Universidade Federal de Ouro Preto
Escola de Minas
Engenharia de Controle e Automação

Vinícius Gonçalves dos Santos

Arquitetura IoT-Fog-Tangle para garantia de integridade de dados de sensores

Ouro Preto
2026

Vinícius Gonçalves dos Santos

Arquitetura IoT-Fog-Tangle para garantia de integridade de dados de sensores

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro(a) de Controle e Automação.

Orientador: Prof. M.e Diógenes Viegas Mendes Ferreira

Ouro Preto
2026

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Vinícius Gonçalves dos Santos

Arquitetura IoT-Fog-Tangle para garantia de integridade de dados de sensores

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 3 de agosto de 2026.

Membros da banca

Me. Diógenes Viegas Mendes Ferreira - Orientador - DECOM - Universidade Federal de Ouro Preto

Dra. Adrielle de Carvalho Santana - Convidada - DECAT - Universidade Federal de Ouro Preto

Dra. Luciana Gomes Castanheira - Convidada - DECAT - Universidade Federal de Ouro Preto

Diógenes Viegas Mendes Ferreira, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 03/08/2026

Documento assinado eletronicamente por **Diogenes Viegas Mendes Ferreira**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 03/08/2026, às 15:57, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1146384** e o código CRC **BADB18F6**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Agradecimentos

Ao final desta etapa, levo comigo o legado de tudo o que vivi, aprendi e compartilhei ao longo desta caminhada. Agradeço a Deus pelo caminho trilhado até aqui. Aos meus pais, Deiverson e Andreia, agradeço por todo apoio, dedicação e amor incondicional. Vocês são meu alicerce, meu porto seguro e minha inspiração diária. À minha avó Teresa e à minha irmã Lavinnia, pelo carinho e pela presença constantes, que tornaram cada desafio mais leve. Aos meus padrinhos, tios e primos, agradeço pelo apoio.

Ao meu orientador, Prof. M.e Diógenes Viegas Mendes Ferreira, pela disponibilidade, pelas orientações precisas e pela confiança depositada na condução deste trabalho.

Resumo

A crescente dependência de dados de sensores em sistemas baseados em Internet das Coisas (IoT) torna necessária a garantia de sua integridade, sobretudo em cenários nos quais tais dados funcionam como prova e podem ser adulterados retroativamente pelo próprio operador da infraestrutura que os armazena. Mecanismos tradicionais de segurança protegem os dados em trânsito, mas não impedem a alteração de registros já armazenados. Este trabalho propõe e demonstra, por meio de uma prova de conceito, uma arquitetura em quatro camadas: IoT, computação em *Fog*, Rede Tangle e *Cloud*, na qual o resumo criptográfico de cada leitura é ancorado em um registro distribuído privado, funcionando como âncora de confiança externa ao ambiente de armazenamento. Adotou-se como caso de uso a detecção de violação de lacre, com aquisição em um dispositivo ESP32, processamento e cálculo de resumo SHA-256 na camada *Fog*, e verificação por comparação com o registro imutável. A avaliação, conduzida sobre 150 eventos, submeteu os registros a três cenários de adulteração. Os resultados demonstram que, sob comprometimento consistente do armazenamento, a detecção ocorre exclusivamente pela comparação com o resumo ancorado no Tangle, evidenciando o papel decisivo do *ledger*; a taxa de detecção foi de 96,7%, sem falsos positivos, e a latência de processamento na borda manteve mediana de 5,67 ms. Discute-se ainda a tensão entre a ausência de custos e o grau de descentralização nas implantações de Tangle, evidenciada pela migração da rede pública IOTA para o protocolo Rebased em 2025.

Palavras-chave: Internet das Coisas. Integridade de dados. Tangle. IOTA. Fog Computing. Ledger distribuído.

Abstract

The growing reliance on sensor data in Internet of Things (IoT) systems makes data integrity a relevant concern, especially in scenarios where such data serves as evidence and may be retroactively tampered with by the very operator of the infrastructure that stores it. Traditional security mechanisms protect data in transit but do not prevent the alteration of records once stored. This work proposes and demonstrates, through a proof of concept, a four-layer architecture: IoT, Fog computing, the Tangle network, and cloud, in which the cryptographic digest of each reading is anchored in a private distributed ledger, acting as a trust anchor external to the storage environment. Seal-tampering detection was adopted as the use case, with acquisition on an ESP32 device, processing and SHA-256 digest computation at the Fog layer, and verification by comparison against the immutable record. The evaluation, carried out over 150 events, subjected the records to three tampering scenarios. The results show that, under consistent compromise of the storage, detection occurs exclusively through comparison with the digest anchored in the Tangle, evidencing the decisive role of the ledger; the detection rate was 96.7%, with no false positives, and the edge processing latency had a median of 5.67 ms. The tension between fee-free operation and the degree of decentralization in Tangle deployments, made evident by the 2025 migration of the public IOTA network to the Rebased protocol, is also discussed.

Key-words: Internet of Things. Data integrity. Tangle. IOTA. Fog computing. Distributed ledger.

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
ADC	*Analog-to-Digital Converter*
API	*Application Programming Interface*
CLP	Controlador Lógico Programável
CYD	*Cheap Yellow Display*
DAG	*Directed Acyclic Graph*
DLT	*Distributed Ledger Technology*
IoT	*Internet of Things*
JSON	*JavaScript Object Notation*
LDR	*Light Dependent Resistor*
MQTT	*Message Queuing Telemetry Transport*
NIST	*National Institute of Standards and Technology*
NTP	*Network Time Protocol*
SCADA	*Supervisory Control and Data Acquisition*
SDK	*Software Development Kit*
SHA	*Secure Hash Algorithm*

Sumário

1 Introdução

O aumento do uso de dispositivos conectados à Internet tem transformado os paradigmas de comunicação, automação e controle de processos industriais. Com o avanço da Indústria 4.0, sensores, atuadores, controladores lógicos programáveis (CLPs) e sistemas de supervisão e aquisição de dados (SCADA) passaram a integrar redes IP capazes de transmitir informações em tempo real para plataformas de computação em *Cloud*, habilitando o monitoramento remoto, a manutenção preditiva e a tomada de decisão baseada em dados (BURHAN *et al.*, 2023).

A adoção massiva de dispositivos IoT industriais, contudo, não foi acompanhada de medidas de segurança proporcionais à criticidade dos ambientes em que operam (IARAS EDITORIAL BOARD, 2025). Entre as fragilidades, se destacam a falta de verificação de integridade das informações geradas pelos sensores: dados adulterados em um sistema de supervisão podem induzir decisões incorretas, comprometer auditorias regulatórias e abrir vetores para ataques direcionados: Ao injetar leituras falsas, um agente malicioso pode induzir o sistema de controle a executar ações indevidas como desligar um alarme, mascarar uma falha ou acionar um atuador em momento crítico, transformando a manipulação do dado em um meio de sabotagem da operação física (FERRARIS; FERNANDEZ-GAGO; LOPEZ, 2023).

A arquitetura em camadas IoT–*Fog*–*Cloud*, amplamente adotada para reduzir a latência de processamento e aliviar a carga de comunicação com a *Cloud*, introduz novos pontos de falha e amplia a superfície de ataque disponível para agentes maliciosos (IARAS EDITORIAL BOARD, 2025). A multiplicação de nós intermediários dificulta a implementação de políticas de segurança uniformes e a garantia de que os dados transitados entre as camadas não foram modificados (VALDEZ *et al.*, 2024). Nesse cenário, a questão da integridade dos dados assume caráter central, sobretudo em sua dimensão menos abordada: como garantir que uma leitura de sensor, depois de armazenada, não possa ser adulterada retroativamente sem que isso seja detectável, ainda que o agente capaz de alterá-la seja o próprio responsável pela infraestrutura que a armazena?

No âmbito das soluções para prover rastreabilidade, imutabilidade e auditabilidade em sistemas distribuídos, as Tecnologias de Registro Distribuído (DLT -*Distributed Ledger Technology*) emergem como alternativas promissoras. A *blockchain*, estrutura de dados encadeada em blocos imutáveis com validação por consenso distribuído, popularizou-se com o protocolo Bitcoin (NAKAMOTO, 2008) e expandiu-se para aplicações além das criptomoedas, incluindo contratos inteligentes com a plataforma Ethereum (BUTERIN, 2014). Desde então, a tecnologia tem sido aplicada em diversos domínios industriais e de pesquisa, como

a rastreabilidade de produtos em cadeias de suprimentos (SILVANO; MARCELINO; VIGIL, 2021) e a criação de mercados descentralizados de recursos (SOUZA *et al.*, 2022), sempre explorando a mesma propriedade fundamental: a manutenção de um registro compartilhado e descentralizado, que assegura integridade e transparência dos dados sem depender de uma autoridade central (ZUBAYDI; VARGA; MOLNAR, 2023). *Surveys* recentes apontam que a integração de DLTs a sistemas IoT é um campo em franca expansão, capaz de prover camadas de confiança sem depender de uma autoridade central (ZUBAYDI; VARGA; MOLNAR, 2023; PAJOOH *et al.*, 2021). Contudo, as *blockchains* tradicionais apresentam limitações relevantes para ambientes IoT industriais: taxas de transação elevadas, requisitos computacionais incompatíveis com dispositivos de baixo consumo e degradação de desempenho em cenários com grande número de transações simultâneas.

A Rede Tangle, estrutura de dados subjacente ao protocolo IOTA, proposta por Popov (2018), surge como uma alternativa diretamente endereçada às restrições dos ambientes IoT. Baseada em um Grafo Acíclico Direcionado (DAG — *Directed Acyclic Graph*), a Tangle não possui mineradores nem validadores centralizados: cada nova transação valida duas transações anteriores, eliminando taxas e permitindo que o desempenho da rede escale positivamente com o aumento do volume de participantes (POPOV, 2018; MÜLLER *et al.*, 2022). Essas características técnicas tornam a Tangle particularmente adequada ao perfil de ambientes IoT industriais, nos quais dezenas, centenas ou milhares de dispositivos de baixo consumo realizam transmissões contínuas de pequenos volumes de dados (CARELLI *et al.*, 2022; SEALEY; AIJAZ; HOLDEN, 2022).

Estudos empíricos sobre o comportamento da Tangle em condições reais confirmam tempos de confirmação de transações compatíveis com aplicações industriais e resistência a ataques de duplo gasto, nos quais um mesmo ativo digital é reutilizado em duas transações conflitantes por meio de uma cadeia paralela que subverte a ordem de validação da rede (FAN *et al.*, 2021; DONG *et al.*, 2020). Cabe ressalvar que essas características referem-se ao protocolo original da Tangle. A migração da rede pública IOTA para o protocolo Rebased, em 2025, reintroduziu taxas de transação, de modo que a ausência de custos se mantém hoje sobretudo em implantações privadas ou permissionadas (NZAKUNA; PACIELLO *et al.*, 2025).

No contexto brasileiro, o interesse pela integração de IoT e DLTs tem crescido tanto no meio acadêmico quanto no âmbito regulatório. O Decreto n.º 9.854, de 25 de junho de 2019, instituiu o Plano Nacional de Internet das Coisas, reconhecendo a tecnologia como vetor estratégico para setores como indústria, cidades inteligentes, saúde e agronegócio (BRASIL. MCTI, 2019). O *Roadmap* Tecnológico que subsidiou esse plano identifica a segurança da informação como um dos principais desafios da área e aponta, como tendência central, a adoção do princípio de *security by design*, isto é, a incorporação da segurança desde a arquitetura do sistema, seguindo os pilares de confidencialidade, integridade, dis-

ponibilidade, autenticidade e privacidade (BNDES; MCTIC, 2017). O mesmo documento reconhece o potencial da tecnologia *blockchain* para elevar a segurança em IoT, ao descentralizar a confiança por características como imutabilidade e auditabilidade (BNDES; MCTIC, 2017).

Pesquisas desenvolvidas em universidades brasileiras têm explorado aplicações de *blockchain* e Tangle em IoT (SILVANO; MARCELINO, 2020; MARTINEZ; RODRIGUES, 2020; GOMES, 2021), evidenciando a atualidade do tema. Este trabalho insere-se nesse esforço, endereçando uma lacuna ainda pouco explorada: o uso da Rede Tangle como âncora de integridade contra a adulteração retroativa de registros já armazenados, e não apenas contra ataques em trânsito, nas arquiteturas IoT–*Fog*–*Cloud*.

1.1 Objetivo Geral

Propor e demonstrar, por meio de uma prova de conceito, uma arquitetura integrada de IoT, computação em *Fog*, Rede Tangle e *Cloud* para garantia de integridade e rastreabilidade de dados em sistemas de monitoramento baseados em sensores.

1.2 Objetivos Específicos

1. Propor e implementar uma arquitetura de quatro camadas (IoT, *Fog*, Tangle e *Cloud*) com sensor conectado a um ESP32, processamento em contêiner Docker e ancoragem em Tangle privado.

2. Garantir a integridade dos dados por meio do registro de *hashes* SHA-256 na Tangle e da verificação por comparação com o registro imutável ancorado.

3. Demonstrar a detecção de adulteração de registros, inclusive sob comprometimento consistente do armazenamento, e mensurar a latência de processamento na camada *Fog*.

1.3 Justificativas e Relevância

A ausência de mecanismos formais de verificação de integridade em fluxos de dados IoT industriais representa um risco real e crescente à confiabilidade de sistemas de supervisão e controle (IARAS EDITORIAL BOARD, 2025). Esse risco, contudo, não se restringe à adulteração de dados em trânsito, já tratada por mecanismos de criptografia: em muitos cenários, o dado de sensor é utilizado posteriormente como prova em auditorias regulatórias, perícias técnicas ou disputas contratuais, e precisa permanecer íntegro depois de armazenado. Nesses casos, a criptografia de transporte e o controle de acesso não são

suficientes, pois não impedem que um registro já gravado seja alterado retroativamente, sobretudo quando o agente capaz de alterá-lo é o próprio responsável pela infraestrutura que o armazena.

A utilização da Rede Tangle como camada adicional de confiança, integrada a uma arquitetura IoT–*Fog*–*Cloud*, endereça diretamente esse problema. Ao registrar o *hash* criptográfico de cada leitura na Tangle, obtém-se uma referência imutável e externa ao ambiente de armazenamento: qualquer modificação posterior nos dados mantidos na *Cloud* passa a ser detectável por comparação com o registro distribuído, ainda que o atacante tenha alterado o dado de forma consistente no próprio armazenamento (CARELLI *et al.*, 2022; MARGARIA; VESCO; CARELLI, 2024). Diferente das abordagens baseadas em *blockchain* convencional, a Tangle dispensa taxas e infraestrutura de mineração em implantações privadas, tornando-a viável para uso industrial (POPOV, 2018).

Justifica-se, assim, o desenvolvimento de uma prova de conceito que demonstre essa propriedade de forma concreta. Adotou-se como caso de uso a detecção de violação de lacre, na qual o registro de que um invólucro permaneceu fechado constitui, ele próprio, uma informação passível de contestação e de adulteração posterior. A avaliação por meio de ferramentas de código aberto torna o trabalho replicável e de interesse prático para a comunidade de engenharia (GOMES, 2021).

1.4 Organização do documento

Este trabalho está organizado em cinco capítulos. O presente capítulo apresentou a contextualização do tema, o problema de pesquisa, os objetivos e a justificativa do estudo. O Capítulo 2 apresenta a revisão de literatura, abordando os conceitos fundamentais de Internet das Coisas, computação em *Fog*, computação em *Cloud*, tecnologias de registro distribuído e a Rede Tangle, além dos mecanismos de integridade empregados. O Capítulo 3 descreve a metodologia, detalhando a arquitetura proposta, as tecnologias utilizadas e o procedimento de avaliação. O Capítulo 4 apresenta e discute os resultados obtidos quanto à detecção de adulteração e à latência de processamento. Por fim, o Capítulo 5 traz as considerações finais, retomando os objetivos, apontando as limitações do estudo e sugerindo direções para trabalhos futuros.

2 Revisão de literatura

Este capítulo apresenta os fundamentos teóricos que sustentam a arquitetura proposta. São abordados os conceitos de Internet das Coisas, computação em *Fog* e em *Cloud*, tecnologias de registro distribuído e a Rede Tangle, além dos mecanismos de comunicação e de integridade criptográfica empregados neste trabalho.

2.1 Internet das Coisas (IoT)

A Internet das Coisas (IoT) é caracterizada por uma rede expansiva de dispositivos interconectados capazes de coletar, processar e compartilhar dados em tempo real de forma contínua. Segundo Atzori, Iera e Morabito (2010), esse paradigma atravessa a simples conectividade da internet tradicional, estabelecendo uma comunicação direta entre objetos físicos e sistemas digitais que transforma ambientes cotidianos e industriais. Com o avanço da Indústria 4.0, a IoT consolidou-se como uma malha amplamente utilizada na qual sensores e atuadores embarcados colaboram para otimizar processos de forma distribuída, reduzindo a necessidade de intervenção humana direta na geração e na transmissão das informações (BURHAN *et al.*, 2023).

2.1.1 Crescimento e Projeções

O impacto e a adoção dessa tecnologia têm apresentado crescimento significativo em escala global. Estimativas de mercado apontam que o número de dispositivos IoT em operação alcançará a ordem de dezenas de bilhões de unidades nos próximos anos, ampliando as fronteiras da automação residencial e industrial (BURHAN *et al.*, 2023). Essa difusão gera um grande volume de dados e exige infraestruturas de rede cada vez mais robustas e escaláveis para suportar o tráfego intenso de um número crescente de nós interligados (IARAS EDITORIAL BOARD, 2025).

2.1.2 Desafios de Segurança

O crescimento acelerado da IoT, descrito na seção anterior, traz consigo desafios críticos de segurança da informação. A arquitetura predominante da IoT depende fortemente de servidores em *Cloud* para o armazenamento e o processamento do tráfego gerado, o que deixa os dados amplamente expostos à interceptação e à adulteração (IARAS EDITORIAL BOARD, 2025). Essa fragilidade é agravada pelo fato de que o baixo poder computacional da maioria dos sensores de borda dificulta a implementação de criptografia robusta nos pontos de coleta (BURHAN *et al.*, 2023).

Essa vulnerabilidade à adulteração representa risco severo à integridade dos sistemas, uma vez que informações corrompidas na *Cloud* podem comprometer a operação de infraestruturas inteiras e a privacidade dos usuários (FERRARIS; FERNANDEZ-GAGO; LOPEZ, 2023). A comunicação entre a borda da rede e os servidores centrais consolidou-se como um dos principais alvos de agentes maliciosos, o que reforça, na literatura, a necessidade de novos padrões de proteção e de mecanismos de verificação de integridade que assegurem a autenticidade dos dados coletados ao longo de toda a cadeia (IARAS EDITORIAL BOARD, 2025).

2.1.3 Soluções Propostas

Para mitigar esses desafios, a literatura recente propõe arquiteturas em camadas que introduzem processamento intermediário entre os dispositivos IoT e a *Cloud* central. O modelo de *Fog Computing* processa dados localmente, reduzindo a latência e filtrando o tráfego antes do envio à *Cloud* (BONOMI *et al.*, 2012). De forma complementar, tecnologias de registro imutável, como as *blockchains* e os grafos acíclicos dirigidos (DAGs), oferecem meios de registrar dados de forma permanente e auditável, garantindo sua integridade (ZUBAYDI; VARGA; MOLNAR, 2023).

2.1.4 Trabalhos Relacionados

Diversos trabalhos têm explorado a aplicação de IoT e de tecnologias de registro distribuído. No contexto nacional, Silvano e Marcelino (2020) investigam o uso da Tangle para comunicação de dados entre dispositivos IoT, enquanto Gomes (2021) explora a rede no contexto de Economia das Coisas, e Martinez e Rodrigues (2020) discutem a segurança de transações no ecossistema IoT com *blockchain* e Tangle. No contexto internacional, Wu *et al.* (2025) propõem um esquema baseado em *blockchain* para transações seguras e preservação de privacidade em sistemas IoT, no qual a integridade dos dados é assegurada pela ancoragem de identificadores de conteúdo em um registro distribuído, de forma conceitualmente próxima à ancoragem de resumos criptográficos adotada neste trabalho. Esses estudos, contudo, concentram-se majoritariamente na proteção dos dados durante a transmissão, no controle de acesso ou no emprego de *blockchains* convencionais. Permanece pouco investigada a utilização da Tangle especificamente como âncora de integridade contra a adulteração retroativa de registros já armazenados situação em que o próprio operador da infraestrutura é o agente da adulteração, lacuna que motiva o presente trabalho.

2.2 *Cloud Computing*

A computação em *Cloud* é definida oficialmente pelo *National Institute of Standards and Technology* (NIST) como um modelo que habilita acesso sob demanda, via rede, a um conjunto compartilhado de recursos computacionais configuráveis de redes, servidores, armazenamento, aplicações e serviços, rapidamente provisionados e liberados com mínimo esforço de gerenciamento (MELL; GRANCE, 2011).

A essência da computação em *Cloud* reside na entrega de serviços de processamento, armazenamento e software pela internet, em vez da dependência de hardware local. Esse modelo se relaciona diretamente com dispositivos IoT, cuja capacidade de armazenamento e processamento é reduzida e que dependem da escalabilidade das plataformas em *Cloud* para gerenciar grandes volumes de requisições (BURHAN *et al.*, 2023).

2.2.1 Modelos de Serviço

Cloud Computing oferece três modelos principais de serviço:

1. IaaS (Infraestrutura como Serviço): Provisão de servidores virtuais, armazenamento e redes. Usuário gerencia o sistema operacional e aplicações. Exemplos: AWS EC2, Azure VMs.
2. PaaS (Plataforma como Serviço): Ambiente de desenvolvimento com OS e bibliotecas. Usuário foca em código. Exemplos: Google App Engine, AWS Lambda.
3. SaaS (Software como Serviço): Aplicações completas via navegador. Exemplos: Google Workspace, Microsoft 365.

A Tabela 1 sintetiza a divisão de responsabilidades entre usuário e provedor em cada um desses modelos.

Tabela 1 – Comparação entre modelos de serviço *Cloud*

Modelo	**Usuário gerencia**	**Provedor gerencia**	**Controle**
IaaS	OS, Aplicações, Dados	Hardware, Redes	Alto
PaaS	Aplicações, Dados	OS, Hardware, Redes	Médio
SaaS	Nenhum (só uso)	Tudo	Baixo

2.2.2 Vantagens

Cloud Computing oferece benefícios significativos: escalabilidade (recursos ajustáveis conforme demanda), custo equilibrado (pagamento por uso), alta disponibilidade

(99,9% de *uptime* com redundância geográfica) e colaboração (dados acessíveis de qualquer localização). A *Cloud* viabiliza o processamento complexo de dados para automação e inteligência artificial, permitindo análise de grandes volumes em tempo real.

2.2.3 Limitações

Apesar dos benefícios, a computação em *Cloud* apresenta limitações críticas para aplicações IoT em tempo real. A integração profunda entre IoT e *Cloud* amplia a superfície de ataque, elevando o risco de violação de privacidade e de exposição a ataques cibernéticos (FERRARIS; FERNANDEZ-GAGO; LOPEZ, 2023). Além disso, a latência entre o dispositivo e o servidor central pode inviabilizar aplicações que exigem resposta imediata, o que tem motivado o deslocamento do processamento para as bordas da rede. Esse deslocamento caracteriza o paradigma frequentemente descrito como *continuum* borda-*Cloud*, no qual o processamento se distribui ao longo de um contínuo entre os dispositivos e a *Cloud* (VALDEZ *et al.*, 2024). Esse cenário justifica a introdução da camada *Fog* entre os dispositivos IoT e a *Cloud*.

2.3 *Fog/Edge Computing*

2.3.1 Definição e Arquitetura

O *Fog Computing* é definido como uma extensão da *Cloud* para a borda da rede, posicionando recursos de processamento, comunicação e armazenamento mais próximos dos dispositivos finais, sem substituir completamente a infraestrutura centralizada (BONOMI *et al.*, 2012).

A arquitetura típica de *Fog* Computing segue o modelo em três camadas: dispositivos IoT (sensores e atuadores) $\rightarrow$ nó *Fog* (*gateway* ou servidor de borda) $\rightarrow$ *Cloud* (servidor central). Essa estrutura distribuída permite que dados brutos sejam filtrados e processados localmente, reduzindo a quantidade de tráfego transmitida para a *Cloud* e minimizando a latência em aplicações que exigem resposta em tempo real.

2.3.2 Redução de Latência

A principal vantagem do *Fog Computing* é a redução significativa da latência de resposta. Ao executar a computação localmente, o nó *Fog* elimina a necessidade de transmitir todos os dados a servidores distantes, diminuindo o tempo entre a coleta e o processamento e viabilizando aplicações sensíveis ao tempo, como monitoramento de saúde, automação industrial e controle de veículos autônomos (BONOMI *et al.*, 2012). O processamento na borda também permite a tomada de decisão sem dependência contínua de

conexão com a internet, garantindo continuidade operacional mesmo em cenários de falha de rede (VALDEZ *et al.*, 2024).

2.3.3 Segurança e Filtro de Ataques

A literatura também destaca o papel do *Fog* na segurança e na proteção de dados sensíveis: ao processar localmente informações pessoais, coordenadas ou estados operacionais de infraestrutura crítica, os nós de borda reduzem a exposição da *Cloud* a ataques cibernéticos (BURHAN *et al.*, 2023). Essa camada intermediária atua ainda como ponto de defesa, validando as requisições antes do despacho e bloqueando tráfego anômalo, o que aumenta a confiabilidade do sistema sem exigir elevado poder computacional dos sensores de borda (VALDEZ *et al.*, 2024).

2.4 Tangle (IOTA)

2.4.1 Definição e Funcionamento

O Tangle é a estrutura de dados em grafo acíclico dirigido (DAG — *Directed Acyclic Graph*) subjacente ao protocolo IOTA, proposta por Popov (2018) como alternativa à *blockchain* tradicional para aplicações IoT. Em vez de organizar as transações em blocos lineares consecutivos, o Tangle representa cada transação como um nó do grafo, em que cada nova transação valida duas transações anteriores, formando uma estrutura paralela e potencialmente mais escalável.

Em sua concepção original, essa arquitetura dispensava mineradores e taxas de transação: para emitir uma transação, o próprio usuário validava duas transações anteriores, distribuindo o trabalho de consenso entre os participantes da rede. Popov (2018) argumenta que, sob esse modelo, o desempenho da rede tenderia a crescer com o número de participantes, em contraste com o gargalo de validação das *blockchains* lineares.

2.4.2 Diferença entre *Blockchain* e Tangle

A *blockchain* tradicional organiza as transações em uma cadeia linear, na qual cada bloco referencia o anterior. Esse formato limita a escalabilidade, pois a validação de cada bloco concentra-se em um conjunto restrito de mineradores, o que gera competição e custos de transação. O Tangle, em sua proposta original, organizava as transações em uma estrutura paralela (DAG), em que cada transação validava duas anteriores, permitindo validação simultânea e, em tese, dispensando taxas (POPOV, 2018).

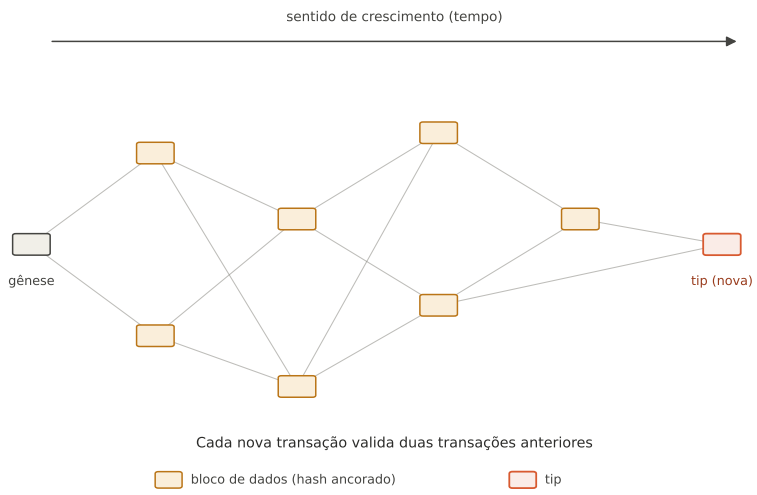
Vale lembrar que esse modelo foi substituído na rede pública IOTA com a migração para o protocolo Rebased, em 2025, detalhada na Seção 2.4.3. Assim, as características de ausência de taxas e de validação totalmente distribuída descritas a seguir aplicam-se

ao protocolo original e, no estado atual, se sustentam somente em implantações privadas ou permissionadas, como a adotada neste trabalho.

Formalmente, o Tangle pode ser representado por um grafo acíclico dirigido (DAG) $G = (V, E)$, em que o conjunto de vértices V corresponde às transações registradas e o conjunto de arestas dirigidas $E \subseteq V \times V$ representa as relações de aprovação entre elas. Uma aresta $(x, y) \in E$ indica que a transação x aprova diretamente a transação y, com y tendo sido registrada anteriormente a x. No modelo original, cada nova transação aprova duas transações precedentes, de modo que todo vértice $x \in V$ (exceto a transação inicial) possui grau de saída igual a dois.

A propriedade da inexistência de ciclos dirigidos em G é o que garante a ordenação temporal do registro: como uma transação só pode aprovar transações já existentes, não há como uma transação ser, direta ou indiretamente, aprovadora de si mesma. Essa propriedade é essencial para a imutabilidade, pois impede a reescrita da ordem histórica dos registros. A Figura 1 ilustra essa estrutura.

Figura 1 – Representação esquemática da estrutura do Tangle.



Fonte: Adaptado de Popov (2018).

2.4.3 Evolução Recente (Coordicide ao Rebased)

Entre 2018 e 2021, a IOTA Foundation desenvolveu o projeto Coordicide, cujo objetivo era remover o *Coordinator*, componente centralizado que validava transações de forma temporária, e descentralizar plenamente a rede, com mecanismos como o consenso probabilístico rápido (FPC) e a comunicação P2P (MÜLLER *et al.*, 2022). Esse caminho, porém, não se consolidou como originalmente previsto. Em maio de 2025, a rede pública migrou para o protocolo Rebased, que abandonou o modelo anterior em favor de um *ledger* baseado em objetos com a *Move Virtual Machine*, consenso Mysticeti por prova de

participação delegada e a reintrodução de taxas de transação (NZAKUNA; PACIELLO *et al.*, 2025).

Essa mudança tem impacto direto para aplicações IoT: a propriedade de ausência de taxas, historicamente apontada como principal vantagem do Tangle para o registro de microdados, deixou de valer na rede pública. Em consequência, cenários que exijam custo nulo passam a depender de implantações privadas ou permissionadas, que recuperam a ausência de taxas ao custo de uma centralização parcial da confiança e um conflito de projeto assumido neste trabalho.

2.4.4 Formalização da estrutura

Definindo a relação de aprovação direta como o conjunto de arestas E, e a aprovação indireta como seu fecho transitivo. Dizendo que uma transação x aprova indiretamente uma transação z se existe um caminho dirigido de x a z em G, isto é, uma sequência de transações $x = w_0, w_1, \ldots, w_k = z$ tal que $(w_i, w_{i+1}) \in E$ para todo $0 \leq i < k$. Por transitividade, uma transação aprova indiretamente todo o histórico alcançável a partir das transações que aprova diretamente (POPOV, 2018).

É essa estrutura de aprovação cumulativa que sustenta a imutabilidade explorada neste trabalho: à medida que novas transações são adicionadas ao grafo, uma transação antiga passa a ser aprovada, direta ou indiretamente, por um número crescente de transações posteriores. Alterar o conteúdo de uma transação já consolidada exigiria, portanto, refazer toda a cadeia de aprovações que dela dependem tarefa computacionalmente proibitiva em uma rede ativa. No contexto da arquitetura proposta, é precisamente essa imutabilidade que confere ao resumo criptográfico ancorado o papel de referência externa inviolável.

Cabe ressalvar que a formalização anterior descreve a estrutura de dados do Tangle, comum às suas diferentes versões. Os mecanismos que o protocolo original emprega para selecionar quais transações aprovar e para quantificar a confiança em cada registro não são utilizados neste trabalho. No modelo descrito por Popov (2018), essa seleção é feita por uma caminhada aleatória sobre o grafo, e a confiança em uma transação cresce conforme ela é aprovada por novas transações, uma medida conhecida como peso acumulado (*cumulative weight*). No ambiente de rede privada adotado neste trabalho, esses mecanismos são substituídos por um esquema de confirmação mais simples e determinístico: um nó central, denominado *coordinator*, emite periodicamente transações especiais, os *milestones*, que confirmam as transações válidas do grafo. Esse esquema, detalhado na metodologia, dispensa a dinâmica de consenso do modelo original.

2.5 Comunicação e integridade: MQTT e funções de resumo criptográfico

2.5.1 Protocolo MQTT

A comunicação entre dispositivos de borda e as camadas superiores em arquiteturas IoT utiliza, com frequência, o protocolo MQTT (*Message Queuing Telemetry Transport*). Trata-se de um protocolo de mensageria leve, baseado no modelo de publicação e assinatura (*publish/subscribe*), no qual os dispositivos produtores publicam mensagens em tópicos, e os consumidores interessados as recebem por meio de um intermediário denominado *broker*. Essa divisão entre produtor e consumidor, somando ao baixo consumo de banda e de recursos computacionais, torna o MQTT particularmente adequado a dispositivos com restrições de energia e processamento, característicos da Internet das Coisas (BURHAN *et al.*, 2023). A leveza do protocolo, contudo, não inclui mecanismos de garantia de integridade do conteúdo transportado: o MQTT assegura a entrega da mensagem, mas não que seu conteúdo permaneça íntegro após o armazenamento, o que reforça a necessidade de mecanismos complementares de verificação.

2.5.2 Funções de resumo criptográfico

A garantia de integridade de dados se baseia, neste trabalho, em funções de resumo criptográfico (*hash*). Uma função de resumo mapeia uma entrada de tamanho arbitrário em uma saída de tamanho fixo, o *digest*, de modo que três propriedades sejam asseguradas: a mesma entrada produz sempre o mesmo resumo (determinismo); é computacionalmente inviável recuperar a entrada a partir do resumo (resistência à pré-imagem); e qualquer alteração, ainda que mínima, na entrada produz um resumo radicalmente diferente, sendo inviável encontrar duas entradas distintas com o mesmo resumo (resistência à colisão). É essa última propriedade que sustenta o uso da *hash* como mecanismo de detecção de adulteração: se o dado for modificado, o resumo recalculado deixará de coincidir com o resumo de referência.

Neste trabalho emprega-se o algoritmo SHA-256, pertencente à família SHA-2, que produz um resumo de 256 bits e é amplamente adotado em aplicações de segurança, como nos mecanismos de consenso de tecnologias de registro distribuído (PAJOOH *et al.*, 2021). Vale ressaltar uma limitação inerente ao mecanismo: a comparação entre o resumo recalculado e o resumo de referência só é confiável se este último estiver armazenado fora do alcance de quem pode adulterar o dado. Caso o atacante controle tanto o dado quanto o seu resumo de referência, ele pode recalcular um resumo coerente com o dado falsificado, tornando a verificação local inócua. É precisamente essa limitação que motiva o uso de uma âncora externa de confiança, papel do registro distribuído discutido na Seção 2.4.

3 Metodologia

Este capítulo apresenta a metodologia adotada para o desenvolvimento e a avaliação da arquitetura proposta, detalhando a classificação da pesquisa, a arquitetura de quatro camadas, as tecnologias empregadas em cada camada, o desenho da avaliação experimental e as limitações metodológicas.

3.1 Classificação e abordagem da pesquisa

Quanto à natureza, esta pesquisa caracteriza-se como aplicada, por desenvolver um artefato funcional voltado a um problema concreto de integridade de dados. Quanto aos objetivos, é de caráter exploratório e descritivo. Quanto aos procedimentos, adota a forma de uma prova de conceito (*proof of concept*): a arquitetura proposta é implementada e submetida a uma avaliação controlada, com o propósito de demonstrar a viabilidade técnica de suas propriedades centrais, a detecção de adulteração de registros e o baixo custo de processamento na borda, e não de validar seu desempenho em escala ou em planta industrial real. Essa delimitação é deliberada e orienta a interpretação dos resultados apresentados no Capítulo 4.

3.2 Visão geral da arquitetura

A solução proposta se organiza em quatro camadas, segundo o fluxo IoT → *Fog* → Tangle → *Cloud*, conforme ilustrado na Figura 2. A camada IoT é responsável pela aquisição do dado; a camada *Fog*, pelo processamento na borda, pelo cálculo do resumo criptográfico e pelo roteamento, a camada Tangle, pela ancoragem imutável do resumo em uma rede de registro distribuído privada, e a camada *Cloud*, pelo armazenamento histórico do dado e pela disponibilização para consulta e verificação.

A premissa de projeto que diferencia a arquitetura é o tratamento da integridade como propriedade verificável posteriormente: o dado original é armazenado na *Cloud*, enquanto apenas o seu resumo criptográfico é ancorado no Tangle. Dessa forma, qualquer divergência entre o dado armazenado e o resumo imutável se torna detectável, mesmo que o adulterador detenha controle sobre o ambiente de armazenamento.

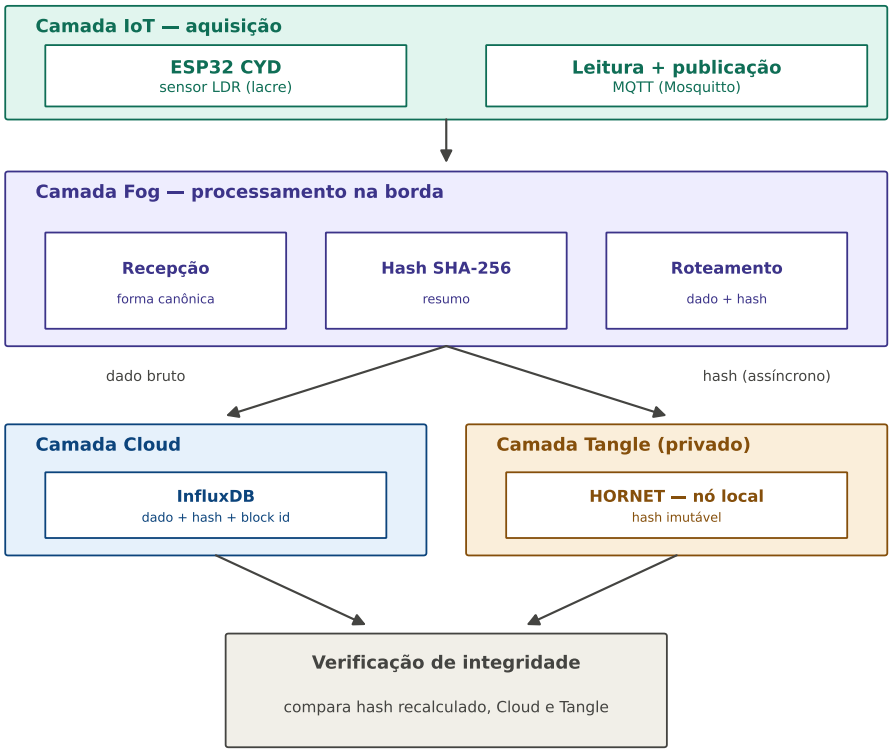


Figura 2 – Arquitetura de quatro camadas proposta.

Fonte: Elaborado pelo autor (2026).

3.3 Camada IoT: aquisição e detecção de violação

A camada de aquisição foi implementada com um microcontrolador ESP32, na variante *Cheap Yellow Display* (CYD), utilizando o sensor de luminosidade (LDR) embarcado na placa, conectado à porta GPIO34. O sensor é empregado como detector de violação de lacre: em um invólucro fechado, a ausência de luz corresponde ao estado `LACRADO`; a abertura do invólucro permite a entrada de luz, caracterizando o estado `VIOLADO`. A escolha desse caso de uso justifica-se por ser uma instância concreta de dado cuja integridade funciona como prova de custódia, alinhada à fundamentação apresentada no Capítulo 2.

A conversão do valor lido em estado emprega uma zona morta, para evitar oscilação na fronteira entre os estados. A partir da calibração descrita na Seção 3.3.1, definiram-se dois limiares: leituras com valor bruto (`luz_raw`) superior a 1750 são classificadas como `LACRADO`; inferiores a 1400, como `VIOLADO`; e a faixa intermediária constitui uma zona morta, na qual o estado anterior é mantido. A cada leitura, o dispositivo monta um pacote no formato JSON contendo o identificador do sensor, o instante da leitura (sincronizado por NTP), um número de sequência, o valor bruto de luminosidade e o estado, e o publica

via protocolo MQTT.

3.3.1 Calibração do sensor

Os limiares de decisão foram determinados empiricamente, medindo-se o valor bruto do LDR em três condições de iluminação controladas, conforme a Tabela 2. Observa-se a relação física esperada para o LDR da placa: quanto maior a incidência de luz, menor o valor bruto. A separação ampla entre o estado de invólucro fechado (escuro) e o de invólucro aberto (luz ambiente) da ordem de oitocentas unidades assegura margem suficiente para os limiares adotados.

Tabela 2 – Calibração do LDR e mapeamento de estados

Condição de iluminação	`luz_raw`	**Estado**
Sensor coberto (invólucro fechado)	≈ 1987	LACRADO
Invólucro aberto (luz ambiente)	≈ 1189	VIOLADO
Incidência de luz direta (saturação)	≈ 0	VIOLADO

Limiares adotados: LACRADO acima de 1750; VIOLADO abaixo de 1400; zona morta (histerese) entre 1400 e 1750.

3.4 Camada *Fog*: processamento e ancoragem

A camada *Fog* foi implementada em Python e cumpre duas funções, separadas em dois caminhos de execução. O caminho rápido, no qual a latência é medida, recebe a mensagem do *broker MQTT*, reconstrói a forma canônica do dado, calcula seu resumo criptográfico e grava o dado original na camada *Cloud* imediatamente. O caminho assíncrono, executado por um *worker* em segundo plano, enfileira o resumo e o ancora no Tangle, gravando posteriormente o identificador do bloco retornado.

A confirmação de um registro no *ledger* ocorre na escala de segundos, ao passo que o processamento na borda ocorre na escala de milissegundos. Manter a ancoragem fora do caminho de dados é o que permite que a latência de processamento seja baixa, sem que o tempo de confirmação do *ledger* a contamine. Em consequência, a métrica de latência reportada neste trabalho refere exclusivamente ao processamento na camada *Fog* (recepção, cálculo do resumo e gravação), e não ao tempo de confirmação no Tangle.

3.4.1 Forma canônica e resumo criptográfico

A integridade se apoia em uma representação determinística do dado, denominada forma canônica, obtida pela junção dos campos na ordem: `sensor_id|ts|seq|luz_raw|estado`. Cada campo dessa cadeia corresponde a um elemento do pacote publicado pela camada IoT (Seção 3.3.1): `sensor_id` identifica o dispositivo que originou a leitura; `ts` é o instante

em que a leitura foi realizada, sincronizado por NTP; `seq` é o número de sequência que ordena os eventos publicados por um mesmo sensor; `luz_raw` é o valor bruto de luminosidade capturado pelo LDR; e `estado` é a classificação resultante desse valor (`LACRADO` ou `VIOLADO`), conforme os limiares definidos na Seção 3.3.1. Sobre essa cadeia, aplica a função de resumo SHA-256, gerando o *hash* que identifica a leitura. A reprodutibilidade da verificação exige que os três componentes do sistema (gerador de eventos, nó *Fog* e verificador) construam a forma canônica de maneira idêntica e assim qualquer divergência de formatação invalidaria a comparação por construção.

3.5 Camada Tangle: rede de registro distribuído privada

A ancoragem dos resumos foi realizada em uma rede Tangle privada, executada localmente por meio do nó HORNET (protocolo Stardust), e não na rede pública do IOTA. Essa escolha é justificada por três fatores. Primeiro, a reprodutibilidade: a rede privada é determinística e independe de conectividade externa. Segundo, a ausência de custos: por controlar a emissão, a rede privada é efetivamente isenta de taxas, propriedade que, como discutido no Capítulo 2, deixou de valer na rede pública após a migração para o protocolo Rebased (NZAKUNA; PACIELLO *et al.*, 2025). Terceiro, a compatibilidade do kit de desenvolvimento de software (*Software Development Kit*, SDK) utilizado para comunicação com o nó local.

Cada resumo é ancorado como um bloco de dados etiquetado (*tagged data payload*), um modo de operação do protocolo destinado ao registro de dados arbitrários que não constituem transações de valor. Reconhece-se que a rede privada empregada utiliza um nó coordenador (*coordinator*) único, responsável por confirmar as transações, o que representa uma centralização parcial da confiança. Essa limitação é discutida na Seção 3.13.

3.6 Camada *Cloud*: armazenamento e verificação

O armazenamento histórico foi implementado com o banco de dados de séries temporais InfluxDB, executado localmente, que cumpre o papel de *Cloud* privada da arquitetura. Para cada leitura, armazenam-se o dado original, o resumo criptográfico calculado pelo *Fog* e, posteriormente, o identificador do bloco correspondente no Tangle. A visualização das leituras e dos estados do lacre é feita pela ferramenta Grafana, conectada ao InfluxDB.

Cabe esclarecer o escopo da proteção oferecida por esta arquitetura. O mecanismo proposto garante a integridade dos dados em repouso, isto é, protege os registros contra adulteração após seu armazenamento, inclusive por parte de um agente que controle a própria infraestrutura de armazenamento. A proteção da integridade e da confidenciali-

dade durante a transmissão, contra ataques como o de intermediário (*man-in-the-middle*), é um problema distinto e complementar, tratado por mecanismos consolidados de segurança de canal, como o uso de protocolos de comunicação cifrados, e não constitui o foco deste trabalho. Armazena-se o dado original porque o objetivo é permitir sua verificação posterior: a qualquer momento, recalcula-se o resumo do dado armazenado e compara-se com o resumo imutável ancorado no Tangle, o que revela qualquer alteração ocorrida no repositório.

A verificação de integridade, executada por um módulo dedicado, percorre os registros armazenados e, para cada um, reconstrói a forma canônica, recalcula o resumo, recupera o resumo ancorado no Tangle por meio do identificador do bloco e compara os três valores. A coincidência entre o resumo recalculado, o resumo armazenado na *Cloud* e o resumo ancorado no Tangle atesta a integridade; qualquer divergência indica adulteração.

3.7 Ambiente experimental

Todo o sistema foi executado em um único computador portátil, com processador Intel Core i7-7500U (2,70 GHz) e 8 GB de memória RAM, sob o sistema operacional Ubuntu 24.04 LTS em ambiente WSL2 sobre Windows. Os serviços de infraestrutura nó HORNET, *broker MQTT*, banco de dados e ferramenta de visualização foram executados em contêineres Docker, enquanto o nó *Fog* e os *scripts* de avaliação foram executados diretamente no sistema hospedeiro, fora dos contêineres. As versões dos componentes utilizados são apresentadas na Tabela 3, visando à reprodutibilidade do experimento.

Tabela 3 – Componentes de software e respectivas versões

Componente	**Versão**
Sistema operacional	Ubuntu 24.04 LTS (WSL2)
Docker	25.0.3
Python	3.12.3
Nó IOTA (HORNET, Stardust)	2.0.2
InfluxDB	2.7
Mosquitto (*broker* MQTT)	2.0
Grafana	11.1.0
Biblioteca iota-sdk	1.1.4
Biblioteca influxdb-client	1.43.0
Biblioteca paho-mqtt	1.6.1

3.8 Detalhamento da implementação e tecnologias

Esta seção apresenta os elementos centrais da implementação, com trechos do código que materializam os conceitos descritos anteriormente. O objetivo não é reproduzir

os programas integralmente, mas evidenciar como a integridade é assegurada em cada etapa, da recepção da leitura à verificação final.

3.8.1 Orquestração do ambiente local

A infraestrutura de apoio *broker MQTT*, banco de dados, ferramenta de visualização e nó do Tangle foi conteinerizada com Docker, o que garante a reprodutibilidade do ambiente e o isolamento entre os serviços. Cada serviço é declarado com sua imagem e versão fixadas (conforme a Tabela 3) e exposto em sua respectiva porta, de modo que o ambiente completo possa ser instanciado em qualquer máquina com Docker, sem instalação manual de dependências. O nó *Fog* e os *scripts* de avaliação foram executados diretamente no *host*, conectando-se aos serviços conteinerizados pelas portas mapeadas.

3.8.2 Nó *Fog*: recepção e geração do resumo

A camada *Fog* assina o tópico MQTT e, a cada leitura recebida, executa o caminho rápido: reconstrói a forma canônica do dado, calcula seu resumo criptográfico e grava o registro no banco de dados. A forma canônica é uma representação determinística, construída pela concatenação dos campos em ordem fixa, empregando apenas inteiros e *strings* para evitar divergências de formatação que invalidariam a comparação posterior. Sobre essa cadeia aplica-se a função SHA-256.

```
def canonical_form(sensor_id, ts, seq, luz_raw, estado):
# Ordem fixa: só inteiros e strings (sem float) para ser determinística
return f"{sensor_id}|{int(ts)}|{int(seq)}|{int(luz_raw)}|{estado}"

def sha256_hex(text):
return hashlib.sha256(text.encode("utf-8")).hexdigest()

def on_message(client, userdata, msg):
t0 = time.perf_counter()                          # marca início (latência)
p = json.loads(msg.payload.decode("utf-8"))       # leitura recebida via MQTT
canon = canonical_form(p["sensor_id"], p["ts"], p["seq"],
p["luz_raw"], p["estado"])
data_hash = sha256_hex(canon)                     # resumo da leitura
# grava dado + hash no InfluxDB (caminho rápido) ...
item = {"sensor_id": ..., "ts": ..., "hash": data_hash}
anchor_queue.put(item)  # enfileira para ancoragem assíncrona
```

Algoritmo 3.1 – Reconstrução da forma canônica e cálculo do resumo SHA-256.

O resumo é calculado antes de o dado ser persistido, e o instante de início (‘t0‘) marca o começo da medição da latência de processamento. Ao final, o *hash* é colocado em uma fila, para ancoragem assíncrona, sem bloquear o caminho rápido.

3.8.3 Ancoragem assíncrona no Tangle

A ancoragem do resumo é o elo que confere imutabilidade ao registro. Diferentemente de uma transação de valor, que transfere unidades da criptomoeda (*tokens*) entre participantes, o resumo é gravado como um bloco de dados etiquetado (*tagged data payload*), um modo de operação do protocolo destinado ao registro de dados arbitrários, e não de valores financeiros, identificados por uma etiqueta. Utilizou-se a biblioteca `iota-sdk` (versão 1.1.4) para codificar a etiqueta e o resumo em hexadecimal e submetê-los ao nó HORNET local, que retorna o identificador único do bloco gerado (*block id*).

```
iota = Client(nodes=[IOTA_NODE_URL], ignore_node_health=True)

def anchor_to_tangle(item):
# Encapsula o hash como tagged data payload e posta no nó HORNET
result = iota.build_and_post_block(
tag=utf8_to_hex(IOTA_TAG),        # etiqueta: "SENSOR-INTEGRITY"
data=utf8_to_hex(item["hash"]),   # conteúdo: o SHA-256 da leitura
)
return result[0]                        # block_id retornado pelo nó
```

Algoritmo 3.2 – Ancoragem do resumo no Tangle.

A submissão é executada por um *worker* em segundo plano; o *block id* retornado é gravado no banco de dados, associando cada leitura ao seu bloco imutável. Cabe registrar uma particularidade da operação em rede privada: por ser um nó único, sem pares, o HORNET reporta-se como não sincronizado (`isHealthy: false`). Ainda que confirme *milestones*, normalmente essa métrica pressupõe uma rede distribuída, inexistente em ambiente isolado. Foi necessário, portanto, instruir o cliente a não recusar a submissão com base nessa métrica, por meio do parâmetro `ignore_node_health`, sem o qual a ancoragem seria bloqueada apesar de o nó estar operacional. Trata-se de decorrência do uso de rede privada de nó único, e não de supressão de verificação relevante à integridade.

3.8.4 Simulação de adulteração da *Cloud*

A avaliação da propriedade central submete os registros já ancorados a três cenários de adulteração, todos aplicados exclusivamente sobre o banco de dados, jamais sobre o Tangle, coerente com a hipótese de um adversário que controla o armazenamento, mas não o *ledger*. O ataque encenado consiste em ocultar uma violação: pontos no estado VIOLADO são reescritos como LACRADO. A distinção entre os cenários está em o atacante atualizar, ou não, o resumo armazenado.

No cenário A (*só-dado*), o atacante altera a leitura mas não o resumo, tornando a fraude detectável localmente. No cenário B (completo), o atacante recalcula e regrava o resumo, deixando a *Cloud* internamente consistente. É justamente esse o caso que só o Tangle detecta. O cenário C mantém pontos íntegros, para medir falsos positivos.

```
def adulterar(row, completo):
novo_raw = LACRADO_RAW + random.randint(-3, 3)    # forja um "lacrado"
novo_estado = "LACRADO"
if completo:                                       # Cenário B
canon = canonical_form(row["sensor_id"], row["ts"], row["seq"],
novo_raw, novo_estado)
novo_hash = sha256_hex(canon)                  # recalcula hash coerente
escrever_campos(row, luz_raw=novo_raw, estado=novo_estado,
data_hash=novo_hash)          # nuvem fica consistente
else:                                              # Cenário A
escrever_campos(row, luz_raw=novo_raw, estado=novo_estado)
# deixa o hash antigo -> SHA local já detecta
```

Algoritmo 3.3 – Aplicação das adulterações nos cenários A e B.

3.8.5 Algoritmo de verificação de integridade

A verificação reconstrói a forma canônica de cada registro, recalcula o resumo, recupera o *hash* ancorado no Tangle por meio do *block id* e confronta os três valores. A coincidência entre os três atesta a integridade; qualquer divergência indica adulteração, e o ponto em que a divergência ocorre identifica a camada que a detectou.

```
def fetch_anchored_hash(block_id):
# Lê o bloco no Tangle e extrai o hash do tagged data payload
block = iota.get_block_data(block_id)
data_hex = block.payload.data
return hex_to_utf8(data_hex)

def verify_row(row):
canon = canonical_form(row["sensor_id"], row["ts"], row["seq"],
row["luz_raw"], row["estado"])
recomputed = sha256_hex(canon)           # hash recalculado agora
in_influx  = row.get("data_hash")        # hash guardado na nuvem
in_tangle  = fetch_anchored_hash(row["block_id"])  # hash imutável
if recomputed == in_influx == in_tangle:
return ("OK", "")
return ("ADULTERADO", ...)               # algum dos três divergiu
```

Algoritmo 3.4 – Verificação de integridade por comparação de três vias.

A lógica de três vias é o que distingue as camadas de detecção. Quando o resumo recalculado difere do armazenado na *Cloud*, a adulteração é detectada localmente, sem o *ledger* (cenário A). Quando o recalculado coincide com o da *Cloud*, mas difere do ancorado no Tangle, a fraude só é revelada pela âncora externa (cenário B) evidência empírica do papel decisivo do *ledger*.

3.9 Desenho da avaliação

A avaliação foi organizada em duas frentes complementares, cada uma respondendo a uma questão distinta.

3.9.1 Demonstração de operação com hardware real

A primeira frente verifica o funcionamento da arquitetura fim a fim com o dispositivo físico. O ESP32 CYD, conectado à rede sem fio, publica leituras reais do LDR, provocada a transição entre os estados LACRADO e VIOLADO pela alteração da iluminação sobre o sensor, observando a propagação do evento pela cadeia até a ancoragem no Tangle e a visualização no Grafana. Essa frente é de natureza qualitativa: seu objetivo é evidenciar a viabilidade operacional do sistema com hardware, e não produzir medidas estatísticas. A Figura 3 resume os cenários de adulteração empregados na segunda frente de avaliação, descritos a seguir.

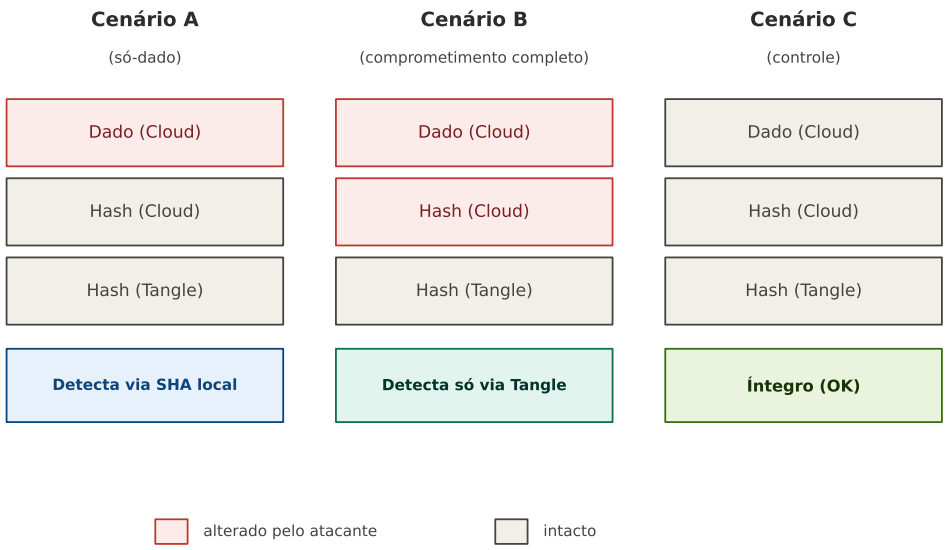


Figura 3 – Cenários de adulteração e camada responsável pela detecção.

Fonte: Elaborado pelo autor (2026).

3.9.2 Avaliação sistemática da integridade

A segunda frente avalia, de forma controlada e em volume, a propriedade central do trabalho: a detecção de adulteração. Para assegurar reprodutibilidade e cobertura dos cenários de ataque, a avaliação incide sobre o processamento, a ancoragem e a verificação, etapas que independem da origem física do dado, e essa escolha metodológica é declarada explicitamente.

Foram registrados 150 eventos, processados e ancorados no Tangle. Sobre o conjunto, definiram-se três cenários, aplicados diretamente sobre os registros armazenados na *Cloud*:

1. Cenário A (adulteração de dado): altera-se o registro armazenado sem atualizar o resumo correspondente. A divergência entre o resumo recalculado e o resumo armazenado torna a adulteração detectável localmente, sem necessidade do Tangle.

2. Cenário B (comprometimento consistente): altera-se o registro e recalcula-se o resumo armazenado, de modo que a *Cloud* permaneça internamente consistente. Nesse caso, a verificação local não acusa divergência, e somente a comparação com o resumo imutável ancorado no Tangle revela a adulteração. Este cenário é o que demonstra a necessidade do *ledger*.

3. Cenário C (controle): registros mantidos íntegros, que devem ser verificados como válidos. Este grupo mede a taxa de falsos positivos.

Aplicaram-se 30 adulterações no Cenário A, 30 no Cenário B e mantiveram-se 90 registros como controle. Para cada registro, determinou-se se a adulteração foi detectada e por qual mecanismo: comparação local de resumo ou comparação com o registro ancorado, permitindo quantificar tanto a taxa de detecção quanto a taxa de falsos positivos. Em todos os cenários, a manipulação restringiu-se aos dados na *Cloud*, sem qualquer alteração no Tangle, em conformidade com a hipótese de um adversário que controla o armazenamento, mas não o *ledger*.

3.9.3 Métrica de latência

Paralelamente, mediu-se a latência de processamento na camada *Fog* para cada leitura, registrada pelo próprio nó *Fog* como o intervalo entre a recepção da mensagem e a conclusão da gravação na *Cloud*. Reitera-se que essa métrica não inclui o tempo de transmissão entre o dispositivo e a borda nem o tempo de confirmação no Tangle, conforme justificado na Seção 3.9.

3.10 Procedimentos de Coleta de Dados

A avaliação foi conduzida em uma sessão controlada, com o ambiente reinicializado para garantir que os dados analisados fossem exclusivos do experimento. O nó *Fog* foi posto em execução e, em seguida, o gerador publicou os 150 eventos no tópico MQTT, no formato de mensagem descrito na Seção 3.9. Cada evento foi processado pela camada *Fog*, que calculou o resumo criptográfico, registrou a latência de processamento e gravou o dado na *Cloud* e, de forma assíncrona, teve seu resumo ancorado no Tangle. Aguardou-se a conclusão da ancoragem de todos os eventos, confirmada pela presença dos respectivos identificadores de bloco, antes da aplicação dos cenários de adulteração. A demonstração de operação com o dispositivo físico foi realizada em sessão separada, para não interferir na composição controlada do conjunto sistemático.

3.11 Métricas e tratamento dos dados

A avaliação apoia-se em duas métricas objetivas, definidas a seguir.

A primeira é a latência de processamento na camada *Fog*, medida em milissegundos como o intervalo entre a recepção da mensagem pelo nó *Fog* e a conclusão da gravação do dado na *Cloud*. Sobre o conjunto de medições, calcularam-se medidas de posição e dispersão: média, mediana, desvio-padrão, valores mínimo e máximo e os percentis 95 e 99, adotando-se a mediana como medida central de referência, por ser menos sensível a valores extremos associados à inicialização do sistema.

A segunda é a detecção de adulteração, avaliada sobre os três cenários definidos na Seção 3.9. Para cada registro, registrou-se se a adulteração foi detectada e por qual mecanismo: comparação local de resumo ou comparação com o registro ancorado no Tangle, bem como a ocorrência de falsos positivos no grupo de controle. A partir dessas contagens, computaram-se a taxa de detecção e a taxa de falsos positivos.

O tratamento e a sumarização dos dados foram realizados com o auxílio das bibliotecas `pandas` (agregação) e `matplotlib` (visualização) da linguagem Python.

3.12 Critérios de análise dos resultados

Os resultados obtidos são analisados à luz das duas métricas definidas na Seção 3.11, segundo os critérios descritos a seguir, sem que se estabeleçam, a priori, valores de referência numéricos coerente com a natureza exploratória de uma prova de conceito.

A detecção de adulteração é analisada de forma desagregada por cenário, distinguindo-se o mecanismo responsável por cada detecção. Interessa, em particular, verificar se há cenários em que a adulteração escapa à verificação local e só é revelada pela comparação com o resumo ancorado no Tangle, situação que evidenciaria empiricamente o papel do *ledger* como âncora externa. A interpretação considera conjuntamente a taxa de detecção e a taxa de falsos positivos, uma vez que uma elevada taxa de detecção isolada careceria de significado caso o sistema também sinalizasse registros íntegros como adulterados.

A latência de processamento na camada *Fog* é analisada por suas medidas de posição e dispersão, adotando-se a mediana como referência central, por sua robustez a valores extremos. A análise busca caracterizar a ordem de grandeza do processamento na borda e verificar sua consistência ao longo da execução, identificando e discutindo eventuais valores atípicos. Ressalta-se que, por se tratar de uma prova de conceito sem comparação controlada com uma configuração exclusivamente em *Cloud*, a latência é interpretada em termos absolutos e qualitativos, e não como evidência de superioridade quantitativa sobre outra arquitetura.

Por fim, a análise contempla a operação fim a fim do sistema, de natureza qualitativa, verificando se a arquitetura cumpre o fluxo projetado da aquisição no dispositivo à ancoragem no *ledger* e à verificação em conformidade com os objetivos estabelecidos.

3.13 Limitações da metodologia

A metodologia adotada apresenta limitações que delimitam o alcance das conclusões e que devem ser consideradas na interpretação dos resultados.

- Ambiente controlado: os testes foram realizados em laboratório, e não em uma planta industrial real, sujeita a variáveis imprevisíveis como ruído elétrico, interferência e falhas intermitentes de rede.
- Centralização da rede privada: a rede Tangle empregada é privada e utiliza um *coordinator* único, o que representa uma centralização parcial da confiança. A plena descentralização exigiria a rede pública, que, por sua vez, passou a cobrar taxas após a migração para o protocolo Rebased (NZAKUNA; PACIELLO *et al.*, 2025).
- Escala limitada: o volume avaliado é adequado a uma prova de conceito, mas insuficiente para caracterizar o desempenho do sistema sob carga elevada ou com grande número de dispositivos simultâneos.
- Infraestrutura de nó único: o nó *Fog* foi executado em um único computador, não representando um agrupamento (*cluster*) *Fog* distribuído geograficamente, como ocorreria em uma implantação real.
- Hardware de prova de conceito: o ESP32 CYD e o sensor LDR empregados são componentes de baixo custo, adequados à demonstração do mecanismo, mas não certificados para uso industrial. Uma implantação em planta real exigiria sensores robustecidos (por exemplo, sensores de violação baseados em RFID ou lacres eletrônicos industriais), com maior precisão, imunidade a ruído eletromagnético e suporte a protocolos industriais (como Modbus ou OPC-UA), além de compatibilidade com certificações de área classificada quando aplicável. Os parâmetros de calibração adotados (Seção 3.3.1) são específicos deste sensor e não devem ser extrapolados diretamente para outros dispositivos ou escalas sem nova calibração.

4 Resultados e Discussão

Este capítulo apresenta os resultados obtidos com a prova de conceito descrita no Capítulo 3. A avaliação foi conduzida sobre uma sessão controlada de 150 eventos, integralmente processados pela camada *Fog* e ancorados na rede Tangle privada. Os resultados são organizados em três frentes: a operação fim a fim do sistema, a detecção de adulteração e a latência de processamento.

4.1 Operação da arquitetura

A arquitetura operou de ponta a ponta conforme projetado. Cada evento publicado foi recebido pela camada *Fog*, teve seu resumo criptográfico calculado e o dado original gravado na camada *Cloud*, e teve esse resumo ancorado de forma assíncrona na rede Tangle, com o respectivo identificador de bloco registrado de volta. Ao final da sessão, a verificação de integridade confirmou que a totalidade dos 150 registros estavam ancorada e íntegra, sem ocorrência de erros de leitura no *ledger*. A operação com o dispositivo físico foi igualmente verificada: o ESP32 CYD, ao ter a iluminação sobre o sensor alterada, produziu corretamente a transição entre os estados LACRADO e VIOLADO, propagada pela cadeia até a ancoragem no Tangle e a visualização no painel.

As Figuras 4 a 6 ilustram a montagem física do dispositivo utilizado como detector de violação de lacre.

4.2 Detecção de adulteração

A avaliação da detecção de adulteração constitui o resultado central deste trabalho. Os três cenários definidos na metodologia foram aplicados sobre os registros já ancorados, e o resultado é sintetizado na Tabela 4 e na Figura 7.

Tabela 4 – Resultado dos cenários de adulteração

Cenário	**N**	**Detectados**	**Via SHA local**	**Via Tangle**
A — adulteração de dado	30	30	30	0
B — comprometimento consistente	30	28	0	28
C — controle (íntegro)	90	0	—	—

No Cenário A, em que o dado foi alterado sem a atualização do resumo armazenado, a totalidade das 30 adulterações foi detectada localmente, pela divergência entre o resumo recalculado e o resumo mantido na *Cloud*. Esse resultado confirma o funcionamento do

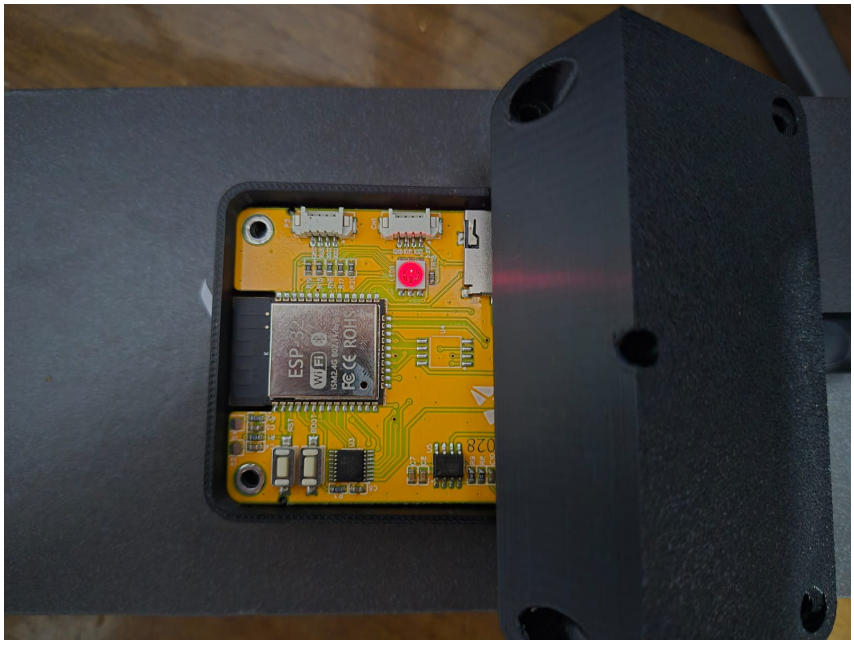


Figura 4 – Placa ESP32 CYD posicionada no invólucro, com o sensor de luminosidade (LDR) exposto antes do fechamento.

Fonte: Elaborado pelo autor (2026).

mecanismo de resumo criptográfico, mas não evidencia, por si só, a necessidade do *ledger*: a detecção ocorreria ainda que o Tangle não existisse.

O Cenário B é o que demonstra a contribuição do trabalho. Nele, o adversário alterou o dado e recalculou o resumo armazenado, de modo que a *Cloud* permaneceu internamente consistente. Como esperado, a verificação local não acusou divergência em nenhum dos casos: o resumo recalculado coincidia com o resumo armazenado. A detecção ocorreu exclusivamente pela comparação com o resumo imutável ancorado no Tangle, que o adversário não tem como alterar. Das 30 adulterações desse tipo, 28 foram detectadas somente por essa via, evidenciando empiricamente que, sob comprometimento consistente do armazenamento, somente a âncora externa é capaz de revelar a fraude.

No Cenário C, composto por 90 registros mantidos íntegros, nenhum foi sinalizado como adulterado, resultando em uma taxa de falsos positivos nula. Esse resultado é relevante porque confere significado à taxa de detecção: um sistema que sinalizasse adulteração indiscriminadamente também alcançaria detecção elevada, mas seria inútil na prática.

Considerados os dois cenários de adulteração em conjunto, a taxa global de detecção foi de 96,7% (58 de 60), com 0% de falsos positivos sobre o grupo de controle.

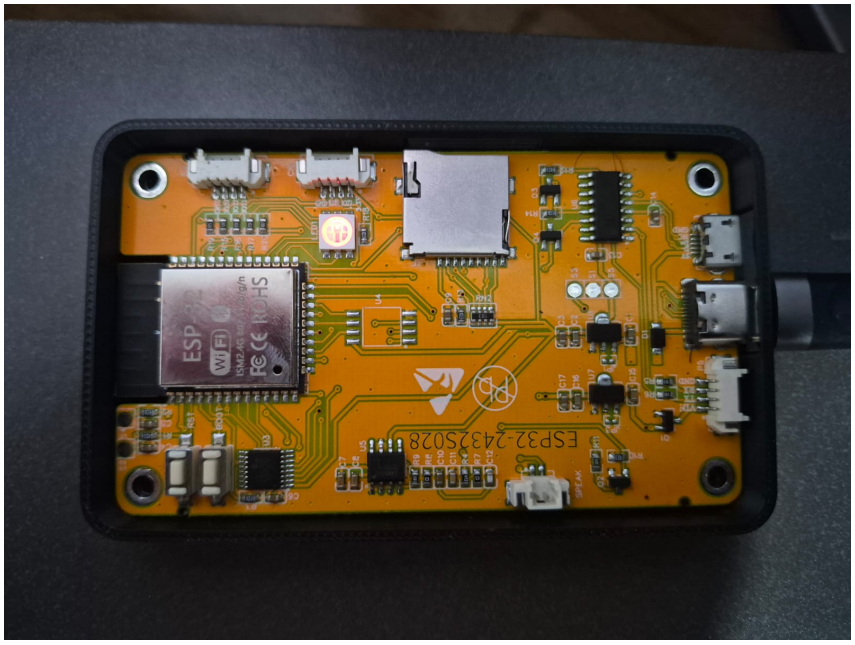


Figura 5 – Vista superior da placa ESP32 CYD, com o módulo Wi-Fi/Bluetooth, o LDR e as portas de conexão.

Fonte: Elaborado pelo autor (2026).

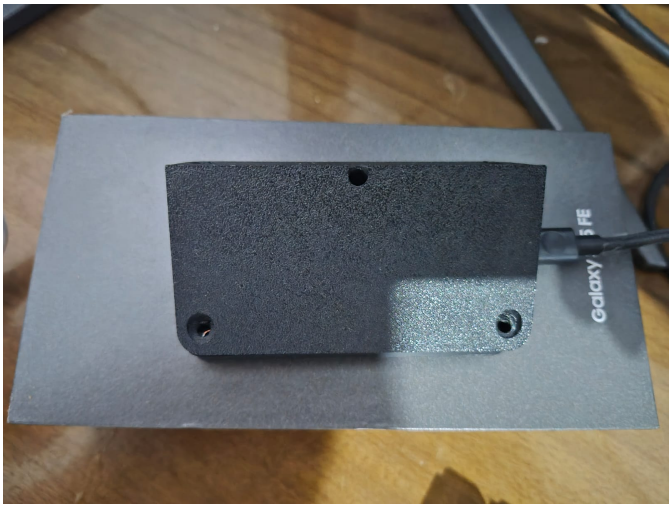


Figura 6 – Invólucro fechado sobre o dispositivo, representando o estado `LACRADO` utilizado na aquisição das leituras.

Fonte: Elaborado pelo autor (2026).

4.2.1 Discussão das duas adulterações não detectadas

As duas adulterações não detectadas no Cenário B merecem análise, por não decorrerem de falha do mecanismo de ancoragem. Os dois casos referem-se a registros cuja verificação não pôde ser concluída em razão da granularidade temporal de um segundo adotada no armazenamento: quando dois eventos compartilham o mesmo instante de

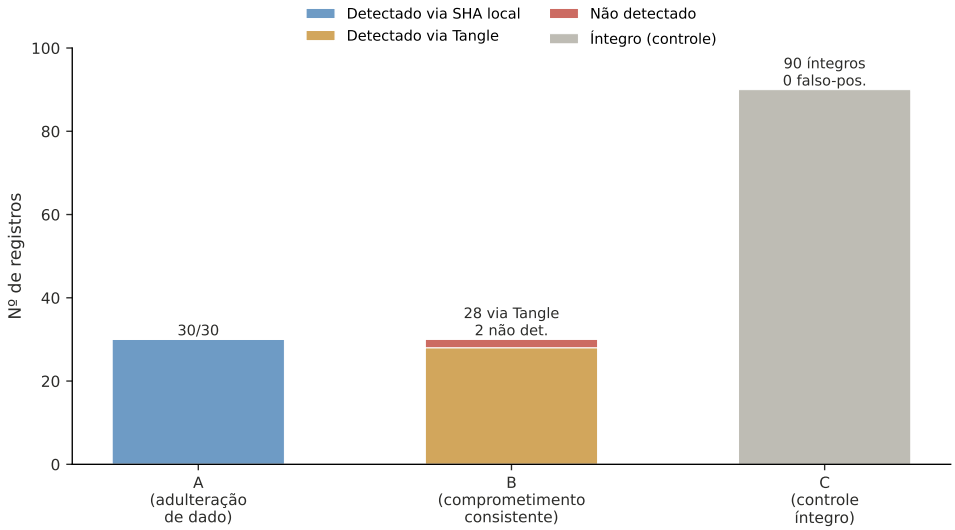


Figura 7 – Detecção de adulteração por cenário, segundo a camada responsável.

Fonte: Elaborado pelo autor (2026).

tempo, a estrutura de armazenamento os consolida, comprometendo a reconstrução individual da forma canônica daquele ponto específico. Trata-se, portanto, de uma limitação de implementação na geração e no registro dos eventos, e não de uma falha do princípio de verificação por âncora externa que, nos 28 casos restantes, operou conforme o esperado. A adoção de granularidade temporal mais fina (por exemplo, milissegundos) eliminaria essa colisão, e fica registrada como melhoria para trabalhos futuros.

4.3 Latência de processamento

A latência de processamento na camada *Fog* foi medida para os 150 eventos, e suas estatísticas são apresentadas na Tabela 5. Reitera-se que a métrica corresponde ao intervalo entre a recepção da mensagem pelo nó *Fog* e a conclusão da gravação na *Cloud*, não incluindo o tempo de transmissão do dispositivo até a borda nem o tempo de confirmação no Tangle.

Tabela 5 – Latência de processamento na camada *Fog* (n = 150)

Métrica	**Valor (ms)**
Mediana	5,67
Média	9,29
Desvio-padrão	16,61
Mínimo	4,97
Percentil 95	12,40
Percentil 99	120,04
Máximo	126,91

A mediana de 5,67 ms e o percentil 95 de 12,40 ms indicam que o processamento na borda ocorre de forma consistente na faixa de poucos milissegundos. Observa-se, contudo, que a média (9,29 ms) é superior à mediana e que o desvio-padrão é elevado (16,61 ms), reflexo de um pequeno número de valores extremos, cujo máximo atinge 126,91 ms. Esses valores concentram-se na fase inicial da execução, associados ao custo de inicialização das conexões com os serviços, e não representam o regime estacionário de operação. Por essa razão, adota-se a mediana como medida central representativa do desempenho do sistema. A Figura 8 apresenta a distribuição completa das medições.

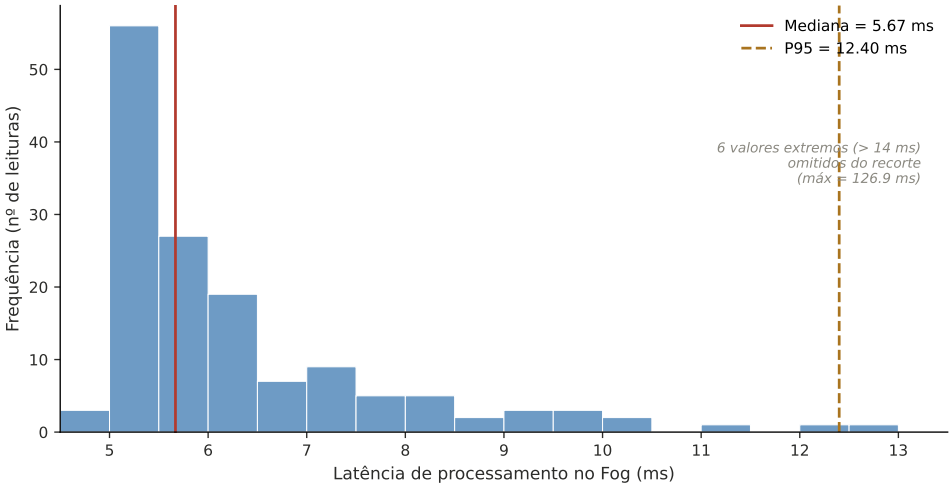


Figura 8 – Distribuição da latência de processamento na camada *Fog*.

Fonte: Elaborado pelo autor (2026).

A ordem de grandeza obtida processamento na faixa de poucos milissegundos na borda é consistente com a motivação da arquitetura em camadas discutida no Capítulo 2, na qual o processamento local visa a evitar a latência associada à comunicação com servidores remotos. Ressalva-se, todavia, que este trabalho não realizou uma comparação controlada com uma configuração exclusivamente em *Cloud*; os valores de latência de *Cloud* mencionados na literatura referem-se a estudos de terceiros e não a medições conduzidas neste experimento.

4.4 Síntese dos resultados

Em conjunto, os resultados demonstram a viabilidade técnica da arquitetura proposta. O sistema operou de ponta a ponta com hardware real; a detecção de adulteração atingiu 96,7%, com ausência de falsos positivos; e, sobretudo, o Cenário B evidenciou empiricamente o papel decisivo do *ledger* como âncora externa de confiança diante de um adversário que controla o armazenamento. A latência de processamento na borda manteve-se na faixa de poucos milissegundos. Esses resultados, obtidos em ambiente de

prova de conceito, sustentam as conclusões apresentadas no capítulo seguinte, observadas as limitações já declaradas.

5 Considerações Finais

Este capítulo encerra o trabalho, retomando os objetivos propostos, sintetizando os principais resultados e contribuições, e discutindo as limitações do estudo e as direções para trabalhos futuros.

Este trabalho propôs e demonstrou, por meio de uma prova de conceito, uma arquitetura em quatro camadas IoT, *Fog*, Tangle e *Cloud* voltada à garantia de integridade de dados de sensores contra adulteração retroativa, mesmo sob comprometimento do ambiente de armazenamento. A motivação partiu de uma lacuna específica: enquanto os mecanismos tradicionais de segurança protegem os dados em trânsito, a integridade dos registros já armazenados permanece vulnerável quando o próprio operador da infraestrutura é capaz de alterá-los, situação relevante em contextos nos quais o dado funciona como prova.

5.1 Atendimento aos objetivos

O objetivo geral foi alcançado. A arquitetura de quatro camadas foi implementada e operou de ponta a ponta, com aquisição em um dispositivo ESP32, processamento e cálculo de resumo criptográfico na camada *Fog*, ancoragem em rede Tangle privada e armazenamento na camada *Cloud*. Quanto aos objetivos específicos, o primeiro foi cumprido com a implementação integral da arquitetura e a demonstração de seu funcionamento com hardware real; o segundo, com o registro dos resumos SHA-256 no Tangle e a verificação por comparação com o registro imutável ancorado; e o terceiro, com a avaliação sistemática da detecção de adulteração e a mensuração da latência de processamento na camada *Fog*.

5.2 Principais resultados e contribuição

O principal resultado do trabalho é a demonstração empírica de que a ancoragem de resumos em um *ledger* distribuído cumpre o papel de âncora externa de confiança. Sob o cenário de comprometimento consistente do armazenamento no qual o adversário altera tanto o dado quanto o resumo mantido na *Cloud*, a verificação local mostrou-se incapaz de detectar a fraude, e a detecção ocorreu exclusivamente por meio da comparação com o resumo imutável registrado no Tangle. Esse resultado evidencia que a contribuição do *ledger* não reside em detectar a adulteração em si, função do resumo criptográfico, mas em fornecer uma referência que o operador do armazenamento não pode reescrever.

A avaliação registrou taxa de detecção de 96,7% com ausência de falsos positivos, e a latência de processamento na borda manteve-se na faixa de poucos milissegundos.

Um achado relevante, de natureza qualitativa, diz respeito à própria tecnologia empregada. Ao longo do desenvolvimento, constatou-se que a migração da rede pública IOTA para o protocolo Rebased, em 2025, reintroduziu taxas de transação, descaracterizando a ausência de custos historicamente apontada como principal vantagem do Tangle para aplicações IoT. Em consequência, a propriedade de custo nulo só se sustenta, atualmente, em implantações privadas ou permissionadas como a adotada neste trabalho, ao custo de uma centralização parcial da confiança no *coordinator*. Essa tensão entre ausência de custos e descentralização constitui uma contribuição de discussão deste trabalho, frequentemente ausente em abordagens que tratam o Tangle apenas por suas características originais.

5.3 Limitações

Os resultados devem ser interpretados à luz das limitações já discutidas. A avaliação foi conduzida em ambiente de laboratório e em escala compatível com uma prova de conceito, não caracterizando o desempenho do sistema sob carga elevada ou em planta industrial real. A avaliação sistemática empregou eventos gerados por software, e a infraestrutura foi executada em um único computador, sem distribuição geográfica dos nós *Fog*. A rede Tangle utilizada é privada e centralizada em um *coordinator* único. O ESP32 CYD e o sensor LDR utilizados são componentes de baixo custo voltados à demonstração do mecanismo, não certificados para uso industrial; uma implantação real exigiria sensores robustecidos e compatíveis com protocolos e certificações industriais, além de nova calibração para os parâmetros específicos do dispositivo empregado. Por fim, a granularidade temporal de um segundo adotada no armazenamento ocasionou colisões pontuais, responsáveis pelos dois casos não verificados no cenário de comprometimento consistente.

5.4 Trabalhos futuros

Como desdobramentos, identificam-se as seguintes direções.

- Granularidade temporal mais fina: a adoção de granularidade na ordem de milissegundos eliminaria as colisões observadas e permitiria a verificação íntegra de todos os registros.
- Avaliação em escala ampliada: um maior volume de eventos e múltiplos dispositivos simultâneos permitiria caracterizar o desempenho sob carga.

- Distribuição geográfica da camada *Fog*: a execução da camada *Fog* em um agrupamento distribuído geograficamente aproximaria a implementação de um cenário industrial realista.
- Operação concorrente: a avaliação da latência de processamento sob operação concorrente de múltiplos dispositivos publicando simultaneamente permitiria verificar o comportamento do sistema sob carga distribuída, aproximando o experimento das condições de uma implantação real com dezenas de sensores ativos.
- Rede pública sob o protocolo Rebased: a investigação do uso da rede pública IOTA sob o protocolo Rebased, com a respectiva análise de custos, contraporia o modelo privado adotado.
- Aplicação a sensores industriais reais: a aplicação da arquitetura a sensores de fato sujeitos a requisitos de auditoria, como medição de emissões ou parâmetros de cadeia do frio, permitiria avaliar a proposta em um de seus cenários de uso mais pertinentes.

Referências

ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The Internet of Things: A survey. *Computer Networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. DOI: 10.1016/j.comnet.2010.05.010. Disponível em: https://www.cs.mun.ca/courses/cs6910/IoT-Survey-Atzori-2010.pdf. Acesso em: 18 mar. 2026. Citado 1 vez na página 15.

BANCO NACIONAL DE DESENVOLVIMENTO ECONÔMICO E SOCIAL; MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES. *Internet das Coisas: um plano de ação para o Brasil — Produto 2: Roadmap Tecnológico*. Brasília, 2017. Versão 2.0. Disponível em: https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/transformacaodigital/arquivosinternetdascoisas. Acesso em: 15 abr. 2026. Citado 2 vezes na página 13.

BONOMI, Flavio *et al.* Fog computing and its role in the internet of things. *In:* PROCEEDINGS of the First Edition of the MCC Workshop on Mobile Cloud Computing. ACM, 2012. p. 13–16. DOI: 10.1145/2342509.2342513. Disponível em: https://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf. Acesso em: 15 abr. 2026. Citado 3 vezes nas páginas 16, 18.

BRASIL. MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES. *Plano Nacional de Internet das Coisas — Decreto n.º 9.854, de 25 de junho de 2019*. 2019. Disponível em: https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/transformacaodigital/internet-das-coisas. Acesso em: 21 abr. 2026. Citado 1 vez na página 12.

BURHAN, Muhammad *et al.* A Comprehensive Survey on the Cooperation of Fog Computing Paradigm-Based IoT Applications: Layered Architecture, Real-Time Security Issues, and Solutions. *IEEE Access*, IEEE, v. 11, p. 73006–73057, 2023. DOI: 10.1109/ACCESS.2023.3294479. Disponível em: https://osuva.uwasa.fi/bitstream/handle/10024/16325/Osuva_Burhan_Alam_Arsalan_Rehman_Anwar_Faheem_Ashraf_2023.pdf. Acesso em: 27 mar. 2026. Citado 7 vezes nas páginas 11, 15, 17, 19, 22.

BUTERIN, Vitalik. *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. 2014. Disponível em: https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf. Acesso em: 11 abr. 2026. Citado 1 vez na página 11.

CARELLI, Alberto *et al.* Enabling Secure Data Exchange through the IOTA Tangle for IoT Constrained Devices. *Sensors*, MDPI, v. 22, n. 4, p. 1384, 2022. DOI: 10.3390/s22041384. Disponível em: https://pmc.ncbi.nlm.nih.gov/articles/PMC8963087/. Acesso em: 3 abr. 2026. Citado 2 vezes nas páginas 12, 14.

DONG, Serafina *et al.* Characterizing IOTA Tangle with Empirical Data. *In:* IEEE Global Communications Conference (GLOBECOM). IEEE, 2020. Disponível em: https://dsg.

tuwien.ac.at/team/sd/papers/Globecom_2020_SD_ohne_ISBN.pdf. Acesso em: 13 mar. 2026. Citado 1 vez na página 12.

FAN, Chuanzheng *et al.* Performance Analysis of the IOTA DAG-Based Distributed Ledger. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, ACM, v. 7, n. 1-4, p. 1–30, 2021. DOI: 10.1145/3485188. Disponível em: https://dl.acm.org/doi/10.1145/3485188. Acesso em: 29 mar. 2026. Citado 1 vez na página 12.

FERRARIS, Davide; FERNANDEZ-GAGO, Carmen; LOPEZ, Javier. A survey on IoT trust model frameworks. *The Journal of Supercomputing*, Springer, 2023. Disponível em: https://www.nics.uma.es/wp-content/papers/surveyIoTrust2023.pdf. Acesso em: 26 mar. 2026. Citado 3 vezes nas páginas 11, 16, 18.

GOMES, Eduarda Oliveira. *Utilizando a rede Tangle para comunicação e iteração de dispositivos IoT dentro de um contexto de Economia das Coisas*. 2021. Monografia (Trabalho de Conclusão de Curso) – Universidade Federal de Santa Catarina, Florianópolis. Disponível em: https://repositorio.ufsc.br/handle/123456789/225838. Acesso em: 11 fev. 2026. Citado 3 vezes nas páginas 13, 14, 16.

IARAS EDITORIAL BOARD. A Survey on IoT Security: Attacks, Threat Classification, and Countermeasures. *International Journal of Information Technology and Web Engineering*, 2025. Disponível em: https://www.iaras.org/iaras/filedownloads/ijitws/2025/022-0010(2025).pdf. Acesso em: 22 abr. 2026. Citado 6 vezes nas páginas 11, 13, 15, 16.

MARGARIA, Davide; VESCO, Andrea; CARELLI, Alberto. *Building Trust in Data for IoT Systems*. 2024. arXiv: 2403.02225. Disponível em: https://arxiv.org/abs/2403.02225. Acesso em: 19 abr. 2026. Citado 1 vez na página 14.

MARTINEZ, Guilherme Gabriel; RODRIGUES, Cintia Kalyla. Blockchain and Tangle: The Transaction Security in the IoT Ecosystem. *Revista de Sistemas de Computação*, UNIFACS, 2020. Disponível em: https://revistas.unifacs.br/index.php/rsc/article/download/6565/4032. Acesso em: 5 fev. 2026. Citado 2 vezes nas páginas 13, 16.

MELL, Peter; GRANCE, Timothy. *The NIST Definition of Cloud Computing*. Gaithersburg, MD, 2011. Disponível em: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf. Acesso em: 12 fev. 2026. Citado 1 vez na página 17.

MÜLLER, Sebastian *et al.* Tangle 2.0 Leaderless Nakamoto Consensus on the Heaviest DAG. *IEEE Access*, IEEE, v. 10, p. 105807–105842, 2022. DOI: 10.1109/ACCESS.2022.3211019. Disponível em: https://arxiv.org/abs/2205.02177. Acesso em: 18 fev. 2026. Citado 2 vezes nas páginas 12, 20.

NAKAMOTO, Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 10 fev. 2026. Citado 1 vez na página 11.

NZAKUNA, Steve; PACIELLO, Giulio *et al.* From IOTA Tangle 2.0 to Rebased: A Comparative Analysis of Protocols. *Systems*, MDPI, 2025. Disponível em: https://pmc.ncbi.nlm.nih.gov/articles/PMC12157984/. Acesso em: 14 abr. 2026. Citado 4 vezes nas páginas 12, 21, 26, 34.

PAJOOH, Houshyar Honar *et al.* Blockchain and IoT Integration: A Systematic Survey. *Sensors*, MDPI, v. 21, n. 17, p. 5982, 2021. DOI: 10.3390/s21175982. Disponível em: https://pdfs.semanticscholar.org/3830/57f972b11b99cbc8c0d3e6c47170e9d95c1c.pdf. Acesso em: 7 fev. 2026. Citado 2 vezes nas páginas 12, 22.

POPOV, Serguei. *The Tangle*. 2018. IOTA Foundation Whitepaper. Disponível em: https://api-new.whitepaper.io/documents/pdf?id=SkUUXUdaf. Acesso em: 13 abr. 2026. Citado 8 vezes nas páginas 12, 14, 19, 21.

SEALEY, Nathan; AIJAZ, Adnan; HOLDEN, Ben. IOTA Tangle 2.0: Toward a Scalable, Decentralized, Smart, and Autonomous IoT Ecosystem. *In:* 2022 IEEE International Conference on Smart Networks and Communications (SmartNets). IEEE, 2022. DOI: 10.48550/arXiv.2209.04959. Disponível em: https://arxiv.org/abs/2209.04959. Acesso em: 17 mar. 2026. Citado 1 vez na página 12.

SILVANO, Willian Friderichs; MARCELINO, Roderval. IOTA Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future Generation Computer Systems*, Elsevier, v. 112, p. 307–319, 2020. DOI: 10.1016/j.future.2020.05.047. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0167739X19329048. Acesso em: 10 abr. 2026. Citado 2 vezes nas páginas 13, 16.

SILVANO, Willian Friderichs; MARCELINO, Roderval; VIGIL, Martin. Tecnologia Blockchain-IOTA aplicada à rastreabilidade de produtos. *Revista de Tecnologia e Sociedade*, UTFPR, 2021. Disponível em: https://periodicos.utfpr.edu.br/rts/article/viewFile/12091/8100. Acesso em: 6 abr. 2026. Citado 1 vez na página 12.

SOUZA, Anderson *et al.* Mercado de Espectro com IOTA. *In:* WORKSHOP em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain/SBC). SBC, 2022. Disponível em: https://sol.sbc.org.br/index.php/wblockchain/article/view/21468. Acesso em: 23 abr. 2026. Citado 1 vez na página 12.

VALDEZ, Andres *et al.* Exploring the Synergy of Fog Computing, Blockchain, and Federated Learning for IoT: A Systematic Literature Review. *IEEE Access*, IEEE, 2024. Disponível em: https://bpb-eu-w2.wpmucdn.com/blogs.bristol.ac.uk/dist/f/1002/files/2024/09/SLR_Paper_BC_FC_FL__Full_version__IEEE_access.pdf. Acesso em: 15 fev. 2026. Citado 4 vezes nas páginas 11, 18, 19.

WU, Jing *et al.* A Blockchain-Based Secure Data Transaction and Privacy Preservation Scheme in IoT System. *Sensors*, MDPI, v. 25, n. 15, p. 4854, 2025. DOI: 10.3390/s25154854. Disponível em: https://pmc.ncbi.nlm.nih.gov/articles/PMC12349439/. Acesso em: 27 jul. 2026. Citado 1 vez na página 16.

ZUBAYDI, Haider Dhia; VARGA, Pal; MOLNAR, Sandor. Leveraging Blockchain Technology for Ensuring Security and Privacy Aspects in Internet of Things: A Systematic Literature Review. *Sensors*, MDPI, v. 23, n. 2, p. 788, 2023. DOI: 10.3390/s23020788. Disponível em: https://pmc.ncbi.nlm.nih.gov/articles/PMC9867322/. Acesso em: 4 mar. 2026. Citado 3 vezes nas páginas 12, 16.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Declaro que, na elaboração deste trabalho, foi utilizada a ferramenta de Inteligência Artificial (IA) generativa Claude, da Anthropic, como apoio a tarefas específicas, tais como: revisão textual e gramatical, formatação do documento em LaTeX e apoio à depuração de trechos de código-fonte. A concepção da pesquisa, a definição da arquitetura proposta, a implementação central do sistema, a condução dos experimentos, a análise crítica dos resultados e as conclusões apresentadas são de inteira autoria e responsabilidade do autor deste trabalho.