yang, T., Ma, C., and Mi, X. (2024). The transformative potential of blockchain technology in developing green supply chain: An evolutionary perspective on complex networks. *Computers Industrial Engineering*, 197:110548.