ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

EDUARDA CRISTINA PISSOLATTO

EDNA: UMA ARQUITETURA DE MARKETPLACE DESCENTRALIZADO PARA O SETOR AUTOMOTIVO

Porto Alegre
2025

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

EDNA: UMA ARQUITETURA DE MARKETPLACE DESCENTRALIZADO PARA O SETOR AUTOMOTIVO

EDUARDA CRISTINA PISSOLATTO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestra em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Fabiano Passuelo Hessel

Porto Alegre
2025

Ficha Catalográfica

P678e Pissolatto, Eduarda Cristina

eDNA : Uma Arquitetura de Marketplace Descentralizado para o Setor Automotivo / Eduarda Cristina Pissolatto. – 2025.
76 p.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Fabiano Passuelo Hessel.

1. Blockchain. 2. Indústria Automotiva. 3. IoT. 4. IPFS. 5. Polygon. I. Hessel, Fabiano Passuelo. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

EDUARDA CRISTINA PISSOLATTO

EDNA: UMA ARQUITETURA DE MARKETPLACE DESCENTRALIZADO PARA O SETOR AUTOMOTIVO

Dissertação apresentada como requisito parcial para obtenção do grau de Mestra em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado(a) em 26 de fevereiro de 2025.

BANCA EXAMINADORA:

Prof. Dr. Jorge Luis Victoria Barbosa (Unisinos)

Prof. Dr. Avelino Francisco Zorzo (PPGCC/PUCRS)

Prof. Dr. Fabiano Passuelo Hessel (PPGCC/PUCRS - Orientador)

DEDICATÓRIA

Às mulheres na computação, que, com coragem e determinação, abrem caminhos em um campo ainda marcado por desafios. Que esta dissertação seja um pequeno testemunho do impacto que podemos causar e uma inspiração para que mais mulheres ocupem, transformem e brilhem nesta área.

“Quanto mais estudo, mais sinto que minha mente nisso é insaciável.”
(Ada Lovelace)

AGRADECIMENTOS

Agradeço, com todo o meu carinho, ao meu namorado e companheiro de vida, Victor, por ser meu alicerce nos momentos difíceis, por acreditar em mim quando eu mesma duvidei e por toda a motivação e apoio incondicional ao longo desta jornada.

Aos meus pais, cuja dedicação e amor sempre foram a base de tudo que conquistei. Vocês nunca mediram esforços para que eu pudesse alcançar meus objetivos, e por isso, serei eternamente grata.

À minha família, mesmo à distância, pelo incentivo constante e pelo apoio em todos os momentos. Sua presença em meu coração fez toda a diferença durante essa caminhada.

Ao meu orientador, professor Fabiano Hessel, pela paciência e orientação em um percurso repleto de desafios. Seu conhecimento, generosidade e confiança foram fundamentais para a realização deste trabalho.

Por fim, agradeço a todos aqueles que, de alguma forma, contribuíram para minha pesquisa, seja com palavras de encorajamento, ideias valiosas ou simplesmente estando ao meu lado. Muito obrigado!

EDNA: UMA ARQUITETURA DE MARKETPLACE DESCENTRALIZADO PARA O SETOR AUTOMOTIVO

RESUMO

No contexto da transformação digital impulsionada pela Internet das Coisas (*IoT*), vários setores da economia, incluindo o automotivo, estão passando por renovações. Com o avanço da *Web 3.0*, soluções inovadoras em gestão de dados, segurança da informação e propriedade de dados estão surgindo. Devido à adoção de conceitos de *IoT* no setor automotivo, uma grande quantidade de dados é coletada e transmitida aos fabricantes sem o consentimento dos proprietários. Este trabalho propõe a arquitetura *eletronic* DNA (eDNA), baseada em *blockchain* e *IoT*, para criar um mercado descentralizado de dados automotivos. O eDNA utiliza a rede Polygon, *IPFS/IPNS* e contratos inteligentes para armazenar, proteger e compartilhar dados de forma segura e transparente, promovendo a inclusão dos proprietários no mercado de dados automotivos. Na implementação do eDNA, utilizando dados sintéticos, as taxas de transação permaneceram consistentes e acessíveis, provando a viabilidade da arquitetura. Além disso, o resultado dos testes validam a arquitetura *eDNA* como uma solução robusta e escalável para o futuro da transformação digital no setor automotivo, onde a propriedade e transparência dos dados são essenciais.

Palavras-Chave: Internet das Coisas, *blockchain*, indústria automotiva, *IPFS*, *IPNS*, *IDD*, *Polygon*.

EDNA: A DECENTRALIZED MARKETPLACE ARCHITECTURE FOR THE AUTOMOTIVE SECTOR

ABSTRACT

In the context of the digital transformation driven by the Internet of Things (*IoT*), various sectors of the economy, including the automotive industry, are undergoing renewals. With the advancement of *Web 3.0*, innovative solutions in data management, information security, and data ownership are emerging. Due to the adoption of *IoT* concepts in the automotive sector, a large volume of data is collected and transmitted to manufacturers without the owners' consent. This work proposes the eletronic DNA (eDNA) architecture, based on *blockchain* and *IoT*, to create a decentralized automotive data marketplace. eDNA leverages the Polygon network, *IPFS/IPNS*, and smart contracts to securely and transparently store, protect, and share data, promoting the inclusion of owners in the automotive data market. In the implementation of eDNA using synthetic data, transaction fees remained consistent and affordable, proving the viability of the architecture. Moreover, the test results validate the eDNA architecture as a robust and scalable solution for the future of digital transformation in the automotive sector, where data ownership and transparency are essential.

Keywords: Internet of Things, *blockchain*, automotive industry, *IPFS*, *IPNS*, *IDD*, *Polygon*.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

EDNA – *eletronic DNA*

IOT – Internet das Coisas (do inglês, *Internet of Things*)

IPFS – Sistema de Arquivos Interplanetário (do inglês, *InterPlanetary File System*)

IPNS – Sistema de Nomes Interplanetário (do inglês, *InterPlanetary Name System*)

DNFT – Tokens Não Fungíveis Dinâmicos (do inglês, *dynamic Non-Fungible Tokens*)

DAO – Organizações Autônomas Descentralizadas (do inglês, *Decentralized Autonomous Organization*)

C2D – *Compute-to-Data*

ZKP – Prova de Conhecimento Zero (do inglês, *Zero-Knowledge Proof*)

VC – Credenciais Verificáveis (do inglês, *Verifiable Credentials*)

DID – Identidade Descentralizada (do inglês, *Decentralized Identifier*)

GDPR – Regulamento Geral de Proteção de Dados (do inglês, *General Data Protection Regulation*)

LGPD – Lei Geral de Proteção de Dados

EVM – Máquina Virtual Ethereum (do inglês, *Ethereum Virtual Machine*)

UBI – Seguro Baseado em Uso (do inglês, *Usage-Based Insurance*)

SDK – Kit de Desenvolvimento de Software (do inglês, *Software Development Kit*)

DAPP – Aplicações Descentralizadas (do inglês, *Decentralized Applications*)

CLI – Interface de Linha de Comando (do inglês, *Command Line Interface*)

POS – Prova de Participação (do inglês, *Proof of Stake*)

API – Interface de Programação de Aplicação (do inglês, *Application Programming Interface*)

NFT – Token Não Fungível (do inglês, *Non-Fungible Token*)

MAP – Sensor de Pressão Absoluta do Coletor de Admissão (do inglês, *Manifold Absolute Pressure*)

DAG – Grafo Acíclico Dirigido (do inglês, *Directed Acyclic Graph*)

CID – Identificador de Conteúdo (do inglês, *Content Identifier*)

POL – Moeda nativa da *Polygon*, substituindo *MATIC*

UBI – Seguros Baseados no Uso (do inglês, *Usage-Based Insurance*)

SUMÁRIO

1. INTRODUÇÃO

A transformação digital impulsionada pela Internet das Coisas (do inglês, *Internet of Things (IoT)*) está remodelando vários setores, incluindo o automotivo. Com o avanço da *Web 3.0*, onde a propriedade dos dados pertence aos geradores, surgem soluções inovadoras para gestão, segurança e uso de dados.

Nas últimas décadas, a adoção da *IoT* no setor automotivo resultou na coleta maciça de dados, que são transmitidos aos fabricantes sem o consentimento dos proprietários. Esses dados podem incluir informações como estilo de condução, manutenção e uso do cinto de segurança, possibilitando traçar o perfil de condução e uso do veículo. A partir da análise destes dados, é possível, também, oferecer serviços personalizados como por exemplo a oferta de seguros veiculares.

No entanto, os proprietários dos dados raramente compartilham dos lucros gerados por suas informações, destacando a necessidade de uma inclusão mais significativa nesse mercado. A adoção de tecnologias emergentes, como a *blockchain*, oferece soluções promissoras para enfrentar esses desafios. A *blockchain* assegura uma infraestrutura segura e transparente para rastrear e gerenciar o ciclo de vida dos veículos, garantindo a integridade dos dados e a confiabilidade dos registros. A combinação da *blockchain* com a *IoT* propõe uma abordagem descentralizada e inovadora, superando as limitações dos sistemas atuais, como vulnerabilidades a ataques cibernéticos e falta de privacidade e transparência. Essas tecnologias promovem maior segurança, privacidade e escalabilidade no mercado automotivo, melhorando sua eficiência geral.

Este trabalho visa abordar os desafios existentes na gestão de dados e segurança da informação no contexto automotivo e explorar o potencial das tecnologias emergentes. O objetivo é propor o DNA eletrônico do veículo (*eDNA*), sendo uma arquitetura de mercado de dados para o setor automotivo, baseada na tecnologia *blockchain* em conjunto com a *IoT*. Essa abordagem busca superar as limitações dos sistemas atuais e oferecer um modelo descentralizado, seguro e escalável para a gestão de identidades digitais e dados, com uma aplicação prática e impactante no setor automotivo. A arquitetura *eDNA* emprega a tecnologia *blockchain*, juntamente com Sistema de Arquivos Interplanetário (do inglês, *InterPlanetary File System (IPFS)*) e Sistema de Nomes Interplanetário (do inglês, *InterPlanetary Name System (IPNS)*) para armazenar dados de telemetria, e Tokens Não Fungíveis Dinâmicos (do inglês, *dynamic Non-Fungible Tokens* (*dNFT*)) para encapsular e proteger dados esporádicos sensíveis.

A arquitetura permite o armazenamento descentralizado e a gestão segura de identidades digitais através do *Privado ID* (anteriormente *Polygon ID*), facilitando a autenticação e verificação de dados no ecossistema. O *eDNA* possibilita a monetização de

dados na forma de modelos de dados enriquecidos, oferecendo, por exemplo, a possibilidade de seguros parametrizados.

A estrutura deste trabalho está organizada para apresentar o tema de forma lógica e sequencial. Primeiramente, são abordados os conceitos fundamentais e as tecnologias relacionadas, seguidos pela revisão de trabalhos relacionados. Em seguida, é detalhada a arquitetura proposta e sua implementação, culminando em uma análise dos resultados e considerações finais.

1.1 Problema de Pesquisa

Com a urbanização acelerada, mudanças no comportamento de consumo e a crescente popularidade de modelos de negócios como o aluguel de carros, torna-se imperativo adotar soluções que garantam a segurança, a privacidade e a eficiência na gestão de identidades e dados. Nesse contexto, o setor automotivo emerge como um campo fértil para a aplicação de novas tecnologias de gerenciamento de dados. Assim, este trabalho propõe uma arquitetura pensada para a coleta e comercialização de dados no setor automotivo, tendo como problemas de pesquisa:

P.1. Como o veículo deve ser monitorado através de sensores *IoT* no ambiente cotidiano;

P.2. Como ter um registro das ocorrências relacionadas ao veículo de forma segura, transparente e inviolável;

P.3. Como estes registros devem ser inseridos em modelos de dados enriquecidos, posteriormente disponíveis para a venda a interessados externos, bem como para o próprio proprietário do veículo;

P.4. A centralização tradicional dos sistemas tem se mostrado inadequada diante das demandas de segurança, escalabilidade e disponibilidade persistentes no ambiente interconectado. As limitações desses sistemas centralizados, como pontos únicos de falha, vulnerabilidades a ataques e dificuldades em escalar proporcionalmente ao crescimento do número de dispositivos, evidenciam a necessidade de novas abordagens;

A partir desses problemas, o presente trabalho investiga as seguintes hipóteses:

H.1. A implementação de sensores *IoT* em veículos melhora significativamente a coleta de dados em tempo real, aumentando a eficiência na gestão de uso dos veículos.

H.2. Tecnologias de *blockchain* podem garantir um registro de ocorrências relacionadas a veículos que seja seguro, transparente e inviolável, minimizando fraudes e disputas.

H.3. A integração de dados de veículos coletados em modelos enriquecidos aumenta o valor comercial desses dados. Existe um modelo de negócios que permite o compartilhamento destes dados com entidades terceiras sem comprometer a privacidade do proprietário.

H.4. Sistemas descentralizados oferecem melhor segurança, escalabilidade e disponibilidade para a gestão de dados em veículos, superando as limitações dos sistemas centralizados.

Para validar essas hipóteses, o presente trabalho investigou o seguinte conjunto de questões de pesquisa:

QP.1. De que maneira os sensores *IoT* podem ser utilizados no monitoramento cotidiano de veículos de forma eficaz?

QP.2. Como a tecnologia de *blockchain* pode ser aplicada para criar um sistema de registros veiculares que seja ao mesmo tempo seguro e acessível para as partes interessadas?

QP.3. De que forma os dados coletados de veículos podem ser enriquecidos e preparados para venda, garantindo privacidade e agregando valor para o proprietário e para terceiros?

QP.4. Quais são as principais vantagens e desafios na adoção de sistemas descentralizados?

1.2 Objetivos

Diante das considerações, o objetivo deste trabalho é propor o *eDNA*, uma arquitetura de *data marketplace* para o setor automotivo, baseada em tecnologia *blockchain* em conjunto com o *IoT*. Esta abordagem busca superar as limitações dos sistemas atuais e oferecer um modelo descentralizado, seguro e escalável para a gestão de identidades digitais e dados no vasto ecossistema da *IoT*, com uma aplicação prática e impactante no setor automotivo. Os objetivos específicos consistem em:

1. Desenvolver uma arquitetura baseada em *blockchain*, em conjunto com *IoT*, aplicada ao setor automotivo.
2. Explorar a utilização de Sistema de Arquivos Interplanetário (do inglês, *InterPlanetary File System (IPFS)*), Sistema de Nomes Interplanetário (do inglês, *InterPlanetary Name System (IPNS)*), Identidade Descentralizada (do inglês, *Decentralized Identifiers* (DID)), Token Não Fungível Dinâmico (do inglês, *dynamic Non-Fungible Tokens (dNFT)*) e um modelo de *marketplace*.

3. Explorar a obtenção de modelos de dados enriquecidos, a partir de dados armazenados no *IPFS/IPNS*, para comercialização através do *marketplace*.
4. Validar a arquitetura proposta, simulando cenários de casos reais.
5. Documentar os resultados obtidos e publicá-los em artigo.

1.3 Estrutura do Documento

A estrutura deste trabalho está organizada em sete capítulos, cada um com um propósito específico.

O Capítulo 1 apresenta o contexto geral do tema abordado, discutindo a relevância da transformação digital no setor automotivo, impulsionada pela *IoT* e pela *Web3*. Além disso, são introduzidos o problema de pesquisa, os objetivos gerais e específicos, bem como as hipóteses que orientaram a elaboração da proposta. Este capítulo também destaca as justificativas para a escolha do tema, evidenciando sua importância acadêmica e prática.

No Capítulo 2, são descritos os fundamentos teóricos e técnicos necessários para a compreensão do trabalho. São abordados temas como a *IoT*, a *Web3*, a tecnologia *blockchain* e sistemas de arquivos distribuídos, incluindo o *IPFS* e o *IPNS*. Esses conceitos fornecem o embasamento necessário para o entendimento da proposta apresentada nos capítulos subsequentes, contextualizando quanto às ferramentas e tecnologias utilizadas.

O Capítulo 3 realiza uma análise detalhada da literatura existente, focando em soluções que combinam *blockchain* e *IoT*, com ênfase particular no setor automotivo. Essa revisão inclui uma descrição das contribuições já realizadas, bem como a identificação de lacunas e limitações nos trabalhos previamente publicados. Essa análise estabelece as bases para a formulação da proposta, evidenciando sua originalidade e potencial impacto.

O Capítulo 4 é o núcleo deste trabalho e apresenta, em detalhes, a arquitetura denominada *eDNA*. Inicialmente, são descritas as camadas que compõem a solução, incluindo sensores, blockchain e aplicações descentralizadas. São explorados também os principais componentes técnicos, como a utilização de *DID*, *dNFTs* e o sistema de armazenamento *IPFS/IPNS*. Este capítulo descreve ainda o fluxo de dados e a integração dos diversos elementos tecnológicos, demonstrando como eles se conectam para oferecer uma solução inovadora e eficiente.

No Capítulo 5, o trabalho se aprofunda na aplicação do conceito de *Compute-to-Data (C2D)*, um modelo que permite processar dados sensíveis sem comprometer sua privacidade. São descritas as vantagens dessa abordagem no contexto do mercado de dados automotivos, destacando como ela promove a monetização segura de informações, respeitando regulamentações de privacidade. Além disso, o capítulo explora a utilização

de identificadores descentralizados e provas de conhecimento zero (*ZKP*) para reforçar a segurança e a privacidade dos dados. Este capítulo detalha como esses elementos são integrados à arquitetura para autenticar e verificar identidades de forma eficiente, garantindo que informações sensíveis sejam protegidas contra acessos indevidos.

O Capítulo 6 detalha os procedimentos de validação e avaliação de desempenho da arquitetura *eDNA* proposta, abordando a aplicabilidade e eficácia da solução no contexto do setor automotivo. A seção inicial discute o *dataset* utilizado, seguido pela descrição da implementação da arquitetura e uma análise detalhada dos custos de transação relacionados à implementação da solução. O capítulo também descreve os testes de desempenho, com o objetivo de validar a viabilidade técnica da arquitetura.

Por fim, o Capítulo 7 conclui o trabalho, apresentando uma síntese dos resultados alcançados e discutindo as contribuições do estudo para a área. São analisadas as limitações da solução proposta e sugeridas possíveis direções para pesquisas futuras, incentivando o avanço contínuo do uso de *blockchain* e *IoT* no setor automotivo

2. CONCEITOS BÁSICOS

Este capítulo apresenta conceitos básicos técnicos das ferramentas utilizadas neste trabalho. A Seção 2.1 apresenta o conceito de *Web3*; a Seção 2.2 traz aspectos relacionados a *blockchain*; e a Seção 2.3 introduz aos conceitos relacionados a Sistema de Arquivos Distribuídos, referindo-se a *IPFS* e *IPNS*.

2.1 *Web3*

Web3 representa uma nova fase na evolução da internet. Esta era é marcada por uma forte inclinação para descentralização, transparência e participação do usuário. Enquanto as versões anteriores da *web* eram dominadas por empresas e estruturas centralizadas, a *Web3* muda o foco para uma rede distribuída onde os usuários têm mais controle e propriedade sobre seus dados e interações.

Um elemento-chave da *Web3* é a tecnologia *blockchain*, que possibilita essa descentralização. A *blockchain* não é apenas uma tecnologia para criptomoedas, mas um pilar fundamental para a criação de um ecossistema digital mais democrático e equitativo. Isso inclui a possibilidade de realizar transações de forma segura e transparente, sem a necessidade de intermediários [33].

As Organizações Autônomas Descentralizadas (do inglês, *Decentralized Autonomous Organization* (DAO)) são outra inovação importante. Elas representam uma nova forma de governança e operação organizacional, funcionando com base em regras pré-definidas e executadas automaticamente por meio de contratos inteligentes. Isso abre portas para uma gestão mais democrática e transparente de projetos e organizações, refletindo o espírito colaborativo e participativo da *Web3*. Essas transformações sugerem um futuro em que a internet será mais um espaço de colaboração equitativa e propriedade compartilhada, contrastando com as versões anteriores da *web*, onde o controle e a propriedade estavam nas mãos de poucos. *Web3*, portanto, representa não apenas uma mudança tecnológica, mas também uma mudança cultural e social no uso e na governança da internet.

A Figura 2.1 apresenta um comparativo entre o modelo atual de internet, conhecido como *Web2*, e o emergente modelo *Web3*. Na figura, é possível identificar diferenças cruciais em termos de privacidade, controle de dados, acesso e liberdade de escolha dos usuários. Enquanto a *Web2* é caracterizada por um controle centralizado, onde grandes corporações detêm a maior parte dos dados e da infraestrutura, a *Web3* propõe um sistema descentralizado que enfatiza a propriedade e o controle dos dados pelos próprios usuários. Através de tecnologias como *blockchain* e contratos inteligentes, a *Web3* visa

proporcionar um ambiente mais seguro e transparente, onde os usuários têm maior autonomia e responsabilidade sobre suas informações e interações na rede. A Figura 2.1 destaca ainda como a *Web3* pode potencialmente transformar a experiência *online*, oferecendo uma maior resistência à censura e promovendo uma economia digital mais inclusiva e equitativa.

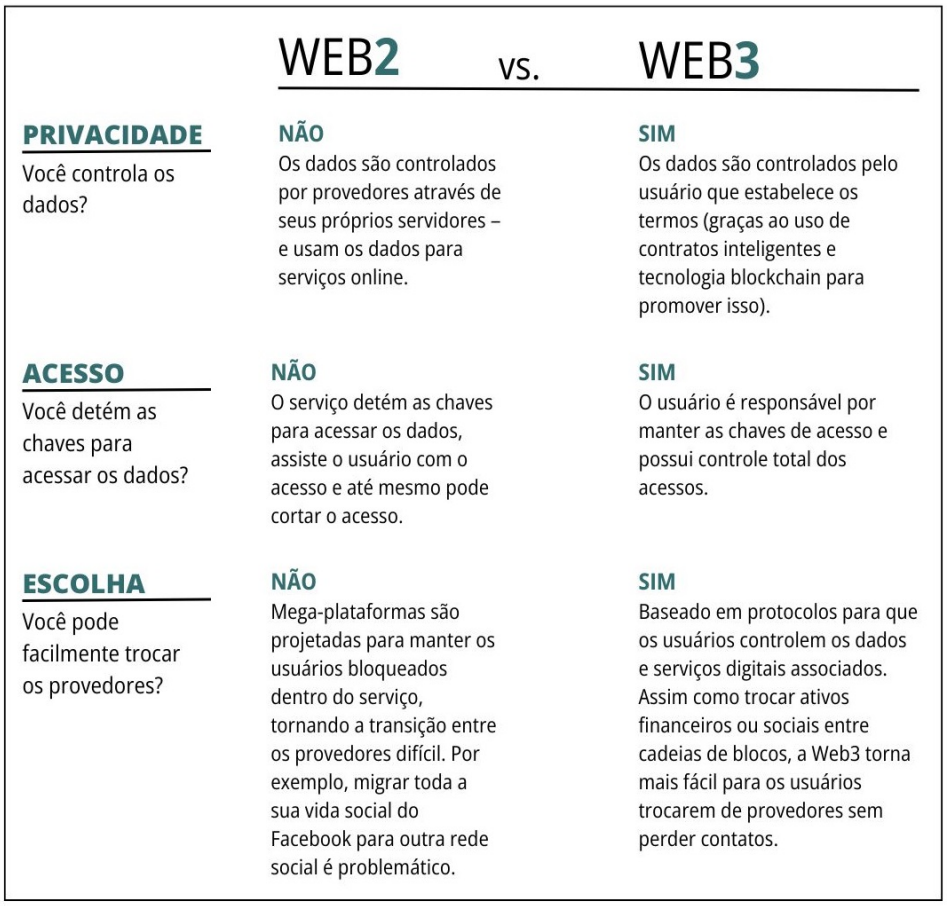

Figura 2.1 – Comparativo *Web2* vs. *Web3* [26]

2.2 *Blockchain*

Blockchain é uma tecnologia que reformula a maneira como os dados são armazenados e gerenciados. É composta por uma série de blocos conectados que registram transações de maneira descentralizada [50]. Esta estrutura oferece características como persistência, anonimato e auditabilidade. [19] salienta a importância da descentralização na blockchain, que elimina intermediários e reduz custos associados às transações. Além disso, a natureza descentralizada da *blockchain* proporciona um nível de segurança e transparência sem precedentes. Isso é crucial para proteger contra fraudes e para estabelecer um sistema confiável de transações. A descentralização também significa que os

dados na *blockchain* não são controlados por uma única entidade, o que é essencial para a integridade e a confiança no sistema.

2.2.1 Estrutura da *Blockchain*

A estrutura da *blockchain* é um dos seus aspectos mais notáveis. Cada bloco na cadeia contém um *hash* criptográfico do bloco anterior, além de um carimbo de data/hora e os dados da transação [50]. Isso cria um registro histórico inalterável e rastreável de todas as transações. Essa configuração garante a imutabilidade dos dados, um aspecto crucial para a segurança da *blockchain* [19].

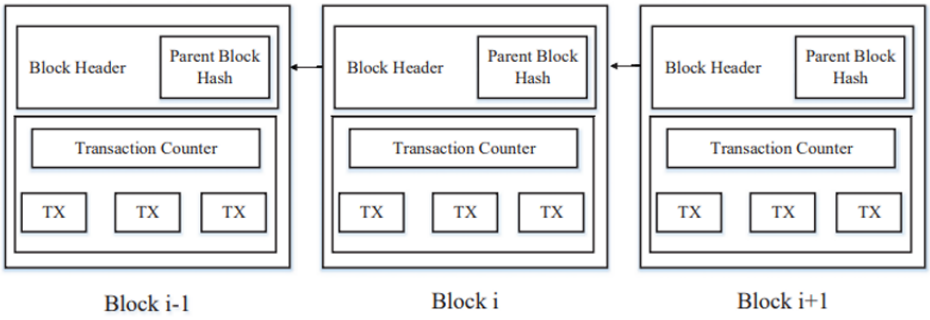

Figura 2.2 – Exemplo de *blockchain*, uma sequência contínua de blocos [50]

2.2.2 Tipos de *Blockchain*

Os tipos de *blockchain* são categorizados, principalmente em *blockchains* públicas e privadas, cada uma com suas características e aplicações específicas.

As *blockchains* públicas, são abertas a todos, permitindo que qualquer pessoa participe e verifique as transações [50]. Esta transparência e acessibilidade são ideais para criptomoedas, onde a segurança e a imutabilidade são cruciais.

As *blockchains* privadas, por outro lado, são restritas a um grupo específico de usuários. São frequentemente adotadas em ambientes corporativos onde a privacidade e o controle centralizado são essenciais. Essas *blockchains* oferecem maior segurança em termos de acesso e manipulação dos dados, adequando-se a casos de uso onde a confidencialidade é prioritária [19].

A escolha entre *blockchain* pública e privada depende das necessidades específicas do caso de uso. Enquanto as *blockchains* públicas são preferidas para aplicações que requerem maior transparência e imutabilidade, as privadas são escolhidas por organizações que necessitam de maior controle sobre seus dados e operações.

2.2.3 Identidade Descentralizada

A *DID* na *blockchain* é uma inovação significativa, proporcionando controle e segurança para os usuários. Esse sistema permite que os usuários gerenciem suas próprias identidades digitais sem depender de uma autoridade central [12]. Esta abordagem contrasta fortemente com os sistemas de identidade tradicionais e é particularmente benéfica no contexto da *IoT*, onde a segurança e a privacidade dos dados são cruciais.

A informação de identidade de um indivíduo ou entidade é armazenada, gerenciada e verificada de forma distribuída em uma rede *blockchain*, em vez de depender de uma autoridade central ou um intermediário. Isso significa que a informação de identidade não é controlada por uma única entidade, mas é mantida de forma compartilhada e confiável em toda a rede. A descentralização da identidade também oferece vantagens em termos de interoperabilidade e eficiência. Ao permitir que os usuários compartilhem suas informações de forma segura e direta, elimina-se a necessidade de intermediários, reduzindo assim a vulnerabilidade a ataques cibernéticos e falhas de segurança. Isso é particularmente relevante em aplicações que exigem confiança e verificação de identidade, como transações financeiras e serviços governamentais.

Como exemplo, podemos citar a implementação de sistemas de *DID* em países como a Estônia, permitindo aos cidadãos acessar serviços governamentais online, assinar documentos eletronicamente e iniciar negócios digitalmente. O sistema é conhecido como *e-Residency* [11]. A *blockchain* garante a segurança e a autenticidade das identidades digitais. Além disso, as *DIDs* podem ser utilizadas para sistemas de votação *online*, oferecendo uma solução segura e transparente para eleições, assegurando que os votos sejam inalteráveis e anonimamente registrados. Isso não apenas aumenta a confiança no processo eleitoral, mas também facilita a participação de eleitores remotos ou de países com infraestrutura de votação limitada.

2.2.4 Token Não Fungível

Os tokens não fungíveis (do inglês, *non-fungible token (NFT)*) representam uma mudança paradigmática no mundo digital. Cada *NFT* é um ativo digital único, não intercambiável, que pode representar desde arte digital até direitos autorais [29]. Essa singularidade abre novas possibilidades para artistas e criadores de conteúdo, oferecendo uma forma de propriedade digital verificável e segura.

Além disso, os *NFTs* estão criando mercados e formas de valorização de bens digitais. Eles permitem que os usuários tenham um senso de propriedade e autenticidade que não era possível com ativos digitais tradicionais. Isso está não apenas transformando

o mercado de arte, mas também influenciando áreas como colecionáveis, jogos e até mesmo identidade digital.

2.2.5 Token Não Fungível Dinâmico

Os tokens não fungíveis dinâmicos (do inglês, *dynamic non-fungible token (dNFT)*) são uma evolução dos *NFTs*, permitindo que eles mudem ou se adaptem ao longo do tempo, com base em determinadas circunstâncias. Esses *tokens* podem ser programados para refletir mudanças ocorridas nas condições externas ou em seus ativos subjacentes .

Essa capacidade de adaptabilidade e interatividade amplia o escopo dos *NFTs*, tornando-os adequados para uma variedade de aplicações inovadoras. Isso significa que os contratos inteligentes permitem que os *NFTs* sejam alterados ao longo do tempo. Portanto, os contratos inteligentes usam dados fora da cadeia e dentro da cadeia para descobrir se um *token* não fungível deve mudar e, se for o caso, atualizar os metadados de um *NFT* dinâmico. Por exemplo, em jogos ou em colecionáveis digitais, um *token* não fungível dinâmico pode evoluir com base nas ações do usuário ou em eventos do mundo real. Isso não apenas aumenta o engajamento dos usuários, mas também adiciona uma camada de complexidade e valor aos ativos digitais, abrindo novas fronteiras no mundo digital.

Neste ambiente, o uso da tecnologia *blockchain* ajuda a manter seguros e confiáveis os *NFTs* que podem mudar com o tempo. Essa tecnologia permite que todas as alterações feitas em um *NFT* sejam guardadas de forma clara e permanente. Isso torna mais fácil para todos verificar a origem e o histórico de um item digital, construindo uma base de confiança entre quem cria e quem compra ou coleciona esses itens. Além disso, essa tecnologia abre espaço para o uso desses *NFTs* em novas áreas, como na representação de objetos reais que precisam ser atualizados com frequência. Assim, a combinação de mudanças controladas, interatividade e segurança oferece novas oportunidades para o mundo digital, criando um futuro em que ativos digitais podem ser ao mesmo tempo adaptáveis e seguros.

2.2.6 Contratos inteligentes

Contratos inteligentes são uma das aplicações mais poderosas da *blockchain*. Eles são protocolos autoexecutáveis que operam na *blockchain* e executam automaticamente os termos de um contrato quando certas condições são atendidas [37]. Esses contratos reduzem a necessidade de intermediários, aumentando a eficiência e a segurança nas transações. Os contratos inteligentes têm um vasto leque de aplicações, desde a automação de processos em cadeias de suprimentos até a execução de acordos legais.

Eles são particularmente úteis em ambientes onde a confiança e a transparência são essenciais, oferecendo um método seguro e confiável para garantir que os termos de um acordo sejam cumpridos.

2.2.7 Integração *Blockchain-IoT*

IoT é um paradigma tecnológico emergente que transforma objetos comuns em entidades interconectadas capazes de coletar, enviar e processar dados. Essa conectividade estende-se desde dispositivos domésticos comuns até complexos sistemas industriais, criando uma rede vasta e interativa que permeia diversos aspectos da vida diária e do ambiente de trabalho. O uso de *IoT* em diferentes setores econômicos tem sido explorado ao longo das últimas décadas, a ponto de se tornar algo comum e de fácil compreensão pela população em geral. No entanto, as questões relacionadas ao uso dos dados gerados pelos dispositivos *IoT* para fins econômicos é algo que está começando a se desenvolver.

Os dispositivos *IoT* geram grandes volumes de dados sensíveis e confidenciais, apresentando desafios significativos em termos de segurança e escalabilidade. Dados são criados a uma taxa exponencial, à medida que aumentam as interações entre pessoas e empresas no mundo digital. A *blockchain* então, surge para descentralizar o compartilhamento de dados *IoT*, oferecendo uma abordagem mais segura e transparente para o gerenciamento de informações [49]. Também, os contratos inteligentes podem ser usados para autenticação e integridade dos dados, abordando desafios comuns em redes IoT, como pontos únicos de falha e questões de autenticação [8].

Além disso, a monetização de dados *IoT* é explorada, propondo um *framework blockchain* para compartilhamento descentralizado e privativo de dados em troca de serviços monetários [4]. Destaca-se a democratização dos dados *IoT*, permitindo aos usuários controlar e monetizar suas próprias informações. Esses estudos ilustram como a *IoT* está remodelando a interação entre o mundo físico e o digital, ao mesmo tempo em que destacam a necessidade de abordagens robustas de segurança e privacidade para proteger os dados gerados por essa vasta rede de dispositivos conectados.

A integração da tecnologia *blockchain* no *IoT* surge como uma solução promissora, oferecendo maior segurança, descentralização e oportunidades para a monetização de dados. A combinação *blockchain* com *IoT* pode melhorar significativamente a segurança, a confiabilidade e a eficiência das redes de dispositivos conectados [44]. Essa integração permite a criação de sistemas mais seguros para *IoT*, onde as transações e os dados podem ser registrados de forma transparente e imutável.

2.3 Sistema de Arquivos Distribuídos

A evolução no armazenamento e compartilhamento de dados digitais tem sido significativamente influenciada pelo desenvolvimento de Sistemas de Arquivos Distribuídos. O *IPFS* como um paradigma emergente, é uma das tecnologias existentes que possibilita o uso deste tipo de paradigma. *IPFS* é uma rede *peer-to-peer* que armazena dados em um formato distribuído, onde o conteúdo é endereçado pelo seu *hash*, não pela localização. Esta abordagem resolve problemas de centralização e eficiência, características limitantes de sistemas de armazenamento convencionais [46].

A integração da *blockchain* com sistemas de armazenamento descentralizados, como o *IPFS*, proporciona um ambiente mais seguro e confiável para o compartilhamento de dados. Essa combinação traz vantagens como controle de acesso mais granular, garantindo privacidade e segurança, sem a existência de um ente centralizador.

2.3.1 Sistema de Arquivo Interplanetário

O *IPFS*, uma inovação no armazenamento descentralizado, aborda a ineficiência dos sistemas de armazenamento centralizados, armazenando dados em uma rede *peer-to-peer*. Os arquivos são divididos em blocos e distribuídos pela rede, acessíveis através de um *hash* único. Essa metodologia não apenas aumenta a resistência a falhas e ataques, mas também reduz a redundância de dados.

A integração do *IPFS* com a *blockchain*, amplia suas aplicações, incluindo sistemas de identidade descentralizados e mercados de dados. Essa combinação oferece um nível de segurança e transparência inédito em sistemas de armazenamento. Um exemplo envolve sistemas de identidade descentralizados, onde o *IPFS* e a *blockchain* são usados juntos para criar um sistema seguro e resistente a censura para o armazenamento e gerenciamento de identidades digitais.

Nesse sistema, os dados de identidade de um usuário são armazenados no *IPFS*, proporcionando uma maneira segura e acessível de acessar esses dados de qualquer lugar, sem depender de uma autoridade central. A *blockchain* é utilizada para registrar e validar transações relacionadas a essas identidades, como a verificação de identidade ou a atualização de dados pessoais, garantindo a integridade e a imutabilidade do registro de identidade. Esse modelo promove maior controle e privacidade para os usuários, permitindo-lhes provar sua identidade *online* sem expor informações sensíveis desnecessariamente.

2.3.2 Sistema de Nome Interplanetário

O *IPNS* é uma extensão crítica do *IPFS*. O *IPNS* aborda o desafio da mutabilidade dos dados no *IPFS*. No *IPFS*, o *hash* de um arquivo muda se seu conteúdo for alterado, o que pode criar desafios na manutenção de *links* consistentes. O *IPNS* resolve isso ao fornecer identificadores estáveis que podem apontar para diferentes versões de um arquivo ao longo do tempo. Essa funcionalidade do *IPNS* é essencial para a gestão eficaz de arquivos em redes descentralizadas, permitindo atualizações e referências consistentes a dados ou documentos, mesmo com alterações de conteúdo. Esse mecanismo oferece flexibilidade e eficiência significativas, especialmente para aplicações que exigem referências consistentes a longo prazo, como sistemas de publicação de conteúdo e registros de identidade digital.

Adicionalmente, em contextos de *IoT* nos quais dispositivos realizam transmissões de dados com alta frequência, cada nova transmissão resulta na criação de um *hash* exclusivo no *IPFS*. Quando um contrato inteligente regula o acesso a esses dados, torna-se imprescindível a sua atualização com a inserção de cada novo *hash* gerado, o que exige a criação de um novo contrato a cada atualização. Conforme os dados são transmitidos e seus respectivos *hashes* vinculados ao contrato, observa-se um aumento no seu volume, resultando em um aumento nos custos e no tempo necessário para a operação.

Por outro lado, ao optar pela integração do *IPFS* ao *IPNS*, o contrato mantém-se inalterado, pois o *hash* direciona para um identificador *IPNS* fixo, capaz de apontar para versões atualizadas do arquivo sem requerer modificações no contrato inteligente. Essa abordagem otimiza o gerenciamento de dados mutáveis em ambientes de *IoT* e promove uma redução significativa tanto nos custos quanto na complexidade operacional, viabilizando que os contratos inteligentes façam referência a um ponto de dados estável, o qual pode ser renovado no *IPNS* quando necessário.

3. TRABALHOS RELACIONADOS

Recentes investigações têm se focado em desenvolver soluções inovadoras para a integração de *blockchain* e *IoT*, com ênfase particular no setor automotivo. Este capítulo aborda esses esforços, iniciando com uma revisão de literatura específica, seguida pela descrição e comparação de diversas abordagens significativas nesse campo. A revisão da literatura foi especificamente direcionada para identificar avanços no uso de *blockchain* e *IoT* no setor automotivo. Foram utilizadas as bases de dados *Association for Computing Machinery* (ACM), *IEEE Explorer*, *Google Schoolar* e *Springer*, com uma *string* de pesquisa bem definida: *vehicular OR car OR automobile AND blockchain AND IPFS*. A pesquisa foi limitada a publicações entre 2019 e 2024, buscando literatura que não apenas respondesse às questões de pesquisa, mas também estivesse alinhada com o desenvolvimento de soluções escaláveis para a integração de *blockchain* e *IoT* no setor automotivo.

As tecnologias *blockchain* e *IoT* estão revolucionando o setor automotivo de diversas maneiras, introduzindo soluções inovadoras para melhorar a privacidade, segurança e eficiência. Estudos recentes destacam a integração da *blockchain* em Redes Veiculares Ad-Hoc (*VANETs*) para garantir comunicações seguras e privadas entre veículos [47]. O uso do *gRPC*, uma plataforma de chamadas de procedimento remoto de alto desempenho, enfatiza a necessidade de preservar a identidade do usuário em *VANETs* e oferece uma solução de *blockchain* para mitigar problemas de segurança.

Um caso interessante utilizando *blockchain*, é o de tokenização de carros compartilhados em Viena, Áustria [4]. Cada carro da frota compartilhada possui sua própria identidade digital, enquanto a propriedade é distribuída entre todos os carros tokenizados da frota. Dessa forma, cada investidor possui uma participação na frota proporcional ao seu investimento. Essa abordagem proporciona uma maneira única de integrar a tecnologia *blockchain* no setor automotivo, especialmente no contexto dos serviços de compartilhamento de carros.

A integração de tecnologias *blockchain* e *IoT* no setor automotivo não se limita à tokenização de veículos ou gestão de seguros. Também se estende à gestão de dados e à manutenção da privacidade e segurança dos usuários. A arquitetura em camadas da *blockchain* facilita a interação segura entre seguradoras e clientes [9]. Essa abordagem possibilita a implementação de Seguros Baseados no Uso (do inglês, *Usage-Based Insurance (UBI)*), onde as taxas de seguro são ajustadas com base no uso do veículo.

O *UBI* é um exemplo claro de como a coleta e análise de dados podem ser utilizadas para fornecer serviços mais personalizados e eficientes. Dados coletados por veículos conectados, como localização via *GPS*, aceleração e uso geral, são essenciais para avaliar comportamentos de condução e ajustar os prêmios de seguro de forma justa. No entanto, o desafio reside em como coletar esses dados de maneira confiável e segura, com o con-

sentimento dos usuários dos carros e com o estabelecimento de uma relação justa entre as empresas interessadas e os proprietários dos dados.

Também há foco na aplicação da tecnologia *blockchain* combinada com o *IPFS* para o setor de seguros automotivos [48]. Usando contratos inteligentes na *blockchain Ethereum* e armazenamento de dados no *IPFS*, o modelo visa reduzir fraudes, acelerar o processamento de sinistros e melhorar a confiança entre seguradoras e segurados, destacando a potencial revolução na administração dos seguros automotivos.

Além disso, a *blockchain* oferece soluções para outros desafios no setor automotivo, como evidenciado pelo *CarChain* [35]. Este sistema propõe um método confiável e transparente para relatar o histórico dos veículos, abordando um problema generalizado enfrentado por compradores e vendedores.

A *blockchain* também pode ser aplicada em situações de acidentes de trânsito [13]. Após um acidente, os veículos envolvidos podem enviar seus dados para um contrato inteligente dedicado, permitindo uma análise rápida e precisa do incidente. Essa implementação não só acelera o processamento de sinistros, como também garante a privacidade e segurança dos dados coletados.

Modelos inovadores para comércio de dados e energia em veículos elétricos também estão sendo estudados, propondo um sistema onde os veículos elétricos não só consomem energia, mas também podem funcionar como fornecedores, criando um ecossistema energético dinâmico e interconectado [23]. A integração da *blockchain* nesse contexto garante transações confiáveis e seguras, promovendo um mercado descentralizado para energia e dados.

Além disso, há a importância do controle de acesso granular aos dados em sistemas de armazenamento descentralizados [15]. Essa abordagem é crucial para garantir a segurança e privacidade dos dados em um ambiente onde múltiplos usuários e dispositivos estão constantemente interagindo e compartilhando informações. O controle de acesso granular permite uma gestão detalhada de permissões, permitindo que os administradores do sistema definam precisamente quem pode acessar, modificar ou compartilhar dados específicos. Isso é essencial em contextos que requerem proteção rigorosa de informações sensíveis, permitindo que apenas usuários autorizados acessem determinados dados. Implementar um *framework* baseado em *blockchain* para controle de acesso aos dados oferece uma solução robusta para gerir o compartilhamento de dados em grande escala. Usando contratos inteligentes, regras claras e automatizadas para o acesso aos dados podem ser estabelecidas, aproveitando a transparência e segurança proporcionadas pela *blockchain*. Essa abordagem não só garante a privacidade e segurança dos dados armazenados de maneira descentralizada, como também otimiza a eficiência e escalabilidade do controle de acesso, facilitando a gestão de permissões em um ecossistema amplo e diversificado.

Esses estudos reforçam a ideia de que a tecnologia *blockchain*, juntamente com a *IoT*, está remodelando o setor automotivo de maneiras profundas e variadas. Desde a gestão de seguros e relatório de histórico de veículos até o comércio de dados e energia, essas tecnologias estão estabelecendo novos paradigmas para a eficiência, segurança e sustentabilidade no setor automotivo.

A *eDNA* visa abordar desafios específicos no setor automotivo relacionados à segurança, privacidade e eficiência na gestão de dados. A arquitetura integra tecnologias como a *blockchain* e *IoT*, com o objetivo de superar as limitações dos sistemas centralizados tradicionais, ou seja, arquiteturas *Web 2.0*, como pontos únicos de falha, vulnerabilidades a ataques e dificuldades em escalar de acordo com o crescimento do número de dispositivos e veículos conectados. Essa abordagem técnica baseada em evidências garante que a proposta não só atenda às necessidades atuais de segurança, privacidade e eficiência na gestão de dados veiculares, como também estabeleça um novo paradigma para o uso de tecnologias descentralizadas no setor automotivo. Ao fornecer uma solução escalável e segura para a gestão de dados, a proposta tem o potencial de facilitar a adoção de práticas mais inovadoras e sustentáveis no setor.

A proposta *eDNA* surge da observação de que os trabalhos existentes, embora valiosos em muitos aspectos, ainda deixam lacunas significativas. Em particular, eles abordam questões como tokenização de veículos, gestão de seguros e histórico de veículos, mas não aprofundam suficientemente na proteção da privacidade dos dados dos usuários e na segurança das transações. Nos modelos tradicionais, há uma transferência de dados para terceiros, o que representa um risco potencial para a segurança e privacidade dos usuários.

A diferenciação crucial da proposta *eDNA* reside na abordagem *Compute-to-Data (C2D)* e no foco na privacidade dos dados. Ao adotar um modelo onde os dados permanecem localizados nos dispositivos dos usuários e são processados de maneira distribuída, elimina-se a necessidade de transferência de dados para terceiros, mitigando assim muitos dos riscos associados à segurança e privacidade. Isso não só assegura maior controle dos usuários sobre seus dados, como também contribui para a construção de um ambiente mais confiável e seguro para a troca de informações no setor automotivo. Ao estabelecer um novo padrão para a gestão descentralizada de dados, a *eDNA* oferece uma solução escalável e robusta que pode catalisar a adoção de práticas mais inovadoras e sustentáveis na indústria automotiva, alinhando-se assim aos princípios fundamentais de segurança, privacidade e eficiência.

4. ARQUITETURA PROPOSTA

A arquitetura *eDNA* (Figura 4.1) utiliza uma abordagem baseada em *blockchain* pública para proposição de um marketplace de dados automotivos. Esta arquitetura foi projetada para abordar os desafios de escalabilidade e privacidade frequentemente encontrados em aplicações de *blockchain* para a Internet das Coisas (*BIoT*), mantendo um nível de segurança e eficiência característico das soluções baseadas em *blockchain*.

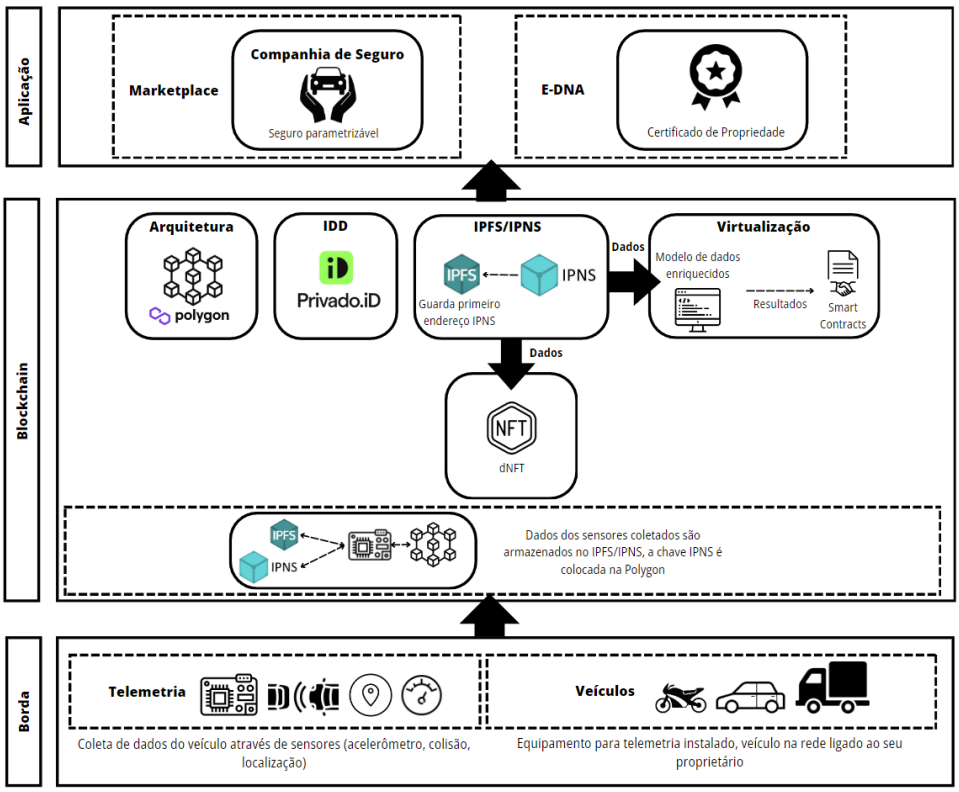

Figura 4.1 – Diagrama da arquitetura.

4.1 Visão geral da arquitetura

A arquitetura proposta é construída sobre a *blockchain Polygon*, uma solução de escalabilidade de camada 2 compatível com a *Ethereum Virtual Machine* (*EVM*), oferecendo maior velocidade e custos de transação mais baixos em comparação com a rede principal da *Ethereum*.

A arquitetura *eDNA* é composta por três camadas: borda, *blockchain* e aplicação. A camada de borda contém sensores, atuadores e dispositivos de telemetria responsáveis pela coleta dos dados automotivos. A camada *blockchain*, executada sobre a *blockchain Polygon*, é responsável por receber os dados coletados pela camada borda, armazená-los

no *IPFS/IPNS* (Figura 4.3) e por enriquecer esses dados através do conceito de *C2D* (Figura 4.4). Além disso, essa camada verifica a identificação dos usuários e dos automóveis através do uso do conceito de identidade distribuída e descentralizada implenentado pela *Privado ID*, antiga *Polygon ID*. Por fim, um *dNFT* é criado para permitir que seus metadados possam ser alterados com base em condições externas, como quilometragem do veículo, proprietários anteriores e manutenções realizadas. Finalmente, se tem a camada de aplicação que é responsável por fornecer as *APIs* para as aplicações descentralizadas (*dAPPs*).

Essas requisitos são essenciais para atender às demandas das aplicações *BIoT*, que requerem respostas rápidas e operações de baixo custo. Ao operar na rede *Polygon*, a *eDNA* torna-se uma plataforma descentralizada e transparente, elementos fundamentais para garantir a integridade e confiabilidade nas aplicações *BIoT*. Essa escolha estratégica fortalece a base da *eDNA*, estabelecendo-a como uma solução inovadora e adaptável no campo da *blockchain* aplicada à *IoT*.

4.2 Identidade Descentralizada

A implementação da Identidade Descentralizada (*DID*) é baseada no uso da API disponibilizada pela *Privado ID*, que facilita a integração de aplicações e serviços com as funcionalidades de identificação descentralizada. O uso de *DID* permite realizar a verificação da identidade de um usuário sem a necessidade de expor suas informações pessoais.

O objetivo de usar *DID* na *eDNA* é sua capacidade de reforçar a segurança e a privacidade no contexto de gestão de identidade e dados. A *DID* permite a verificação de credenciais sem revelar informações sensíveis ou armazenar usuário/senha como ocorre em um sistema centralizado. Isto é possível porque a identidade é armazenada na *blockchain* e pode ser verificada através do uso do conceito de prova de conhecimento zero (*ZKP*). Isso reduz o risco de conformidade e protege contra bots e ataques *Sybil*.

A implementação da *DID* permite que os usuários criem e gerenciem sua própria identidade digital, controlando como seus dados pessoais são acessados e utilizados no ambiente digital (identidade autossoberana). Isso é relevante na gestão do histórico de dados do veículo, até na integração com sistemas de seguros e financeiros, onde estas informações são críticas.

4.3 Tokens Não Fungíveis Dinâmicos

Um *dNFT* é um *token* não fungível dinâmico que expande o espaço de design que os *NFTs* podem abordar através de sua capacidade de se adaptar e mudar em res-

posta a eventos externos e dados. Isso significa que a lógica codificada nos contratos inteligentes permite que os *NFTs* sejam alterados ao longo do tempo, preservando todas as propriedades originais do *NFT*. Contratos inteligentes usam dados *on-chain* e *off-chain* para determinar se um *token* não fungível deve mudar ou não e, em caso de mudança, atualizar os metadados de um dNFT. Na arquitetura proposta, o *dNFT* desempenha o papel de armazenar dados importantes sobre o veículo, tais como um acidente onde o seguro foi acionado, quilometragem percorrida pelo veículo, antigos proprietários, entre outros dados.

Um exemplo que demonstra a aplicabilidade do *dNFT* foi criado pela *Chainlink* [6] em parceria com a *Tesla* (Figura 4.2). Neste exemplo, para cada automóvel é criado um *dNFT* e os metadados são atualizados para refletir, ao longo do tempo, os registros de serviço realizados, quilometragem, relatório de acidentes e valor de mercado.

Figura 4.2 – dNFT Tesla e Chainlink [6].

Para a criação do *dNFT*, a configuração de um *token* ERC721 é utilizada, projetado como uma *commodity* rara e usada para criar edições limitadas. Sua principal característica é a singularidade. O contrato inteligente interage de maneira que o *dNFT* seja alterado com base nas variáveis definidas, permitindo a atualização contínua dos metadados do *dNFT* para refletir mudanças importantes, garantindo não apenas a precisão dos dados representados pelo dNFT, mas também aumentando a confiança e a transparência para todas as partes interessadas, desde proprietários de veículos até seguradoras e compradores potenciais.

Além disso, usar contratos inteligentes para gerenciar as atualizações do *dNFT* garante que as mudanças sejam feitas de forma segura e automática. Isso não só melhora a integridade dos dados veiculares, como também facilita uma gama mais ampla de aplicações, como a personalização de seguros com base no comportamento de condução ou histórico do veículo, e o desenvolvimento de novos modelos de negócios no setor automotivo.

4.4 Sistema de Armazenamento *IPNS/IPFS*

O *IPFS*, como um sistema de arquivos distribuído e imutável, é usado para armazenar os dados gerados pela camada borda na arquitetura *eDNA*. Cada dado carregado no *IPFS* é identificado por um Identificador de Conteúdo (do inglês, *Content Identifier (CID)*), derivado de um *hash* do conteúdo dos dados, garantindo que qualquer mudança nos dados resultará em uma mudança no seu *CID*, promovendo a imutabilidade e integridade dos dados.

Dada a natureza dinâmica e o volume substancial de dados gerados em sistemas *BIoT*, como na arquitetura *eDNA*, o uso exclusivo do *IPFS* pode não ser adequado. Cada novo dado gerado pelos dispositivos *IoT* na *eDNA* requer uma transação para atualizar o estado do contrato inteligente correspondente com o *CID* atual no *IPFS*, levando a taxas de transação recorrentes e possíveis problemas de escalabilidade. Para superar esse problema, o *IPNS* é integrado à arquitetura *eDNA*, fornecendo ponteiros mutáveis que podem ser atualizados sem perder a referência original ou mudar o endereço.

Na prática, isso significa que, em vez de atualizar continuamente o conteúdo de um contrato inteligente com um novo *CID* do *IPFS* para cada nova transmissão de dados, um ponteiro *IPNS* fixo pode ser associado a um contrato inteligente específico, como na Figura 4.3. Esse ponteiro sempre apontará para o último elemento da lista ligada no *IPFS*, facilitando o acesso aos dados mais recentes e permitindo a rastreabilidade completa da sequência de dados, além de trazer várias vantagens para a arquitetura.

Esta abordagem reduz a necessidade de transações frequentes na *blockchain*, mitigando problemas relacionados a custos e escalabilidade. Além disso, ela mantém a rastreabilidade e integridade dos dados, garantindo que todo o histórico de dados de um dispositivo possa ser acessado e verificado. Também, ao manter a lista de dados no *IPFS* e IPNS, a arquitetura *eDNA* se beneficia da descentralização e eficiência do sistema de arquivos distribuído, garantindo acesso rápido e seguro aos dados armazenados.

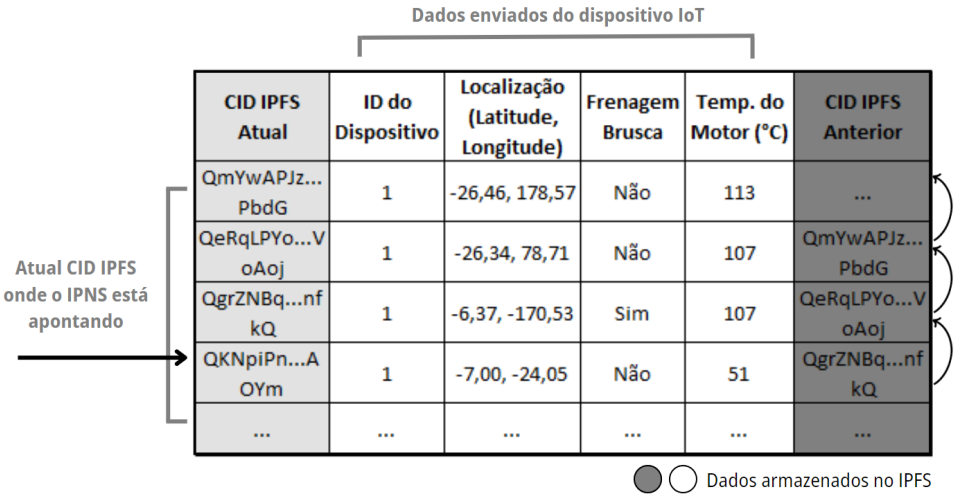

Figura 4.3 – Esquema *IPFS/IPNS*. A chave pública do *IPNS* aponta para o último elemento da lista encadeada armazenada no *IPFS*.

4.5 Virtualização de Dados

A virtualização de dados é um conceito que envolve a criação de uma representação abstrata de dados confidenciais. Em contextos onde a venda de dados e o acesso limitado são relevantes, a virtualização de dados se torna particularmente importante. Além de facilitar a análise e o acesso a grandes volumes de dados, ela também fornece um meio de controlar e limitar o acesso a esses dados, garantindo segurança e conformidade com regulamentos de privacidade.

No contexto do *eDNA*, a virtualização de dados emerge como um mecanismo essencial para equilibrar as necessidades de acesso e análise de dados com requisitos de segurança e privacidade. Torna-se especialmente relevante em um ambiente onde a quantidade e a variedade de dados veiculares estão crescendo exponencialmente, incluindo informações sensíveis como padrões de condução, localização *GPS* e histórico de manutenção.

A capacidade de acessar esses dados para análises avançadas ou treinamento de modelos de *IA* sem comprometer a segurança ou a privacidade é uma vantagem significativa. O conceito principal que diferencia o *eDNA* de outras arquiteturas é que terceiros interessados não têm acesso aos dados em si, mas sim às informações geradas pelo processamento dos dados. Isso é possível através do conceito de *C2D* (Figura 4.4), originalmente proposto pelo *Ocean Protocol* [41], onde as partes interessadas enviam algoritmos que serão executados nos dados, recebendo como resultado os insights gerados.

O *Ocean Protocol* é uma plataforma que combina tecnologia *blockchain* com economia de dados. Ele permite que proprietários de dados compartilhem seus dados de forma segura e controlada, enquanto terceiros interessados podem acessar e utilizar es-

ses dados para diversos fins, como por exemplo análise ou treinamento de modelos de inteligência artificial (IA), de acordo com as regras de negócio estabelecidas pelo proprietário dos dados. O *C2D* é uma inovação chave dentro do *Ocean Protocol*, pois resolve o problema de acessar dados valiosos sem comprometer a privacidade ou segurança.

Ao invés de permitir o acesso aos dados pela parte interessada para que ela possa realizar o processamento destes dados, o *C2D* possibilita que o processamento ocorra onde os dados estão armazenados, sem que a parte interessada tenha acesso a estes dados. Assim, os algoritmos são enviados aos dados, mas os dados em si nunca são compartilhados ou transferidos. Isso é especialmente útil em situações onde os dados são sensíveis ou quando existem regulamentações rigorosas sobre como os dados podem ser compartilhados.

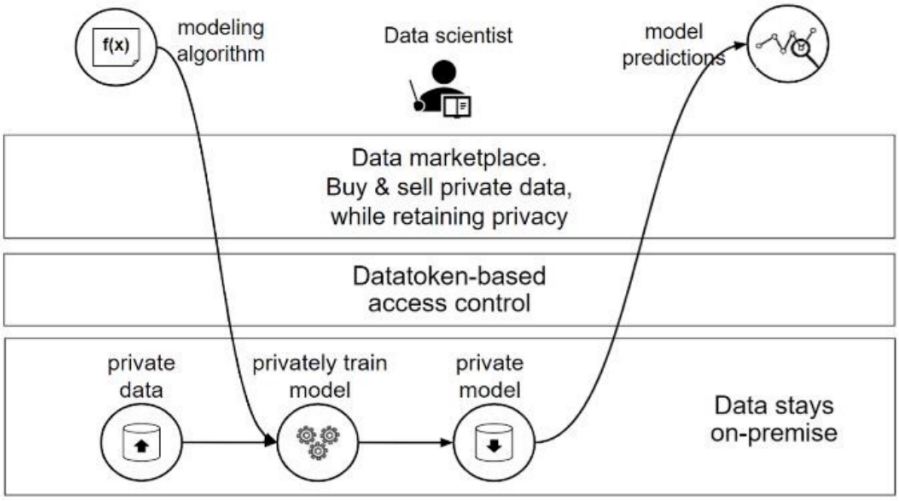

Figura 4.4 – Modelo *Compute-to-Data* (*C2D*) [41].

Na arquitetura *eDNA*, a combinação de armazenamento descentralizado via *IPFS/IPNS* com o conceito *C2D*, visto na Figura 4.4, complementado pela virtualização de dados, cria um ecossistema para gerenciamento, análise e compartilhamento de dados no setor automotivo. Este sistema permite que os dados sejam armazenados, acessados e analisados de forma segura, eficiente e em conformidade com regulamentos de privacidade, sem comprometer a integridade e confidencialidade de dados sensíveis dos usuários.

A decisão de confiar em determinados algoritmos para o processamento de dados é uma prerrogativa exclusiva do proprietário dos dados. Eles têm a liberdade de analisar o código, incluindo testá-lo em um ambiente controlado para verificar suas dependências, comunicações e uso de recursos. Assim, a entidade que detém os dados é a mesma que avalia os riscos e benefícios ao escolher em quais algoritmos confiar, com base em uma avaliação de risco versus recompensa.

O proprietário dos dados também é responsável por precificá-los na criptomoeda de sua escolha e determinar o período de disponibilidade desses dados a partir da compra.

Conjuntos de dados podem ser armazenados em várias localizações, abrangendo provedores centralizados ou descentralizados, como servidores locais ou *IPFS*. O único requisito é a disponibilidade de um *URL* para acessar o conjunto de dados.

A responsabilidade pela manutenção e hospedagem dos dados fica inteiramente com o detentor do conjunto de dados. O uso destes conceitos na arquitetura *eDNA* representam um avanço em direção a um gerenciamento de dados mais seguro, privado e eficiente no setor automotivo, abrindo novas possibilidades para o desenvolvimento e implementação de serviços automotivos avançados e personalizados.

4.6 Contratos Inteligentes

Os contratos inteligentes foram desenvolvidos para regular as trocas de dados e gerenciar transações financeiras entre nós solicitantes e proprietários de dados. A *Polygon* perimite a programação de contratos inteligentes através do uso da linguagem *Solidity*, que pode ser facilmente vinculada ao nível de aplicação usando scripts como *Node.js* ou *Python*, por exemplo.

Neste contexto, foi utilizado o *OpenZeppelin* [32], uma biblioteca para facilitar e acelerar o desenvolvimento de contratos inteligentes na rede Ethereum e *blockchains* compatíveis, como a *Polygon*, oferecendo uma série de funcionalidades cruciais para o sucesso da arquitetura, como a capacidade de auditar o contrato para verificar falhas de segurança. A arquitetura de contratos inteligentes foi construída com base em dois tipos de contratos, Frota e Veículo.

1) Vehicles Fleet: O Contrato Inteligente *Vehicles Fleet* é a interface onde nós solicitantes podem procurar e adquirir novos dados dos dispositivos veiculares. Ele contém uma chave pública *IPNS* fixa que aponta para o primeiro elemento da lista encadeada de veículos disponíveis. Esta lista também contém os endereços de cada Contrato Inteligente de Veículo, com os quais o nó solicitante pode interagir diretamente.

2) Vehicle: Para cada veículo conectado ao *eDNA*, um Contrato Inteligente *Vehicle* é criado e pertence ao agente que fornece os dados, ou seja, o proprietário do veículo. Os dados são armazenados no *IPFS* e podem ser acessados através da chave pública *IPNS* armazenada no contrato inteligente.

4.7 Fluxograma da *eDNA*

A arquitetura *eDNA* é projetada para garantir um sistema de dados seguro, eficiente e descentralizado para o setor automotivo. Esta seção descreve visualmente o fluxo

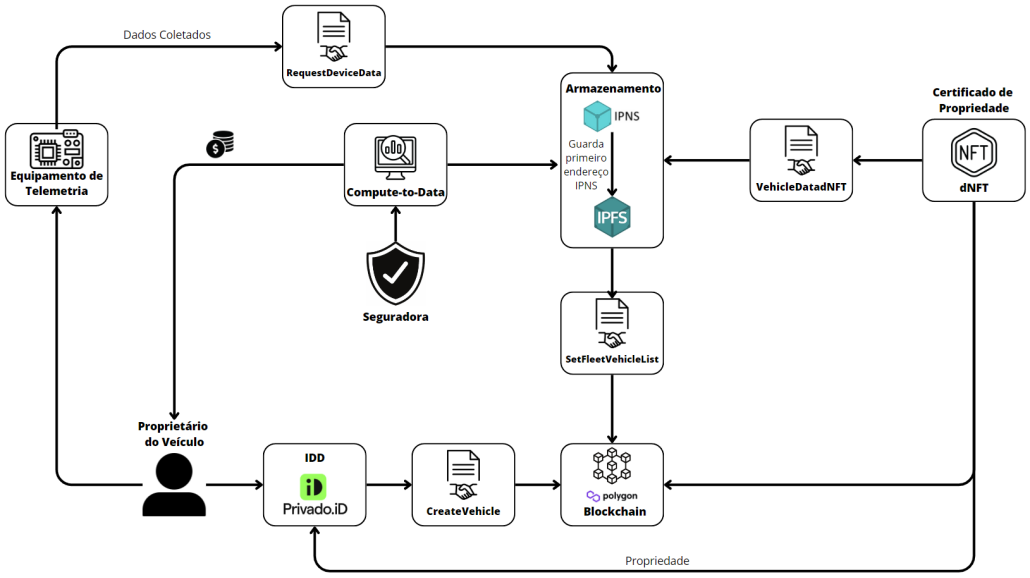

Figura 4.5 – Fluxograma da *eDNA*.

de dados e interações entre os componentes da arquitetura *eDNA*, conforme ilustrado na Figura 4.5.

O fluxograma da *eDNA* (Figura 4.5) está dividido nas seguintes etapas principais:

1. **Coleta de Dados:** O dispositivo de telemetria coleta os dados automotivos, como, por exemplo, velocidade, RPM, marcha, freio, temperatura do motor e nível de combustível. Os dados coletados ficam na camada de borda, onde são preparados para a transmissão segura. Os dados são transmitidos da camada de borda para a camada *blockchain* (Figura 4.1), garantindo a integridade e a privacidade durante o transporte.

2. **Armazenamento Descentralizado:** Na camada *blockchain*, os dados são armazenados no *IPFS/IPNS*.

3. **Criação de *dNFTs*:** Com base nos dados armazenados, *dNFTs* são gerados, encapsulando informações dinâmicas e protegendo dados sensíveis através de contratos inteligentes.

4. **Gerenciamento de Identidades:** Através da *Privado ID*, as identidades dos proprietários dos dados e dos veículos são verificadas e gerenciadas de forma descentralizada, utilizando provas de conhecimento zero (*ZKP*).

5. **Compartilhamento de Dados:** Os dados encapsulados nos *dNFTs* podem ser compartilhados de forma segura e transparente com terceiros, como seguradoras, utilizando contratos inteligentes que garantem a integridade das transações.

6. **Monetização de Dados:** Os proprietários dos dados podem monetizar suas informações, permitindo que terceiros executem algoritmos sobre os dados (*C2D*) sem

que esses dados precisem ser transferidos, assegurando a privacidade e a segurança.

O fluxograma da *eDNA*, portanto, representa uma abordagem robusta para a gestão de dados automotivos, combinando tecnologias para desenvolver uma solução segura, eficiente e escalável.

5. EXPANDINDO A ARQUITETURA *COMPUTE-TO-DATA* E A IDENTIDADE DESCENTRALIZADA COM *PRIVADO ID*

5.1 *Compute-to-Data*

Os dados privados possuem um valor imenso, pois podem melhorar significativamente os resultados de pesquisa e negócios. No entanto, preocupações com a privacidade e o controle frequentemente impedem seu acesso. A abordagem de *Compute-to-Data* (*C2D*) trata esse desafio concedendo acesso específico aos dados privados sem compartilhá-los diretamente. Essa abordagem é útil em vários domínios, incluindo pesquisa científica, avanços tecnológicos e mercados onde dados privados podem ser comercializdos de forma segura, enquanto preservam a privacidade. As empresas e os indivíduos podem aproveitar a oportunidade de monetizar seus ativos de dados, garantindo a máxima proteção das informações sensíveis.

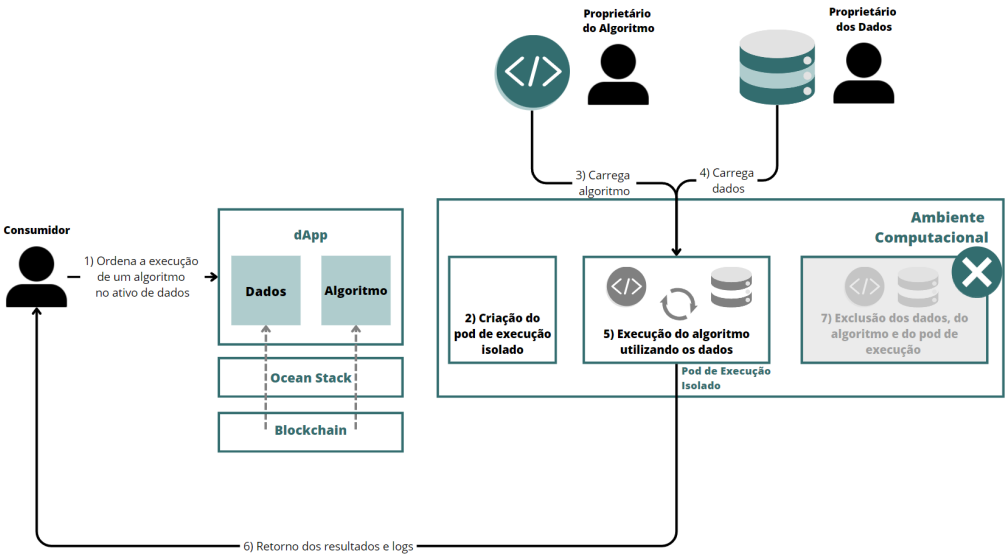

Figura 5.1 – Visão Geral do Fluxo *Compute-to-Data* (*C2D*) (adaptado de [41]).

5.1.1 Descrição do Paradigma

A arquitetura *C2D* adota um paradigma de processamento de dados onde a computação é realizada de forma segura e a privacidade dos dados sensíveis é preservada. O fluxo de trabalho *C2D* é esquematizado na Figura 5.1 e compreende as seguintes etapas, de forma geral:

- O consumidor inicia o processo selecionando o conjunto de dados desejado e o algoritmo que deseja executar sob estes dados, a partir do catalogo de dados e algoritmos disponíveis no marketplace. Em alguns casos, o consumidor fornece o algoritmo que deseja executar sob os dados, ao invés de selecionar um algortimo do catálogo de algoritmos disponíveis.

- A partir da seleção, a solicitação é validada pelo proprietário dos dados via *dApp*, ou seja, o proprietário informa se o algoritmo escolhido pode ser utilizado no conjunto de dados ou não. Se o proprietário dos dados autorizar o uso do algoritmo, o sistema executa o contrato inteligente subjacente que estabelece as regras de negócio para a transação. Caso o proprietário dos dados negue a autorização, o negócio é desfeito.

- Caso seja aprovada pelo proprietário dos dados, um *pod* de execução dedicado e isolado é criado para a execução da demanda. No contexto do *C2D*, um *pod* refere-se a uma unidade computacional efêmera, frequentemente baseada em contêineres, responsável por processar os dados de maneira segura e temporária. Ele garante que os dados sensíveis nunca saiam do ambiente controlado e sejam acessíveis apenas para o processamento autorizado.

- O *pod* carrega o algoritmo que será executado e o conjunto de dados selecionado para processamento em seu ambiente e realiza o processamento.

- Os resultados e logs gerados pelo algoritmo são retornados ao consumidor.

- O *pod* de execução exclui o conjunto de dados do ambiente de execução, o algoritmo e ele mesmo para garantir a privacidade e a segurança dos dados.

Já na Figura 5.2, destaca-se o processo detalhado pelo qual os dados são processados, começando pela solicitação de acesso até a entrega de resultados, sem a transferência direta de dados sensíveis.

O fluxo inicia com a seleção de dados pelo consumidor, que escolhe quais conjuntos de dados deseja processar e os algoritmos a serem aplicados (etapas de 1 a 3 da figura 5.2). Após a aprovação do proprietário dos dados, o *C2D* inicia um ambiente de execução isolado, onde os algoritmos são aplicados aos dados sem que estes sejam expostos ou transferidos para fora do ambiente seguro (etapas de 4 a 35 da figura 5.2).

Os resultados gerados pelo algoritmo são então retornados ao consumidor, garantindo que apenas as informações processadas, e não os dados brutos, sejam compartilhadas (etapas 33 e 34 da figura 5.2). Este fluxo não apenas protege a privacidade dos dados, mas também reduz os riscos associados à transferência e ao armazenamento de dados sensíveis.

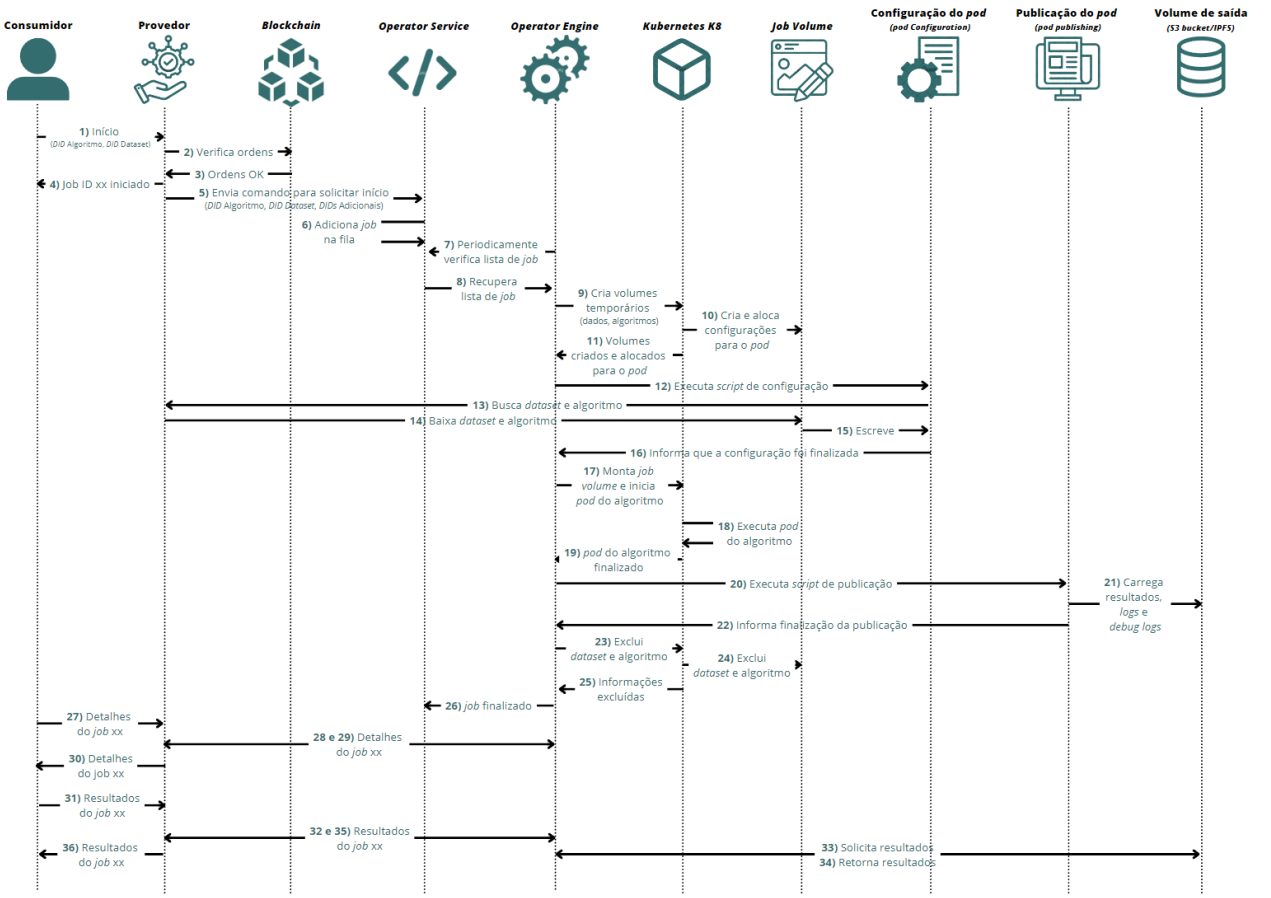

Figura 5.2 – Fluxo completo do *Compute-to-Data* (*C2D*) (adaptado de [41]).)

5.1.2 Interação e Componentes do Sistema

A interação entre o consumidor e o provedor (proprietário dos dados) segue um fluxo específico (figura 5.2). Para iniciar o processo, o consumidor contata o provedor invocando a função `start(did, algorithm, aDIDtionalDIDs)` com parâmetros como o identificador de dados (*DID*), o algoritmo e *DIDs* adicionais, se necessário. Ao receber essa solicitação, o provedor gera um identificador de trabalho único e o retorna ao consumidor. O provedor então assume a responsabilidade de supervisionar as etapas restantes.

Durante o processo de computação, o consumidor pode verificar o status do trabalho consultando o provedor através do uso da função `getJobDetails(jobId)`, fornecendo o identificador do trabalho como referência.

O consumidor tem a opção de iniciar um trabalho de computação usando um ou mais ativos de dados. Para explorar essa funcionalidade, é possível utilizar as bibliotecas `ocean.py`[43] e `ocean.js`[42].

5.1.3 Componentes e Pré-condições para o Processamento

Os principais atores/componentes do fluxo de trabalho *C2D* incluem:

- **Consumidores**: Usuários finais que desejam utilizar os serviços de processamento oferecidos pelo publicador de dados.
- ***Operator-Service***: Micro-serviço que lida com as solicitações de processamento.
- ***Operator-Engine***: Sistemas de computação onde o processamento será executado.
- ***Kubernetes***: *Cluster* K8.

Algumas pré-condições devem ser atendidas, antes que o fluxo *C2D* possa iniciar:

- O Objeto de Dados Descentralizado (*DDO*) do Ativo deve ter um serviço de computação definido. O *DDO* é um Identidade Descentralizada que representa um ativo de dados e contém diversos atributos que descrevem e gerenciam o ativo, incluindo metadados, serviços e credenciais. Ele garante que os dados estão preparados para serem processados conforme as especificações do sistema. todotem um pequeno exemplo disto?
- O serviço de computação do *DDO* do Ativo deve permitir a execução de algoritmos. Isso significa que os algoritmos podem ser executados diretamente sobre os dados, sem a necessidade de transferi-los para fora do ambiente controlado, mantendo a privacidade e a segurança dos dados.
- O *DDO* do Ativo deve especificar um *endpoint* do Provedor Ocean (seção 5.1.4) exposto pelo Publicador. Esse *endpoint* é necessário para que o *Operator-Service* e o *Operator-Engine* possam acessar e processar os dados conforme solicitado.

Basicamente, um *DDO* é um Identidade Descentralizada que representa um ativo de dados. Cada *DDO* contém diversos atributos que descrevem e gerenciam o ativo. Exemplificando, usando o *DDO*, o proprietário do veículo pode permitir que sejam executados algoritmos específicos nos dados de telemetria sem transferir os dados em si. O comprador dos dados envia um algoritmo ao *DDO*, que executa a computação nos dados locais e retorna os resultados ao comprador, garantindo que os dados originais nunca saiam do controle do proprietário.

Essa estrutura assegura que a privacidade dos dados do proprietário do veículo seja mantida, enquanto permite que o comprador dos dados obtenha *insights* valiosos. O *DDO* age como um mediador que realiza o processamento necessário sem expor os dados brutos a terceiros.

5.1.4 Provedor Ocean

O Provedor *Ocean* atua como um intermediário entre os usuários e a infraestrutura de dados. Este provedor gerencia as interações seguras, autorizando e autenticando o acesso aos dados e serviços de computação. Compondo a Figura 5.2, ele facilita uma integração organizada no ecossistema do *C2D*, atuando como uma ponte essencial para todos os processos de computação e acesso a dados. As responsabilidades principais do Provedor *Ocean* incluem autenticação e autorização de usuários, gerenciamento da infraestrutura, e fornecimento de *APIs* transparentes e acessíveis para os consumidores solicitarem serviços de dados e computação.

Dentro deste provedor, o *Operator Service*, um micro-serviço vital, gerencia o ciclo de vida das solicitações de processamento de dados. Este serviço é fundamental para a execução eficaz dos pedidos de computação, garantindo que todos os aspectos do processamento sejam seguros e eficientes. Ele expõe *APIs HTTP* que permitem aos usuários executar algoritmos sobre dados, acessar pontos de extremidade e gerenciar suas solicitações. Além disso, o *Operator Service* coordena com a infraestrutura subjacente, seja na nuvem ou *on-premise*, utilizando as credenciais fornecidas pelo publicador de dados para garantir operações seguras. Ele também monitora a execução das tarefas computacionais e gerencia a coleta e armazenamento de *logs* para revisão de segurança e análise posterior.

O *Operator Engine* complementa o *Operator Service*, ao comandar os recursos de computação no *backend*. Utilizando *Kubernetes*, ele gerencia a execução dos trabalhos de computação em ambientes isolados (*pods*), configurando e gerenciando-os para assegurar a privacidade e segurança dos dados. Este componente também é responsável pela alocação eficiente de recursos, otimizando o desempenho e minimizando custos, além de supervisionar as cargas de trabalho para garantir que os fluxos de trabalho sejam executados conforme os parâmetros especificados e cumpram com as políticas de segurança.

No contexto da organização de *pods*, o repositório *Pod-Configuration* e a ferramenta *Pod Publishing* são essenciais para a preparação e conclusão dos trabalhos de computação. O *Pod-Configuration* configura dinamicamente o ambiente de execução, assegurando que todos os recursos e dados necessários estejam prontamente disponíveis e seguros. Esta automação simplifica o processo de configuração, reduzindo o potencial para erros humanos e acelerando o tempo de preparação. Por outro lado, o *Pod Publishing* gerencia os resultados do processamento, coletando e enviando os dados processados de volta para os consumidores de forma segura e eficiente.

Após a conclusão do processamento, essa ferramenta também garante que todos os dados temporários sejam destruídos, mantendo a integridade e confidencialidade

dos dados originais e prevenindo qualquer vazamento de informações sensíveis após o término das operações.

5.1.5 Armazenamento de Dados

No *C2D*, cada algoritmo é executado em um *pod Kubernetes*, que é configurado com volumes específicos para armazenar dados de forma segura e isolada. Os conjuntos de dados de entrada são armazenados no volume `/data/inputs`, organizados em pastas indexadas pelo identificador do conjunto de dados (`did`) e pelo identificador do serviço (`service_id`), garantindo acesso restrito ao algoritmo em execução no *pod*. Os *DDOs*, que contêm metadados e informações de controle sobre os dados e o algoritmo, são armazenados em `/data/ddos` como arquivos *JSON*.

Os resultados do processamento são mantidos temporariamente em `/data/outputs` antes de serem transferidos para um armazenamento em nuvem, e as *URLs* correspondentes são enviadas ao consumidor. Além disso, todos os *logs* gerados, como saídas de impressão e registros de console, são coletados em `/data/logs` e também disponibilizados ao consumidor.

5.1.6 Metadados do Algoritmo

Dentro *C2D*, os algoritmos são tratados como ativos digitais e têm seus atributos definidos no campo `metadata.algorithm`. Estes metadados descrevem aspectos essenciais do algoritmo, como a linguagem de programação usada (atributo *language*), a versão do *software* em notação *SemVer* (atributo *version*), os parâmetros que o consumidor deve fornecer antes da execução (atributo *consumerParameters*), e a imagem do *Docker* que será utilizada para executar o algoritmo (atributo *container*).

Esta estrutura detalhada garante que o algoritmo possa ser executado de forma precisa e segura no ambiente designado. A Tabela 5.1 apresenta um exemplo sintético de metadados para um algoritmo de Análise Preditiva, modelado para prever falhas em veículos com base em dados de telemetria.

Essa estrutura garante que o algoritmo possa ser executado dentro do *C2D* de forma modular, segura e replicável, assegurando que o processamento seja realizado sem a exposição dos dados brutos.

```
{
  "metadata": {
    "algorithm": {
      "name": "AnalisePreditiva",
      "description": "Modelo de aprendizado de máquina para prever falhas em veículos com base em telemetria.",
      "language": "Python",
      "version": "1.2.0",
      "consumerParameters": [
        {
          "name": "janela_temporal",
          "type": "string",
          "default": "30 dias",
          "description": "Período de tempo dos dados a serem analisados."
        },
        {
          "name": "modelo",
          "type": "string",
          "default": "XGBoost",
          "description": "Algoritmo de aprendizado de máquina utilizado."
        },
        {
          "name": "threshold_alerta",
          "type": "float",
          "default": 0.85,
          "description": "Valor mínimo para acionar um alerta de falha."
        }
      ],
      "container": {
        "image": "registry.example.com/c2d/analise-preditiva:latest",
        "tag": "latest",
        "entrypoint": "python main.py",
        "requirements": ["pandas", "numpy", "scikit-learn", "xgboost"]
      }
    }
  }
}
```

Algoritmo 5.1 – Exemplo de metadados de um algoritmo para *Compute-to-Data (C2D)*

5.1.7 Opções de Computação

Os ativos de computação podem ser configurados com opções adicionais que definem a segurança e a funcionalidade do processo de computação. Essas opções incluem a permissão para executar algoritmos em texto bruto (*allowRawAlgorithm*), o controle de acesso à rede durante a execução do algoritmo (*allowNetworkAccess*), e listas de algo-

ritmos e publicadores confiáveis (*publisherTrustedAlgorithmPublishers* e *publisherTrustedAlgorithms*).

A estrutura do *publisherTrustedAlgorithms* especifica, detalhadamente, quais algoritmos podem ser usados, incluindo o *DID* do algoritmo, o *checksum* dos arquivos, e o *checksum* da seção do contêiner. Esta verificação de integridade é essencial para garantir que o algoritmo não foi alterado e está conforme as especificações do publicador.

O Algoritmo 5.2 exemplifica a configuração das opções de computação. Essas configurações são essencias para manter a segurança dos dados e a integridade do processamento, permitindo que apenas algoritmos verificados e aprovados pelos publicadores especificados sejam utilizados.

```
{
  "computeOptions": {
    "allowRawAlgorithm": false,
    "allowNetworkAccess": false,
    "publisherTrustedAlgorithms": [
      {
        "did": "did:op:123456789abcdef",
        "filesChecksum": "a1b2c3d4e5f6g7h8i9j0",
        "containerChecksum": "z9y8x7w6v5u4t3s2r1"
      }
    ],
    "publisherTrustedAlgorithmPublishers": [
      "0x123456789abcdef",
      "0xa1b2c3d4e5f6g7h8i9j0"
    ]
  }
}
```

Algoritmo 5.2 – Exemplo de configuração das opções de computação no modelo *Compute-to-Data (C2D)*

5.1.8 Desenvolvimento de Algoritmos para *Compute-to-Data*

No *stack* do *Ocean Protocol*, os algoritmos são tratados como ativos distintos e estruturados a partir de componentes essenciais, como o código-fonte, a imagem *Docker* associada e o ponto de entrada para execução. Esses algoritmos podem ser publicados, descobertos e executados de forma segura dentro da plataforma, garantindo a preservação da privacidade dos dados e o processamento confiável.

Para que um algoritmo seja corretamente registrado e utilizado no ambiente *Compute-to-Data*, é necessário definir sua configuração de execução. Isso envolve a especificação da imagem base do *Docker*, que determina o ambiente de execução do código, e

a definição de tags e versões que permitem o rastreamento e a reprodutibilidade das execuções. Além disso, é preciso configurar os parâmetros de *runtime*, como limites de uso de memória e CPU, garantindo eficiência e segurança na execução distribuída. Também devem ser incluídas as dependências do algoritmo, listadas em um arquivo de requisitos, assegurando que todas as bibliotecas necessárias estejam disponíveis no ambiente de execução. Por fim, o ponto de entrada deve ser claramente definido, indicando o script responsável por iniciar o processamento dos dados e gerar os resultados esperados.

Após a configuração desses elementos, o algoritmo pode ser registrado no serviço de metadados do *Ocean Protocol*, tornando-se acessível para execução dentro do fluxo *Compute-to-Data*. Esse processo assegura que os dados sensíveis nunca sejam expostos diretamente ao consumidor do processamento, permitindo que apenas os resultados da computação sejam compartilhados. Um exemplo prático e detalhado de implementação de um algoritmo para *Compute-to-Data* será apresentado na Seção 6.

5.2 Identidade Descentralizada com *Privado ID*

A *Privado ID*, anteriormente conhecida como *Polygon ID*, evoluiu para uma solução independente de identidade descentralizada e distribuída (*IDD*). A mudança estratégica visou manter a neutralidade e independência do ecossistema *Polygon*, com o objetivo de capturar um mercado mais amplo e facilitar a interoperabilidade com diferentes protocolos de *blockchain* [16].

Embora a *Privado ID* tenha sido adquirida pela *Polygon*, a empresa mantém suas raízes neutras e agnósticas em relação ao ecossistema de *blockchain*. Essa neutralidade é crucial para proporcionar uma maneira confiável e padronizada de lidar com dados confidenciais, permitindo que a empresa se alinhe aos protocolos existentes e se torne uma ferramenta valiosa no ecossistema *blockchain*.

5.2.1 Identificador Distribuido Descentralizado

Cada identidade é identificada por um DID, um identificador único que representa todas as informações baseadas em identidade através de Credenciais Verificáveis (do inglês, *Verifiable Credentials*). Um *VC* pode representar qualquer tipo de informação relacionada a um indivíduo, empresa ou objeto, desde a idade de uma entidade até um certificado de membro emitido por uma *DAO*.

5.2.2 Triângulo da Confiança

A arquitetura do *framework* de identidade descentralizada (Figura 5.3) é composta por três módulos principais:

1. **Detentor da Identidade (*Identity Holder*):** A entidade que detém as reivindicações em sua carteira digital (*Wallet*). O Detentor gera Provas de Conhecimento Zero (*ZKP*) dos *VCs* emitidos e apresenta essas provas ao Verificador, utilizando zk-SNARKs pela sua eficácia em garantir a privacidade sem revelar detalhes subjacentes das credenciais [25].

2. **Emissor (*Issuer*):** A entidade que emite *VCs* para os Detentores, assinados criptograficamente.

3. **Verificador (*Verifier*):** A entidade que verifica a prova apresentada pelo Detentor, realizando verificações específicas para garantir a autenticidade e a conformidade das provas.

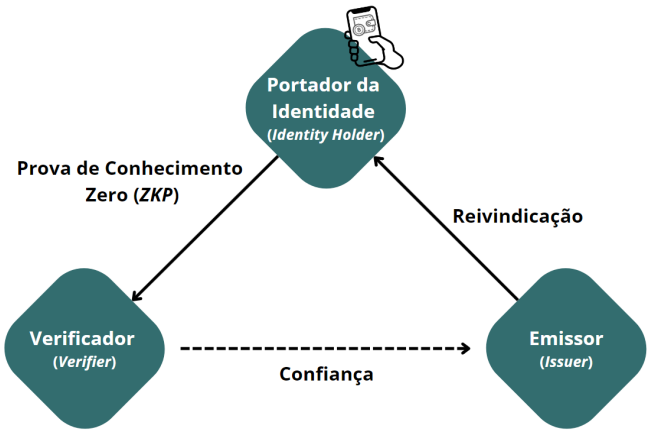

Figura 5.3 – Triângulo da Confiança (adaptado de [16]).

5.2.3 Provas de Conhecimento Zero (*ZKP*) e a *Privado ID*

A *Privado ID* emprega um esquema de Provas de Conhecimento Zero (do inglês, *Zero-Knowledge Proof*), conhecido como *zk-SNARKs* (*Zero-Knowledge Succinct Non-Interactive Argument of Knowledge*), para garantir a privacidade e a segurança das informações de identidade em um ambiente descentralizado. Este método é destacado por sua eficiência em ambientes de *blockchain*, conforme explorado por [25], e é particularmente adequado para o sistema de identidade descentralizada devido às suas características de não interatividade, compactação de provas e eficiência de verificação.

O *zk-SNARK* permite que um usuário (*prover*) demonstre a posse de certas informações sem revelar esses dados ao verificador (*verifier*), garantindo assim a privacidade do usuário. O processo é dividido em quatro fases principais:

1. ***Setup*:** Criação de parâmetros públicos a partir de um segredo inicial que é destruído após esta etapa para garantir a segurança.
2. **Geração de Provas:** O usuário (*prover*) gera uma prova de que possui certas informações, utilizando os parâmetros gerados na fase de setup.
3. **Verificação:** O verificador (*verifier*) pode confirmar a validade da prova usando apenas os parâmetros públicos, sem necessidade de interagir com o *prover*.
4. **Compactação:** As provas geradas são sucintas, facilitando sua transmissão e verificação sem comprometer a segurança.

A escolha por *zk-SNARKs* na *Privado ID* deve-se à sua capacidade de fornecer provas eficientes e não interativas que são essenciais em ambientes de *blockchain*, onde cada transação ou verificação deve ser concluída de maneira rápida e segura [25]. Este método de *ZKP* permite verificações de credenciais sem comprometer informações pessoais sensíveis, e posicionando a *Privado ID* como uma ferramenta confiável no ecossistema de *blockchain*.

5.2.4 Digital Wallet

A *digital wallet* desempenha um papel crucial na gestão das Credenciais Verificáveis (*VCs*) dentro do ecossistema da *Privado ID*. Além de armazenar a chave privada do usuário e buscar as *VCs* emitidas pelo Emissor, a carteira digital é essencial na aplicação das Provas de Conhecimento Zero (*ZPK*). A carteira digital gera provas que demonstram a posse de uma credencial sem revelar informações sensíveis diretamente ao Verificador. Este processo assegura que a identidade e os dados do Detentor sejam protegidos, mantendo a confidencialidade e a integridade das informações ao longo de todas as transações.

A *digital wallet* faz uso da tecnologia *zk-SNARKs*. Isto significa que a carteira digital utiliza as *zk-SNARKs* para gerar provas de conhecimento zero. A *digital wallet* então, permite que os usuários interajam de forma segura e privada dentro do ecossistema *blockchain*, confirmando que possuem certas credenciais ou direitos, sem expor informações sensíveis. Para acessar e utilizar a *Privado ID*, os usuários precisam baixar o aplicativo correspondente, criar uma carteira digital e configurar um número *PIN*.

6. VALIDAÇÃO E AVALIAÇÃO DA ARQUITETURA PROPOSTA

6.1 Validação da Arquitetura

Para validar a arquitetura proposta, foi necessário configurar um ambiente de testes que reproduzisse as condições esperadas de operação do sistema. Dados sintéticos foram gerados para simular o fluxo de informações entre os componentes da arquitetura, permitindo a análise da funcionalidade e do desempenho da solução.

6.1.1 *Dataset* Utilizado

O cenário de teste envolveu o uso de dados gerados por um veículo, coletados com o auxílio de um sistema de injeção eletrônica de combustível programável. No total, oito tipos de dados foram coletados em uma amostra de aproximadamente 40 minutos. Para os testes, foram utilizados os primeiros 10 minutos, em uma frequência de 200Hz. Os dados utilizados incluem RPM do motor, Distância Percorrida, Sensor de Pressão Absoluta do Coletor de Admissão (do inglês, *Manifold Absolute Pressure* (*MAP*)), Temperatura do Motor, Velocidade, Bateria, Marca e Freio Ativo, totalizando 374.248 registros de dados na taxa de amostragem, que foram inseridos na arquitetura e utilizados para os testes.

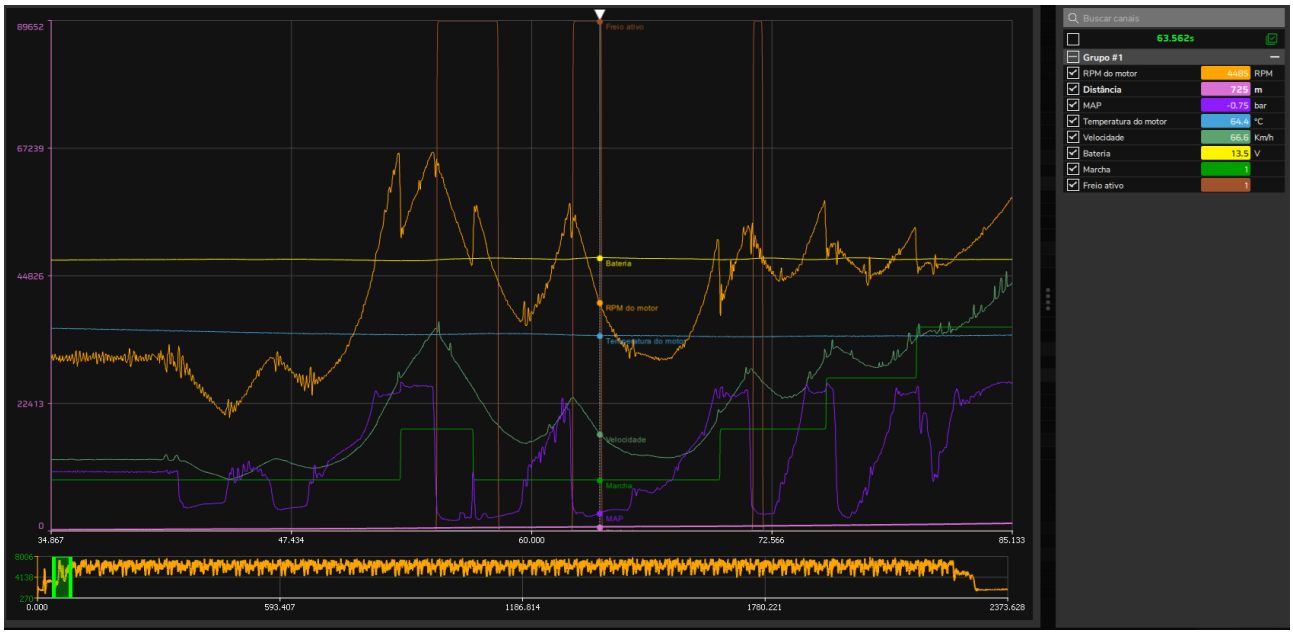

Figura 6.1 – Gráfico das amostras coletadas em determinado intervalo de tempo

6.1.2 Implementação da Arquitetura

A implementação da arquitetura para validação foi realizada através da rede de testes *Polygon Amoy* [2]. Com o objetivo de acelerar o processo de desenvolvimento e simplificar a interação com a testnet da *Polygon*, foi escolhido o *Alchemy* [3], um provedor de nós de *blockchain*. Os testes foram conduzidos em um ambiente controlado, utilizando dados sintéticos representativos do setor automotivo.

O armazenamento de dados foi realizado no *IPFS/IPNS*, garantindo descentralização, integridade e persistência das informações. Para evitar a necessidade de atualizações frequentes nos contratos inteligentes a cada novo dado registrado, o *IPNS* foi utilizado para manter uma referência fixa que aponta para os dados mais recentes no *IPFS*, reduzindo custos transacionais e melhorando a escalabilidade da solução.

Figura 6.2 – Primeiras 10 linhas do dataset coletado a partir do comando no terminal, utilizando o *CID*.

Na Figura 6.2, a execução do comando exibe as primeiras 10 linhas do conjunto de dados armazenado no IPFS. A listagem confirma que os dados podem ser recuperados de forma confiável através do *CID*, garantindo a integridade e acessibilidade da informação.

Já na Figura 6.3, a estrutura de Grafo Acíclico Dirigido (do inglês, *Directed Acyclic Graph* (DAG)) do *IPFS* é evidenciada pela fragmentação do arquivo em múltiplos blocos, representados pelos links listados na interface. O *hash* da raiz aponta para os demais blocos que compõem o arquivo.

Além do armazenamento descentralizado, a arquitetura incorporou o modelo *C2D* do *Ocean Protocol*, permitindo o processamento seguro dos dados sem a necessidade de compartilhamento direto. Nesse modelo, os interessados enviam algoritmos que são executados nos dados armazenados, recebendo apenas os resultados das análises, sem expor as informações brutas dos veículos. Isso garante maior privacidade e controle sobre os dados, permitindo sua monetização sem comprometer a segurança dos usuários.

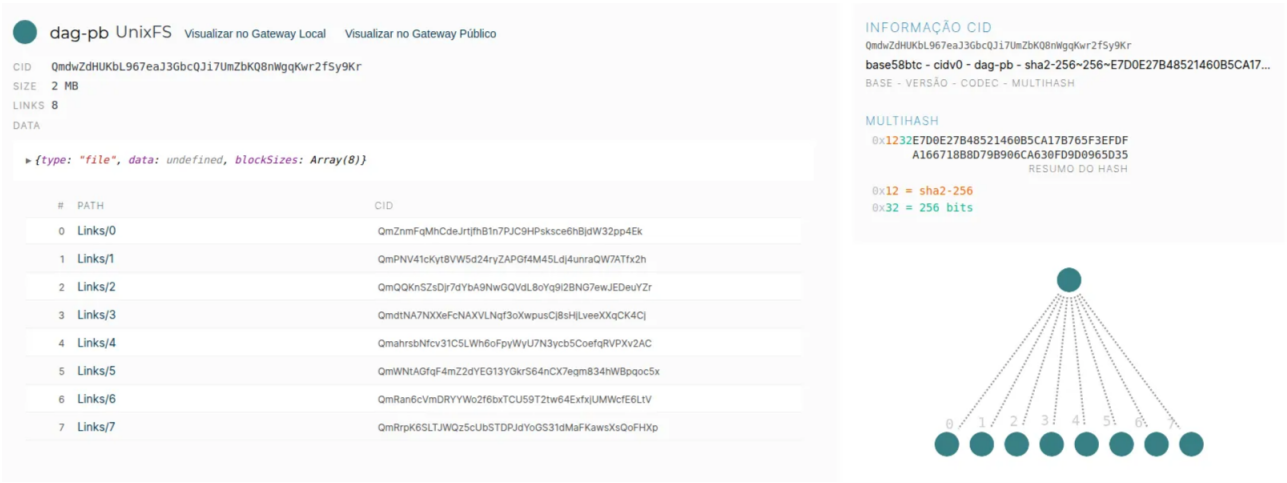

Figura 6.3 – Informações coletadas a partir do *CID*.

Na Figura 6.4, observa-se a interface do *Ocean Market*, onde o conjunto de dados armazenado no *IPFS* foi publicado para comercialização. Os metadados incluem informações sobre o autor, proprietário e os atributos do *dataset*, permitindo que terceiros interessados adquiram acesso ao conjunto de dados de forma segura. A precificação do ativo é definida, assim como um prazo de acesso, onde os dados podem ser analisados sem necessidade de compartilhamento direto.

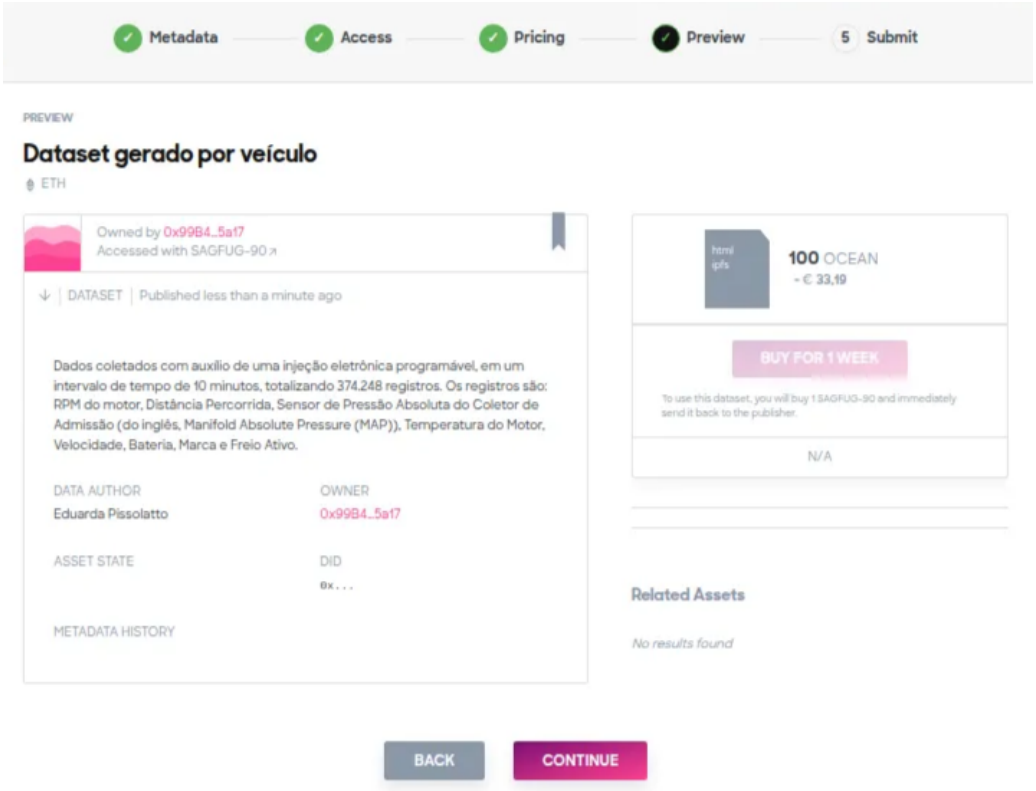

Figura 6.4 – Publicando os dados para venda no *Ocean Markert*.

O modelo *C2D* permite a execução de algoritmos diretamente onde os dados estão armazenados, sem que esses dados precisem ser transferidos para terceiros. Esse

conceito garante maior segurança e privacidade, sendo amplamente utilizado para análise de dados sensíveis.

O desenvolvimento de algoritmos para *C2D* envolve três etapas principais:

1. Definição dos dados de entrada e seu armazenamento em um ambiente seguro.
2. Implementação do algoritmo que será executado dentro do ambiente computacional isolado.
3. Publicação do resultado do processamento de volta ao repositório seguro.

Um esqueleto básico de um algoritmo para *C2D*, que realiza um processamento simples de dados, como média aritmética, pode ser observado no Apêndice B.

Basicamente, o ambiente *C2D* executa o script, nesse caso em *Python* em um contêiner isolado. O resultado do processamento é publicado novamente no repositório descentralizado, retornando um novo *CID* para o usuário. Após a execução, o ambiente é automaticamente descartado, garantindo que nenhum dado residual permaneça, garantindo um bom nível de segurança e controle sobre os dados, permitindo a comercialização de insights sem comprometer a privacidade do proprietário original dos dados.

O funcionamento do *C2D* foi simulado utilizando uma estrutura baseada em *Docker* para garantir a execução isolada dos algoritmos e a interação segura com os dados armazenados no *IPFS/IPNS*. Essa abordagem permitiu que os dados fossem processados sem que precisassem ser diretamente compartilhados com terceiros, garantindo privacidade e segurança no modelo de monetização de dados.

A implementação foi dividida em três principais componentes dentro do ambiente *Docker*:

- **Definição do ambiente *Docker*:** Para padronizar a execução dos experimentos, foi criado um `Dockerfile` que configura um contêiner baseado em `Python 3.9 slim`, garantindo compatibilidade com os *scripts* utilizados. Além disso, o *IPFS* foi instalado dentro do contêiner para permitir a publicação e recuperação dos dados descentralizados. Durante a inicialização do contêiner, o *daemon* do *IPFS* é iniciado, possibilitando a interação com a rede. O Dockerfile pode ser observado no Apêndice A.
- **Gerenciamento de dependências:** Para evitar problemas de compatibilidade e facilitar a reprodução do experimento, as bibliotecas necessárias foram definidas em um arquivo `requirements.txt`, garantindo que todas as dependências fossem instaladas automaticamente durante a construção da imagem *Docker*. Esse método evita que pacotes sejam instalados manualmente dentro do código *Python*, otimizando a execução e reduzindo erros. Por exemplo, o arquivo `requirements.txt` teve como

conteúdo a declaração da biblioteca *Numpy*, seguida de sua versão. Dessa forma, o conteúdo do arquivo corresponde a `numpy==1.23.0`.

- **Execução do algoritmo e manipulação de dados via *IPFS*:** O *script* principal (`script.py`) foi responsável por toda a interação com os dados armazenados no *IPFS/IPNS*. Ele implementou funções para baixar *datasets* e algoritmos publicados no IPFS, executar a análise sobre os dados e publicar os resultados de volta na rede descentralizada. Para isso, foram utilizadas chamadas diretas ao *CLI* do *IPFS* via subprocessos *Python*. Esse fluxo permite que diferentes usuários ou entidades pudessem processar os dados sem que fosse necessário acessá-los diretamente. O código pode ser observado no Apêndice B.

O processo completo de execução do *Compute-to-Data* seguiu as seguintes etapas:

1. O usuário publica um *dataset* no *IPFS* e disponibiliza o *CID*.
2. O interessado, dentro do ambiente *Docker*, recupera os dados utilizando o *CID* e baixa o algoritmo de processamento.
3. O algoritmo é executado dentro do contêiner, garantindo o isolamento do ambiente e evitando exposição dos dados.
4. Após o processamento, os resultados são publicados novamente no *IPFS*, retornando um novo *CID* ao usuário.
5. O contêiner é finalizado e removido automaticamente após a execução, evitando armazenamento desnecessário de dados e garantindo que o ambiente sempre seja inicializado em um estado limpo.

A implementação da *dNFT* foi baseada na combinação de contratos inteligentes na *blockchain Polygon*, armazenamento de metadados no *IPFS/IPNS* e integração com *Chainlink VRF* [7]. Essa abordagem foi adaptada ao contexto automotivo, permitindo o registro descentralizado da quilometragem e outras informações relevantes dos veículos.

No modelo proposto, cada veículo recebe um *dNFT* único, contendo dados como o modelo do automóvel, o nome do proprietário, o semestre atual do ano e a quilometragem registrada. No início de cada novo semestre, um novo arquivo *JSON* é gerado no *IPFS* contendo a quilometragem atualizada do veículo, garantindo que o *dNFT* reflita sempre os dados mais recentes semestralmente. Esse mecanismo reduz o custo de execução de transações na *blockchain*, otimizando o armazenamento e a escalabilidade do sistema.

A implementação do contrato foi realizada em *Solidity*, seguindo o padrão ERC-721, com suporte para atualizações dinâmicas dos metadados, que pode ser observado

no Apêndice C. Já a observação da *dNFT* foi realizada com auxilio da ferramenta *OpenSea* [31]. A Figura 6.5 ilustra a atualização de um *dNFT* de veículo em dois períodos distintos. No exemplo apresentado, o *dNFT* do veículo foi inicialmente registrado no primeiro semestre de 2024, contendo os dados do modelo, do proprietário e a quilometragem registrada naquele momento. No início do segundo semestre de 2024, um novo registro foi gerado, contendo a quilometragem atualizada do veículo.

Figura 6.5 – Exemplo de *dNFT* gerado a partir do veículo de teste.

A autenticação descentralizada dos proprietários de veículos foi realizada utilizando *Privado ID*, garantindo a verificação de identidade e a posse do ativo digital sem expor dados sensíveis. A implementação foi conduzida com o SDK `iden3-js` [1], que permite a criação de Identidades Descentralizadas e a emissão de Credenciais Verificáveis utilizando provas de conhecimento zero. Essas credenciais foram utilizadas para autenticar transações relacionadas ao *dNFT*, como a transferência de propriedade.

A implementação seguiu três etapas principais: criação de identidade do proprietário do veículo, emissão da credencial vinculada ao veículo e prova de posse do veículo e autenticação segura. A Tabela 6.1 apresenta as funções utilizadas para essa integração.

A criação do *DID* para o proprietário foi realizada utilizando o módulo `@iden3/js-sdk`, que gera chaves criptográficas e define um identificador único para o usuário. Com a identidade descentralizada criada, foi possível emitir uma credencial vinculada ao veículo, registrando dados essenciais de forma segura e verificável. Para evitar exposição desnecessária dos dados do usuário, utilizou-se *ZKP* para provar a posse do veículo sem revelar informações confidenciais. Por fim, a autenticação do proprietário foi implementada utilizando a função `handleAuthRequest`, garantindo que apenas o dono le-

Tabela 6.1 – Funções do *Privado ID* utilizadas na integração com *dNFT* [1].

Funções do SDK *Privado ID*	
Função	**Descrição**
identityCreation	Gera um *DID* único para cada proprietário, assegurando a identidade descentralizada.
issueCredential	Emite uma Credencial Verificável associada ao veículo e ao seu proprietário. Essa credencial é assinada e armazenada *on-chain*.
generateProofs	Cria provas de conhecimento zero para que um usuário prove a posse do veículo sem revelar seus dados.
handleAuthRequest	Permite autenticar o usuário no momento da transferência de propriedade do *dNFT*, garantindo que apenas o dono legítimo possa realizar a operação.

gítimo pudesse executar operações críticas sobre o *dNFT*, como a transferência para um novo proprietário, por exemplo.

6.1.3 Avaliação de Custos

A avaliação das taxas de transação para o *eDNA*, presente na tabela 6.2[1] , foi realizada utilizando o *Remix Ethereum IDE* [10] e *MetaMask* [27] para interação com os contratos inteligentes, aproveitando a compatibilidade do *IDE* com a rede *Polygon*. Na rede *Polygon*, taxas são aplicadas apenas a operações que modificam valores em um contrato inteligente ou realizam transferências de fundos entre contas. Operações de leitura, por outro lado, não incorrem em custos e são executadas gratuitamente.

CreateVehicle é a função para criar um Contrato Inteligente de Frota de Veículos, construída com dois parâmetros: a chave pública *IPNS* e o ID do dispositivo; e *SetFleetVehicleList* que configura a chave pública *IPNS* que apontará para a lista de veículos armazenada no *IPFS*. Embora a arquitetura *eDNA* tenha a capacidade de acessar todos os dados gerados na rede em detalhes, o sistema implementado para obter os resultados, apresentados nesta seção, foi projetado especificamente para consultar dois tipos de dados: "dados do veículo", que engloba todas as informações geradas por um veículo, e "dados do dispositivo", que se refere aos dados transmitidos por um dispositivo *IoT* específico.

[1] Na data de 12.01.2025, 1 POL equivale a aproximadamente R$ 2,8

Ambos os tipos de dados podem ser solicitados usando a função *RequestDeviceData*, passando como parâmetro o *ID* do dispositivo, e a função *RequestVehicleData*, que não requer parâmetros. *VehicleDataNFT* é um contrato inteligente que implementa um padrão de *NFT* na *blockchain*. Ele permite associar informações específicas do veículo, neste caso de teste foram utilizados os momentos em que o seguro foi ativado e quilômetros percorridos no último semestre.

Considerando que a arquitetura *eDNA*, do lado do proprietário do veículo, requer apenas uma atualização nos seus contratos inteligentes, as taxas de transação permanecerão constantes mesmo se mais dispositivos, veículos ou nós solicitantes se juntarem ao sistema.

Tabela 6.2 – Taxas de Transação das Funções dos Contratos Inteligentes.

Contrato Inteligente *Vehicle*		
Função	**Gás (Unidades)**	**Preço em *POL***
RequestDeviceData()	224911	0,00056
RequestVehicleData()	720794	0,00179
VehicleDatadNFT()	1534400	0,15344
Contrato Inteligente *Vehicles Fleet*		
Função	**Gás (Unidades)**	**Preço em *POL***
CreateVehicle()	579813	0,00144
SetFleetVehicleList()	458334	0,00114

Para o registro de domínio e sua manutenção na rede *Privado ID*, são aplicadas taxas específicas tanto para o registro inicial quanto para a renovação subsequente. No caso de um domínio com a extensão `.id`, a taxa de registro é de 0,02 POL, enquanto a taxa de renovação anual é definida em 0,01 POL. É importante observar que, além dessas taxas, cada transação na rede incorre em uma taxa adicional de gás, que é determinada pela carga atual da rede *Polygon*.

Por exemplo, ao registrar um domínio `.id` por um período de um ano, o custo imediato seria a taxa de registro de 0,02 POL. No entanto, deve-se considerar também a taxa de gás associada à transação, que, embora variável, geralmente é inferior a 0,01 POL em condições normais de rede. Assim, o custo total estimado para registrar um domínio `.id` por um ano geralmente não ultrapassa 0,03 POL. Da mesma forma, para a renovação anual de um domínio `.id`, a taxa necessária é de 0,01 POL, mais a taxa de gás necessária para executar a transação.

Mais uma vez, se a taxa de gás permanecer abaixo de 0,01 POL, o custo total para renovação não deve ultrapassar 0,02 POL. Esse modelo de custo garante que o processo de registro e renovação de domínios na rede *Polygon* seja acessível, ao mesmo tempo que se adapta às flutuações na demanda da rede, refletidas nas taxas de gás.

Além das taxas associadas ao registro e renovação de domínios na *Privado ID*, o modelo proposto no *eDNA* também envolve operações essenciais para a atualização e transferência de ativos digitais vinculados à identidade descentralizada. A emissão de *VCs* permite associar informações relevantes ao proprietário e ao veículo, enquanto a utilização de *ZKP* garante autenticação segura sem exposição desnecessária de dados sensíveis.

Cada uma dessas operações requer uma transação na rede *Polygon*, incorrendo em taxas de gás proporcionais à complexidade computacional do processo. A Tabela 6.3 apresenta uma visão geral das taxas de transação, refletindo os custos envolvidos na administração de identidades descentralizadas e ativos digitais dinâmicos.

Tabela 6.3 – Estimativa de Custos de Transação com *Privado ID*.

Operação	**Custo (POL)**
Registro de domínio *.id*	0,02
Renovação anual do domínio *.id*	0,01
Emissão de Credencial Verificável (*VC*)	0,015
Geração de Prova *ZKP* para autenticação	0,012
Transferência de *dNFT* para novo proprietário	0,025
Atualização de metadados do *dNFT*	0,018

Em cenários de uso real, espera-se que um veículo tenha múltiplas atualizações ao longo de seu ciclo de vida, como mudanças de quilometragem, novas credenciais de manutenção e transferências de propriedade. Considerando uma atualização semestral do *dNFT* e uma transferência de proprietário a cada dois anos, o custo médio anual estimado para a manutenção da Identidade Descentralizada e atualização dos metadados do *dNFT* é de aproximadamente 0,06 POL. Esse valor pode variar conforme a demanda da rede.

Sendo assim, as taxas de transação foram avaliadas, especialmente aquelas relacionadas ao registro e renovação de domínios na rede *Privado ID*, garantindo acessibilidade e adaptabilidade às flutuações na demanda da rede. Esse processo de implementação representa um passo significativo na criação de um *framework* robusto e eficiente para a arquitetura *eDNA*, permitindo acesso seguro e manipulação dos dados gerados na rede.

6.2 Avaliação de Desempenho da Arquitetura

Esta seção apresenta a metodologia e os resultados obtidos na avaliação de desempenho da arquitetura proposta. O objetivo é analisar latência, escalabilidade e eficiência computacional, garantindo que a solução seja viável e eficiente.

6.2.1 Metodologia de Avaliação

A avaliação foi realizada em um ambiente de testes configurado em um computador pessoal, equipado com um processador Intel Core i7-11800H @ 2.30GHz e 16GB de RAM. Os dados foram armazenados no IPFS/IPNS, enquanto a rede *blockchain* de testes *Polygon Amoy* foi utilizada para garantir escalabilidade e baixos custos transacionais. Os testes utilizaram dados sintéticos representativos do setor automotivo, coletados de um sistema de injeção eletrônica de combustível programável, com frequência de 200Hz durante 10 minutos, totalizando 374.248 registros.

6.2.2 Resultados

A seguir, são apresentados os resultados de três testes realizados com o objetivo de avaliar a eficácia e a viabilidade da solução proposta, baseada na arquitetura *eDNA*, em diferentes aspectos: latência, escalabilidade e eficiência computacional.

6.2.2.1 Teste de Latência

Com o intuito de avaliar a eficiência da solução proposta, testes de latência foram aplicados para mensurar os tempos de armazenamento e recuperação de dados na *blockchain*. Esses testes foram conduzidos na *Polygon Amoy Testnet*, simulando interações reais com contratos inteligentes dentro do contexto de compartilhamento seguro de informações.

Para a mensuração da latência, um conjunto de cinquenta interações consecutivas foi implementado, onde cada execução consistiu em duas etapas principais. A primeira etapa envolveu o armazenamento do dado na *blockchain*, onde informações são gravadas para garantir sua imutabilidade. A segunda etapa consistiu na recuperação do dado previamente armazenado, validando a eficiência do acesso às informações registradas.

A medição dos tempos de execução foi realizada diretamente no ambiente de testes, registrando o tempo necessário para cada transação e compilando os resultados para análise posterior. Após a execução de todas as interações, os dados coletados foram utilizados para a construção de gráficos que evidenciam a variação da latência ao longo das execuções.

Os resultados demonstram que a latência de armazenamento na *blockchain* é diretamente influenciada pelo tempo de validação das transações, gerando variações em redes públicas e descentralizadas. Apesar disso, a recuperação dos dados apresentou tempos mais consistentes, reforçando a viabilidade do uso da *blockchain* para leitura eficiente de informações.

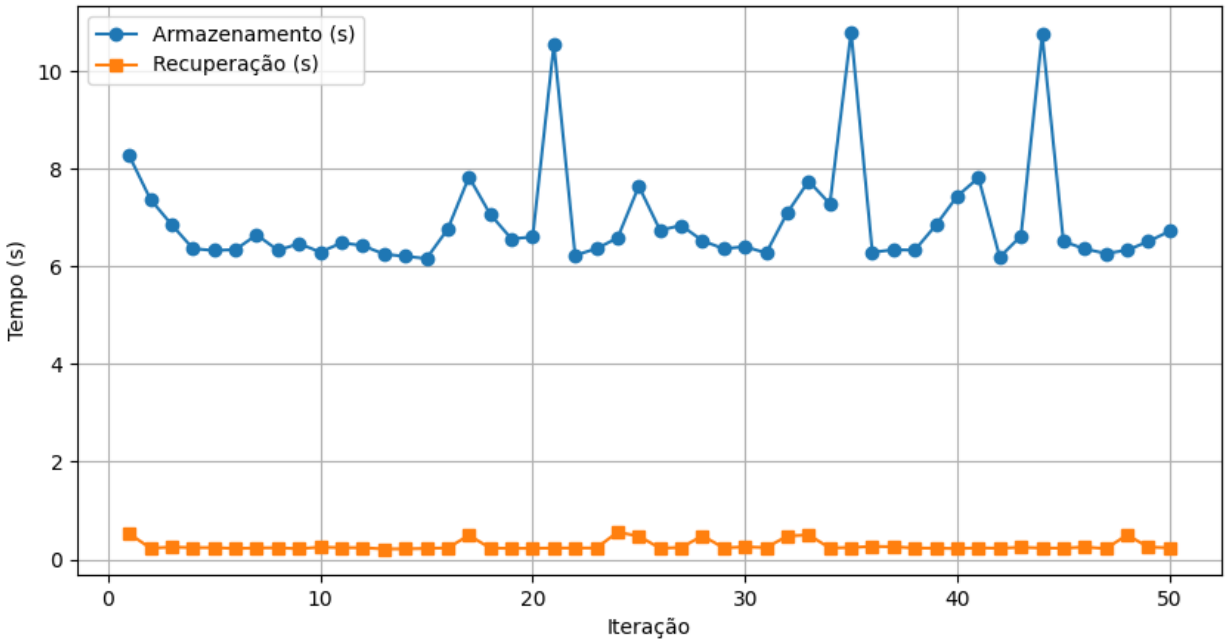

Figura 6.6 – Latência de armazenamento e recuperação na *blockchain*.

A análise da Figura 6.6 revela algumas tendências significativas. Observa-se que o tempo de armazenamento apresenta oscilações, com picos em determinadas iterações. Isso sugere que a latência da escrita pode ser influenciada por fatores como carga da rede, congestionamento momentâneo e tempo de confirmação das transações pelos validadores. Em contrapartida, o tempo de recuperação se mantém relativamente estável e significativamente inferior ao tempo de armazenamento, permanecendo abaixo de 1 segundo em todas as execuções. Esse comportamento reforça que a leitura de dados na *blockchain* é uma operação mais eficiente, uma vez que não depende do processo de mineração ou validação por múltiplos nós.

Os picos observados no armazenamento podem estar associados a períodos de maior demanda na rede, flutuações no preço do gás ou a presença de um número elevado de transações sendo processadas simultaneamente. Apesar dessas variações, a análise

geral indica que a rede apresenta um desempenho previsível na maioria das execuções, garantindo uma latência de armazenamento dentro de um intervalo relativamente constante. Assim, embora o tempo de escrita na *blockchain* seja mais elevado quando comparado a soluções centralizadas, o tempo de leitura se mantém competitivo, tornando a tecnologia viável para cenários que exigem alta disponibilidade e segurança das informações armazenadas.

6.2.2.2 Teste de Escalabilidade

Com o objetivo de avaliar a capacidade de escalabilidade da solução proposta, foram realizados testes para mensurar o desempenho da *blockchain* em termos de tempo de confirmação de transações sob alta carga.

Para a realização do teste, foi implementado um conjunto de cem transações consecutivas, sendo cada transação registrada na *blockchain* e monitorada para medir seu tempo de confirmação. Cada execução consistiu em duas etapas principais: a primeira etapa envolveu o envio de dados para a *blockchain*, onde as informações foram armazenadas de forma imutável. A segunda etapa consistiu na medição do tempo necessário para que cada transação fosse confirmada e registrada de forma permanente.

Os tempos de execução foram medidos diretamente durante o teste, registrando o tempo necessário para cada transação e compilando os resultados para análise posterior. Após a execução das transações, os dados obtidos foram utilizados para a construção de gráficos que demonstram a variação do tempo de confirmação ao longo das transações.

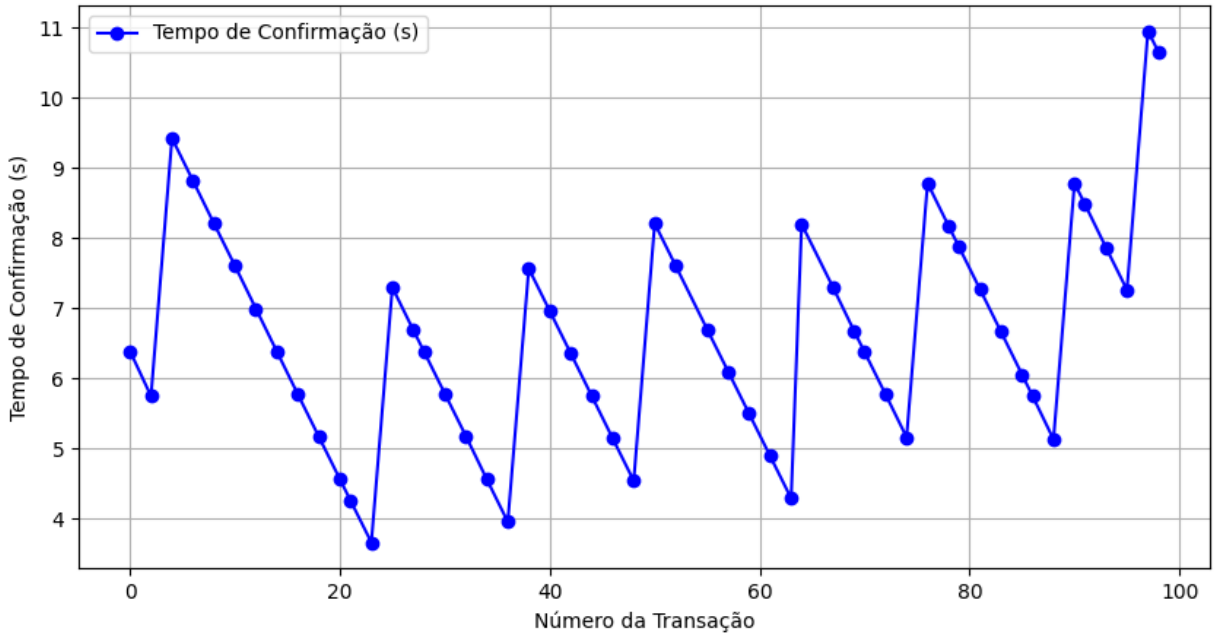

Figura 6.7 – Teste de Escalabilidade: Tempo de Confirmação por Transação

A análise da Figura 6.7 revela algumas tendências significativas. O tempo de confirmação apresenta picos em determinadas transações, o que sugere que o desempenho da rede pode ser influenciado por fatores como a carga momentânea da rede e a priorização do gás para determinadas transações.

Apesar das hipóteses levantadas, não foi possível identificar com precisão os fatores determinantes para esses picos, o que indica a necessidade de investigações futuras em ambientes controlados ou com o apoio de ferramentas de monitoramento de rede mais robustas.

Os picos observados podem estar relacionados a períodos de alta demanda, congestionamentos momentâneos ou flutuações nas taxas de gás. Ainda assim, a análise geral aponta que a rede apresentou boa capacidade de escalabilidade dentro do cenário de transações simuladas.

6.2.2.3 Teste de Eficiência Computacional

O principal objetivo desse teste foi avaliar a eficiência e a privacidade proporcionadas pela execução de operações sobre dados armazenados no *IPFS*, sem a necessidade de acessar ou expor os dados diretamente.

Durante o teste, todo o *dataset* foi processado. Os cálculos executados incluíram medições como a temperatura do motor, distância percorrida, velocidade, e outros parâmetros relacionados, contidos no *dataset*. O tempo total de execução foi de 607.081ms, o que indicou um tempo de processamento eficiente para o volume de dados analisado.

```
duda@VM-Duda:~/blockchain-tests$ node compute_to_data.js
Inicializando o IPFS...
Acessando dados do CID...
Dados lidos com sucesso!
Total de linhas a processar (incluindo cabeçalho): 46782
Exibindo a cada 5000 linhas...
Processando a linha 2: Marcha = N, Distância = 0, Bateria = 9.6, RPM = 270, MAP = -0.051, Temperatura = 65.1, Velocidade = 0, Freio ativo = 0
Processando a linha 5002: Marcha = 1, Distância = 699, Bateria = 13.5, RPM = 5943, MAP = -0.746, Temperatura = 64.6, Velocidade = 88, Freio ativo = 1
Processando a linha 10002: Marcha = 3, Distância = 2853.5, Bateria = 13.5, RPM = 5737, MAP = 0.599, Temperatura = 67.9, Velocidade = 131, Freio ativo = 0
Processando a linha 15002: Marcha = 2, Distância = 5254.2, Bateria = 13.6, RPM = 4964, MAP = -0.507, Temperatura = 70.7, Velocidade = 94, Freio ativo = 0
Processando a linha 20002: Marcha = 3, Distância = 7648.5, Bateria = 13.6, RPM = 6154, MAP = -0.782, Temperatura = 72.8, Velocidade = 139, Freio ativo = 0
Processando a linha 25002: Marcha = 3, Distância = 10085.6, Bateria = 13.5, RPM = 5919, MAP = 0.515, Temperatura = 74.3, Velocidade = 128, Freio ativo = 0
Processando a linha 30002: Marcha = 4, Distância = 12534.4, Bateria = 13.5, RPM = 6527, MAP = 1.044, Temperatura = 74.9, Velocidade = 169, Freio ativo = 0
Processando a linha 35002: Marcha = 5, Distância = 15077.9, Bateria = 13.5, RPM = 7199, MAP = 1.178, Temperatura = 75.8, Velocidade = 223, Freio ativo = 0
Processando a linha 40002: Marcha = 2, Distância = 17536.5, Bateria = 13.5, RPM = 4578, MAP = -0.703, Temperatura = 76.5, Velocidade = 88, Freio ativo = 0
Processando a linha 45002: Marcha = 3, Distância = 19927.7, Bateria = 13.5, RPM = 6711, MAP = 1.021, Temperatura = 75.6, Velocidade = 147, Freio ativo = 0
Média da Temperatura do Motor: 72.24 °C
Média da Velocidade: 132.48 km/h
Média da Distância: 9468.02 km
Média da Bateria: 13.45 %
Média do RPM do Motor: 5858.41
Média do MAP: 0.35
Execução IPFS: 607.081ms
duda@VM-Duda:~/blockchain-tests$
```

Figura 6.8 – Teste de eficiência computacional no *C2D*

A Figura 6.8 revela que o tempo de execução foi diretamente influenciado pela taxa de recuperação dos dados do *IPFS*. Esse tempo de recuperação dos dados depende de fatores como a carga da rede *IPFS* e o tamanho do arquivo armazenado. No entanto, como o processo de execução ocorreu sem exposição dos dados sensíveis, a privacidade das informações foi completamente preservada.

As medições realizadas mostram que o processo de execução de algoritmos sobre dados armazenados de forma segura, utilizando *C2D*, pode ser tão eficiente quanto a análise de dados tradicionais, mesmo em uma rede descentralizada, onde os dados são mantidos imutáveis e privados.

7. CONSIDERAÇÕES FINAIS

7.1 Resumo Geral

Este trabalho apresentou a arquitetura *eDNA*, uma proposta voltada para a criação de um marketplace descentralizado de dados automotivos. Utilizando tecnologias como *blockchain*, *IoT*, *IPFS/IPNS* e *dNFTs*, o objetivo foi enfrentar as limitações dos sistemas centralizados tradicionais, que frequentemente sofrem com vulnerabilidades de segurança, pontos únicos de falha e falta de escalabilidade. A arquitetura foi projetada para oferecer um modelo descentralizado, seguro e escalável, promovendo a inclusão dos proprietários de veículos no mercado de dados automotivos.

A estrutura da *eDNA* foi desenvolvida sobre a rede *blockchain Polygon,* reconhecida por sua alta escalabilidade e custos reduzidos de transação. A solução combina coleta de dados em tempo real por sensores *IoT*, armazenamento descentralizado e imutável no *IPFS/IPNS* e encapsulamento de informações dinâmicas em *dNFTs*. Além disso, utiliza o modelo *Compute-to-Data (C2D)* para possibilitar análises diretamente nos dados armazenados, eliminando a necessidade de compartilhá-los com terceiros e garantindo a privacidade do proprietário. Essa abordagem resolve problemas comuns em arquiteturas centralizadas e até mesmo em algumas soluções descentralizadas, como a dificuldade de atualizar informações de forma eficiente e segura.

Durante a implementação, o sistema foi avaliado utilizando dados sintéticos, simulando cenários reais. Os testes demonstraram que as taxas de transação permanecem consistentemente acessíveis, mesmo em operações frequentes, comprovando a viabilidade do modelo em larga escala. A integração do *IPFS* com o *IPNS* mostrou-se essencial para superar limitações do armazenamento descentralizado, permitindo a atualização eficiente de registros sem a necessidade de constantes interações com a *blockchain*, o que reduz custos e complexidade operacional.

Os resultados obtidos confirmam que a arquitetura *eDNA* oferece uma solução robusta para os desafios do setor automotivo, estabelecendo um novo padrão para a gestão de dados de veículos. Além de proporcionar segurança e eficiência, a arquitetura permite que os proprietários de veículos monetizem seus dados de maneira transparente, promovendo uma redistribuição mais justa do valor gerado por essas informações. A pesquisa também demonstrou que a integração de tecnologias descentralizadas é uma abordagem viável para superar as limitações dos sistemas centralizados, enquanto garante a escalabilidade necessária para atender à crescente demanda por soluções digitais no setor automotivo.

Em síntese, a arquitetura *eDNA* não apenas alcançou os objetivos propostos, como também se revelou uma solução versátil, com potencial para ser aplicada em outras indústrias que demandem segurança, privacidade e eficiência na gestão e comercialização de dados *IoT*.

7.2 Revisitando a proposta de pesquisa

Esta seção revisita os problemas, hipóteses e questões de pesquisa apresentados no início deste documento. Também estabelece um paralelo entre o que foi proposto e o que foi alcançado.

7.2.1 Problemas e Objetivos de Pesquisa

P.1. **Como o veículo deve ser monitorado através de sensores *IoT* no ambiente cotidiano:** A arquitetura eDNA possibilita uma solução robusta para o monitoramento contínuo e em tempo real de veículos. A Camada de Borda, abordada na Figura 4.1, coleta dados por meio de sensores presentes nos veículos. Embora a pesquisa tenha utilizado dados sintéticos para testes, a estrutura é plenamente compatível com dados reais, permitindo sua implementação em ambientes automotivos reais.

P.2. **Como ter um registro das ocorrências relacionadas ao veículo de forma segura, transparente e inviolável:** O uso de blockchain no *eDNA*, em conjunto com o armazenamento descentralizado via *IPFS/IPNS*, permite que registros de dados sejam mantidos de forma imutável e auditável. Essa abordagem elimina vulnerabilidades associadas a sistemas centralizados.

P.3. **Como estes registros devem ser inseridos em modelos de dados enriquecidos, posteriormente disponíveis para a venda a interessados externos, bem como para o próprio proprietário do veículo:** Os registros devem ser integrados a modelos de dados enriquecidos para comercialização e análise. A arquitetura combina o conceito de *dNFT* e o modelo *Compute-to-Data (C2D)* para encapsular e processar os dados, permitindo sua monetização sem comprometer a privacidade do proprietário.

P.4. **A centralização tradicional dos sistemas tem se mostrado inadequada diante das demandas de segurança, escalabilidade e disponibilidade persistentes no ambiente interconectado. As limitações desses sistemas centralizados, como pontos únicos de falha, vulnerabilidades a ataques e difi-**

culdades em escalar proporcionalmente ao crescimento do número de dispositivos, evidenciam a necessidade de novas abordagens: Sistemas centralizados apresentam falhas em termos de segurança e escalabilidade, enquanto soluções *Web3* enfrentam desafios de adoção ampla. A *eDNA* supera os desafios de escalabilidade ao integrar a *blockchain* e o *IPNS*. A avaliação prática demonstrou a eficiência da abordagem, validando sua aplicação no contexto automotivo.

7.2.2 Hipóteses e Questões de Pesquisa

Na presente pesquisa, as hipóteses que foram validadas incluem:

H.1. A implementação de sensores *IoT* em veículos melhora significativamente a coleta de dados em tempo real, aumentando a eficiência na gestão de uso dos veículos: A arquitetura eDNA confirmou a validade dessa hipótese ao demonstrar que os sensores *IoT* instalados nos veículos permitem a coleta eficiente de dados em tempo real. Durante os testes, foi possível observar a captura de informações precisas como velocidade, RPM, nível de combustível e temperatura do motor, essenciais para uma gestão detalhada e eficaz do veículo.

H.2. Tecnologias de *blockchain* podem garantir um registro de ocorrências relacionadas a veículos que seja seguro, transparente e inviolável, minimizando fraudes e disputas: A pesquisa validou essa hipótese ao implementar o armazenamento de registros na rede *blockchain Polygon*, garantindo imutabilidade e transparência. Os testes confirmaram que os dados armazenados são protegidos contra alterações e acessíveis apenas por meio de contratos inteligentes, minimizando riscos de fraudes e disputas entre as partes interessadas.

H.3. A integração de dados de veículos coletados em modelos enriquecidos aumenta o valor comercial desses dados para partes externas, sem comprometer a privacidade do proprietário: Essa hipótese foi confirmada com a implementação do modelo *Compute-to-Data (C2D)*, que permite a execução de algoritmos diretamente nos dados armazenados, sem que estes sejam compartilhados com terceiros. Esse modelo viabiliza a comercialização segura e agrega valor aos dados sem expor informações sensíveis do proprietário.

H.4. Sistemas descentralizados oferecem melhor segurança, escalabilidade e disponibilidade para a gestão de dados em veículos, superando as limitações dos sistemas centralizados: A arquitetura *eDNA* demonstrou que sistemas descentralizados são mais seguros e escaláveis do que sistemas centralizados. O uso combinado de *blockchain*, *IPFS/IPNS* e inteligentes contratos garantiu que o sis-

tema mantivesse alta disponibilidade e integridade dos dados, mesmo em cenários de alta demanda simulados durante os testes.

As questões de pesquisa exploradas no presente trabalho incluem:

QP.1. De que maneira os sensores *IoT* podem ser utilizados na monitoração cotidiana de veículos de forma eficaz? A arquitetura eDNA mostrou que sensores *IoT* conectados à camada de borda são eficazes para coletar e transmitir dados em tempo real. Essa coleta contínua permite que o sistema capture informações críticas sobre o veículo, como condições de operação e histórico de uso, promovendo uma gestão mais eficiente, utilizando a tecnologias presentes no ramo automotivo, como o *OBD*.

QP.2. Como a tecnologia de *blockchain* pode ser aplicada para criar um sistema de registros veiculares que seja ao mesmo tempo seguro e acessível para as partes interessadas? A tecnologia *blockchain* foi aplicada para criar registros veiculares seguros e acessíveis, utilizando a rede *Polygon*. O uso de contratos inteligentes assegurou que as partes interessadas tivessem acesso controlado e auditável aos registros, promovendo confiança e transparência nas transações de dados.

QP.3. De que forma os dados coletados de veículos podem ser enriquecidos e preparados para venda, garantindo privacidade e agregando valor para o proprietário e para terceiros? O modelo *Compute-to-Data (C2D)* implementado na arquitetura *eDNA* possibilitou o enriquecimento de dados para venda, ao permitir que algoritmos analíticos fossem executados diretamente nos dados armazenados. Isso garantiu a privacidade do proprietário, pois os dados brutos nunca foram compartilhados, apenas os resultados processados foram disponibilizados para os interessados.

QP.4. Quais são as principais vantagens e desafios na adoção de sistemas descentralizados? A pesquisa identificou várias vantagens, incluindo maior segurança, transparência e escalabilidade, proporcionadas pela descentralização. Contudo, desafios como a complexidade de implementação e os custos iniciais de integração foram observados, além dos custos para manutenção dos dados no IPFS, no caso do sistema evoluir para um MVP. Estes custos são importantes, pois durante a realização da PoC não foi utilizada a opção de manutenção dos dados no IPFS, caso o servidor que armazena estes dados venha a ser excluído da rede. Estes custos podem ser de até 100 dólares por mês, considerando um espaço de armazenamento de 5TB (0,035 dólares para cada GB extra). Ferramentas como o Pinata [34] oferecem soluções de pinagem de dados. Estes custos podem ser reduzidos, no caso de utilização em escala real, através do envio de um conjunto de dados ao IPFS, ao invés do envio de dados individuais. Assim, poderia ser definido que os dados de um

veículo seriam enviados a cada x unidades de tempo, por exemplo a cada hora, dia ou semana.

7.3 Trabalhos Futuros

Embora este trabalho tenha apresentado avanços na integração de tecnologias descentralizadas para o setor automotivo, há oportunidades para expandir e aprofundar os resultados obtidos. Os principais direcionamentos para trabalhos futuros incluem:

7.3.1 Análise de Impacto Regulatório e de Conformidade

Como a privacidade de dados é um tema central, estudos futuros poderiam analisar a conformidade da arquitetura com regulamentações emergentes, como a Lei Geral de Proteção de Dados Pessoais (LGPD) no Brasil e regulamentos internacionais como o *GDPR*. Também desenvolver *frameworks* de monitoramento para assegurar a auditoria e a conformidade contínuas em ambientes descentralizados. Além disso, investigar abordagens de governança descentralizada para estabelecer padrões e protocolos para o uso ético e seguro de dados veiculares.

7.3.2 Validação e Testes em Ambientes Reais

Para consolidar os resultados deste trabalho, estudos futuros podem implementar e testar a arquitetura proposta em ambientes reais, como parcerias com montadoras para validação da arquitetura em veículos conectados, avaliando sua eficiência em cenários de uso prático. Realizar simulações em larga escala para medir a performance da arquitetura em redes de alta densidade, como frotas de veículos compartilhados. Por fim, realizar a avaliação da experiência do usuário ao interagir com aplicações descentralizadas, investigando barreiras de adoção e usabilidade.

Os direcionamentos propostos reforçam o compromisso com a inovação e a sustentabilidade, garantindo que a arquitetura desenvolvida permaneça relevante em um cenário tecnológico em constante evolução.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] 0xPolygonID. “Polygon ID JS SDK Examples”. Capturado em: https://github.com/0xPolygonID/js-sdk-examples, Dez 2024.

[2] Alchemy. “Launching Support for the Polygon PoS Amoy Testnet”. Capturado em: https://www.alchemy.com/blog/polygon-amoy-is-live, Dez 2024.

[3] Alchemy. “Alchemy Platform”. Capturado em: https://www.alchemy.com/, Jan 2025.

[4] Ali, M. S.; Vecchio, M.; Antonelli, F. “A blockchain-based framework for IoT data monetization services”, *The Computer Journal*, vol. 64–2, 2021, pp. 195–210.

[5] Besançon, L.; Silva, C. F. D.; Ghodous, P.; Gelas, J.-P. “A blockchain ontology for dApps development”, *IEEE Access*, vol. 10, 2022, pp. 49905–49933.

[6] Chainlink. “Chainlink”. Capturado em: https://chain.link, Dez 2023.

[7] Chainlink. “Chainlink VRF Subscription Manager – Polygon Amoy”. Capturado em: https://vrf.chain.link/polygon-amoy, Dez 2024.

[8] Choi, S. S.; Burm, J. W.; Sung, W.; Jang, J. W.; Reo, Y. J. “A Blockchain-based Secure IoT Control Scheme”. In: 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2018, pp. 74–78.

[9] de Brito Gonçalves, J. P.; Spelta, G.; da Silva Villaca, R.; Gomes, R. L. “IoT data storage on a blockchain using smart contracts and IPFS”. In: 2022 IEEE International Conference on blockchain (Blockchain), 2022, pp. 508–511.

[10] Ethereum. “Remix - Ethereum IDE”. Capturado em: https://remix.ethereum.org/, Dez 2023.

[11] Fassoni, R. “e-Residency: passo a passo para a Cidadania Digital Estoniana”. Capturado em: https://estoniahub.com.br/empreendedorismo/e-residency-passo-a-passo-para-a-cidadania-digital-estoniana/, Dez 2023.

[12] Gebresilassie, S. K.; Rafferty, J.; Morrow, P.; Chen, L.; Abu-Tair, M.; Cui, Z. “Distributed, secure, self-sovereign identity for IoT devices”. In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2020, pp. 1–6.

[13] Gerrits, L.; Kromes, R.; Verdier, F. “A True Decentralized Implementation Based on IoT and Blockchain: a Vehicle Accident Use Case”. In: 2020 International Conference on Omni-layer Intelligent Systems (COINS), 2020, pp. 1–6.

[14] Gonçalves, H.; Hessel, F. “IoChain: A Decentralized Multichain-Based Architecture for IoT Smart Agriculture Using IPNS”. In: 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), 2023, pp. 1–6.

[15] Hao, X.; Yeoh, P. L.; Wu, T.; Yu, Y.; Li, Y.; Vucetic, B. “Scalable double blockchain architecture for IoT information and reputation management”. In: 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), 2021, pp. 171–176.

[16] ID, P. “Privado ID Documentation”. Capturado em: https://devs.polygonid.com, Jun 2024.

[17] IoTeX. “What are Decentralized Physical Infrastructure Networks (DePIN)?” Capturado em: https://iotex.io/blog/what-are-decentralized-physical-infrastructure-networks-depin/, Dez 2023.

[18] Jayabalan, J.; Jeyanthi, N. “Scalable blockchain model using offchain IPFS storage for healthcare data security and privacy”, *Journal of Parallel and Distributed Computing*, vol. 164, 2022, pp. 152–167.

[19] Kshetri, N. “Can blockchain strengthen the Internet of Things?”, *IT Professional*, vol. 19–4, 2017, pp. 68–72.

[20] Kumar, R.; Goyal, R. “On cloud security requirements, threats, vulnerabilities and countermeasures: A survey”, *Computer Science Review*, vol. 33, 2019, pp. 1–48.

[21] Kumar, R.; Tripathi, R.; Marchang, N.; Srivastava, G.; Gadekallu, T. R.; Xiong, N. N. “A secured distributed detection system based on IPFS and blockchain for industrial image and video data security”, *Journal of Parallel and Distributed Computing*, vol. 152, 2021, pp. 128–143.

[22] Labs, P. “IPFS”. Capturado em: https://ipfs.tech, Dez 2023.

[23] Lee, J.-L.; BusiReddyGari, P.; Thompson, B. “A lightweight smart meter framework using a scalable blockchain for smart cities”. In: 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), 2021, pp. 433–438.

[24] Lei, K.; Du, M.; Huang, J.; Jin, T. “Groupchain: Towards a scalable public blockchain in fog computing of IoT services computing”, *IEEE Transactions on Services Computing*, vol. 13–2, 2020, pp. 252–262.

[25] Luong, D. A.; Park, J. H. “Privacy-Preserving Blockchain-Based Healthcare System for IoT Devices Using zk-SNARK”, *IEEE Access*, vol. 10, 2022, pp. 55739–55752.

[26] McCauley, A. “What is the difference between Web2 and Web3?” Capturado em: https://www.linkedin.com/feed/update/urn:li:activity:6897586526239236096/, Fev 2024.

[27] MetaMask. “MetaMask”. Capturado em: https://metamask.io, Dez 2023.

[28] Monrat, A. A.; Schelen, O.; Andersson, K. “Performance evaluation of permissioned blockchain platforms”. In: 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2020, pp. 1–8.

[29] Nakamoto, S. “Bitcoin: A Peer-to-Peer Electronic Cash System”, *IEEE Transactions on Computers*, vol. 58–9, Sept 2009, pp. 1073–1083.

[30] Niya, S. R.; Schiller, E.; Cepilov, I.; Stiller, B. “Biit: Standardization of blockchain-based IoT systems in the i4 era”. In: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, 2020, pp. 1–9.

[31] OpenSea. “Testnets OpenSea”. Capturado em: https://testnets.opensea.io/, Dez 2024.

[32] OpenZeppelin. “OpenZeppelin”. Capturado em: https://www.openzeppelin.com/, Dez 2023.

[33] Peng, L.; Feng, W.; Yan, Z.; Li, Y.; Zhou, X.; Shimizu, S. “Privacy preservation in permissionless blockchain: A survey”, *Digital Communications and Networks*, vol. 7–3, 2021, pp. 295–307.

[34] Pinata. “Pinata Documentation”. Capturado em: https://pinata.cloud/pricing, Jan 2025.

[35] Pincheira, M.; Donini, E.; Vecchio, M.; Kanhere, S. “A decentralized architecture for trusted dataset sharing using smart contracts and distributed storage”, *Sensors*, vol. 22–23, 2022.

[36] Pissolatto, E. C.; Hessel, F. “eDNA: A Decentralized Marketplace Architecture for the Automotive Sector”. In: 2024 IEEE 10th World Forum on Internet of Things (WF-IoT), 2024, pp. 858–863.

[37] Pithadia, H.; Fenoglio, E.; Batrinca, B.; Treleaven, P.; Echim, R.; Bubutanu, A.; Kerrigan, C. “Data Assets: Tokenization and Valuation”, Relatório Técnico, Social Science Research Network (SSRN), 2023, 32p, available at: http://dx.doi.org/10.2139/ssrn.4419590 or https://ssrn.com/abstract=4419590.

[38] Polygon. “Polygon Technology”. Capturado em: https://polygon.technology, Dez 2023.

[39] Polygon. “Polygon zkEVM”. Capturado em: https://polygon.technology/polygonzkevm, Dez 2023.

[40] Popchev, I.; Doukovska, L.; Radeva, I. “A framework of blockchain/IPFS-based platform for smart crop production”. In: 2022 International Conference Automatics and Informatics (ICAI), 2022, pp. 265–270.

[41] Protocol, O. “Compute-to-Data”. Capturado em: https://docs.oceanprotocol.com/developers/Compute-to-Data, Dez 2023.

[42] Protocol, O. “Ocean.js Documentation”. Capturado em: https://github.com/oceanprotocol/ocean.js, Jan 2025.

[43] Protocol, O. “Ocean.py Documentation”. Capturado em: https://github.com/oceanprotocol/ocean.py, Jan 2025.

[44] Ramachandran, G. S.; Radhakrishnan, R.; Krishnamachari, B. “Towards a Decentralized Data Marketplace for Smart Cities”. In: 2018 IEEE International Smart Cities Conference (ISC2), 2018, pp. 1–8.

[45] Rasolroveicy, M.; Fokaefs, M. “Performance and cost evaluation of public blockchain: An NFT marketplace case study”. In: 2022 4th Conference on blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2022, pp. 79–86.

[46] Sangeeta, N.; Nam, S. Y. “Blockchain and Interplanetary File System (IPFS)-Based Data Storage System for Vehicular Networks with Keyword Search Capability”, *Electronics*, vol. 12–7, 2023.

[47] Schiller, E.; Niya, S. R.; Surbeck, T.; Stiller, B. “Scalable transport mechanisms for blockchain IoT applications”. In: 2019 IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium), 2019, pp. 34–41.

[48] Shahjalal, M.; Islam, M. M.; Alam, M. M.; Jang, Y. M. “Implementation of a secure LoRaWAN system for industrial internet of things integrated with IPFS and blockchain”, *IEEE Systems Journal*, vol. 16–4, 2022, pp. 5455–5464.

[49] Truong, H. T. T.; Almeida, M.; Karame, G.; Soriente, C. “Towards Secure and Decentralized Sharing of IoT Data”. In: 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 176–183.

[50] Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. “An overview of blockchain technology: Architecture, consensus, and future trends”. In: 2017 IEEE International Congress on Big Data (BigData Congress), 2017, pp. 557–564.

APÊNDICE A – CONFIGURAÇÃO DO AMBIENTE DOCKER

```
1: # Imagem base do contêiner
2: FROM python:3.9-slim
3: # Definir o diretório de trabalho dentro do contêiner
4: WORKDIR /app
5: # Instalar IPFS
6: RUN apt-get update
7: RUN apt-get install -y ipfs
8: # Copiar os arquivos para o contêiner
9: COPY script.py .
10: COPY requirements.txt .
11: # Instalar as dependências
12: RUN pip install –no-cache-dir -r requirements.txt
13: # Inicializar o daemon do IPFS
14: RUN ipfs init
15: RUN ipfs daemon &
16: # Definir o comando de execução padrão do contêiner
17: ENTRYPOINT [“python”, “script.py”]
```

APÊNDICE B – ALGORITMO DE CÁLCULO DA MÉDIA ARITMÉTICA COM INTEGRAÇÃO *IPFS* PARA *C2D*

```
1: function ProcessData(data)
2: values ← data.get("values", [])
3: if not values then
4:   return {"error":  "Nenhum valor numérico fornecido."}
5: end if
6: mean_value ← sum(values) / len(values)
7: return {"mean":  mean_value}
8:
9: function FetchDataFromIPFS(cid)
10: {Baixa o dataset armazenado no IPFS}
11: command ← "ipfs cat " + cid
12: result ← subprocess.run(command, capture_output=True, text=True)
13: return json.loads(result.stdout)
14:
15: function PublishToIPFS(data)
16: {Publica os resultados de volta no IPFS}
17: json_data ← json.dumps(data)
18: command ← "echo '" + json_data + "' | ipfs add -Q"
19: result ← subprocess.run(command, capture_output=True, text=True)
20: return result.stdout.strip() {Retorna o CID gerado}
21:
22: function Main()
23: {Obtém os dados a partir do IPFS}
24: cid ← "QmExampleCID" {CID de exemplo}
25: input_data ← FetchDataFromIPFS(cid)
26: {Processa os dados}
27: result ← ProcessData(input_data)
28: {Publica o resultado no IPFS}
29: new_cid ← PublishToIPFS(result)
30: {Retorna o CID do resultado}
31: print("Resultado armazenado em:", new_cid)
32:
33: if __name__ == "__main__" then
34:   Main()
35: end if
```

APÊNDICE C – CONTRATO SOLIDITY PARA DNFTS COM PRIVADO ID

```
1: pragma solidity ^0.8.10;
2:
3: {Importação das bibliotecas OpenZeppelin}
4: import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
5: import "@openzeppelin/contracts/token/ERC721/extensions/
6: ERC721URIStorage.sol";
7: import "@openzeppelin/contracts/utils/Strings.sol";
8: import "@openzeppelin/contracts/utils/Base64.sol";
9:
10: {Interface para verificação de DID com Privado ID}
11: interface IPrivadoID
12: function verifyDID(address user, string memory did)
13: external view returns (bool);
14:
15: {Definição do contrato principal}
16: contract CarDynamicNFT is ERC721, ERC721URIStorage
17: {Estrutura dos dados do veículo}
18: struct CarInfo
19: string owner;
20: string model;
21: string imageCID;
22: uint256 lastDistance;
23:
24: {Mapeamento de IDs para informações do veículo}
25: mapping(uint256 => CarInfo) public carData;
26: mapping(uint256 => string) public carDIDs;
27: IPrivadoID private privadoID;
28: address private s_owner;
29:
30: {Evento para atualização dos metadados do dNFT}
31: event NFTUpdated(uint256 tokenId, string newCID);
32:
33: {Construtor do contrato}
34: constructor(address privadoIDAddress) ERC721("Car dNFT", "CAR")
35: s_owner = msg.sender;
36: privadoID = IPrivadoID(privadoIDAddress);
37:
```

```
{Função para mintar um novo dNFT do veículo}
function mintCarNFT(
address to, uint256 tokenId, string memory ownerName,
string memory model, string memory imageCID, uint256 initialDistance,
string memory ownerDID) public
require(privadoID.verifyDID(to, ownerDID), "DID verification
failed");

_safeMint(to, tokenId);
carData[tokenId] = CarInfo(ownerName, model, imageCID,
initialDistance);
carDIDs[tokenId] = ownerDID;
_setTokenURI(tokenId, generateTokenURI(tokenId));

{Função para atualizar a quilometragem do veículo no dNFT}
function updateCarDistance(uint256 tokenId, uint256 newDistance, string memory
userDID) public
require(_exists(tokenId), "Token ID does not exist");
require(privadoID.verifyDID(msg.sender, userDID), "DID verification
failed");
carData[tokenId].lastDistance = newDistance;
string memory newTokenURI = generateTokenURI(tokenId);
_setTokenURI(tokenId, newTokenURI);
emit NFTUpdated(tokenId, newTokenURI);
```

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br