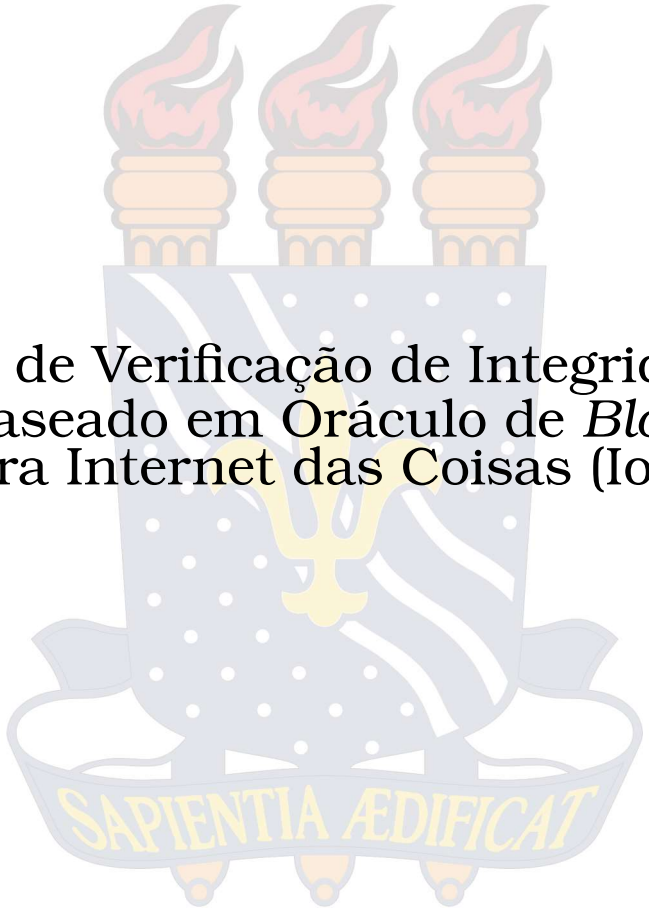
UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Sistema de Verificação de Integridade de Dados Baseado em Oráculo de *Blockchain* para Internet das Coisas (IoT)



Francisco Erberto de Sousa

João Pessoa
2023

Catalogação na publicação
Seção de Catalogação e Classificação

S725s Sousa, Francisco Erberto de.
Sistema de verificação de integridade de dados baseado em oráculo de blockchain para internet das coisas (IoT) / Francisco Erberto de Sousa. - João Pessoa, 2023.
50 f. : il.

Orientação: Cleonilson Protásio de Souza.
Dissertação (Mestrado) - UFPB/CEAR.

1. Segurança da informação. 2. Oráculo - Blockchain. 3. Rede de sensores sem fio. I. Souza, Cleonilson Protásio de. II. Título.

UFPB/BC CDU 004.056(043)

Elaborado por MAGNOLIA FELIX DE ARAUJO - CRB-15/883

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

A Comissão Examinadora, abaixo assinada, aprova a Dissertação

SISTEMA DE VERIFICAÇÃO DE INTEGRIDADE DE DADOS BASEADO EM ORÁCULO DE *BLOCKCHAIN* PARA INTERNET DAS COISAS (IOT)

Elaborada por:

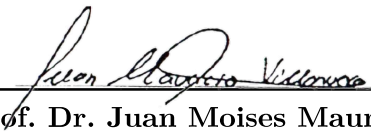
FRANCISCO ERBERTO DE SOUSA

Como requisito parcial para obtenção do grau de

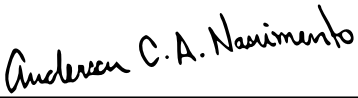
Mestre em Engenharia Elétrica

COMISSÃO EXAMINADORA

Prof. Dr. Cleonilson Protásio de Souza
Orientador



Prof. Dr. Juan Moises Mauricio Villanueva
Avaliador Interno



Prof. Dr. Anderson Clayton Alves Nascimento
Avaliador Externo

João Pessoa

2023

Francisco Erberto de Sousa

SISTEMA DE VERIFICAÇÃO DE INTEGRIDADE DE DADOS BASEADO EM ORÁCULO DE *BLOCKCHAIN* PARA INTERNET DAS COISAS (IOT)

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE da Universidade Federal da Paraíba - UFPB como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Cleonilson Protásio de Souza

João Pessoa

2023

Dedico este trabalho à minha família,
que sempre foi minha base durante toda a vida,
e também a todos os que sempre me apoiaram.

RESUMO

SISTEMA DE VERIFICAÇÃO DE INTEGRIDADE DE DADOS BASEADO EM ORÁCULO DE *BLOCKCHAIN* PARA INTERNET AS COISAS (IoT)

A sociedade vive um processo de evolução tecnológica em que surgem alguns fatores que demandam atenção. Um desses fatores é a geração de dados que tem crescido exponencialmente e presente nos mais diversos ambientes. Com a modernização das tecnologias de comunicação, a Internet das Coisas (IoT) tem ganhado espaço e consequentemente vem produzindo em grande escala o número de dados. As aplicações podem gerar dados de interesse público ou particular, mas um princípio de segurança que, independentemente do público-alvo do sistema IoT, precisa ser monitorado é o da integridade, isso porque os sistemas aguardam valores de medições íntegros dos sensores para poder efetuar suas tomadas de decisões corretas. Diante desse contexto, este trabalho tem como objetivo desenvolver para o cenário de IoT uma solução, capaz de garantir a integridade dos dados medidos, e baseado em oráculo de *Blockchain*, que é uma entidade capaz de garantir que todo dado que entra no *Blockchain* é verdadeiro. Com relação aos resultados obtidos neste trabalho, foi desenvolvido o oráculo de *Blockchain* proposto e que, nos testes experimentais alcançou 100% de aproveitamento e podendo chegar à conclusão de que todos os protocolos de segurança implementados, não apenas evitam o envio de dados com erro de medição para a rede *Blockchain*, mas também é aplicado os métodos de correção quando falhas são identificadas.

Palavras-chave: *Blockchain*, Oráculo, Rede de Sensores Sem Fio, Segurança da Informação.

ABSTRACT

BLOCKCHAIN-ORACLE-BASED DATA INTEGRITY VERIFICATION SYSTEM FOR INTERNET OF THINGS (IOT)

Society is experiencing a process of technological evolution in which some factors arise some demand attention. One of these factors is the generation of data that has been exponentially created and are present in the most environments. With the modernization of communication technologies, the Internet of Things (IoT) has gained space and consequently has been producing large-scale data. Applications can generate data of public or private interest, but a security principle that, regardless of the target audience of the IoT system, needs to be monitored is that of integrity, because the systems await complete measurement values from the sensors in order to be able to simulate their Making the right decisions. In this context, this work aims to develop a solution for the IoT scenario, capable of guaranteeing the integrity of the measured data, and based on a Blockchain oracle, which is an entity capable of guaranteeing that all data that enters the Blockchain is true. With regard to the results obtained in this work, the proposed Blockchain oracle was developed and which, in the experimental tests, reached 100% of use and allowing the conclusion that all the implemented security protocols, not only avoid the sending of data with error from to the Blockchain network, but correction methods are also applied when measurement failures are identified.

Keywords: *Blockchain*, Oracle, Wireless Sensor Network, Information Security.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

CA	*Autoridade de Certificação*
CO	*Monóxido de Carbono*
IBM	*International Business Machines*
ICPs	*Infraestruturas de Chave Pública*
IoT	*Internet das Coisas*
LAN	*Local area network*
NIST	*National Institute of Standards and Technology*
PPS	*Problema do ponto singular*
PoW	*Proof of work*
PoS	*Proof of stake*
RSSF	*Redes de Sensores sem Fio*
TIC	*Tecnologia da Informação e Comunicação*
TTP	*Trusted Third Party*
WAN	*Wide area network*
WLAN	*Wireless Local Area Network*

SUMÁRIO

1 INTRODUÇÃO

Atualmente, a sociedade vive um momento em que a conectividade está cada vez mais presente. Seja ela por meio dos dispositivos móveis ou aplicações para Internet das Coisas (IoT). A partir do avanço dessas ferramentas surgiu um aumento exponencial e massivo de dados. Isso gera a necessidade da implantação de sistemas que assegurem o cumprimento dos pilares da segurança da informação.

Nesse contexto, segundo dados das Nações Unidas (ONU), a expectativa é de que 7 bilhões de pessoas irão viver em áreas urbanas em 2050. Com a evolução dos centros urbanos, surgiu um novo conceito de cidades, as *Smart Cities*, que segundo Giffinger et al. (2007) propõe uma estrutura hierárquica Inteligente, em que o topo da pirâmide é ocupado pela Economia Inteligente, seguida de Governança Inteligente, que logo abaixo vem acompanhado de Ambiente, Mobilidade e Modo de Vida Inteligente. Além desse modelo centrado no cidadão, outra definição utilizada, e que é adotada pela IBM, tem o foco mais tecnológico e focado em uso de Tecnologia da Informação e Comunicação (TIC) nessa estrutura hierárquica, que por sua vez, tem apresentado grandes avanços nos dispositivos e conexões de redes, que logo contribuem para o funcionamento dos grandes centros urbanos.

Os modelos existentes de redes são dos mais variados, como, por exemplo, *Local Area Network* (LAN), *Wide Area Network* (WAN), *Wireless Local Area Network* (WLAN) e Conexão móvel. No entanto, aplicações para IoT requerem redes de longo alcance e com dispositivos de baixo consumo de energia. Logo, as *Low Power Wide Area Network* (LPWAN) são as mais utilizadas nesse contexto.

Além das redes, o ecossistema atual da internet tornou possível trocar dados por meio de praticamente qualquer modelo de rede. No entanto, quando se trata do tráfego de informação, desde que foram implementados os primeiros protocolos de segurança de rede, as maiores preocupações que os usuários têm são: se a mensagem não será alterada e se essa estará disponível apenas para as partes interessadas.

Manter os pilares da segurança da informação nas cidades inteligentes é um desafio que requer um novo modelo e/ou a junção de técnicas capazes de garantir que a integridade, a autenticidade e a legitimidade dos dados sejam mantidas.

1.1 MOTIVAÇÃO

Com a revolução industrial do século XVIII, a população iniciou o processo de migração para os centros urbanos em busca de melhores condições de vida. Consequentemente, a falta de planejamento acabou gerando inúmeros problemas para o meio ambiente e para a sociedade. Na busca de minimizar os problemas acometidos pelo crescimento não

planejado das cidades, estudiosos do século XX iniciaram o debate sobre a possibilidade da criação de Cidades Inteligentes.

Dentro dessa perspectiva, as Cidades Inteligentes (ou *Smart Cities*) surgem como alternativa para viabilizar aplicações que visam solucionar diversos problemas associadas aos centros urbanos, como por exemplo, monitoramento da poluição atmosférica, tráfego, energia, água, serviços e diversos outros. Desta maneira, as cidades inteligentes constituem cenários urbanos que utilizam TIC para melhorar a infraestrutura e a qualidade de vida dos cidadãos (AGUIAR, 2018).

Esse novo modelo de cidade emergiu dos problemas enfrentados pela população, como a poluição do meio ambiente, uma economia pouco sustentável e com uma baixa segurança nos dados e informações gerados. Um exemplo que pode ser citado para ilustrar tais problemas é o seguinte: uma empresa adquire um contrato por meio de uma licitação para gerenciar um aterro sanitário dentro dos padrões estabelecidos pelos órgãos competentes. No entanto, os modelos atuais de monitoramento não são tão eficientes, porque ainda dependem de processos manuais para captação, processamento e envio das informações desse aterro. Logo, isso pode acarretar inúmeras problemas, como: falha no envio, ataques e erros de precisão.

O setor de contratos vem passando por uma modernização que parte do princípio da não participação de um terceiro individuo nas transações geradas. Os Contratos Inteligentes (ou *Smart Contract*) buscam descentralizar os envios e armazenamento dos dados. Com o surgimento desse novo paradigma, Nakamoto (2008) propôs um sistema público de pagamento descentralizado para transações financeiras baseado em *Blockchain*.

Segundo Goa (2018), uma *Blockchain* propõe resolver problemas de confiança em sistemas distribuídos descentralizados, sendo a aplicação mais comum dessa tecnologia a mineração de criptomoedas. Uma rede dessa natureza é composta por diferentes tecnologias, como: armazenamento distribuído de dados, protocolo de rede *peer-to-peer* (P2P) e protocolo de consenso.

O objetivo deste trabalho de Dissertação é o estudo, desenvolvimento e a implementação de uma solução no cenário de IoT e *Blockchain* a fim de garantir a integridade de dados medidos por sensores, por meio do uso de oráculos de *Blockchain*, uma entidade de verificação de integridade de dados e usada como porta de entrada em uma *Blockchain*. A aplicação-alvo deste trabalho será voltada para monitoramento da qualidade do ar. Porém, a solução proposta pode ser utilizada em diversas aplicações que usem sensores sem fio.

1.2 TRABALHOS RELACIONADOS

Em Jr et al. (2021), foi proposto um modelo para aquisição de dados de uma rede *Blockchain* que pode ser implementada sem requerer um sistema de administração único.

Além disso, o modelo permite a geração de múltiplas cadeias de blocos.

Peters et al. (2018) usa rede *Blockchain* para medição de dados meteorológicos aplicando auditoria descentralizada, sistemas de cobrança de tarifas, com um mecanismo para autorização de software e a utilização de Infraestruturas de Chave Pública (ICP).

Miličević et al. (2022) afirma no seu trabalho que as propriedades do se assemelham com as características da metrologia inovadora, que são: rastreabilidade, imutabilidade e documentos legíveis por máquina. Esse trabalho aborda a solução partindo dos dispositivos IoT, que são os oráculos chegando à definição única de unidades de medida. O conceito de modelo de confiança resultante engloba a rastreabilidade vertical e horizontal dos resultados da medição (dados do oráculo), onde os padrões normativos e os requisitos legais são cruciais para a construção da confiança.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral deste trabalho consiste em implementar e analisar um sistema de medição em IoT, baseado em Redes de Sensores Sem Fio (RSSF) a partir do desenvolvimento de uma rede *Blockchain* com o uso de oráculo para garantir que dados medidos somente sejam armazenados em uma *Blockchain* se forem considerados reais ou verdadeiros.

1.3.2 Objetivos Específicos

Dentre os objetivos específicos desta proposta, pode-se destacar:

- Desenvolver um oráculo de *Blockchain* em Hardware para validar dados de medição;
- Implementar protocolos de verificação e validação de dados;
- Desenvolver um nó sensor de gás para RSSF baseado em oráculos; e
- Implementar uma rede *Blockchain* integrando dados de RSSF desenvolvida.

1.4 ORGANIZAÇÃO DO TEXTO

O presente trabalho está organizado em cinco capítulos, de acordo com a seguinte estrutura: no Capítulo 2 é apresentado o conceito de *Blockchain*, Oráculo, Segurança da Informação e Criptografia. No Capítulo 3 é abordado a metodologia utilizada para a implementação do sistema proposto. No Capítulo 4 são apresentados e discutidos os resultados de simulação para os cenários considerados, assim como os resultados experimentais. Por fim, as conclusões deste trabalho e os trabalhos futuros são descritos no Capítulo 5.

2 LEVANTAMENTO BIBLIOGRÁFICO E REFERENCIAL TEÓRICO

Neste capítulo são descritos o conceito de Cidades Inteligentes, Segurança da Informação, *Blockchain*, Oráculo e definições pertinentes relacionadas aos temas abordados nesta pesquisa. Também serão apresentadas as tecnologias utilizadas para compor a solução proposta.

2.1 CIDADES INTELIGENTES

O conceito de Cidades Inteligentes é bastante amplo e diverge entre os estudiosos e entidades, como já foi mencionado no capítulo anterior. Existem dois conceitos que são os mais citados no levantamento bibliográfico realizado, um com uma visão voltada para o cidadão, ou seja, todas as ações e tecnologias desenvolvidas buscam melhorar a qualidade de vida da sociedade, conhecida por ser **holística** e o outro, com uma visão mais tecnológica, cujo foco é em desenvolver soluções tecnológicas nos problemas de uma cidade, denominada **tecnocêntrica**.

Segundo Nam e Pardo (2011), a definição de Cidades Inteligentes foi centralizado em temas relacionados às Tecnologias de Informação e Comunicação (TICs), porém avançou de maneira progressiva para uma visão holística, considerando três fatores principais: tecnologia, pessoas e instituições.

O termo *Smart City* surgiu em 1992 a partir do livro de Gibson com o título *The Technopolis Phenomenon: Smart Cities, Fast Systems, Global Networks*, que tratou das questões do desenvolvimento urbano dependente da globalização e da inovação do ponto de vista econômico.

Para este trabalho foi considerado o modelo holístico, já que a aplicação desenvolvida possui o propósito de garantir seguridade das transações entre os usurários. Essa abordagem prioriza seis pilares capazes de resultar em soluções inteligentes, que são: *Smart Economy*, *Smart Governance*, *Smart Mobility*, *Smart People*, *Smart Environment* e *Smart Living*.

Na realidade, existem poucas cidades com o perfil descrito anteriormente. No entanto, esse tema tem se tornado alvo de grandes pesquisadores, ou seja, uma área que ainda existe inúmeras possibilidades de estudos. Segundo pesquisa realizada pela consultoria Roland Berger (BERGER, 2017) conclui-se que cerca de 41% das Cidades Inteligentes existentes estão situadas da Europa, 27% na Ásia, 24% na América do Norte e 8% nos demais continentes.

2.2 SEGURANÇA DA INFORMAÇÃO

A segurança da Informação surgiu com objetivo de proteger as informações de empresas ou pessoas, ou seja, se aplica a diversos cenários. Segundo Ferreira (2008), todo dado que tenha importância para o usuário do sistema é necessário que sejam utilizados mecanismos de proteção. Eles podem estar armazenados de duas maneiras: para o uso restrito ou exposta ao público para consulta ou aquisição.

2.2.1 Princípios de Segurança da Informação

Segurança e privacidade são princípios basilares de qualquer sistema de informação. Nos referimos a segurança como a combinação de Integridade, Disponibilidade e Confidencialidade. Normalmente é possível obter segurança usando uma combinação de autenticação, autorização e identificação. Esses conceitos são definidos a seguir (STALLINGS, 1995):

- **Integridade:** É a garantia que o dado não foi alterada, exceto por quem tem o direito de realizar estas alterações, ou seja, os indivíduos que são proprietários. Em relação ao *Blockchain* é a garantia de que os dados que estão inseridos nas transações não podem ser modificados intencionalmente ou falhas do sistema. Neste processo, entra a criptografia como mecanismo de verificação;
- **Disponibilidade:** esse princípio garante que os usuários do sistema sempre tenham acesso aos dados quando for necessário. Isso requer que tanto a infraestrutura de comunicação quanto as bases de dados possam ser utilizadas. No contexto deste trabalho é possível alcançar estes objetivos ao permitir que os usuários estabeleçam conexão com vários usuários, já que a rede é descentralizada e mantendo inúmeras cópias dos blocos na rede;
- **Confidencialidade:** Assim como o princípio de disponibilidade garante total acesso aos usuários autorizados sempre que for necessário, o de confidencialidade assegura que a informação não será obtida por pessoas não autorizadas. Isto é, apenas aqueles com os direitos e privilégios necessários serão capazes de acessar a informação, esteja ela armazenada, em processamento ou em trânsito.

Além dos pilares da segurança da informação citados anteriormente, para um sistema de armazenamento e transmissão de dados é necessário outras garantias, como Autenticação, Autorização e Auditoria. Para que um usuário seja reconhecido pelo sistema é imprescindível verificar a identidade de quem estar acessando, além de examinar quais os privilégios do usuário.

2.2.2 Tipos de Criptografia

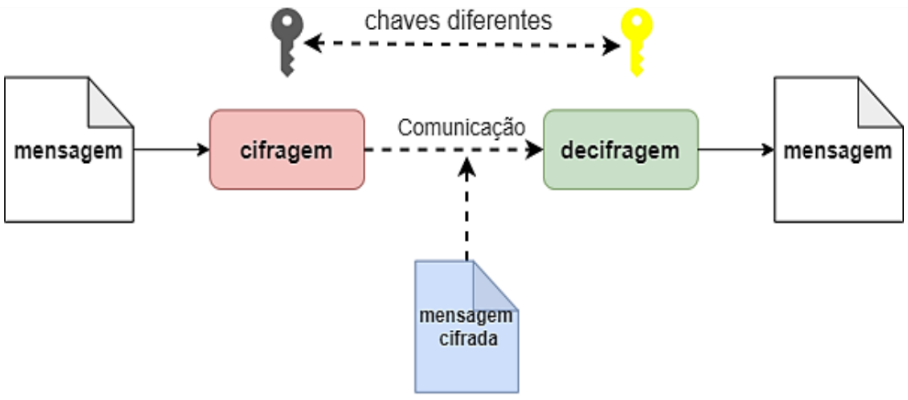
A criptografia é um conjunto de métodos para codificar mensagens, dados, etc. utilizando o conceito de chave de acesso, que pode ser simétrica ou assimétrica, ao aplicá-los a informação torna-se incompreensível. O objetivo desse processo é garantir a segurança do canal de transmissão e da mensagem (TALBOT; WELSH; WELSH, 2006).

O componente mais importante na Segurança da Informação, segundo Stallings, Bressan e Barbosa (2008), é a criptografia, sendo a ciência que mantem as mensagens seguras, através da cifragem, que é o processo cifrar a mensagem original, de tal modo que apenas os usuários com as chaves autorizadas realize a decifragem. A decifragem, por sua vez, é o processo de transformar a mensagem cifrada na mensagem original. A criptografia garante as seguintes propriedades: Integridade, Autenticidade, Não-Repúdio e Sigilo (NAKAMURA; GEUS, 2007).

Para a proteção de dados ou de informações é utilizada uma chave para a codificação e decodificação, a chave é o mecanismo de acesso do dado e esta é única. Em termos criptográficos, chave é a forma de cifrar e decifrar unicamente uma mensagem (BURNETT; PAINE, 2002), existem dois tipos de criptográfica: simétrica e assimétrica, para este trabalho será apresentada apenas o conjunto de chaves assimétricas. Nas Figuras 1 e 2, são apresentados os diagrama de funcionamento de ambos os tipos.

O algoritmo de chave privada utiliza apenas uma chave, ou seja, na etapa de cifrar e decifrar são utilizadas uma única. Essa chave gerada deve ser mantida em segredo para garantir a confidencialidade das mensagens. Apesar disso, o compartilhamento da chave para todos os usuários de forma a mantê-la em segredo é a principal desvantagem destes métodos (MORENO; PEREIRA; CHIARAMONTE, 2005).

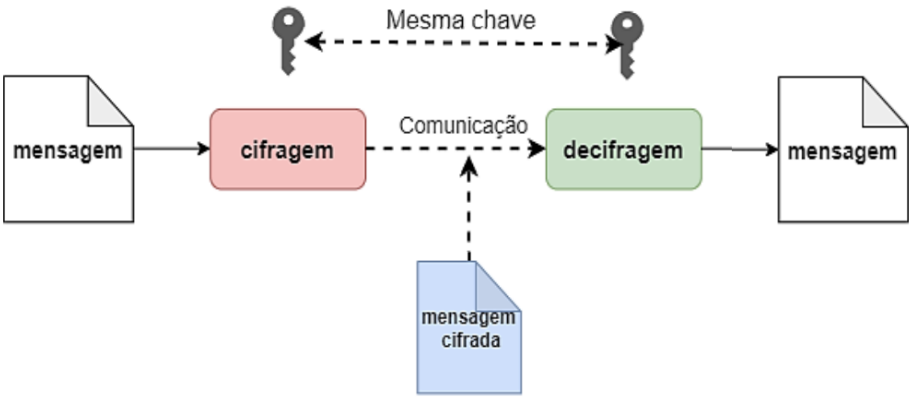
Figura 1 – Diagrama de Criptografia Assimétrica



Fonte: MACORATTI (2016).

O algoritmo de chaves públicas ou assimétrica possibilita a troca de mensagens entre duas entidades, em que cada uma delas contém um par de chaves, público e privado.

Figura 2 – Diagrama de Criptografia Simétrica



Fonte: MACORATTI (2016).

Uma mensagem por exemplo, pode ser cifrada utilizando-se uma chave pública e decifrada utilizando somente a chave privada correspondente ou vice-versa. Dessa forma, dificulta-se a ação de uma entidade externa que queira ler a mensagem, sem que tenha a chave privada da chave pública referente (NAKAMURA; GEUS, 2007).

2.3 *BLOCKCHAIN*

Os sistemas de segurança, na sua grande maioria, propõe um modelo de comunicação centralizado e essa proposta continua com um bom desempenho. No entanto, com o avanço dos dispositivos e da comunicação a possibilidade de invasão torna-se cada vez mais inevitável. A criptografia mais usada em chaves públicas utiliza uma Autoridade de Certificação (CA) (ORMAN, 2018). Com o surgimento da tecnologia *Blockchain*, uma nova forma de implementar a proteção de dados surgiu, agora os dados não precisam estar centralizados e não necessita de uma CA.

Segundo Nakamoto (2008), uma *Blockchain* é um livro-razão que contém transações públicas, que captura dados do mundo real, digitaliza e, que posteriormente são armazenados de forma distribuída. Em 2008, Nakamoto propõe essa tecnologia, que reuni outras para desempenhar papeis importantes, capazes de amenizar e quem sabe até resolver os problemas mais comuns encontrados em proteção e distribuição de dados. Na Figura 3 é apresentada a proposta de Nakamoto em que é possível observar a cadeia de blocos gerada em cada transação dentro do *Blockchain*.

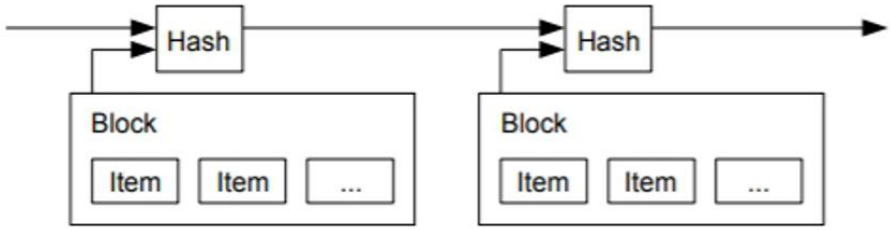
O *Blockchain* é um livro-razão compartilhado e imutável que facilita o processo de registro de transações e o seu rastreamento. Com essa infraestrutura desenvolvida, surge a possibilidade da implantação dos contratos inteligentes (*Smart Contract*). Abaixo é definido cada um desses termos:

- **Livro-razão:** Os participantes da rede têm acesso ao livro-razão distribuído e ao seu registro imutável de transações. Com esse livro-razão compartilhado, as transações são registradas somente uma vez.
- **Registros imutáveis:** Após a inserção do registro na rede, nenhum participante pode modificar uma transação depois de seu registro no livro-razão compartilhado.

A corrente de blocos possui propriedades que proveem benefícios às aplicações e sistemas baseados nesta tecnologia. As principais propriedades da corrente de blocos são (Zheng et al. 2018, Xu et al. 2017,Wüst and Gervais 2018):

- **Descentralização:** As correntes de blocos executam de maneira distribuída, através do estabelecimento de consenso entre todos os participantes da rede. Não há uma entidade centralizadora;
- **Desintermediação:** A corrente de blocos elimina a necessidade de um intermediário confiável para a troca de ativos. Os ativos podem ser trocados diretamente pela rede e a confiança é estabelecida através de consenso;
- **Imutabilidade:** Os dados armazenados em uma corrente de blocos são imutáveis. Não é possível modificar ou recriar qualquer dado incluído na corrente de blocos de forma retroativa. Toda atualização na corrente de blocos é realizada de forma incremental;
- **Irrefutabilidade:** Os dados são armazenados na corrente de blocos em forma de transações assinadas, que não podem ser alteradas devido à propriedade de imutabilidade da corrente de blocos. Portanto, o emissor de uma transação jamais pode negar sua existência;

Figura 3 – Proposta de *Blockchain* de Nakamoto.



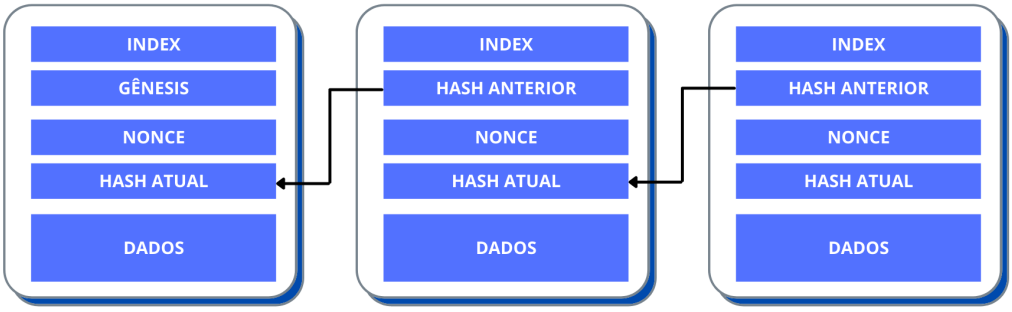
Fonte: (NAKAMOTO, 2008)

- **Transparência:** Todos os dados armazenados na corrente de bloco são acessíveis por todos os participante da rede. Em correntes de blocos públicas, como o Bitcoin e o Ethereum, as transações são abertas para qualquer usuário com acesso à Internet;
- **Auditabilidade:** Como consequência da transparência, todo participante pode verificar, auditar e rastrear os dados inseridos na corrente de blocos para encontrar possíveis erros ou comportamentos maliciosos. No caso de correntes de blocos federadas, o processo de auditagem pode responsabilizar um malfeitor na rede;
- **Disponibilidade:** As correntes de blocos são estruturas replicadas em cada participante da rede e, portanto, a disponibilidade do sistema é garantida mesmo sob falhas, devido à redundância de informações;
- **Anonimidade:** Os usuários e nós mineradores de uma corrente de blocos são identificados por chaves públicas ou identificadores únicos que preservam suas identidades. Ainda, é possível utilizar uma chave pública em cada transação, evitando a rastreabilidade do usuário e conferindo um grau a mais de anonimidade.

2.3.1 Estrutura de um bloco dentro de um *Blockchain*

As partes principais de um bloco são o cabeçalho e as transações. As transações são o agrupamento dos dados que são armazenados no bloco. Por sua vez, o cabeçalho possui diversos campos, dois quais os mais importantes para seu funcionamento são: *hash* do bloco anterior, dificuldade, *nonce*, e raiz da árvore de *Merkle*. Além destes, também é preciso entender dois metadados: altura do bloco e *hash* do cabeçalho, que são armazenados de forma a identificar o bloco e sua posição na cadeia. Estes campos serão detalhados abaixo, pois o correto entendimento da *Blockchain* depende deles. Na Figura 4 são vistos os elementos de um bloco e como esses se interligam em uma *Blockchain*.

Figura 4 – Estrutura de um Bloco



Fonte: Elaborado pelo autor.

A seguir, são descritos os principais elementos de um bloco de uma *Blockchain*.

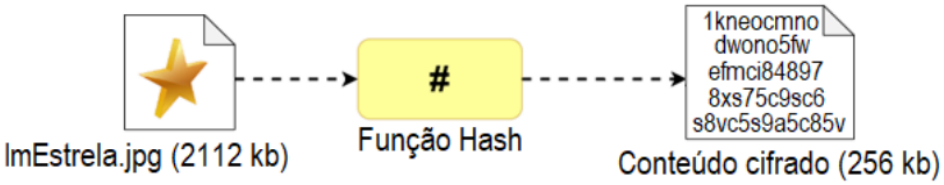
- **Index:** A identificação do bloco segue uma ideia básica que é aplicada em estruturas de dados do tipo FILA, ou seja, segue a data e horário de criação do bloco. No caso para o primeiro bloco criado, ele é chamado de gênesis e possui umas particulares que será apresentada quando for abordado o HASH. Além da numeração do bloco, existe o *timestamp*, que é conhecido como o carimbo de data e hora do bloco. Essa tag garante a ordem de criação cronológica.

- **Nonce:** É um número usado como uma variável para modificar a saída da função hash atual. Esse campo é usado para validar o trabalho do "minerador", ou seja, verificar se o bloco inserido atende as especificações impostas pelo *Blockchain*. Por exemplo, o Hash gerado para o bloco precisar iniciar com 4 zeros a esquerda e essa quantidade de zeros determina o nível de dificuldade para minerar (inserir) um novo bloco na rede.

- **Hash do bloco anterior e Hash atual:** Cada bloco carrega a informação do *hash* anterior, ou seja, é a como a corrente entre os eles é gerada. Com o *hash* anterior é possível localizar o antecessor do bloco. Atualmente, o tamanho utilizado pelo hash é 256, que tem como base o algoritmo HASH 256 desenvolvido pela *National Institute of Standards and Technology* (NIST). O *hash* atual permite que o bloco sucessor do bloco atual consiga acessá-lo. Na Figura 5 é visto um exemplo de aplicação da função *hash* em que uma arquivo de uma Figura no formato JPG é cifrado em um valor de 256 bits.

- **Dado:** É o campo reservado para o armazenamento dos dado e/ou informações coletados no mundo externo.

Figura 5 – Exemplo de Aplicação da Função HASH 256.



Fonte: Elaborado pelo autor.

2.3.2 Mineração

A mineração é o processo responsável por inserir um novo bloco no *Blockchain*. O termo minerador surgiu com o Bitcon, por se tratar de transações financeiras. Ao minerar, eles incluem as transações em um bloco e geram um cabeçalho válido (*hash*) para essas transações. Para criar esse cabeçalho, os mineradores devem calcular a árvore

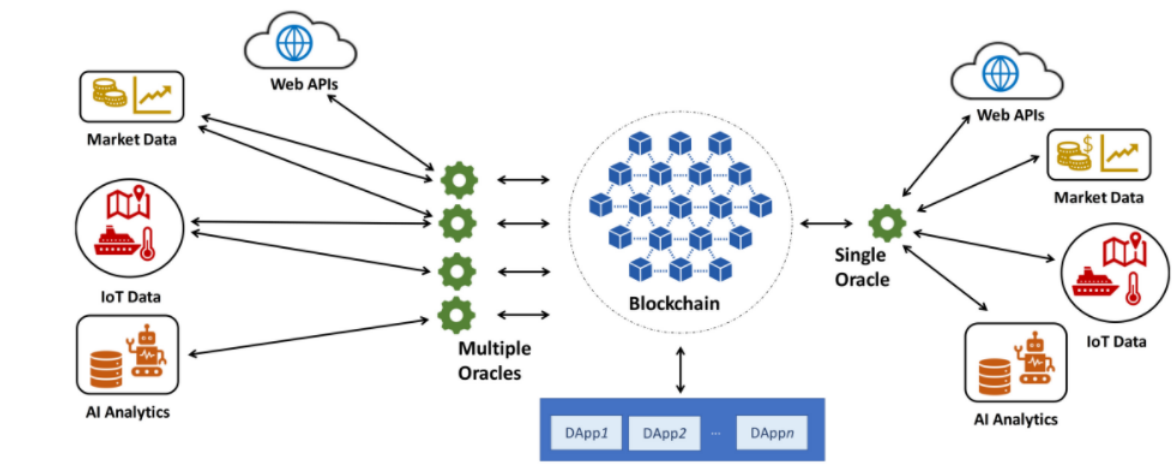
hash, verificar a dificuldade estabelecida, incluir o *timestamp* de tempo e realizar uma série de cálculos a fim de encontrar um *nonce* que satisfaça a dificuldade em vigor. Assim será descrito a importância da dificuldade e como ela se ajusta automaticamente, além de mostrar um passo a passo do processo de mineração.

2.4 ORÁCULO

O termo oráculo começou a ser inserido no contexto de uma rede de *Blockchain* a partir da necessidade de garantir que todo e qualquer dado alimentado na rede siga o princípio da integridade. Um oráculo pode ser então definido como uma entidade capaz de sempre garantir se um dado é verdadeiro. Assim como um oráculo num cenário comum é considerado uma divindade, numa rede *Blockchain* ele assume esse mesmo papel.

As principais aplicações encontradas na literatura apontam para novas criptomoedas ou qualquer ativo digital. No entanto, existem poucas aplicações que capture algum dado ou informação do mundo externo com uso de sensores. Com o crescimento das Cidades Inteligentes será necessário alguma tecnologia capaz de garantir que as leituras de sensores de fato são reais. Uma transação com esse perfil requer uma solução capaz de interagir com o mundo real, que faça leituras e alimente uma rede *Blockchain*. As soluções que realizam essa função em uma rede *Blockchain* são denominados oráculos (MAMMADZADA et al., 2020). A Figura 6 apresenta como é o comportamento de uma rede *Blockchain* com o uso de um ou mais oráculos.

Figura 6 – O papel do Oráculo numa rede *Blockchain*.

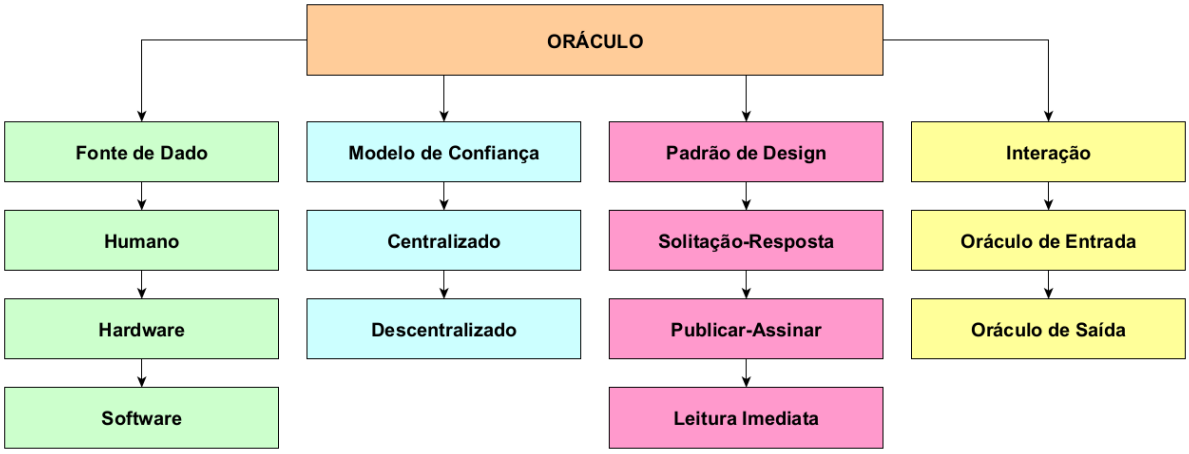


Fonte: (HAMDA, 2020).

Segundo Mammadzada et al. (2020), a partir de uma revisão sistemática, o autor concluiu que, atualmente, a maioria dos oráculos utilizam de uma filosofia contrária proposta por uma rede *Blockchain*, ou seja, eles são implementados com o uso de *Trusted Third Party* (TTP) tornando uma tecnologia centralizada.

Quanto a sua arquitetura, oráculos podem ser classificados em centralizados e descentralizados (Ma et al. 2019). A arquitetura centralizada se baseia em entidades confiáveis que obtém informações do mundo externo quando requisitadas. Uma arquitetura descentralizada, por sua vez, estabelece oráculos como entidades redundantes, que tem a função de prover informações e validá-las, por meio de protocolos envolvendo voto e verificação. Apesar de ser um tema pouco explorado, existe uma classificação bem definida. Na Figura 7 é apresentada a taxonomia do oráculo.

Figura 7 – Classificação dos Oráculos *Blockchain*



Fonte: Adaptado Hamda (2020).

De acordo com Al-Breiki et al. (2020), a taxonomia de um oráculo pode ser definida como:

- **Fonte de Dado**: O oráculo tem três possibilidades de fontes de dados, a saber: por meio de software, esse tipo de oráculo obtém as informações a partir da internet, ou seja, ele navega na rede para extrair a informação desejada. A de hardware, que ao contrário do oráculo de software, é alimentado por informações do mundo externo. E por fim, a fonte humana, que como o próprio nome sugere, para receber uma informação é necessário que um humano a insira.
- **Modelo de Confiança:** O modelo adotado para construção de um oráculo parte da quantidade de nós utilizados. Atualmente existem dois modelos empregados, o centralizado, este depende de uma única fonte de dados, logo é um alvo mais vulnerável de ataques, no entanto, possui uma alta performance. Já o modelo descentralizado resolve o problema do ponto singular (PPS), porque utiliza redundância de oráculos, mas em contrapartida o seu desempenho é reduzido.
- **Padrão de Comunicação:** Existem três modelos que podem ser adotados para a comunicação do oráculo com o *Blockchain*. O primeiro, conhecido por *Request-*

Response, que parte do princípio de que a quantidade de dados gerados pela fonte é grande, mas que essas informações são repassadas apenas quando solicitadas. Em seguida, tem o modelo *Publish-Subscribe*, por sua vez, é utilizado em cenários que existe a necessidade de atualização constantes da informação. E por fim, o modelo *Immediate-Read*, que é usado em oráculos que fornecem dados que são necessários apenas para uma decisão imediata (este modelo armazena os dados uma vez e pode ser atualizado).

- **Interação:** Os oráculos podem ter diferentes interações com o mundo externo. A primeira é inserir dados no *Blockchain* através de sensores ou ativos e o segundo tipo de interação é entregando dados do *Blockchain* para o mundo externo a partir de uma ação de aturadores, por exemplo.

A solução em *Blockchain* possibilita a eliminação da necessidade de uma parte central confiável, ou seja, elimina a necessidade de IPCs e CA e então todos os problemas relacionados ao gerenciamento de chaves são tratados pela implementação do *Blockchain* de maneira descentralizada, em que o *Blockchain* atua como um sistema descentralizado de armazenamento de *key-value*. O gerenciamento descentralizado de chaves pode resolver os problemas com os sistemas de CA por meio da revogação de certificados, eliminando pontos únicos de falha e reagindo rapidamente ao uso indevido de CAs.

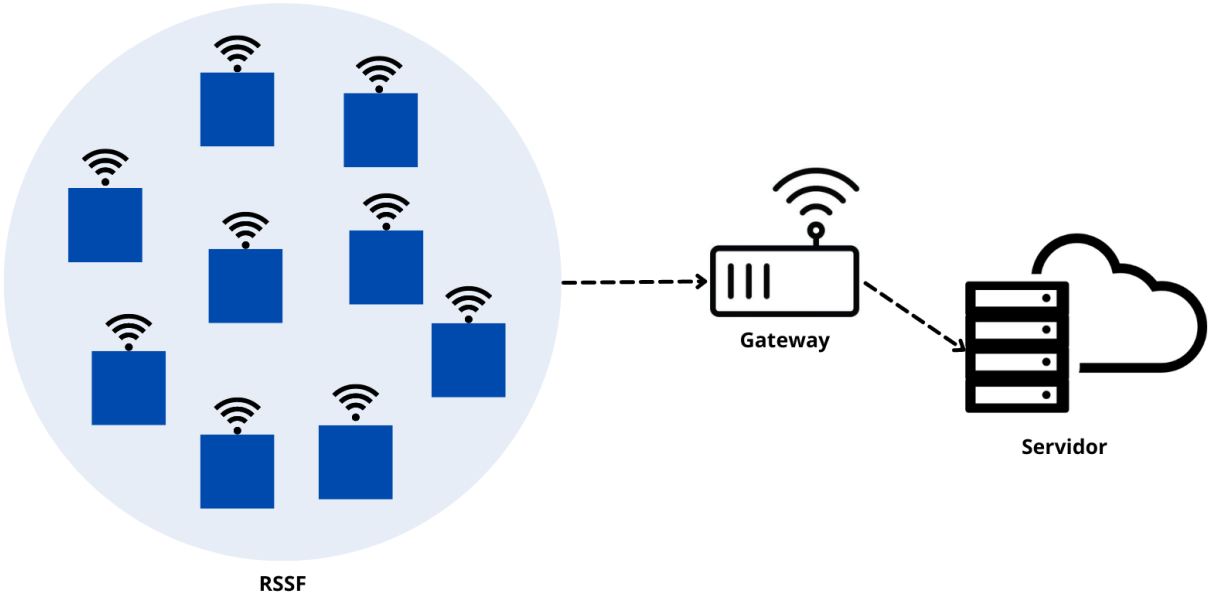
3 PROCEDIMENTO METODOLÓGICO E PROPOSIÇÃO DE UM ORÁCULO PARA UMA REDE *BLOCKCHAIN*

Neste capítulo será descrita uma proposta de implementação de um oráculo e de uma rede *Blockchain*, apresentando as tecnologias empregadas no desenvolvimento e os procedimentos de desenvolvimento. Além disso, também serão descritas as métricas de avaliação utilizadas para análise do desempenho do oráculo proposto.

3.1 PROPOSTA DE UMA RSSF BASEADA EM ORÁCULOS DE *BLOCKCHAIN*

Segundo Aguiar (2018), uma RSSF é composta por nó sensor, nó roteador e o nó concentrador. O nó sensor (emissor) é responsável por realizar a medição e aquisição de dados e enviá-los a partir dos nós intermediários ou nós roteadores até atingir o nó concentrador. A função do nó concentrador é de receber os dados enviados pelos nós da rede e encaminhar para um servidor, que pode estar em nuvem, para que esses dados se tornem disponível para os usuários finais. Atualmente, em geral, o nó concentrador também acumula a função de Gateway, ou seja, interliga a RSSF à internet. Na Figura 8 é apresentado um exemplo de uma RSSF.

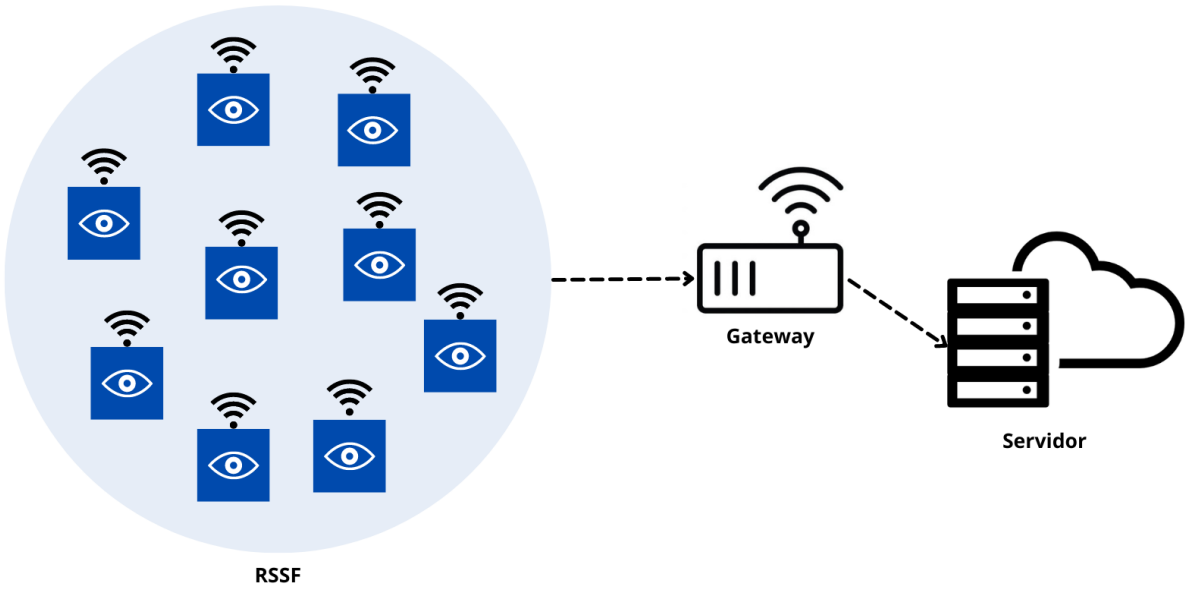
Figura 8 – Modelo de Rede de Sensores Sem Fio.



Fonte: Adaptado de Medeiros et al. (2020).

Na Figura 9 é descrita uma proposta de uma RSSF cujos nós sensores são baseados em uma proposta de oráculo a fim de que todos os dados medidos e enviados à rede sejam íntegros.

Figura 9 – Proposta de uma Rede de Sensores Sem Fio baseada em Oráculo.



Fonte: Elaborado pelo autor.

Na próxima seção é apresentado o oráculo de *Blockchain* proposta para uso em RSSF íntegras.

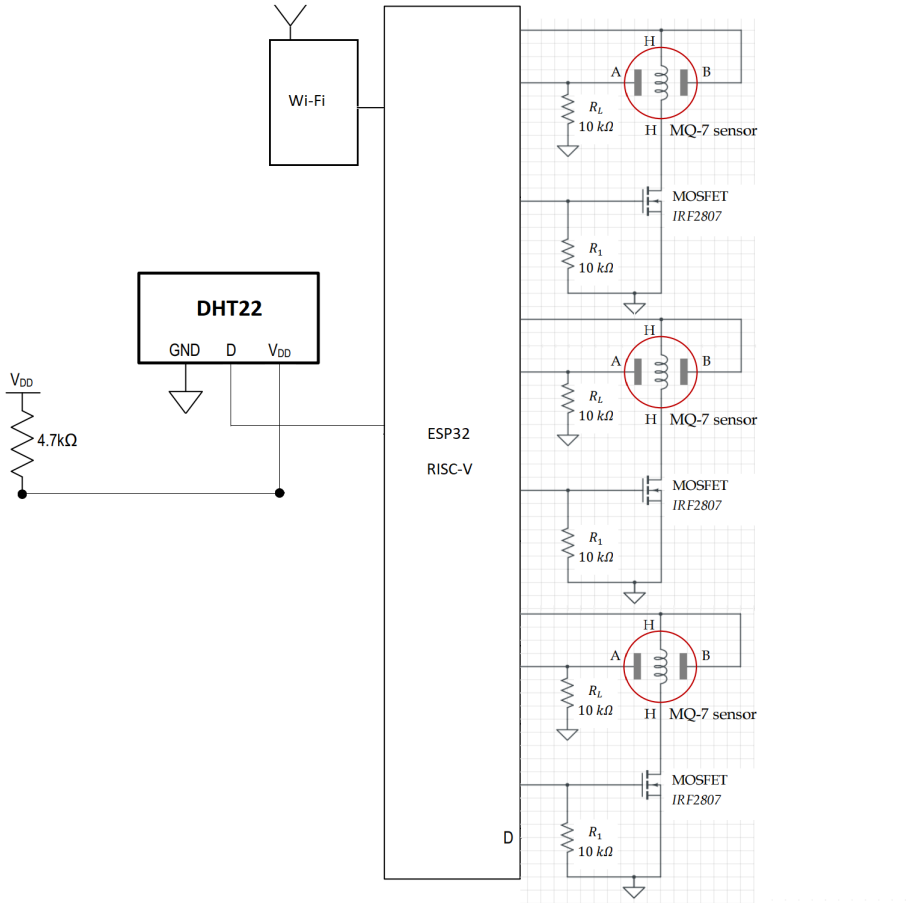
3.2 ORÁCULOS DE *BLOCKCHAIN* PROPOSTO

Para o desenvolvimento da proposta deste trabalho foram selecionadas algumas tecnologias de acordo com a necessidade do cenário determinado, que, neste trabalho, será a de medição de monóxido de carbono (C0) via uma RSSF no contexto de *Blockchain*. Algumas considerações precisam ser elencadas, os tipos e especificações de sensores, módulo de processamento, meio de comunicação, entre outras. Durante este capítulo, será apresentado como cada uma dessas tecnologias são aplicadas no oráculo de *Blockchain*.

O circuito é composto por um microcontrolador ESP32-C3 que possui um microprocessador de arquitetura do tipo RISC, ou seja, as instruções são executadas diretamente no hardware. O ESP32-C3 tem embarcado um módulo *Wifi* e um sensor de temperatura interno para monitorar o aquecimento do chip. A importância desse parâmetro será explanado na Seção 3.2.1. Além dos recursos presentes no microcontrolador, foram inseridos três sensores de CO que são utilizados para fornecer dados ao oráculo e um sensor de temperatura (DHT22), que desempenha um papel importante em umas das etapas validação dos dados. Na Figura 10 é apresentado o diagrama do circuito de oráculo de hardware (nó sensor com integridade) desenvolvido e implementado neste trabalho e na Figura 11 pode ser visto uma foto do circuito implementado.

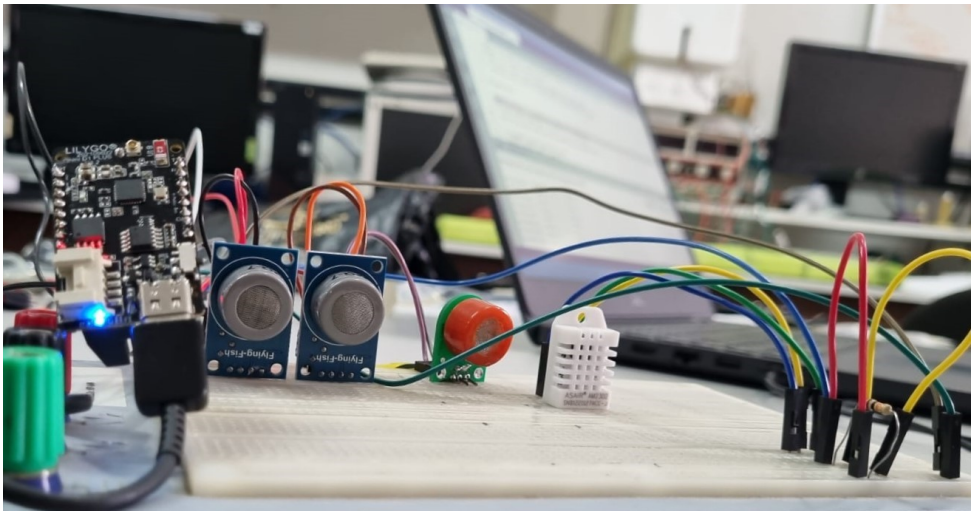
A proposta de nós sensor oráculo de hardware (nó sensor com integridade) vista

Figura 10 – Diagrama de Circuito do Oráculo.



Fonte: Elaborado pelo Autor.

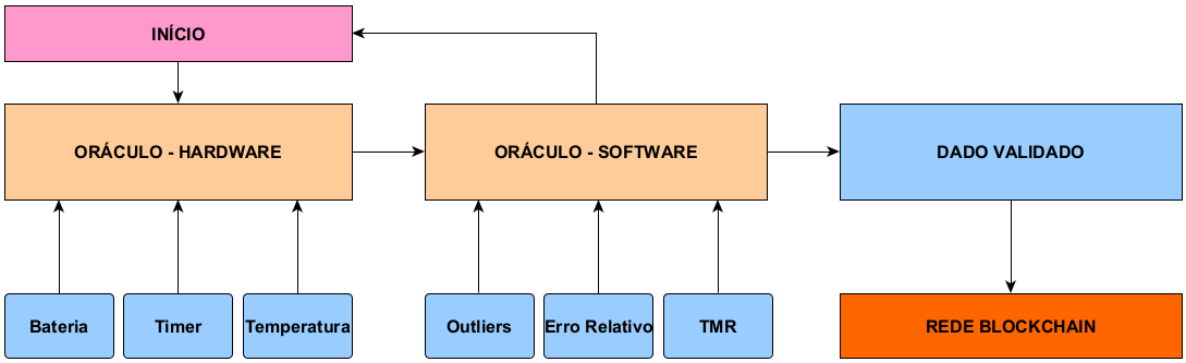
Figura 11 – Protótipo do Oráculo.



Fonte: Elaborado pelo Autor.

na Figura 12 tem sua integridade de dados avaliada em duas etapas: uma em nível de hardware que, consiste em verificar as condições do circuito, como: microcontrolador, bateria e sensores; e outra em nível de software, cujo objetivo é validar os dados por meio de processos lógicos.

Figura 12 – Visão Geral do Oráculo.



Fonte: Elaborado pelo Autor.

3.2.1 Oráculo Proposto: Etapa de Hardware

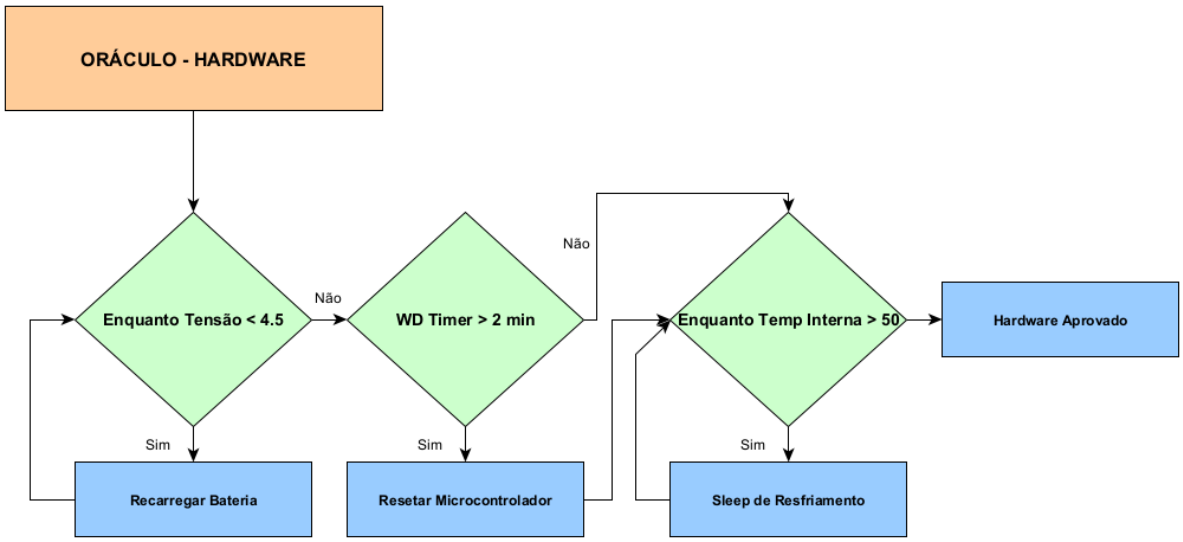
Na etapa de Hardware, o nó sensor utiliza os dados obtidos do hardware para verificar a veracidade dos dados medidos e as circunstâncias apropriadas de medição do dispositivo. Na Figura 13 é apresentado o fluxograma da etapa de hardware. Os componentes escolhidos e que compõem essa análise estão listados abaixo:

- ***Bateria:*** este componente desempenha um importante papel nas soluções IOT e também, é uma limitação pelo tempo de vida útil. Ao reduzir nível de tensão, isso pode acarretar erro nas leituras dos sensores, tendo em vista que, alguns sensores dependem de uma tensão adequada.
- ***WatchDog Timer:*** é um dispositivo que atua quando o microcontrolador apresenta um tempo de execução além do esperado ou entra em uma situação de repetição infinita. O *WatchDog Timer* aplica um reset quando o tempo excede o programado.
- ***Temperatura Interna:*** outro fator que pode ocasionar leitura errôneas ou até mesmo um travamento do nó-sensor é a temperatura do chip. Logo, quando isso ocorre deve ser acionado a rotina de resfriamento.

3.2.2 Oráculo Proposto: Etapa de Software

Na etapa de software, o esquema proposto utiliza certas condições lógicas para determinar a veracidade de dados fornecidas pelo dispositivo. Na Figura 14 é apresentado

Figura 13 – Fluxograma da Etapa de Hardware.



Fonte: Elaborado pelo Autor.

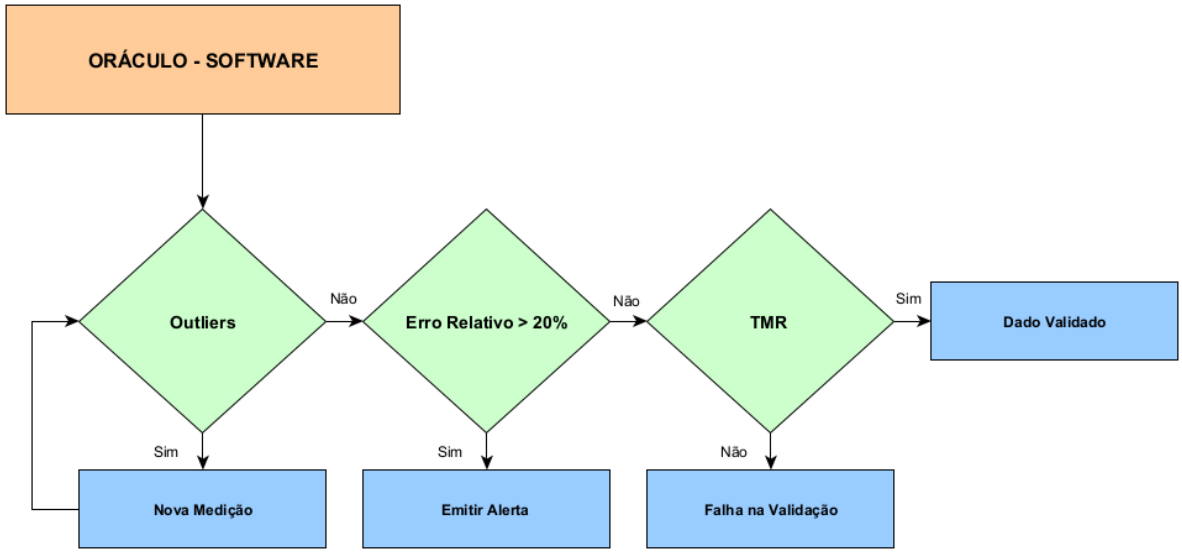
o fluxograma da etapa de software. Os componentes escolhidos e que compõem essa análise estão listados abaixo:

- ***Outliers:*** Outliers são conhecidos por divergirem drasticamente do grupo de dados da qual fazem parte. Segundo o *datasheet* do sensor MQ7 os valores obtidos devem ser entre 20 ppm a 2000 ppm, ou seja, qualquer valor foram dessa margem pode ser considerado um outlier.
- ***TMR:*** é usada para reduzir o número de erros em um sistema, através da triplicação e comparação dos resultados dos processos.
- ***Erro Relativo:*** O erro relativo determina a porcentagem de erro entre os valores medidos pelos sensores.
- ***Temperatura Externa:*** a temperatura externa pode afetar diretamente na medição dos sensores, inclusive o MQ7 que é utilizado neste trabalho.

3.3 IMPLEMENTAÇÃO DO ORÁCULO

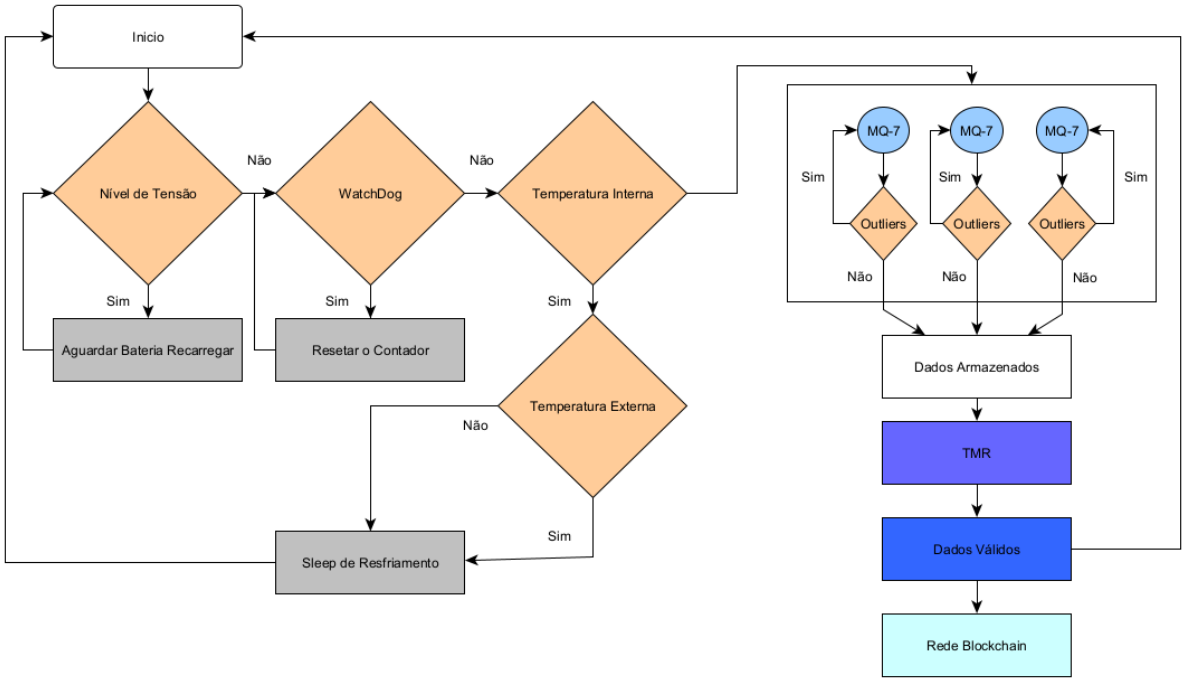
Com a topologia da RSSF definida o próximo passo para o desenvolvimento do oráculo é a escolha da técnica que será implementada. Na Figura 15 é apresentado a lógica aplicada no funcionamento do oráculo.

Figura 14 – Fluxograma do Oráculo do Tipo Software.



Fonte: Elaborado pelo Autor.

Figura 15 – Fluxograma de Funcionamento do Oráculo.



Fonte: Elaborado pelo Autor.

3.3.1 Protocolo de Tensão

O primeiro protocolo de verificação é o nível de tensão. Com o valor da carga da bateria é possível apontar a causa de outliers, tendo em vista que, uma baixa ou alta tensão pode influenciar nos dados mensurados pelos sensores. Quando esta etapa de segurança é acionada, o nó-sensor automaticamente entra em modo sleep e aciona a recarga da bateria

até que o nível de bateria esperado pelo sistema seja atingido.

3.3.2 Protocolo de Temporizador

Após a validação do protocolo de tensão, é acionado o protocolo timer que, tem como objetivo verificar se o software embarcado está funcionando corretamente a partir do watchdog timer. Com essa etapa de segurança implementada, o hardware poderá evitar aquecimentos inesperado em decorrência de comportamento inesperado do software.

3.3.3 Protocolo de Temperatura Interna

A última etapa do oráculo é o protocolo de temperatura. Nesta, é verificado a temperatura interna do chip. Em alguns casos, o protocolo de timer pode falhar e para reforçar a segurança do oráculo é necessário consultar a temperatura interna. Altas temperaturas podem afetar no funcionamento do nó-sensor. Caso isso ocorra, é necessário um resfriamento no dispositivo.

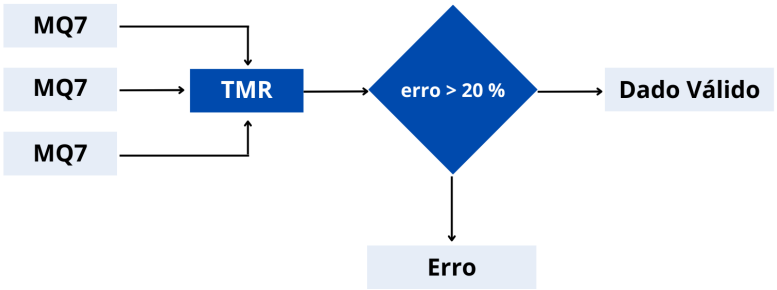
3.3.4 Protocolo de Outliers

Neste ponto, a validação do hardware está aprovada e é iniciado a validação dos dados obtidos pelos sensores MQ7. Aqui é comparada os dados mensurados com informações fornecidas pelo *datasheet*. Caso, algum sensor tenha medido valores fora da margem permitida é informado a presença de outliers e em seguida o sensor realizará uma nova captura.

3.3.5 *Triple Modular Redundancy*

A *Triple Modular Redundancy* (TMR) proposta por Von Neumann, que geralmente é aplicada em sistemas tolerantes a falha é usada para reduzir o número de erros em um sistema, através da triplicação e comparação dos resultados dos processos. Na Figura 16 é apresentada arquitetura da TMR. No contexto da solução proposta, o hardware são os nós sensores e o software que desempenhará o papel de autenticador será o oráculo.

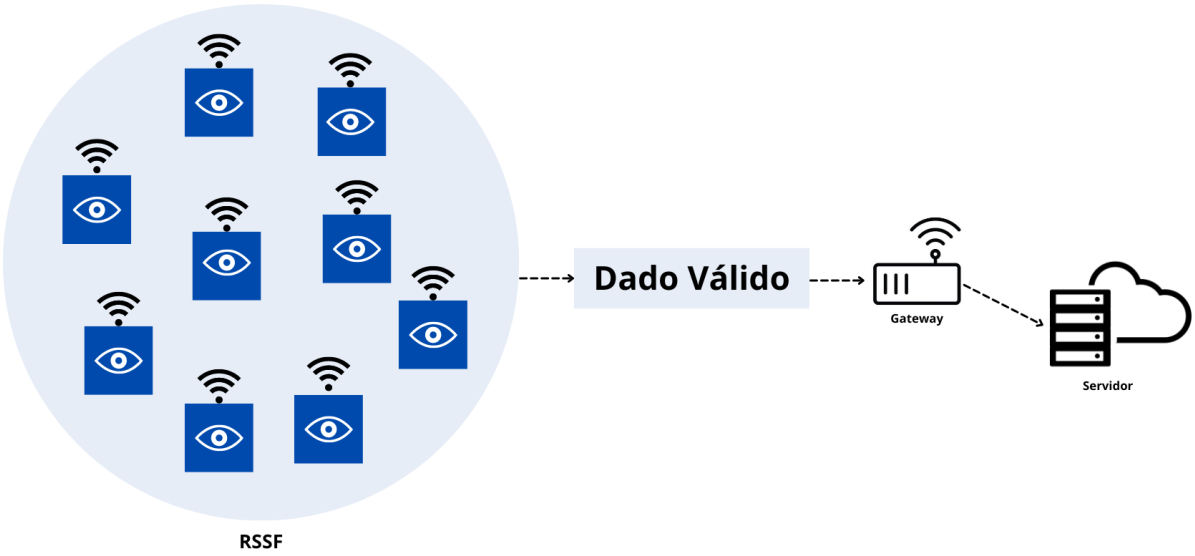
Figura 16 – *Triple Modular Redundancy* (TMR).



Fonte: Elaborado pelo autor.

O oráculo executará uma votação e selecionará a saída de acordo com a maioria dos nós. Portanto, se um sensor falhar, o erro não será refletido na saída do oráculo. Após a votação ser concluída e ter a saída gerada, o dado considerado real deverá seguir o fluxo de uma RSSF como é apresentado na Figura 17.

Figura 17 – Arquitetura da RSSF com Oráculo.



Fonte: Elaborado pelo autor.

Além da votação aplicada com o TMR, é realizada uma análise nos valores fornecidos pelos sensores com objetivo de certificar que não há ocorrência de erros relativos acima do limiar de 20%. Para que esse procedimento seja executado é necessário que seja aplicado uma ordenação de valores dos sensores e logo após os valores possam ser obtidos pela Equação 1.

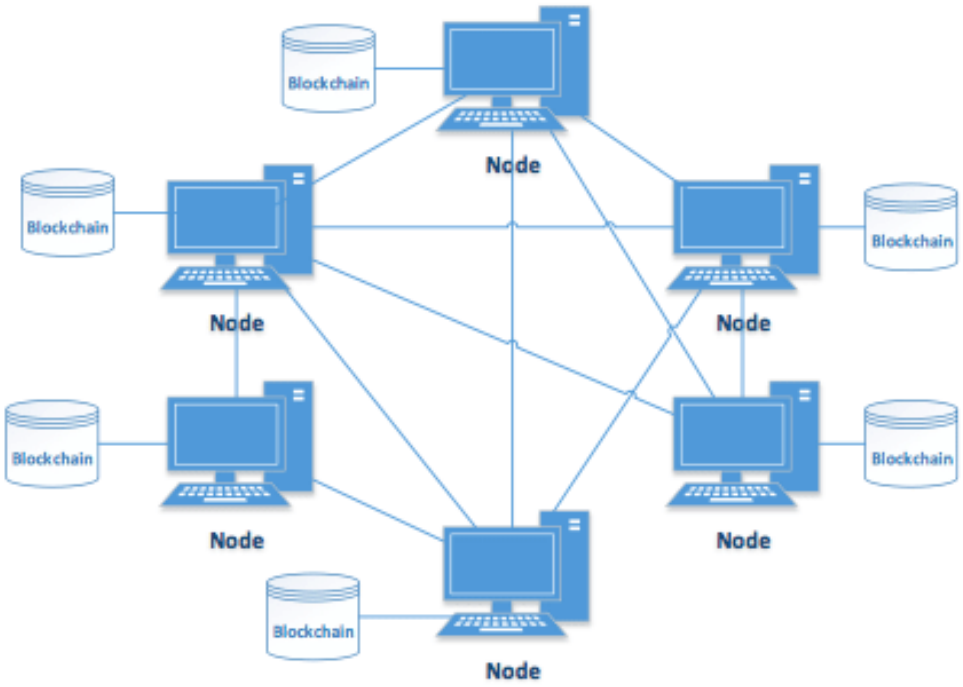
$$erroRelativo(\%) = \frac{maiorValor - menorValor}{maiorValor} \tag{3.1}$$

3.4 REDE *BLOCKCHAIN*

De forma a garantir os princípios da segurança de informação, é usada uma estrutura de rede descentralizada de computadores do tipo *peer-to-peer* (P2P) para a implementação de um *Blockchain* (NAKAMOTO, 2008). Uma rede P2P é uma rede compostas por nós interconectados e que desempenham o papel tanto de cliente quando de servidor, ou seja, é uma rede descentralizada (SCHOLLMEIER, 2001), como vista na Figura 18.

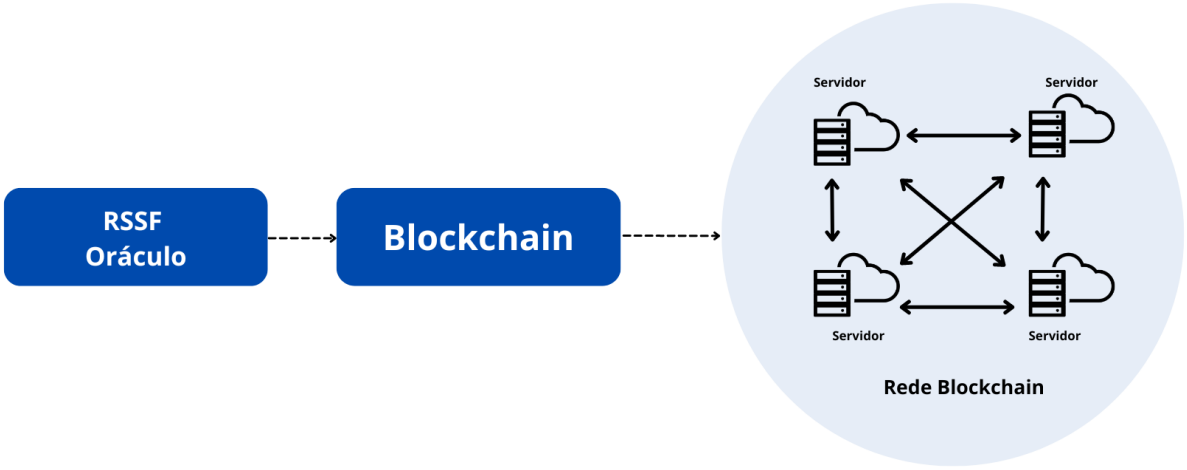
Uma rede *Blockchain* é uma rede P2P em que cada nó da rede possui uma cópia da cadeia de blocos. Dessa forma, se uma ataque ocorrer em um nó e alterar por exemplo um bloco do *Blockchain* (ou seja, uma transação), é aplicado um protocolo de consenso (descrito na Seção 3.4.1) capaz de restaurar ao estado anterior (antes do ataque) do *Blockchain*. Na Figura 19 é apresentado a proposta de desenvolvimento da rede P2P deste trabalho.

Figura 18 – Modelo de rede P2P.



Fonte: (KOTESKA; KARAFILOSKI; MISHEV, 2017)

Figura 19 – Arquitetura da Rede *Blockchain* com oráculo proposta neste trabalho.



Fonte: Elaborado pelo autor.

3.4.1 Protocolo de Consenso

Distribuir cópias de um *Blockchain* em uma rede P2P não é o suficiente para garantir que os dados continuem íntegros, pois podem acontecer falhas na rede ou ataques intencionais. Para aumentar a segurança e confiabilidade de uma rede *Blockchain* é aplicado o mecanismo de consenso que é um algoritmo que tem o objetivo de aumentar a tolerância

a falhas. Ou seja, o mecanismo de consenso evita que os nós da rede contenham *Blockchains* diferentes entre si e são usados para a validação dos dados antes da inserção no *Blockchain* (CHRISTIDIS; DEVETSIKIOTIS, 2016).

O objetivo de um algoritmo de consenso na *Blockchain* é garantir que todos os nós participantes concordam com as transações realizadas na rede, que formam um histórico que é serializado na forma de um *Blockchain* (XIAO et al., 2020). Segundo Baliga (2017), o protocolo de consenso deve apresentar três propriedades, que são:

- ***Safety:*** Todos os nós da rede devem produzir o mesmo resultado para uma transação, e que esse resultado deve ser válido de acordo com o mecanismo de consenso.
- ***Liveness:*** Determina que todos os nós não falhos, que participam do consenso, produzam um valor.
- ***Fault Tolerance:*** Estabelece que o mecanismo deve se recuperar no caso da falha de um dos nós

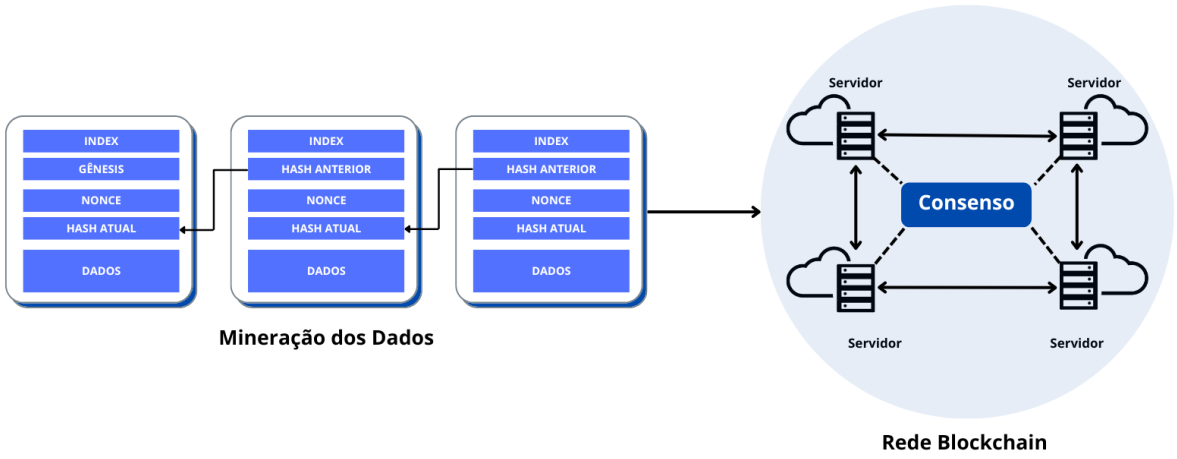
Ainda segundo Baliga (2017), existem inúmeros algoritmos de consenso para *Blockchain*, os dois mais conhecidos são:

- ***Proof of work (PoW):*** Esse mecanismo consiste em encontrar o valor do hash do cabeçalho do bloco de tal forma a seguir os parâmetros definidos pelo parâmetro de dificuldade.
- ***Proof of stake (PoS):*** faz uma escolha do nó minerador que poderá criar um novo bloco. Geralmente é estabelecido um sorteio cuja chance de ganhar é proporcional à quantidade de blocos que o nó já inseriu na rede. Ou seja, o quanto esse nó é confiável.

O mecanismo de consenso selecionado para o desenvolvimento da solução deste trabalho é o PoW, modelo de consenso utilizado pelo *Blockchain* da bitcoin (NAKAMOTO, 2008). A lógica desse protocolo é que, para a criação de um novo bloco, deve ser atendida a dificuldade estabelecida pela rede *Blockchain*. Esse parâmetro representa a quantidade de zeros que o *hash* deve conter à esquerda. Na Figura 20 é apresentado a inserção desse protocolo a solução.

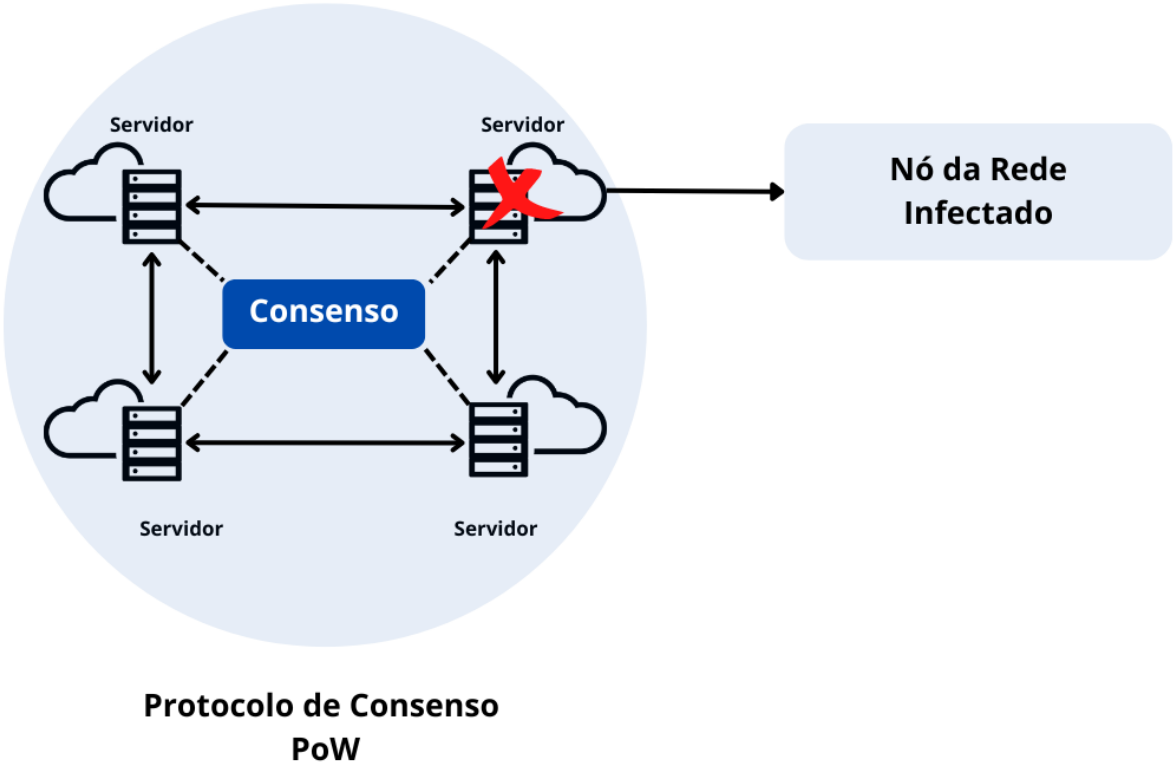
Agora utilizando o exemplo da Figura 21 é possível perceber que um dos nós da rede foi atacado ou apresenta falha, a função do protocolo PoW é de identificar e corrigir baseado nas cópias existentes na rede.

Figura 20 – Protocolo de Consenso PoW.



Fonte: Elaborado pelo autor.

Figura 21 – Exemplo de correção usando PoW.



Fonte: Elaborado pelo autor.

3.4.2 Criptografia SHA 256

Um sistema distribuído como o do *Blockchain* é essencial a utilização de algum mecanismo de criptografia. A criptografia é uma das tecnologias que compõe o *Blockchain*

para cumprir os requisitos de segurança. Dentre os recursos mais utilizados, destacam-se as funções *hash* e as assinaturas digitais (GREVE et al., 2018).

Para a implementação de um *Blockchain* é necessário o desenvolvimento de várias camadas e para não tornar uma solução ineficiente foi selecionado o algoritmo da criptografia Hash 256. Essa função atende as quatro propriedades de uma função de dispersão, que são:

- Deve ser fácil computar o valor de dispersão para qualquer mensagem;
- Deve ser difícil gerar uma mensagem a partir de seu resumo;
- Deve ser difícil modificar a mensagem sem modificar o seu resumo;
- Deve ser difícil encontrar duas mensagens diferentes com o mesmo resumo.

3.5 PROCEDIMENTO DE TESTE DE SOFTWARE

A técnica aplicada para avaliar o comportamento dos protocolos do oráculo é conhecida por estrutural ou teste da caixa-preta. Segundo (PRESSMAN, 2005), o teste de padrão estrutural trabalha com o código fonte, afim de analisar cada componente do software. São considerações na avaliação aspectos tais como:

- Estruturas condições;
- Fluxo de dados;
- Estruturas de repetição;
- Caminhos lógicos.

Na execução dos testes foi gerada uma tabela com os dados de entradas para avaliar cada aspecto do software e analisar as saídas geradas. Todos os valores foram baseado a partir das limitações do hardware e do ambiente em que o oráculo atua, abaixo é descrito os parâmetros.

- **Teste:** é o rótulo de identificação do teste;
- **Tensão:** são os valores de tensão;
- **WDT:** são os valores so temporizador *watchdog timer*;
- **TempIn:** são os valores da temperatura interna;
- **TempEx:** são os valores do sensor da temperatura externa (DHT22);
- **MQ7-1:** são os valores do sensor 1 de CO;

- **MQ7-2:** são os valores do sensor 2 de CO;
- **MQ7-3:** são os valores do sensor 3 de CO;
- **Status:** é o *label* para realizar a identificação de dados íntegros ou não.

O primeiro parâmetro da tabela é a tensão da bateria cujo maior valor aceito é de 5 V, com uma tolerância de até 10% para mais ou para menos. Foram levantadas as mais diversas possibilidades, como: a bateria descarregada, completamente carregada, meia-vida e também a tolerância de 10% em relação a carga total, ou seja, porque foi considerado que no circuito estão presente resistores com margem de precisão de 10%. Com isso, os valores de tensão na tabela de testes variam entre 0,0 V até 5,5 V.

Para parametrizar os são os valores de testes para o *watchdog timer* foi necessário medir o tempo de operação do código-fonte no microcontrolador ESP32-C3, a fim de determinar a média do tempo de execução, que é de 600 milissegundos. Com a média de duração estabelecida, o *watchdog timer* é verificado entre 200 milissegundos até 700 milissegundos. O tempo mínimo foi baseado em nó-sensor sem oráculo e o tempo máximo foi acrescido de 100 milissegundos, ou seja, um valor acima da margem de 10% do tempo médio.

Os valores mínimos e máximos dos sensores de temperatura foram consultados no *datasheet* para verificar a faixa de operação adequada para ambos. Além das informações fornecidas pelo *datasheet* foi considerado o ambiente em que o oráculo está inserido, porque esse fator influencia diretamente o oráculo. Segundo o fabricante do chip, a faixa de operação deve ser -10º C a 120º C. Diante das restrições apresentadas anteriormente, os valores utilizados nos testes de temperatura interna foram entre -5º C até 60ºC.

O sensor DHT22 que é utilizado para verificar a temperatura externa, capta valores entre -40º C até 200ºC, apesar de possui uma escala de medição alta, os valores mínimos e máximos considerados neste trabalho é de 0º C até 44º C, porque é considerado a localização que o oráculo atua, que é na cidade de Campina Grande cuja temperaturas normalmente são entre 15º C até 33º C.

Para selecionar a faixa de teste de medição dos sensores MQ7 foram consideradas algumas orientações do fabricante tais como: o tempo de medição, que neste caso precisa ser de um total de 150 segundos e dois valores de tensão, um de 1,4 V nos primeiros 60 segundos e 5,0 V nos últimos 90 segundos. Além dessas particularidades, a faixa de medição do sensor MQ7 é de 20 a 2000 ppm. Logo, os valores para testes é de 7 ppm até 2790 ppm e neste caso, não é considerado faixa de tolerância de erro.

Ao final de todas as etapas verificadas, cada entrada ao entrar no oráculo emite um parecer sobre os dados recebidos. É utilizado o *label* SIM para os dados aprovados em

todas as etapas ou o *label* NÃO para dados não íntegros ou com falhas na medição. Na Tabela 1 é apresentado parte dos valores testados no oráculo.

Tabela 1 – Padrão de Testes.

Teste	Tensão	Watchdog	TempIn	TempEx	MQ7-1	MQ7-2	MQ7-3	Status
1	4,5	540	35	28	600	800	550	SIM
2	5,0	300	20	30	1000	2100	1100	SIM
3	4,5	350	38	22	2790	826	397	NÃO
4	2,0	200	40	19	1214	1053	57	NÃO
5	4,9	359	27	44	991	701	951	NÃO
6	0,0	597	42	36	653	1732	1320	SIM
7	5,3	500	-5	25	1930	170	178	SIM
8	4,9	230	20	17	30	1490	1106	NÃO
9	5,0	530	55	0	555	1050	473	NÃO
10	4,7	480	32	40	1638	1691	600	SIM
11	5,0	550	38	22	1790	826	1797	SIM
12	4,8	450	40	19	1214	1153	7	SIM
13	4,3	365	27	44	991	701	951	NÃO
14	5,1	580	42	36	653	1732	1720	SIM
15	4,9	400	60	25	1930	970	978	NÃO
16	5,0	700	30	29	1214	1153	1200	NÃO

4 RESULTADOS E DISCUSSÕES

Nesta seção será apresentado os resultados do oráculo e da rede *Blockchain*. Como já foi mencionado anteriormente, a rede *Blockchain* e o Oráculo desenvolvidos neste trabalho são de propósito geral, mas para a obtenção de resultados experimentais, foi considerado dados de medição por sensores de monóxido de carbono CO, como visto na Seção 3.2.

4.1 ORÁCULO

Para a realização dos testes do sistema embarcado foi levantada uma sequência de padrões de teste, quem tem como objetivo testar todos os requisitos funcionais e as funções do programa desenvolvido (*firmware*) afim de avaliar a efetividade do conceito de oráculo de hardware. A sequência de padrões de teste foi estabelecida selecionado valores de parâmetros e de dados de medição dentro de faixas e características consideradas válidas e não-validas. Dessa forma, a eficiência do oráculo proposto pode ser medida avaliando se dados válidos são aceitos e se dados não-válidos são rejeitados (não são enviados ao *Blockchain*.

A seguir são descritas as faixas e condições estabelecidas para características consideradas válidas e não-validas, e na Tabela 2 é apresentado os valores permitidos pelo oráculo:

- **Tensão:** os valores aceitos no protocolo de tensão é de 4,5 V até 5,5 V, considerando uma tolerância de 10%, porque no circuito são aplicados resistores com esse percentual de variação;
- **WDT:** neste protocolo foi calculado a média de tempo de execução do código-fonte para estimar o tempo máximo aceito pelo oráculo que é de 600 milissegundos;
- **TempIn:** na temperatura Interna, o limite de temperatura permitido é de 50º C de acordo com a justificativa apresentada na Seção 3.5;
- **TempEx:** o valor máximo permitido pelo oráculo para a temperatura externa é de 40º C de acordo com a justificativa apresentada na Seção 3.5;
- **MQ7:** O valor máximo e mínimo para as leituras dos sensores MQ7 é de 20 ppm até 2000 ppm, de acordo com as informações do datasheet;

Na execução dos teste utilizando a sequência de padrões de teste estabelecida, o software originalmente não obteve um desempenho ideal, com 18,75% de erro. Algumas das principais falhas foram, os teste da temperatura interna e o cálculo do erro relativo. Após a análise dessas falhas, o código obteve uma atualização para sanar os problemas

Tabela 2 – Os Valores Aceitos Pelo Oráculo.

Parâmetro	Valor Mínimo	Valor Máximo
Tensão	4,5 V	5,5 V
WDT	200 ms	600 ms
TempIn	0º C	50º C
TempEx	15º C	40º C
MQ7-1	20 ppm	2000 ppm
MQ7-2	20 ppm	2000 ppm
MQ7-3	20 ppm	2000 ppm
Erro Relativo	0%	20%

encontradas. Na versão final, o código fonte obteve êxito de 100% em todas as entradas testadas. Foram avaliados todas as funções presentes no software.

A seguir são descritos os critérios de validação de cada característica considerada no oráculo proposto.

4.1.1 Validação do Protocolo de Tensão

Dentre todos os testes realizados, o comportamento deste protocolo foi sempre como esperado. Na Figura 22 é apresentado um dos testes realizados. Como pode ser observado, no caso da bateria está com o nível de tensão abaixo do recomendado o nó-sensor entra em modo sleep pelo tempo necessário para recarregar.

Figura 22 – Teste de Tensão

```
20:03:00.895 -> Tensão: 2.00
20:03:00.895 -> Temperatura Interna: 40.00
20:03:00.895 -> Temperatura Interna: 19.00
20:03:00.940 -> MQ7 1: 1214
20:03:00.940 -> MQ7 2: 57
20:03:00.940 -> MQ7 3: 0
20:03:00.940 ->
20:03:00.940 -> ************************************
20:03:00.940 ->
20:03:00.940 -> Bateria Descarregada! Aguarde um momento
20:03:00.940 -> 2.00V
20:03:00.940 -> ************************************
20:03:00.940 ->
20:03:00.940 -> Going to light-sleep now for 120 seconds...
```

Fonte: Elaborado pelo autor.

4.1.2 Validação do Protocolo do Temporizador *Watchdog*

Em seguida, os testes analisaram o temporizador watchdog timer. Em todos os testes, esse protocolo atuou como esperado, que é resetar o dispositivo afim de recuperar o

nó-sensor. Na situação em que essa etapa não detecta nenhuma falha o oráculo segue para a última verificação, que é a da temperatura interna. Na Figura 23, é visto os testes que foram descritos anteriormente.

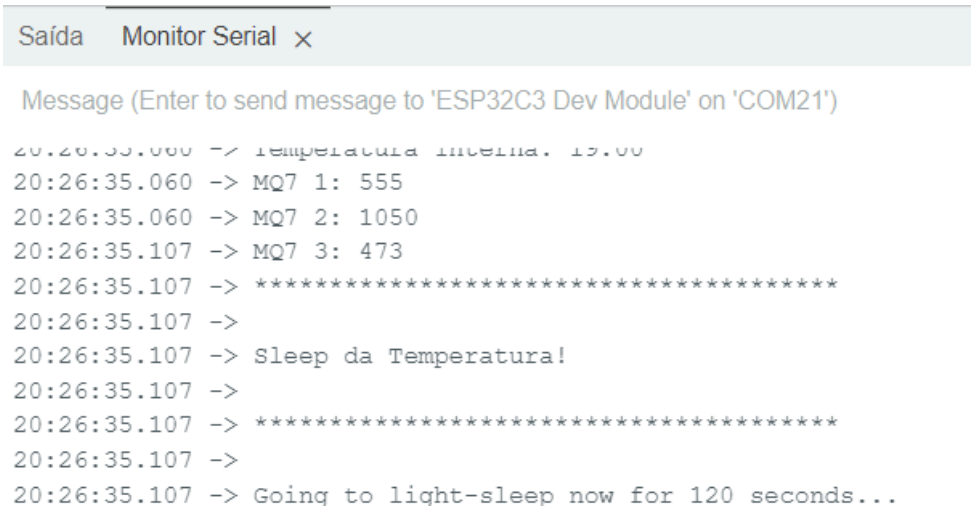
Figura 23 – Teste do WatchDog Time.

Fonte: Elaborado pelo autor.

4.1.3 Validação do Protocolo de Temperatura

O último teste a nível de hardware é o da temperatura interna. Essa informação é fornecida pelo sensor interno do microcontrolador utilizado neste trabalho. Todos os testes em que esse protocolo foi submetido, após as alterações do código-fone, obteve êxito. O teste 9 da Tabela 2 aponta que será necessário um sleep para resfriar a placa e que só seguirá após a temperatura adequada de operação, como pode ser observado na Figura 24.

Figura 24 – Teste de Temperatura Interna



Fonte: Elaborado pelo autor.

4.1.4 Validação de Outliers

Finalizada os testes na etapa do oráculo do hardware foram realizados testes relacionados aos outliers e erro relativo entre os sensores. No teste de 2 da Tabela 2 pode-se observar que o valor fornecido pelo sensor 2 MQ7 está fora da margem permitida. Logo, a resposta condiz com a função implementa, ou seja, refazer a leitura do dado, como é apresentado na Figura 25.

Figura 25 – Teste da Função Outliers

```
07:04:13.607 -> ////////////////////////////////////////////////
07:04:13.607 -> Todos os Testes Aprovados
07:04:13.607 -> Teste de Bateria Aprovado!
07:04:13.607 -> 5.00V
07:04:13.607 -> Teste do WatchDog Aprovado!
07:04:13.607 -> Teste do Temperatura da Interna Aprovado!
07:04:13.607 -> 20.00
07:04:13.607 -> Teste do Temperatura Externa Aprovado!
07:04:13.607 -> 30.00
07:04:13.607 -> Teste do Humidade Aprovado!
07:04:13.607 -> 85.00
07:04:13.654 -> ////////////////////////////////////////////////
07:04:13.654 -> CO : 1000
07:04:13.654 -> CO : 2100
07:04:13.654 -> CO : 1100
07:04:13.654 -> Outiler Detectado em MQ2
07:04:13.654 -> Valor de Erro1: 0.52
07:04:13.654 -> Valor de Erro2: 0.09
07:04:13.654 -> Valor de Erro3: 0.48
07:04:13.654 -> MQ7-1 # MQ7-3
07:04:13.654 -> 1050
```

Fonte: Elaborado pelo autor.

4.1.5 Validação do TMR

O último ponto a ser testado nesta solução é a ocorrência de erros relativos acima de 20%. Pode ocorrer uma situação indesejada que é os três sensores apresentarem leituras divergentes. Neste caso, a confiabilidade do oráculo torna-se comprometida. Este é um caso extremo onde dois sensores apresentam mau funcionamento. A Figura 26 apresenta tal situação e a devida resposta programada.

4.2 REDE *BLOCKCHAIN*

Durante o andamento da pesquisa foram surgindo outras perspectivas para o problema deste trabalho. Algumas análises foram feitas para melhorar a solução, porque ao se analisar os trabalhos relacionados foi possível constatar que geralmente os *Blockchains* são desenvolvidos para receber dados ou informações sem nenhum tratamento e quando isso não ocorre, as demais implementações sugerem a utilização de autenticação de dados através do uso de CA.

No primeiro momento foi implementado um *Blockchain* de uso geral, a princípio uma rede pública, que necessita do uso da mineração para ser adicionado um novo bloco.

Figura 26 – Teste do Erro Relativo

```
20:23:02.731 -> Todos os Testes Aprovados
20:23:02.731 -> Teste de Bateria Aprovado!
20:23:02.731 -> 4.90V
20:23:02.731 -> Teste do WatchDog Aprovado!
20:23:02.731 -> Teste do Temperatura da Interna Aprovado!
20:23:02.731 -> 20.00
20:23:02.731 -> Teste do Temperatura Externa Aprovado!
20:23:02.731 -> 17.00
20:23:02.731 -> Teste do Humidade Aprovado!
20:23:02.731 -> 85.00
20:23:02.731 -> ////////////////////////////////////////////////
20:23:02.731 -> CO : 30
20:23:02.731 -> CO : 1490
20:23:02.731 -> CO : 1106
20:23:02.731 -> Valor de Erro1: 0.98
20:23:02.775 -> Valor de Erro2: 0.97
20:23:02.775 -> Valor de Erro3: 0.26
20:23:02.775 -> ERRO ACIMA DE 20% entre os 3 Sensores!
```

Fonte: Elaborado pelo autor.

Figura 27 – Função que gera um novo bloco

```
def proof_of_work(self, previous_proof):
    new_proof = 1
    check_proof = False

    while check_proof is False:
        hash_operation = hashlib.sha256(str(new_proof**2 - previous_proof**2).encode()).hexdigest()
        if hash_operation[:4] == '0000': #verifica se o hash inicia os 4 primeiros digitos com zero
            check_proof = True

        else:
            new_proof  += 1
    return new_proof
```

Fonte: Elaborado pelo autor.

Para o seu desenvolvimento foi utilizado a linguagem Python que já fornece nas suas bibliotecas a função de criptografia necessária para a criação da cadeia de blocos. Portanto, qualquer tipo de dado, após a configuração da entrada da rede poderá ser inserido. Nas Figuras 27 e 28 é apresentado parte da codificação implementada.

Para implementação da aplicação Web foi adotado um Microframework Flask, que foi criado para utilizar com Python. Uma das vantagens é que não requer total domínio de desenvolvimento Web. A partir de uma requisição (*request*) que entra em contato com o servidor (servidor local) e ele retorna uma reposta (*response*). Na Figura 29 são descritos os passos aplicados o *Blockchain* desenvolvido.

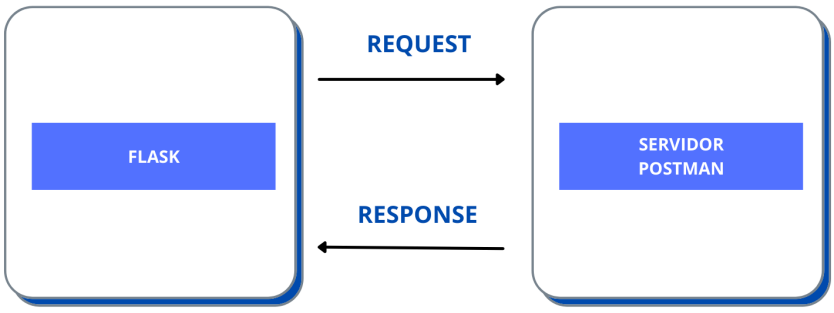
A partir da inserção dessas duas ferramentas apresentadas na Figura 29 foi possível iniciar o processo de criação da rede. O papel desempenhado pelo Postman é integrar a implementação da *Blockchain* e o Flask, ou seja, ele funciona como uma API que recebe

Figura 28 – Função que verifica a Mineração

```
def create_block(self, proof, previous_hash):
    block = {'index': len(self.chain)+1, 'timestamp': str(datetime).now(), 'proof': proof, 'previous_hash': previous_hash}
    self.chain.append(block)
    return block
```

Fonte: Elaborado pelo autor.

Figura 29 – Diagrama de funcionamento do servidor



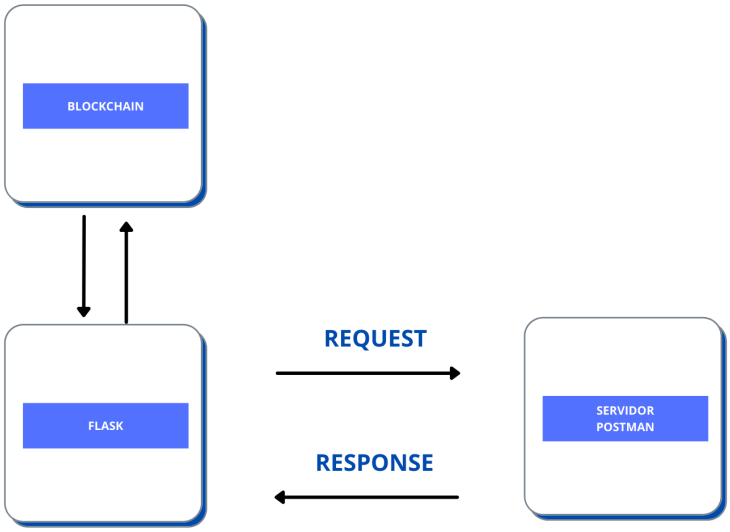
Fonte: Elaborado pelo autor.

as requisições realizadas pela cadeia de blocos e retorna através do Flask a resposta. Na Figura 30 é apresentado todo o processo de criação, mineração e validação da cadeia de blocos.

Após todo sistema configurado, com o Postman e Flask integrado foi possível executar a criação da primeira cadeia de blocos deste trabalho. Na Figura 31 é apresentado o bloco gênesis do *Blockchain*. Nela consta o índice do bloco, o *hash* anterior, a prova de trabalho de mineração, que na implementação foi adotado que será sempre 1 para o primeiro bloco e logo em seguida o *timestamp*.

Na criação de um novo bloco é necessário a junção de algumas tecnologias que são necessárias para a consolidação de um *Blockchain*. Uma delas é a criptografia. Esse processo é composto pela concatenação de todas as informações presentes num bloco. Na Figura 32 é apresentado uma *hash* gerada com uma dificuldade de mineração grau 5. Nela é possível analisar os dados, como: valor do sensor, índice, *hash* anterior, o *nonce* e o *timestamp*.

Figura 30 – Diagrama de funcionamento Completo



Fonte: Elaborado pelo autor.

Figura 31 – Primeiro Bloco da *Blockchain*


```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2022-02-16 18:07:58.086025"
        }
    ],
    "length": 1
```

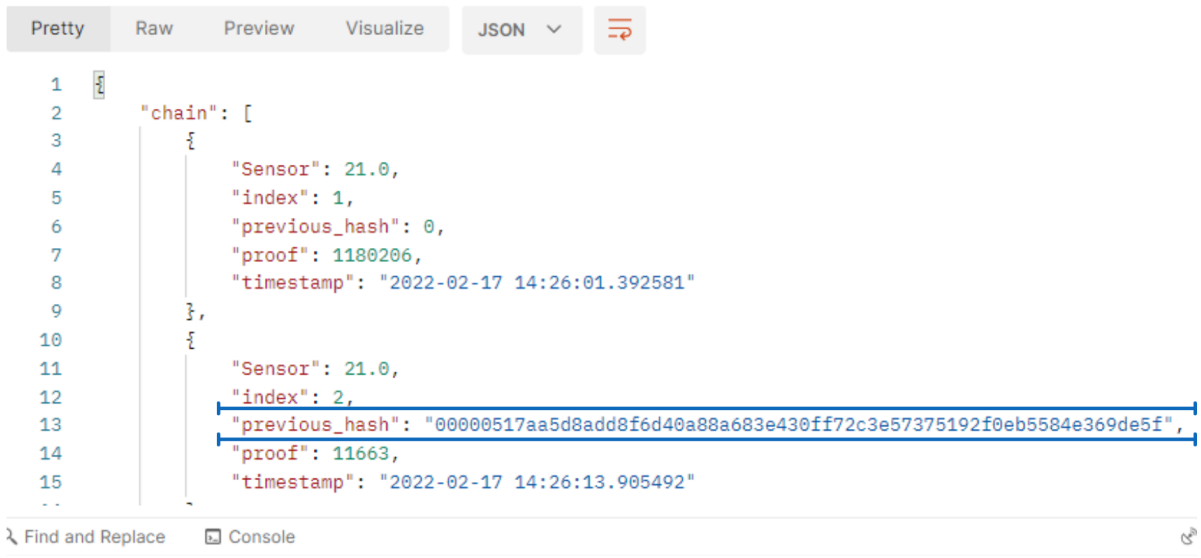

Fonte: Elaborado pelo autor.

Em seguida, temos como resultado uma cadeia de blocos completa, gerada a partir de dados fictícios, como é apresentado na Figura 33 e na Figura 34, um diagrama ilustrando esse *Blockchain*.

O objetivo dos testes foi verificar os principais passos de uma rede *Blockchain*, que são:

- Criar o hash de um bloco concatenando todos os parâmetros do bloco;
- Verificar a partir do protocolo de consenso PoW se o bloco atende o parâmetro Dificuldade, caso isso ocorra, o novo bloco é inserido na corrente;

Figura 32 – Dificuldade de mineração grau 5.



Fonte: Elaborado pelo autor.

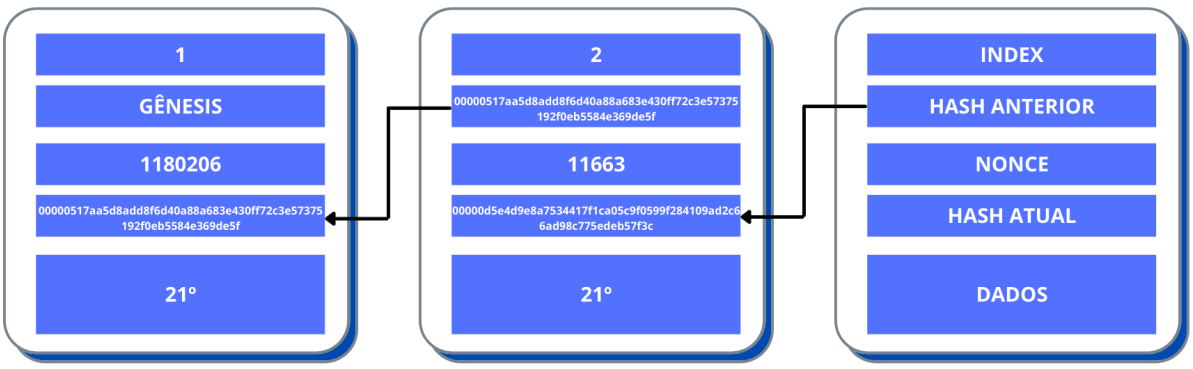
- Posteriormente, o estado atual da cadeia é distribuído na rede P2P;
- Por fim, é esperado que tenha uma sequência de blocos em cadeia com as informações devidamente criptografadas e distribuídas.

Figura 33 – *Blockchain* Completa


```
{
    "chain": [
        {
            "Sensor": 21.0,
            "index": 1,
            "previous_hash": 0,
            "proof": 1180206,
            "timestamp": "2022-02-17 14:26:01.392581"
        },
        {
            "Sensor": 21.0,
            "index": 2,
            "previous_hash": "00000517aa5d8add8f6d40a88a683e430ff72c3e57375192f0eb5584e369de5f",
            "proof": 11663,
            "timestamp": "2022-02-17 14:26:13.905492"
        },
        {
            "Sensor": 21.0,
            "index": 3,
            "previous_hash": "00000d5e4d9e8a7534417f1ca05c9f0599f284109ad2c66ad98c775edeb57f3c",
            "proof": 1407482,
            "timestamp": "2022-02-17 14:26:17.460833"
        },
        {
            "Sensor": 21.0,
            "index": 4,
            "previous_hash": "0000063d4db50728dc45d1ec09e42de6d49a4569c30201c99212571a990fe12f",
            "proof": 2921748,
            "timestamp": "2022-02-17 14:26:39.039507"
        }
    ],
```


Fonte: Elaborado pelo autor.

Figura 34 – Diagrama Ilustrativo do *Blockchain*



Fonte: Elaborado pelo autor.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi proposta uma solução baseada em *Blockchain* e aplicando o conceito de oráculo com o objetivo de desenvolver um sistema IoT em que os dados monitorados sejam íntegros. Foram descritos os principais conceitos envolvidos na solução propostas e os procedimentos de implementação da aplicação.

Como foi proposto por este trabalho, a RSSF e a Rede *Blockchain* foram desenvolvidas, testadas e validadas seguindo todos os procedimentos metodológicos citados anteriormente. Além disso, a conexão entre as duas tecnologias, RSSF e a rede Blockchain também foi realizada e obteve uma performance de 100% de funcionamento. Sendo assim, os dados mensurados pelos sensores MQ7 são enviados ao oráculo, passam pelos protocolos de segurança e em seguida, caso seja um valor válido, é enviado para a rede Blockchain para ser armazenado com segurança.

Os resultados experimentais obtiveram 100% de aproveitamento a partir dos valores fornecidos na entrada do oráculo. Contudo, vale ressaltar que ao inserir o oráculo na rede de sensores sem fio, o tempo de processamento para captar e enviar os valores obtidos pelos sensores tem um acréscimo de aproximadamente 40 milissegundos. Portanto, é viável a inserção do oráculo como mecanismo de segurança em soluções que não tem o tempo como fator crítico.

A rede *Blockchain* foi implementada seguindo os princípios de Nakamoto para garantir que os dados tenham um armazenamento seguro e descentralizado. Além de aplicar a criptografia do tipo SHA-256 nos dados enviados pelo oráculo, a rede opera no tipo P2P e com um protocolo de consenso para manter os dados coerentes.

Como trabalhos futuros propõe-se a inserção de outros parâmetros como protocolos de segurança do oráculo, como, teste da memória EEPROM, por ser um fator ligado diretamente ao microcontrolador e que pode armazenar informações permanentes ligadas a outros protocolos. Além disso, a medição de sinais do ambiente, como umidade para agregar mais integridade nas etapas de verificação.

REFERÊNCIAS

AGUIAR, L. *Uma análise de modelamento de canal e confiabilidade em redes de sensores sem fio em linha*. João Pessoa: Universidade Federal da Paraíba, 2018.

AL-BREIKI, H. et al. Trustworthy blockchain oracles: review, comparison, and open research challenges. *IEEE Access*, IEEE, v. 8, p. 85675–85685, 2020.

BALIGA, A. Understanding blockchain consensus models. *Persistent*, Boca Raton, FL, USA: CRC Press, v. 4, p. 1–14, 2017.

BERGER, R. Think act: Smart city, smart strategy. *Boston: Think act magazine*, 2017.

BURNETT, S.; PAINE, S. Criptografia e segurança–o guia oficial rsa. 1ª edição. *Rio de*, 2002.

CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and smart contracts for the internet of things. *Ieee Access*, Ieee, v. 4, p. 2292–2303, 2016.

FERREIRA, F. N. F. *Política de segurança da informação: guia prático para elaboração e implementação*. [S.l.]: Ciência Moderna, 2008.

GIFFINGER, R. et al. Smart cities. Technische universit‰ t Wien, 2007.

GREVE, F. G. et al. Blockchain e a revolução do consenso sob demanda. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos*, 2018.

JR, W. S. M. et al. Field surveillance of fuel dispensers using iot-based metering and blockchains. *Journal of Network and Computer Applications*, Elsevier, v. 175, p. 102914, 2021.

KOTESKA, B.; KARAFILOSKI, E.; MISHEV, A. Blockchain implementation quality challenges: A literature review. In: . [S.l.: s.n.], 2017.

MAMMADZADA, K. et al. Blockchain oracles: a framework for blockchain-based applications. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2020. p. 19–34.

MILIČEVIĆ, K. et al. Trust model concept for iot blockchain applications as part of the digital transformation of metrology. *Sensors*, MDPI, v. 22, n. 13, p. 4708, 2022.

MORENO, E. D.; PEREIRA, F. D.; CHIARAMONTE, R. B. Criptografia em software e hardware. *São Paulo: Novatec*, p. 21–42, 2005.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, p. 21260, 2008.

NAKAMURA, E. T.; GEUS, P. L. de. *Segurança de redes em ambientes cooperativos*. [S.l.]: Novatec Editora, 2007.

NAM, T.; PARDO, T. A. Smart city as urban innovation: Focusing on management, policy, and context. In: *Proceedings of the 5th international conference on theory and practice of electronic governance*. [S.l.: s.n.], 2011. p. 185–194.

ORMAN, H. Blockchain: the emperors new pki? *IEEE Internet Computing*, IEEE, v. 22, n. 2, p. 23–28, 2018.

PETERS, D. et al. Blockchain applications for legal metrology. In: IEEE. *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. [S.l.], 2018. p. 1–6.

PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: Palgrave macmillan, 2005.

SCHOLLMEIER, R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In: IEEE. *Proceedings First International Conference on Peer-to-Peer Computing*. [S.l.], 2001. p. 101–102.

STALLINGS, W. *Network and internetwork security: principles and practice*. [S.l.]: Prentice-Hall, Inc., 1995.

STALLINGS, W.; BRESSAN, G.; BARBOSA, A. *Criptografia e segurança de redes* . [S.l.]: Pearson Educacion, 2008.

TALBOT, J.; WELSH, D.; WELSH, D. J. A. *Complexity and cryptography: an introduction*. [S.l.]: Cambridge University Press, 2006. v. 13.

XIAO, Y. et al. A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 22, n. 2, p. 1432–1465, 2020.