UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LEIDMAR MAGNUS FESTA

DISPONIBILIDADE DE DADOS PARA SISTEMAS DE INFORMAÇÃO EM CIDADES INTELIGENTES DURANTE DESCONEXÕES DE REDE: UMA ABORDAGEM UTILIZANDO *BLOCKCHAIN*, ORÁCULO E COMPUTAÇÃO EM NÉVOA

CURITIBA

2022

LEIDMAR MAGNUS FESTA

DISPONIBILIDADE DE DADOS PARA SISTEMAS DE INFORMAÇÃO EM CIDADES INTELIGENTES DURANTE DESCONEXÕES DE REDE: UMA ABORDAGEM UTILIZANDO *BLOCKCHAIN*, ORÁCULO E COMPUTAÇÃO EM NÉVOA

Data availability for information systems in smart cities during network outage: an approach using Blockchain, Oracle and Fog Computing

Dissertação apresentado(a) como requisito para obtenção do título de Mestre em Computação Aplicada do Curso de Computação Aplicada da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Daniel Fernando Pigatto

Coorientador: Prof. Dr. Luiz Celso Gomes Junior

CURITIBA

2022

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba

LEIDMAR MAGNUS FESTA

DISPONIBILIDADE DE DADOS PARA SISTEMAS DE INFORMAÇÃO EM CIDADES INTELIGENTES DURANTE DESCONEXÕES DE REDE: UMA ABORDAGEM UTILIZANDO BLOCKCHAIN, ORÁCULO E COMPUTAÇÃO EM NÉVOA

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Computação Aplicada da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Sistemas Computacionais.

Data de aprovação: 06 de Outubro de 2022

Dr. Daniel Fernando Pigatto, Doutorado - Universidade Tecnológica Federal do Paraná

Dra. Ana Cristina Barreiras Kochem Vendramin, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Luis Carlos Erpen De Bona, Doutorado - Universidade Federal do Paraná (Ufpr)

Dr. Luiz Celso Gomes Junior, Doutorado - Universidade Tecnológica Federal do Paraná

Dedico esta dissertação à Deus que me concede diariamente saúde, fé e perseverança.
À minha esposa Priscila, meus filhos Rebeca e Gabriel: obrigado pela compreensão e carinho.
A minha gratidão aos meus pais, Altair e Eulália, que desde a infância me motivavam a estudar.

AGRADECIMENTOS

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem. Certamente esses parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre estas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Agradeço à Deus pela vida e por cuidar de mim em todo este processo.

Obrigado especial à Priscila, minha esposa, companheira, amiga e confidente .

Obrigado aos meus filhos, Rebeca e Gabriel, pelas demonstrações diárias de amor e carinho.

Obrigado ao meu amigo Igor P. Baumann pelas conversas, momentos de descontração e palavras de apoio que foram muito relevantes neste processo! Agradeço a todos os amigos por fazerem parte dessa caminhada, vocês foram essenciais!

Agradeço aos meus orientadores Dr. Daniel F. Pigatto e Dr. Luiz Celso Gomes Jr., que me mostraram os caminhos a serem percorridos e pela confiança depositada.

Agradeço aos membros da banca de defesa, Dr. Luis C. E. De Bona e Dra. Ana Cristina B. K. Vendramin, pelo *feedback* e direcionamento.

Agradeço à *GoLedger* por conceder acesso à plataforma *GoFabric* para realização dos testes propostos e ao Banco do Brasil S.A. pela bolsa parcial concedida.

Enfim, a todos os que de alguma forma contribuíram para a realização deste trabalho.

A aceleração da urbanização da população mundial fomentou o crescimento de problemas econômicos, sociais e ambientais, fato que afeta significativamente as condições de qualidade de vida das pessoas em centros urbanos. O conceito de *cidade inteligente* carrega em sua estrutura soluções para tratar estes problemas através da conciliação entre tecnologia, recursos e desenvolvimento sustentável das cidades. (XIE *et al.*, 2019).

RESUMO

Este trabalho apresenta uma arquitetura que permite o funcionamento de um Sistema de Informação (SI) mesmo em momentos de desconexão da rede, utilizando como caso de uso Unidades Municipais de Saúde (UMS). A arquitetura utiliza uma *blockchain* como base de dados distribuída e um de seus nós está fisicamente localizado dentro de uma UMS, conectado à sua rede local. Este nó é utilizado em momentos de desconexão entre a rede local e a Internet para manter as respostas para as requisições de consultas feitas por usuários autenticados na rede local, funcionando como uma camada de névoa. Também é possível gerar novos dados e, com o auxílio de um *oráculo confiável*, gravar estes em um banco de dados local. Em momentos de desconexão a arquitetura explora o comportamento de um nó da *blockchain* isolado, bem como as funcionalidades de um oráculo para gravar os novos dados gerados. Desta forma é possível garantir que os dados consultados são os mesmos que estão na *blockchain* previamente à desconexão. Também é possível agregar informações suficientes sobre os novos dados gerados para garantir sua integridade. Além disso, estas informações asseguram a rastreabilidade e a auditoria destes dados. Um protótipo da arquitetura foi implementado e usado para realização de testes de execução. Os resultados destes testes mostram que a arquitetura proposta pode manter o funcionamento adequado de um sistema de informação em momentos de desconexão da rede. Desta forma a arquitetura incrementa a resiliência do sistema, sendo sua aplicação adequada para o cenário de uma cidade inteligente, onde se busca maior disponibilidade de dados.

Palavras-chave: *blockchain*; computação em névoa; cidades inteligentes; livro-razão distribuído; oráculo.

ABSTRACT

This work presents an architecture that allows Information Systems functioning even in moments of network disconnection, using the Healthcare Units (HCU) as use case. The architecture uses a blockchain as a distributed database and one of its nodes is physically located inside an HCU, connected to its Local Area Network (LAN). This node is used in disconnection moments between the LAN and the Internet to keep answering the query requests made by authenticated users on the LAN, working as a fog layer. It is also possible to generate new data and, with the help of a *reliable oracle*, write it to a local database. In disconnection moments the architecture explores the blockchain-isolated node behavior, as well as the functionality of an oracle to record the new generated data. In this way it is possible ensure that the data consulted is the same as that on the blockchain prior to disconnection. It is also possible to aggregate enough information about the new generated data to ensure its integrity. In addition, this information ensures the traceability and auditing of this data. An architecture prototype was implemented and used for execution tests. The test results show that the proposed architecture can maintain the information system working properly during some network outage. In this way, the architecture increases the resilience of the system, being its application suitable for a smart city scenario, where data availability is desired.

Keywords: blockchain; fog computing; smart city; distributed ledger; oracle.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

Abreviaturas

cap.	Capítulo
sec.	Seção

Siglas

AC	Autoridade Certificadora
API	*Application Programming Interface*
BDL	Banco de Dados Local
BFT	*Byzantine Fault Tolerance*
BTC	*Bitcoin*
CDV	Carteira Digital de Vacinação
CN	*Challenged Networks*
DiE	Distribuição de Energia
DTN	*Delay Tolerant Network*
EOA	*Externally Owned Account*
ETH	*Ether*
EVM	*Ethereum Virtual Machine*
IDD	*Identidade Digital Descentralizada*
IoT	*Internet of Things*
IBFT	*Istanbul Byzantine Fault Tolerant*
LAN	*Local Area Network*
MEF	Máquina de Estados Finitos
MQTT	*Message Queue Telemetric Transport*
MSP	*Membership Service Provider*

MySQL	My Structured Query Language
OSN	*Ordering Service Nodes*
PBFT	*Practical Byzantine Fault Tolerance*
PoA	*Proof of Authority*
PoET	*Proof of Elapsed Time*
PoS	*Proof of Stake*
PoW	*Proof of Work*
Pu	*Peer* da UMS
RAM	*Random-access Memory*
SP	Serviço de Pedidos
TIC	Tecnologias da Informação e Comunicação
UMS	Unidade Municipal de Saúde
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO

A resiliência de Sistemas de Informação (SI) tem utilidade em várias esferas de atuação nas cidades inteligentes. Nestas cidades a demanda por recursos computacionais de borda e a necessidade de se manter as aplicações em funcionamento mesmo em períodos de desconexão da rede são crescentes. Estas duas necessidades se somam com o volume, tipos de dados e velocidade com que estes são produzidos, bem como a manutenção dos serviços prestados aos cidadãos. Desta forma a resiliência dos SI torna-se relevante e pode ser alcançada através de aplicações que utilizem tecnologias modernas como *blockchain*, computação em névoa e computação em nuvem.

De acordo com Xie *et al.* (2019) características como disponibilidade, confiabilidade e imutabilidade dos dados, bem como o compartilhamento de serviços e recursos computacionais são relevantes no contexto de cidades inteligentes. Estas cidades utilizam as TIC (Tecnologias de Informação e Comunicação) para atingir seus objetivos de desenvolver ambientes urbanos sustentáveis e qualidade de vida dos cidadãos. Em cenários de cidades inteligentes, a ocorrência de uma catástrofe pode causar a desconexão temporária de segmentos da rede. A desconexão é um período no qual o segmento de rede fica sem acesso à Internet. Nesta situação os SI das entidades localizadas naquele segmento podem parar de funcionar e interromper serviços públicos em partes da cidade, causando transtornos para a população (BARON, 2012).

Este trabalho, que utiliza como caso de uso as Unidades Municipais de Saúde (UMS), descreve uma arquitetura que permite manter o funcionamento de um sistema de informação mesmo que a sua rede de dados esteja passando por um momento de desconexão. Aqui a *blockchain* é utilizada como base de dados distribuída e possui um nó chamado *Pu* (*Peer* Ums), o qual está fisicamente localizado dentro da UMS e conectado à sua Rede Local (LAN - *Local Area Network*). Ao ocorrer um momento de desconexão, os usuários que estiverem conectados à LAN e utilizando o SI podem continuar consultando dados da *blockchain* através do *Pu*. Para realizar a inclusão de novos registros é utilizado um módulo de oráculo confiável, com o qual é possível criptografar e incluir novos registros em um banco de dados local. Estes registros serão posteriormente conferidos e efetivados na *blockchain*. Em seu funcionamento normal, o sistema possui uma camada de névoa que se comunica com os usuários e com a *blockchain*. Durante os períodos de desconexão, o nó *Pu* se comporta como uma camada de névoa temporária entre a *blockchain* e os usuários conectados à LAN.

Para realização dos testes da arquitetura foi desenvolvida uma camada de névoa que se comunica com uma *blockchain Hyperledger Fabric* e também um módulo que desempenha o papel de oráculo. A simulação da desconexão de rede foi feita através de requisições apenas para os endereços locais dos nós e a manutenção de apenas um nó (*Pu*) da *blockchain* funcionando. Os resultados dos testes mostram que a arquitetura proposta pode manter o funcionamento adequado de um SI em momentos de desconexão da rede. A arquitetura se mostrou

útil para criar um ambiente virtual resiliente, sendo sua aplicação adequada para o cenário de cidades inteligentes, onde se busca maior disponibilidade de dados.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral do presente trabalho é elaborar um modelo de arquitetura que disponibilize dados para os sistemas de informação em momentos de desconexão da Internet. Para os testes será utilizado o caso de uso de uma Unidade Municipal de Saúde, porém a arquitetura poderia ser aplicada em outros cenários. A arquitetura faz uso das tecnologias de *blockchain*, computação em névoa e oráculo confiável para disponibilizar os dados. O contexto de aplicação deste modelo é o das Cidades Inteligentes, já que na maior parte destas uma das pretensões é buscar uma maior disponibilidade de dados.

1.1.2 Objetivos Específicos

Os objetivos específicos do presente trabalho são:

- Implantar uma rede *blockchain Hyperledger Fabric* com arquitetura tradicional utilizando zonas de acessibilidade diferentes, ou seja, nós localizados em posições geograficamente distintas;
- Simular uma LAN desconectada da Internet inserindo nela um nó da *blockchain* que será usado para leitura de dados;
- Implementar um oráculo confiável no segmento de rede desconectado temporariamente. O oráculo realiza a criptografia/descriptografia dos dados e armazena estes em um banco de dados local. Posteriormente estes dados são persistidos na *blockchain*;
- Implementar o protótipo de um sistema para testes, que consiste no envio de requisições para leitura e escrita na *blockchain*;
- Realizar simulações nas redes criadas, análise e avaliação dos dados obtidos.

1.2 Estrutura do Texto

O cap. 2 apresenta os conceitos fundamentais para compreensão do trabalho, passando por *blockchain*, oráculo confiável, computação em névoa e cidades inteligentes. A seguir o cap. 3 apresenta estudos relacionados ao tema desta pesquisa, mostra aplicações práticas dos conceitos fundamentais e identifica os problemas e desafios relacionados ao presente trabalho. Em

seguida a arquitetura proposta é apresentada no cap. 4, a qual disponibiliza dados para um SI mesmo em momentos de desconexão temporária da rede de dados. Neste capítulo também estão descritos os pontos de falhas e ataques aos quais a arquitetura proposta está sujeita. A seguir, o cap. 5 mostra os experimentos realizados e resultados obtidos. Aqui está descrita a rede *blockchain* implantada para testes, a sua configuração e os gráficos gerados com a medição dos tempos de resposta das requisições. Por fim, o cap. 6 expõe as conclusões e perspectivas sobre este estudo.

2 REFERENCIAL TEÓRICO

Este capítulo revisa os conceitos relacionados à *blockchain*, oráculos confiáveis, computação em névoa, cidades inteligentes e também uma compilação sobre alguns dos principais trabalhos relacionados a esta pesquisa.

2.1 *Blockchain*

Blockchain é uma estrutura de dados idealizada e implementada por Satoshi Nakamoto para resolver o problema do pagamento duplo em uma moeda virtual sem a necessidade do intermédio de uma instituição reguladora confiável. Ela é gerenciada de maneira descentralizada em uma rede de processos pares, ou *peer-to-peer* (NAKAMOTO, 2008). Um dos principais fundamentos da *blockchain* está na sua *cadeia de blocos* que é utilizada para identificar transações realizadas virtualmente. Esta cadeia é registrada e replicada em uma rede para diversos de seus membros, os quais são responsáveis por validar os registros através de um processo de consenso. Além disso, cada bloco da cadeia possui um *hash* de dados dos blocos anteriores, criando a ligação entre os mesmos e provendo mais uma ferramenta para tornar a cadeia segura, imutável e confiável (MOURA; BRAUNER; JANISSEK-MUNIZ, 2020). Esta cadeia de blocos é conhecida como livro-razão distribuído. Swan (2015) caracteriza a *blockchain* como um livro-razão público e descentralizado, o qual pode ser usado para o registro de diversos tipos de dados, conhecidos como *ativos*. Estes podem ser informações sobre validações, interações sociais, registros bancários e outros. Trata-se de um modelo de consenso em escala e, possivelmente, um mecanismo amplamente utilizado para resolver questões de consenso distribuído.

Em relação a participação dos nós na rede, uma *blockchain* pode ser pública (não permissionada) ou privada (permissionada). Em uma *blockchain* pública qualquer nó pode ingressar, acessar todos os dados registrados das transações e atuar como minerador. Exemplos deste tipo de rede são o *Bitcoin* e a *Ethereum* (VALENTA; SANDNER, 2017; WANG *et al.*, 2021). Em redes permissionadas apenas podem ingressar os nós autorizados. A autorização se dá através da concessão de identidades e a *blockchain* possui ferramentas de controle de acesso para manter o sigilo dos dados das transações (WANG *et al.*, 2021).

É necessário entender também que a *blockchain* faz uso massivo de alguns conceitos base. Um destes conceitos é utilizado para proteger informações com o emprego de códigos e cifras, é chamado de *criptografia* (MICROSOFT, 2014). A criptografia moderna utiliza códigos denominados *chaves* para cifrar e decifrar mensagens. Uma chave é utilizada na criptografia de tal forma que não é possível revertê-la sem o conhecimento daquela chave (COULOURIS *et al.*, 2013). As duas principais formas de utilização das chaves são *simétrica* e *assimétrica*. Ao utilizar de forma *simétrica* existe uma única chave que é compartilhada pelo remetente e pelo destinatário. Esta chave é usada para cifrar a mensagem no envio e decifrar a mensagem

no recebimento (CHANDRA *et al.*, 2014). Na forma *assimétrica* existem as chaves *pública* e *privada*. A criptografia pode ser utilizada com o objetivo de *manter a privacidade* das mensagens, para isso o *remetente* codifica a sua mensagem utilizando a *chave pública* divulgada pelo destinatário. Nesta situação o *destinatário* pode decodificar a mensagem usando a sua *chave privada*. Outro objetivo é utilizar uma *assinatura digital*, situação na qual o *remetente* codifica uma mensagem com sua *chave privada*. Assim qualquer destinatário que possuir a chave pública poderá decifrar a mensagem e confirmar a assinatura do remetente (COULOURIS *et al.*, 2013). Estas assinaturas proporcionam o não repúdio do remetente, já que o mesmo não pode negar a assinatura da mensagem, bem como a garantia de autenticidade da mensagem para o destinatário, conforme Zhou e Lam (1999) e Coulouris *et al.* (2013).

Outro conceito é a Máquina de Estados Finitos (MEF), que consiste em um conjunto finito *Q* de estados válidos (considere aqui os estados *si* e *sj*) e um conjunto de transições entre os pares de estados *si* e *sj*. Uma transição ocorre devido a ocorrência de uma *condição* e/ou *ação* de entrada (BEN-ARI; MONDADA, 2017). Além disso, para que uma transição seja válida, ela deve ser realizada entre os estados que fazem parte do conjunto *Q* de estados válidos. Neste caso quem determina as regras de como uma transição é realizada é a *função de transição* (SANTANA *et al.*, 2015). Uma MEF passa para um novo estado sempre devido a um evento de entrada, o qual pode provocar uma mudança de estado e a produção de uma saída. As MEFs são modelos que podem ser utilizados na representação de sistemas, já que possuem um número finito de estados e transições, assim como um sistema computacional real (SOUZA, 2010).

Por fim o conceito de *hash*, que são algoritmos que mapeiam dados grandes e de tamanho variável para pequenos dados de tamanho fixo. Como seu resultado final é um resumo, ele não revela o conteúdo original integral dos dados (COULOURIS *et al.*, 2013). Sua utilização assegura que as informações recebidas sejam exatamente as mesmas informações enviadas, garantindo a integridade das mesmas.

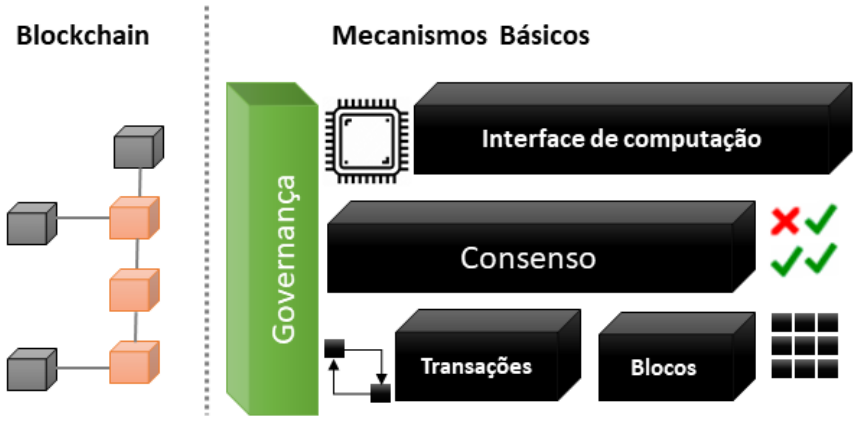
2.1.1 Uma visão geral sobre a *blockchain*

Para compreender o conceito de *blockchain*, é possível considerar a existência de uma infraestrutura de sistemas que consome os recursos daquela *blockchain*, a qual é formada por um conjunto de mecanismos interconectados, como ilustrado na figura 1. Nela é exibida do lado esquerdo uma representação generalista de como se comporta a *blockchain* e do lado direito um detalhamento dos seus mecanismos básicos. No lado esquerdo é possível observar uma cadeia de blocos conectados por uma linha. No lado direito, no nível mais inferior são exibidas as transações assinadas pelos *peers* da rede. Estas representam operações com os ativos da rede, que podem envolver a transferência de valores ou mesmo a simples validação de uma tarefa, dentre outros. Uma transação é realizada entre dois participantes e, após a assinatura digital de pelo menos um destes, ela será disseminada para seus *peers* vizinhos. Um membro

que se conecta à *blockchain* é chamado de *nó*. Porém, existe uma distinção para os *nós* que realizam o processamento da *blockchain*, determinando se uma transação é válida e agrupando as mesmas em blocos, que são chamados de *nós completos* (CASINO; DASAKLIS; PATSAKIS, 2019).

O nível intermediário mostra a camada de *Consenso*, que é utilizada para decidir se uma transação é válida e deve ser mantida na cadeia de blocos (CACHIN; VUKOLIĆ, 2017). Com este intuito, um algoritmo de consenso é executado pelos *nós completos*. O problema computacional que será resolvido pelo consenso exige muito poder de processamento e o seu resultado garante que não houve adulteração dos dados. A camada de *Interface de Computação* oferece funcionalidades que permitem o desenvolvimento de aplicações mais avançadas (CASINO; DASAKLIS; PATSAKIS, 2019), como a REST API - uma ferramenta que possui um conjunto de operações que pode ser aplicado sobre um conjunto de recursos acessados através de um cliente *HTTP (Hypertext Transfer Protocol)* de forma simplificada (SURWASE, 2016). Por fim a camada de *Governança* trata da interação entre a *blockchain* e o usuário final, envolvendo elementos como interfaces gráficas e regras de negócio (CASINO; DASAKLIS; PATSAKIS, 2019).

Figura 1 – Visão geral da arquitetura *blockchain*



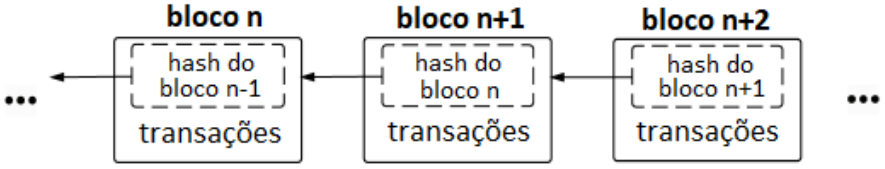
Fonte: Casino, Dasaklis e Patsakis (2019).

2.1.2 O livro-razão distribuído

Como característica essencial, uma *blockchain* mantém uma estrutura chamada de livro-razão distribuído, o qual armazena blocos de dados das transações. Esta estrutura aceita apenas a inclusão de novos registros, os quais são inseridos somente após o último registro armazenado. Utilizando esta lógica, os blocos de transações já armazenados não podem mais ser modificados (KOLB *et al.*, 2020). Além das transações, cada bloco do livro-razão também tem o registro da data e hora de sua criação (*timestamp*), o número do bloco e o *hash* do bloco anterior. A exceção é o primeiro bloco, chamado de *gênesis*, que não possui um *bloco pai* sendo comum a todos os membros da rede.

O agrupamento do *timestamp*, número do bloco e *hash* é chamado de *cabeçalho*. A figura 2 ilustra de forma simplificada a lógica do *hash* em cada bloco. Nesta representação cada bloco possui um *hash* indicado pelo retângulo pontilhado e logo abaixo as transações assinadas. As setas que apontam para a esquerda indicam, por exemplo, que o bloco *n+2* possui o *hash* do bloco *n+1*, que por sua vez possui o *hash* do bloco *n* e assim por diante. Nesta lógica, o bloco subsequente sempre carrega em seu *hash* um pouco das informações de cada um dos blocos anteriores. Esta característica cria a ligação entre os blocos formando a cadeia com registros imutáveis, já que se qualquer dado de blocos anteriores for alterado o *hash* também é alterado e assim invalida o bloco com informações conflitantes.

Figura 2 – Armazenamento do *hash* em cada bloco da *blockchain*



Fonte: Christidis e Devetsikiotis (2016).

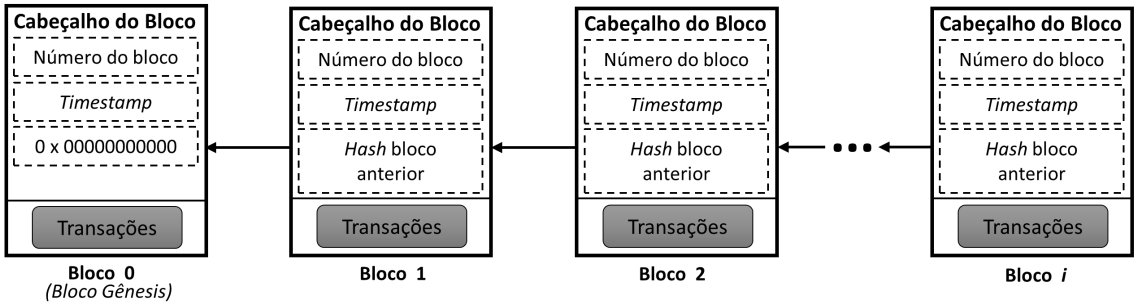
Um exemplo mais detalhado sobre o conteúdo do cabeçalho de cada bloco pode ser observado na figura 3, com destaque para:

- o *bloco 0 (ou gênesis)* mais à esquerda, no início da *blockchain*;
- cada bloco da cadeia possui o espaço no qual o cabeçalho fica armazenado e também o espaço para armazenamento de informações das transações;
- cada cabeçalho é composto pelo número do bloco atual, o *timestamp* e um *hash* do bloco anterior (campos demarcados pelo tracejado). O bloco gênesis possui *hash* definido como zero;
- a seta entre os blocos 0 e 1 indica que o cabeçalho do *bloco 1* possui o *hash* do *bloco 0*. Esta conexão do *hash* entre os blocos se estende por toda a cadeia e faz com que o último bloco incluído possua ligação com todos os seus predecessores.

Ao observar sua rede *peer-to-peer* é possível entender melhor a dinâmica de funcionamento da *blockchain*. Trata-se de um conjunto de nós (membros da rede) que possui uma cópia de todos os blocos já registrados e cada um destes nós pode ser utilizado pelos diversos usuários da rede para solicitar transações. Para exemplificar, considera-se que cada usuário insere suas próprias transações na rede através de seu próprio nó e que a usuária Alice possui 1 BTC *(Bitcoin)* o qual deseja transferir para o usuário Bob (FREY *et al.*, 2018):

- A interação entre usuários e *blockchain* ocorre com a utilização de chaves assimétricas (CHRISTIDIS; DEVETSIKIOTIS, 2016). Pode-se observar na figura 4 que Alice cria

Figura 3 – Detalhamento do cabeçalho dos blocos da *blockchain*

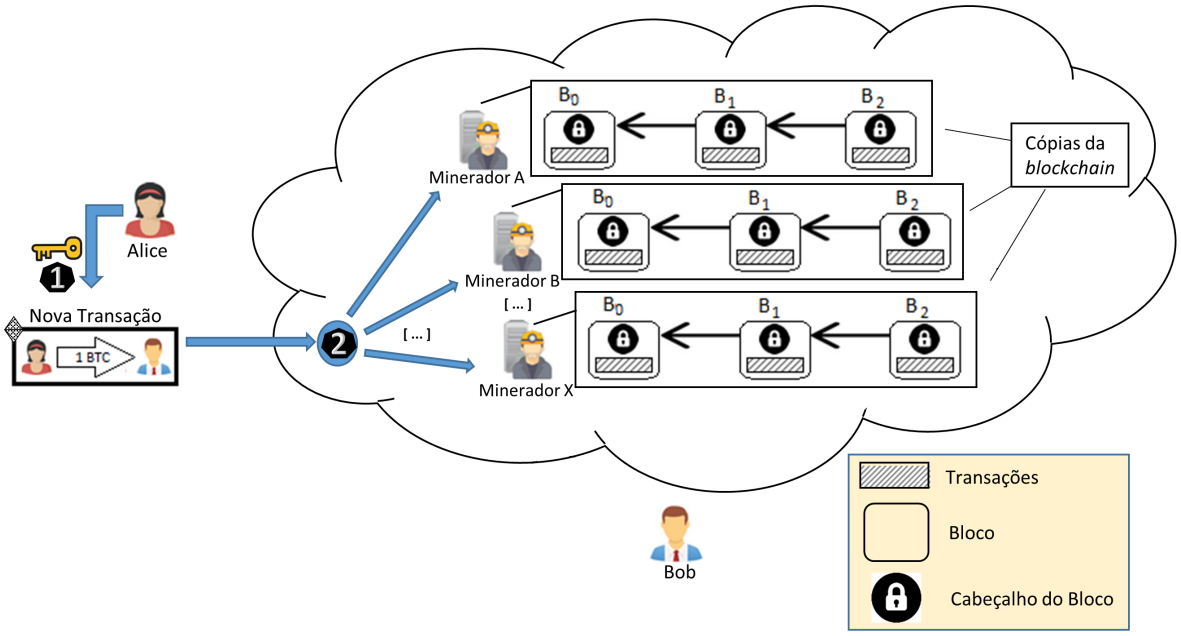


Fonte: Adaptado de Christidis e Devetsikiotis (2016) e Casino, Dasaklis e Patsakis (2019).

uma nova transação assinando esta com sua *chave privada*. Após isso, a saída desta transação é criptografada com a chave pública de Bob, pois desta forma fica indicado que Bob é o favorecido do resultado da transação 1;

- Após isso, Alice transmite a referida transação para a rede de mineradores, também conhecidos como nós validadores, para que ela possa ser incluída na *blockchain* 2;

Figura 4 – Criação de uma nova transação na *blockchain*



Fonte: adaptado de Frey *et al.* (2018).

Na figura 5 estão os próximos passos do processo, onde (CHRISTIDIS; DEVETSIKIOTIS, 2016; FREY *et al.*, 2018):

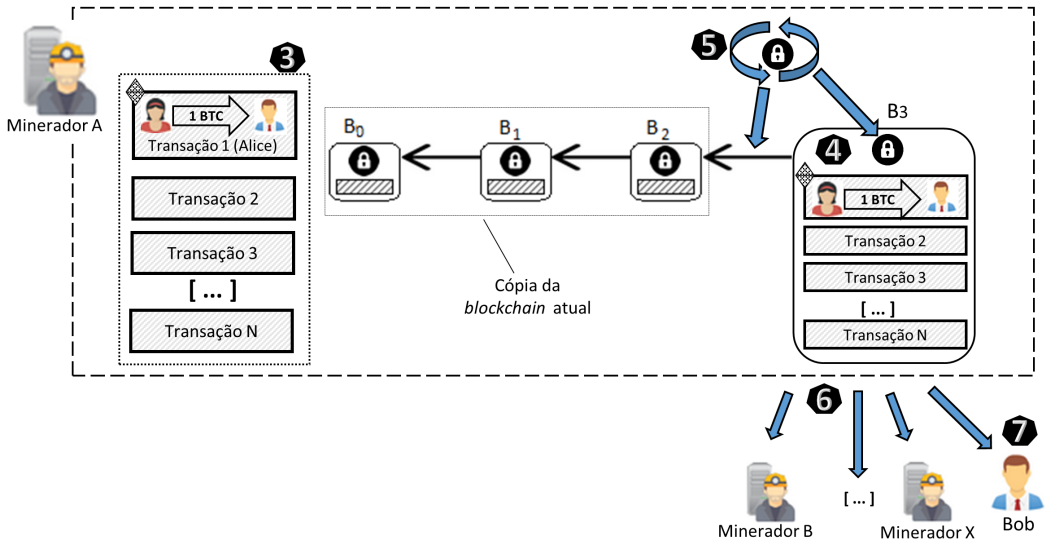
- Os mineradores recebem a nova transação de Alice juntamente com várias outras e verificam se estas são válidas. Considere que o *Minerador A* foi o primeiro a realizar esta tarefa 3. Nesta etapa as transações inválidas são descartadas;

- O *Minerador A* inclui a transação de Alice em um novo bloco proposto, juntamente com várias outras transações que foram coletadas na rede 4, iniciando logo em seguida os cálculos necessários para vincular este bloco proposto na *blockchain* atual 5. Esta etapa prévia exige que o minerador resolva um problema matemático extremamente custoso, o qual exige um elevado tempo de processamento *(intervalo de consenso)* para ser resolvido. O intervalo de consenso determina a periodicidade com a qual um novo bloco será *minerado* (incluído na cadeia) – no *Bitcoin* este intervalo é de 10 minutos. O bloco proposto (B_3) passa a compor a *blockchain* caso o *Minerador A* tenha êxito no processamento. Se isso ocorrer, a composição da cadeia passa a ser B_0, B_1, B_2 e B_3;

- Na sequência o *Minerador A* dissemina o novo bloco B_3 para os outros mineradores da rede 6, que verificam se o bloco contém transações e o *hash* válidos. Se isto for verdadeiro, eles incluem o bloco em suas cópias da cadeia e aplicam as transações contidas nele para atualizar seus dados. Caso as condições não sejam cumpridas, o novo bloco é descartado;

- Se tudo correu bem, o bloco B_3 passa a fazer parte da cadeia e Bob recebe o valor de 1 BTC 7 enviado por Alice.

Figura 5 – Validação e inclusão da nova transação em um bloco



Fonte: adaptado de Frey *et al.* (2018).

2.1.3 Consenso Distribuído

O *Problema dos Generais Bizantinos (BFT)*, descrito por Lamport, Shostak e Pease (1982), é uma das situações que ilustra a complexidade para se chegar a um consenso em um

cenário de sistemas distribuídos onde existem várias entidades com um objetivo em comum. Este problema consiste em decidir se os generais do exército Bizantino, que estão acampados ao redor de uma fortaleza, devem atacar ou não. Para alcançar a vitória todos os generais devem atacar simultaneamente e, para decidir o momento do ataque, eles trocam mensagens entre si através de mensageiros que podem não conseguir entregar as mensagens. Além disso, alguns dos generais podem ser traidores que tentam evitar o consenso sobre o momento do ataque.

Outro problema que surge em sistemas distribuídos que trabalham com moedas virtuais é o do *Gasto Duplo*, ou seja, reutilizar a moeda em duas transações que ocorrem de forma simultânea (DU *et al.*, 2017). O BFT e o Gasto Duplo são duas das situações que podem ocorrer ao se tentar chegar ao consenso em um cenário distribuído. Para tratar este tipo de situação existem os protocolos de consenso, os quais buscam definir regras para que seja possível convergir para uma solução comum (BACH; MIHALJEVIĆ; ZAGAR, 2018). Alguns destes protocolos são:

- ***Proof of Work (PoW):*** introduzido pela primeira vez por Dwork e Naor (1993), este protocolo é utilizado na prevenção de ciberataques e na confirmação de transações na rede *Bitcoin* (ZHANG *et al.*, 2020). A alocação de direitos contábeis e recompensas monetárias de acordo com o poder de processamento empregado por cada *nó minerador* é o princípio do PoW. Os valores alocados são repassados para o minerador que conseguir realizar o processamento e criar um novo bloco aceito pela rede (DU *et al.*, 2017). Diversos destes nós competem entre si para resolver um problema matemático que exige uma grande carga de trabalho *(workload)*. Assim, o primeiro que resolver o problema cria o próximo bloco e recebe uma recompensa cotada na moeda da *blockchain* (ZHANG *et al.*, 2020). Vale ressaltar que o PoW não é amigável no quesito recursos ambientais, tendo em vista a necessidade de um alto poder de processamento (SARAF; SABADRA, 2018; BENTOV; GABIZON; MIZRAHI, 2016), podendo gerar um consumo de energia igual ou maior do que o de alguns países para manter o funcionamento da rede de mineração (VRIES, 2018);

- ***Proof of Stake (PoS):*** assim como no PoW, neste algoritmo também ocorre mineração, porém não há competição entre os nós para realizar este processo. Aqui a própria rede escolhe um *nó validador* (não um minerador), ou seja, aquele que validará a transação. Se por algum motivo o nó validador não conseguir validar a transação, um novo validador é selecionado. Este processo segue até que ocorra a validação da transação por algum nó (SARAF; SABADRA, 2018). O validador é escolhido aleatoriamente, porém os candidatos a validador devem manter um *depósito de segurança*, que é um valor cotado na moeda da *blockchain* que fica em poder da mesma até que a participação do candidato a validador seja encerrada (BENTOV; GABIZON; MIZRAHI, 2016; MOINDROT; BOURNHONESQUE, 2017). Em 2022 a *blockchain Ethereum* passou a utilizar o protocolo de consenso PoS, o qual é considerado mais amigável com o meio

ambiente já que o seu consumo de energia pode ser de até 99,95% menor em relação ao seu predecessor PoW (Ethereum Foundation, 2022);

- ***Proof of Authority (PoA):*** neste protocolo de consenso um grupo de nós da rede é qualificado como *validadores confiáveis*. Existe uma lista pública na *blockchain* com as chaves públicas destes validadores. Durante um período determinado, cada um destes validadores exerce o papel de líder, sendo responsável pela proposição de novos blocos de transações. Caso a maioria dos validadores aprove o bloco proposto ele será aceito pela rede (KOLB *et al.*, 2020). O PoA não possui punição para comportamentos maliciosos como aqueles existentes no PoS ou mesmo os incentivos financeiros existentes no PoW, por este motivo os validadores precisam ser confiáveis e a sua utilização é aconselhável em *blockchains* permissionadas (KOLB *et al.*, 2020; ZHANG *et al.*, 2020).

- ***Proof of Elapsed Time (PoET):*** este protocolo utiliza um *enclave SGX (Software Guard Extension)* da Intel [1] para realizar parte do seu processo. O SGX são funções da CPU que criam um enclave, que é um ambiente seguro que fornece confidencialidade e integridade para execução do código (MCKEEN *et al.*, 2016; YAGA *et al.*, 2018). Dentro de seu enclave local, cada nó minerador inicia a execução de uma instância do PoET. Quando cada um destes nós consegue produzir o seu novo bloco, cada enclave atribui ao seu nó um tempo aleatório e os nós devem aguardar este tempo para propor o novo bloco da cadeia. Nesta lógica, o nó que receber o menor tempo será considerado o líder e vai propor o novo bloco (KOLB *et al.*, 2020). Para comprovar que o líder aguardou o tempo estabelecido, o PoET gera um atestado digital que é exigido para que o novo bloco proposto seja aceito pela rede. Vale destacar que quanto mais módulos SGX um nó possuir, maiores serão as suas chances de se tornar o líder, ou seja, maior será o investimento em hardware (CACHIN; VUKOLIĆ, 2017);

- ***Raft:*** a primeira fase no Raft é a eleição de um nó líder, que é decidida pela maioria dos votos dos nós da rede. O líder eleito se torna responsável pelo gerenciamento do livro-razão e frequentemente envia mensagens para os demais nós para manter sua autoridade. Os demais nós são chamados de seguidores (OUSTERHOUT; ONGARO, 2014). Na segunda fase cabe ao líder receber solicitações de novas transações dos clientes, incluir estas em seu livro-razão e depois replicar estas transações para os nós seguidores (SINGH *et al.*, 2022).

[1] http://software.intel.com/sgx

2.1.4 Arquitetura *ordem-execução* das Transações

Ordem-execução é o nome dado para a arquitetura transacional utilizada pelas plataformas tradicionais de *blockchain* (IBM, 2018). Um exemplo de plataforma é o *Bitcoin*, que utiliza o protocolo de consenso PoW. Outros exemplos são *Quorum* [2] e *Tendermint* [3], cujo consenso é baseado em BFT. Na arquitetura *ordem-execução* as transações são ordenadas pelo protocolo de consenso o qual posteriormente as dissemina para os demais nós completos da rede. Após isso todos estes nós devem executar as transações de forma sequencial.

Mesmo sendo amplamente utilizada devido a sua simplicidade, esta arquitetura possui limitações significativas, como a execução sequencial de todas transações por todos os nós completos. Esta dinâmica de execução pode ser considerada um problema pois limita o desempenho da rede, podendo gerar lentidão nas respostas, o que favorece ataques de *negação de serviço*. Para tratar situações como esta, algumas *blockchain* possuem mecanismos de precificação dos recursos computacionais utilizados no processamento das transações (ANDROULAKI *et al.*, 2018). Desta forma se a transação começa a consumir muitos recursos ela se torna economicamente inviável e o minerador pode rejeitá-la (WOOD, 2014). Outra limitação é a necessidade de que todas as transações executadas após o consenso sejam determinísticas, ou seja, para uma determinada entrada devem produzir sempre o mesmo resultado. Isso evita que os nós da rede fiquem com dados diferentes entre si (*fork*) e seja necessário realizar a verificação do estado de todos – já que isso causa atrasos e lentidão em toda a rede (ANDROULAKI *et al.*, 2018).

2.1.5 *Hyperledger Fabric*

O *Hyperledger*[4] foi criado pela *Linux Foundation* em 2015. Trata-se de um conjunto de projetos de *código aberto*, cujo foco está na criação de um ecossistema baseado em *blockchain* que engloba diversos tipos de indústrias. Atualmente existem seis projetos estáveis e mais nove projetos em incubação. O *Hyperledger Fabric* é um dos projetos estáveis, sendo uma plataforma para criação de redes *blockchain* permissionadas (SARAF; SABADRA, 2018; Hyperledger Foundation, 2016). Isso significa que para um nó fazer parte da rede, deverá receber uma identidade, a qual é fornecida e gerenciada pelo Provedor de Serviços de Associação (MSP - *Membership Service Provider*). Caso o nó não possua a identidade será ignorado pelo restante da rede (GORENFLO *et al.*, 2020). Outro detalhe das redes *Fabric* é que estas funcionam executando *contratos inteligentes* (ou *chaincodes*), os quais são programas que definem as regras de negócio e realizam tarefas de manipulação dos ativos da rede (WANG *et al.*, 2021). Uma tarefa pode ser a gravação ou consulta de dados na rede, o registro de interações sociais, regis-

[2] https://consensys.net/quorum/
[3] https://tendermint.com/
[4] https://www.hyperledger.org

tros de compra e venda de bens, dentre outras (SWAN,2015). A *Fabric* foi a primeira plataforma *blockchain* a aceitar *chaincodes* escritos em linguagens de programação de uso geral e sem a implementação de uma moeda virtual nativa da rede (ANDROULAKI *et al.*, 2018).

Uma rede *blockchain Fabric* é formada por um consórcio de organizações que realizam transações entre si. Cada organização possui um conjunto de nós que desempenham uma ou mais funções distintas (WANG *et al.*, 2021; ANDROULAKI *et al.*, 2018), dentre elas a função de *Autoridade Certificadora* (AC), que se responsabiliza pela entrega das identidades aos nós que solicitam autorização para participar da rede. Por padrão, cada organização participante de uma *blockchain Fabric* possui uma AC e o conjunto de todas as ACs da rede forma o MSP (Hyperledger Fabric, 2020). Outro conjunto de nós é chamado de *peers* e tem a incumbência de manter o livro-razão distribuído, executar as transações propostas e validar estas transações. Esta última tarefa é executada por alguns dos *peers* chamados de *endossadores*, os quais verificam se as transações cumprem os requisitos, dentre eles a assinatura digital do solicitante, e depois geram o endosso de cada transação (ANDROULAKI *et al.*, 2018). Existem também os nós *ordenadores*, que recebem as transações propostas enviadas para a rede e determinam a ordem geral de todas as transações na *Fabric* (BRANDENBURGER *et al.*, 2017; VALENTA; SANDNER, 2017). Por padrão, cada organização da rede possui um nó deste tipo e o conjunto de todos os *ordenadores* forma o *Serviço de Pedidos* (SP). Alguns dos nós são chamados de *clientes* e executam tarefas para os usuários finais. Dentre estas tarefas estão a transmissão das transações propostas para os *ordenadores* e a comunicação com os *peers* (VALENTA; SANDNER, 2017)

Normalmente uma transação em uma rede *Fabric* segue um fluxo básico no qual um nó *cliente* envia uma proposta de transação *W* para os *peers* invocando um *chaincode*. Cada *peer* que executa o *chaincode* produz um resultado chamado de *endosso*, que é anexado à transação *W* e submetido para o SP. Por sua vez o SP agrupa a transação *W*, e outras enviadas para a rede, em um bloco que é transmitido para todos os *peers*. Estes validam as transações e gravam o novo bloco no livro-razão (BRANDENBURGER *et al.*, 2018). Neste fluxo descrito, todos os dados envolvidos são públicos e estão disponíveis para todo o consórcio. Isso gera problemas de privacidade, especialmente quando os dados de uma transação são sensíveis e devem ser visíveis para apenas algumas das organizações, mantendo o sigilo entre as demais (Hyperledger Fabric, 2020). Para situações como esta é possível utilizar no *Fabric* o recurso da *Coleção de Dados Privados* (CDP). Com o CDP o *hash* das transações é incluído no livro-razão, mas apenas as organizações autorizadas têm acesso aos dados destas transações (BROTSIS *et al.*, 2020; WANG *et al.*, 2021).

2.1.6 Oráculos de *Blockchain*

Existe uma limitação dos contratos inteligentes que é foco de muitos projetos: a impossibilidade destes em acessar dados do mundo externo, ou seja, dados fora da *blockchain*

(também chamados de dados *off-chain* ou fora da cadeia). Isso pode ser restritivo para aplicações que acessam dados relacionados a preços de ativos, verificação de identidade, dentre outros (AL-BREIKI *et al.*, 2019). Além disso, conforme Xu *et al.* (2016) um dos desafios que ainda persiste em redes *blockchain* é a limitação do espaço de armazenamento. Esta limitação frequentemente é superada armazenando os conjuntos grandes de dados brutos em bases de dados externas e mantendo na *blockchain* apenas metadados, como em Lone e Mir (2019), Adler *et al.* (2018) e Azaria *et al.* (2016).

Neste contexto, um oráculo é um componente que permite a interação dos contratos inteligentes da *blockchain* com dados fora da cadeia de forma segura e confiável (ADLER *et al.*, 2018). É necessário salientar que um oráculo realiza a busca, verificação, autenticação e disponibilização dos dados de fontes externas confiáveis. Porém, ele não é a fonte destes dados (Beniiche, 2020). Pasdar, Dong e Lee (2021) destaca que um oráculo pode ser classificado de acordo com algumas características relevantes como:

- *Confiança:* pode ser centralizada ou descentralizada;
- *Fonte dos dados:* dados podem ser originados por software, hardware ou humanos;
- *Fluxo dos dados:* indica o sentido no qual a informação trafega em relação à *blockchain*, podendo ser de entrada ou de saída;
- *Padrão da arquitetura:* modelo de disponibilização dos dados.

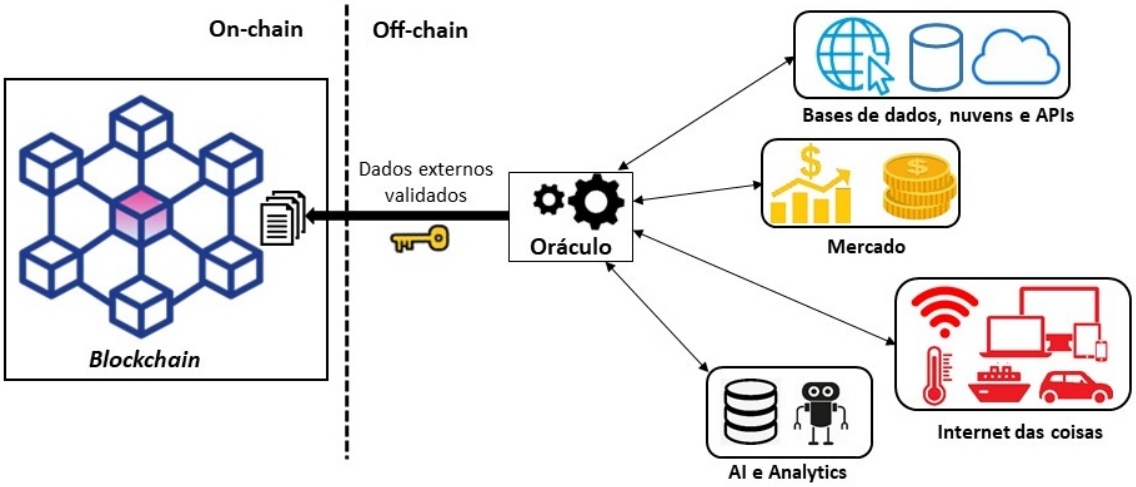
Quanto à *forma de confiança*, os oráculos *centralizados* são mais eficientes e com melhor desempenho, pois uma única entidade realiza o processamento dos dados (PASDAR; DONG; LEE, 2021). Estes possuem problemas inerentes à centralização como: maior facilidade de adulteração e de ataques (KOLINKO, 2014). Os oráculos *descentralizados* resolvem parte destes problemas já havendo soluções comerciais e de código aberto. Alguns exemplos são o *Orisi*[5] que utiliza um grupo de oráculos para chegar a um consenso e o *Provable Things*[6] que implementa contratos inteligentes vinculados a bases de dados independentes (MÜHLBERGER *et al.*, 2020).

Com relação às *fontes de dados*, Pasdar, Dong e Lee (2021) e Beniiche (2020) classificam como: *oráculo de software* quando recebe dados de fontes *online* como servidores ou bases de dados; *Oráculo de hardware* aquele que recebe dados originados no mundo físico como sensores eletrônicos ou dispositivos de borda; *Oráculo humano* é um indivíduo que possui conhecimento especializado, verifica a autenticidade de uma informação e a repassa para o contrato inteligente. No quesito *fluxo dos dados* um oráculo de *entrada* leva dados do mundo externo para dentro da *blockchain*, enquanto um oráculo de *saída* entrega os dados da *blockchain* para o mundo externo (AL-BREIKI *et al.*, 2020). A figura 6 exibe um oráculo de entrada,

[5] https://orisi.org
[6] https://provable.xyz

onde observa-se mais à direita as fontes de dados, no centro está o oráculo que busca os dados nestas fontes, faz a validação e repassa para um contrato inteligente da *blockchain*.

Figura 6 – Padrão de arquitetura de um oráculo centralizado de entrada



Fonte: Adaptado de Xu *et al.* (2018) e Al-Breiki *et al.* (2020).

De acordo com Beniiche (2020) e Al-Breiki *et al.* (2020), o *padrão de arquitetura* pode ser definido como *leitura imediata, publish-subscribe* e *requisição-resposta*. A *leitura imediata* é empregada aos dados disponibilizados para uma decisão momentânea e que provavelmente não serão mais utilizados futuramente. O *publish-subscribe* serve para dados voláteis que precisam ser transmitidos continuamente. O padrão *requisição-resposta* é empregado quando o conjunto de dados é muito grande para ser armazenado na *blockchain* e, por isso, ele permanece fora da cadeia. Neste caso, a cada interação dos usuários apenas uma pequena parte dos dados será utilizada. Por isso existe um módulo de monitor do oráculo que recupera e retorna partes do conjunto de dados em cada requisição.

Kolinko (2014) destaca que existem alguns problemas conhecidos relacionados à confiabilidade e integridade dos oráculos centralizados. Para estes pode ocorrer adulteração ou não haver clareza no processo de validação dos dados. Outros autores como Lo *et al.* (2020), Pasdar, Dong e Lee (2021), Caldarelli (2020) e Egberts (2017) informam que um oráculo cria um *ponto único de falha* ao introduzir uma terceira parte totalmente confiável em um sistema descentralizado como a *blockchain*. Este ponto de falha está relacionado com a necessidade do oráculo em ser aceito como confiável por todos os participantes que endossam as transações na rede. Xu *et al.* (2018) salienta que dados externos voláteis podem criar dificuldades na validação já que as transações da *blockchain* são imutáveis.

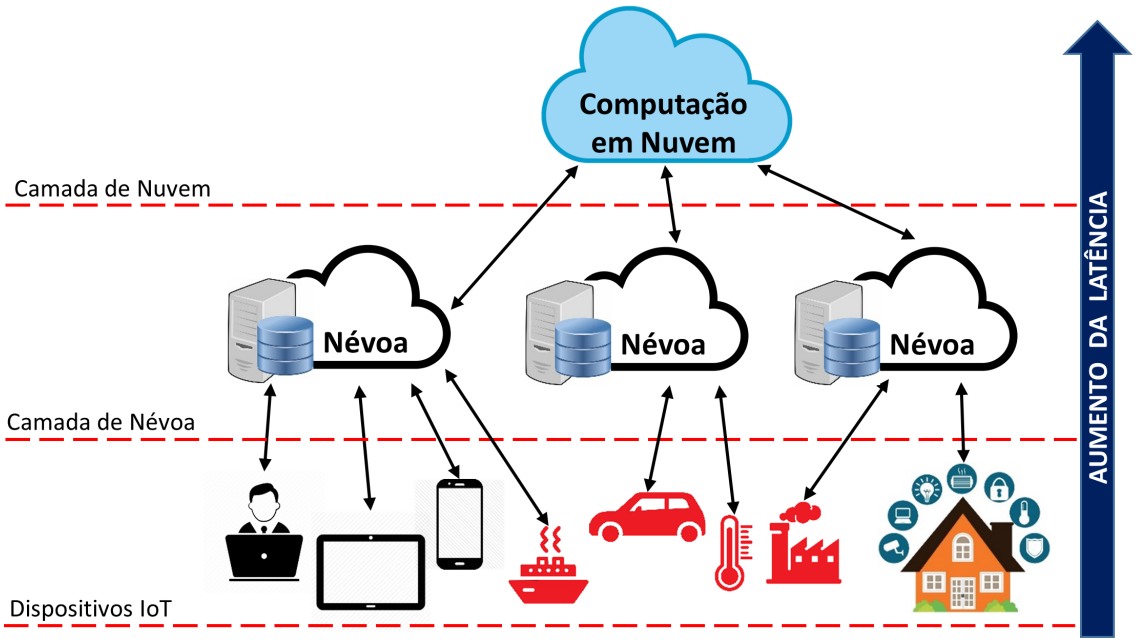
2.2 Computação em Névoa

A computação em névoa, do inglês *fog computing*, foi originada pela Cisco e está entre a camada da borda, onde ficam os dispositivos, e a nuvem. A principal função da névoa é esten-

der as funcionalidades da nuvem, levando estas para perto do usuário final (CISCO, 2015). Em sua estimativa, a *International Data Corporation* informa que haverá em torno de 41,6 bilhões de dispositivos IoT conectados à internet em 2025 e que estes vão gerar 79,4 ZB (zettabytes) de dados (ANMULWAR; GUPTA; DERAWI, 2020). Em um cenário como este, o volume e variedade de dados produzidos, bem como a velocidade com que estes são gerados pelos dispositivos de IoT, estão muito acima da capacidade de processamento e análise dos modelos de computação em nuvem existentes (CISCO, 2015; ALGHAMDI; ALZAHRANI; THAYANANTHAN, 2021; SAURABH; DHANARAJ, 2021; NAVEEN; KOUNTE, 2022).

Em um caso de uso de uma plataforma petrolífera, onde os aplicativos geram terabytes de dados diariamente e exigem processamento em tempo real, a computação em nuvem tradicional focada no processamento em lotes não é suficiente (RABAH; RESEARCH; NAIROBI, 2018). Já a névoa soluciona problemas desta natureza com seu alto grau de virtualização e fornecendo serviços computacionais para a borda com menor latência, prevenção de indisponibilidades, heterogeneidade de dispositivos e recursos geograficamente distribuídos (BONOMI *et al.*, 2012).

Figura 7 – Dinâmica de funcionamento entre os dispositivos IoT e as camadas de névoa e nuvem



Fonte: Adaptado de Mayer (2020), Iorga *et al.* (2018) e Hajibaba e Gorgin (2014) .

A figura 7 mostra de uma forma simplificada a interação entre dispositivos IoT, névoa e nuvem. Ao lado direito a seta azul indica a menor latência nas camadas inferiores (IORGA *et al.*, 2018). Na parte inferior desta figura estão os diversos dispositivos IoT. Estes conectam-se à camada de névoa e utilizam seus serviços de processamento, armazenamento e outros. Na parte central da figura está a camada de névoa a qual é composta por vários nós geograficamente distribuídos (HAJIBABA; GORGIN, 2014). Conforme destacado por Iorga *et al.* (2018), estes nós podem ser físicos (como servidores e *switches*) ou virtuais (máquinas virtuais, *cloudlets* e

outros). Esta camada recebe os dados dos dispositivos IoT e realiza o tratamento adequado. Assim os dados necessários imediatamente, como de sistemas em tempo real, são processados na névoa. Já os dados que podem esperar o tratamento (armazenamento de longo prazo, sistemas assíncronos e outros) podem ser selecionados e tratados posteriormente na névoa ou na nuvem (CISCO, 2015). Na parte superior estão os serviços da computação em nuvem que, conforme Bonomi *et al.* (2012), são centralizados e possuem menos dispositivos.

Algumas das características essenciais da computação em névoa são a *baixa latência*, que se dá em razão da *pulverização geográfica* e do *grande número de seus dispositivos*, os quais estão próximos da borda (BONOMI *et al.*, 2012). Outra característica é o *suporte para redes inteligentes e sensores de uso geral em larga escala*, aplicados no monitoramento de ambientes (CHEN; ZHANG; SHI, 2017) e *interações em tempo real* para as aplicações (HAJIBABA; GORGIN, 2014). No âmbito da computação em névoa um conceito muito usado de aplicações é o do mundo conectado, no qual pessoas, equipamentos e serviços estão interligados (SUGAYAMA; NEGRELLI, 2016). Seguindo este conceito destacam-se aplicações como *veículos conectados*, que se comunicam com serviços através de sensores e dispositivos de conexão a redes (PIZZOLATO, 2019), e as *redes de energia inteligentes*, que controlam a distribuição de energia e integram fontes de energia renovável (JENKINS; LONG; WU, 2015).

Outras aplicações são as *cidades inteligentes* e *casas inteligentes*. Na primeira são utilizadas as Tecnologias de Informação e Comunicação (TIC) para construir um ambiente urbano sustentável (XIE *et al.*, 2019). A segunda utiliza as TIC para viabilizar a comunicação e o uso de serviços exteriores, criar um ambiente monitorado e controlar equipamentos elétricos (HAMED, 2012), as quais buscam a melhoria da qualidade de vida do cidadão. Existem também várias aplicações no ramo da *saúde*. Estas podem manter o histórico médico unificado do paciente (MAYER, 2020), realizar a coleta periódica de dados fisiológicos para acompanhamento (HASSEN; DGHAIS; HAMDI, 2019) ou mesmo realizar um diagnóstico automatizado de acordo com os dados coletados (TULI *et al.*, 2020).

É possível observar que a computação em névoa e a *blockchain* (seção 2.1) possuem características de sistemas distribuídos (HAJIBABA; GORGIN, 2014), portanto existe uma grande convergência das aplicações destas tecnologias (RABAH; RESEARCH; NAIROBI, 2018). Desta forma alguns recursos da *blockchain*, como imutabilidade e transparência dos dados, podem ser utilizados para sanar algumas das deficiências de confiabilidade no compartilhamento de dados entre organizações em ambientes de névoa. Especialmente no contexto de cidades inteligentes, estes recursos podem auxiliar na tomada de decisões conjuntas, criando um ambiente confiável e seguro (XIE *et al.*, 2019).

2.3 Cidades Inteligentes

A ideia de *cidade inteligente* é ampla e vem sendo estudada ao longo dos anos através do levantamento de vários aspectos da vida urbana que podem ser aprimorados e reinventados

(CHOURABI *et al.*, 2012). A conceituação sobre este tema existe há pelo menos duas décadas, mas de acordo com Cocchia (2014), ainda não existe uma definição abrangente aceita por acadêmicos, empresas e instituições sobre o que é *cidade inteligente* bem como quais são seus principais elementos e limites. Ao realizar um levantamento mais amplo, várias definições semelhantes sobre este tema foram encontradas, as quais receberam nomenclaturas distintas em várias situações, como é o caso das apresentadas na Tabela 1:

Tabela 1 – Definições encontradas na literatura para *cidades inteligentes*.

Nomenclatura	Definição
Comunidade Inteligente	Refere-se à utilização do potencial das TIC em benefício da melhoria da vida social e do trabalho da população em uma determinada região geográfica (California Institute for Smart Communities, 1997).
Cidade Virtual	Foca no papel cívico dos cidadãos através da divulgação das suas manifestações, preocupações e ideias por meio de ambientes virtuais conectados às cidades reais (SCHULER, 2002).
Cidade Digital	Trata-se de uma representação virtual das dimensões social, cultural, política, ideológica e teórica de uma cidade real (COUCLELIS, 2004; SCHULER, 2002).
Cidade da Informação	Nestas cidades ocorre a coleta de informações através de canais digitais oficiais e não oficiais e a posterior divulgação através de portais na internet. Dentre estas informações estão as notícias locais, novos serviços aos cidadãos e o contato direto dos responsáveis por cada departamento administrativo (FERGUSON; SAIRAMESH; FELDMAN, 2004) .
Cidade do Conhecimento	O principal objetivo nestas cidades é desenvolver o conhecimento, especialmente através do estímulo à criação, atualização contínua e inovação. Para catalisar estas características são criadas dinâmicas e sistemas para interação contínua entre os cidadãos locais e de outras cidades (ERGAZAKIS; METAXIOTIS; PSARRAS, 2004).
Intelligent City	São cidades que possuem uma boa infraestrutura para instituições que geram conhecimento e realizam uma boa gestão do mesmo, comunicação digital e incentivo à criatividade da população. Tudo isso cria um ambiente com grande capacidade de aprendizagem e inovação (KOMNINOS, 2006) .
Ubiquitous City	Aprimoramento da ideia de *Cidade Digital*, mas neste caso envolve a evolução da tecnologia que passa a ser onipresente em todos os aspectos da cidade (ANTHOPOULOS; FITSILIS, 2010) .

Fonte: Cocchia (2014).

Conforme Xie *et al.* (2019), apesar de todas estas definições uma comumente aceita é que uma cidade inteligente utiliza as TIC com o objetivo de melhorar a qualidade de vida dos cidadãos e também construir um ambiente urbano sustentável. Desta forma as cidades inteligentes passam a buscar melhorias em diversas frentes, como: maior transparência do governo *(smart government)*, incentivo ao envolvimento dos cidadãos *(smart citizens)*, gestão eficaz do tráfego e do transporte público *(smart mobility)*, melhor utilização de recursos *(smart resources)*, maior proteção ambiental *(smart environment)*, dispositivos de controle inteligentes e melhores serviços *(smart services)* de saúde, energia e educação (GIFFINGER *et al.*, 2007; XIE *et al.*, 2019). Com tudo isso em mente, é necessário entender que uma cidade não pode ser chamada de *inteligente* apenas por implementar algumas destas melhorias isoladamente, pois, de fato,

uma cidade inteligente engloba vários destes aspectos de forma cumulativa e integrada (GORI; PARCU; STASI, 2015).

2.3.1 Cidades Inteligentes Resilientes

Conforme Godschalk (2003), uma cidade resiliente possui componentes construídos ou naturais que mantêm seu funcionamento durante uma catástrofe. Estes componentes podem ser infraestrutura, sistemas de comunicação, monitoramento, instalações de energia, cursos d'água e outros que, mesmo com pequenas avarias, permanecem funcionando. Desta forma um local consegue resistir a uma catástrofe com um nível tolerável de perdas, sejam estas danos estruturais, diminuição da produtividade ou qualidade de vida (MILETI, 1999). Na prática, em uma cidade resiliente os impactos de uma catástrofe são atenuados. Neste contexto existem menos desabamentos de edifícios, ocorrem menos cortes de energia, a exposição ao risco é menor para famílias e negócios, ocorrem menos mortes, ferimentos e interrupções de comunicação (GODSCHALK, 2003). Uma das razões pelas quais a implantação deste tipo de cidade é necessária pode ser ilustrada através do relatório elaborado pela Munich Reinsurance Company (2001). Este documento indica que em 2001, mesmo ano do ataque terrorista ao *World Trade Center*, os desastres naturais ao redor do mundo resultaram em 25.000 mortes e US$ 36 bilhões de perdas econômicas. Godschalk (2003) defende que com isso em vista fica clara a necessidade de se implementar soluções que tornem as zonas urbanas resilientes e mitiguem as consequências de catástrofes. Destaca ainda dois fatores importantes sobre a resiliência das cidades:

1. Não é possível prever *onde*, *como* e *quando* ocorrerão os desastres, por isso a necessidade de se projetar cidades que possam tratar de forma eficaz as situações de contingência;

2. Durante momentos de desastres as pessoas e os componentes construídos devem ter uma performance melhor em *cidades resilientes* do que em outros locais (BOLIN; STANFORD, 1998).

2.3.2 Algumas Implementações de *Blockchain* no Contexto de Cidades Inteligentes

Conforme o prospecto de urbanização mundial (United Nations, 2018), em 2007 pela primeira vez mais da metade da população mundial passou a ser urbana. Em 2018 o percentual da população vivendo em áreas urbanas chegou em 55% e a estimativa é que até 2050 ocorra um incremento de 2,46 bilhões de pessoas vivendo nestas áreas, perfazendo quase 70% da população mundial. Nos aspectos relacionados à educação, saúde, economia, trabalho e outros, é possível observar melhorias acarretadas pelo processo de urbanização na qualidade de vida das pessoas (DAVIS, 1965). Porém a alta densidade demográfica urbana também traz novos

desafios e problemas para a qualidade de vida, desafios estes relacionados à escassez dos recursos ambientais, aumento da quantidade de veículos, poluição do ar, dentre outros. Para tratar estas situações, é necessário que todas as entidades que existem nas cidades utilizem a abordagem de *cidade inteligente*, a qual propicia novas formas de se conciliar tecnologia, recursos e desenvolvimento sustentável das cidades, buscando assim a melhoria da qualidade de vida dos cidadãos (XIE *et al.*, 2019).

Desta forma é possível abordar vários problemas, existentes nas cidades, utilizando soluções com *blockchain*, principalmente naqueles onde é necessário coletar dados de vários dispositivos ou disponibilizar dados de forma distribuída. Neste cenário os conceitos de IoT e computação em névoa também ganham espaço (CHRISTIDIS; DEVETSIKIOTIS, 2016). Abaixo seguem algumas aplicações mencionadas por alguns autores que utilizam *blockchain* e que podem ser utilizadas no contexto de cidades inteligentes:

- *Segurança dos dados:* uma cidade inteligente está diretamente relacionada com o conceito de aplicações em tempo real. Estas sincronizam vários recursos e serviços para possibilitar a implementação de melhorias em centros urbanos para que estes sejam sustentáveis e habitáveis. Desta forma as aplicações possuem conectividade entre si e armazenam seus dados de forma centralizada. Dentro desta dinâmica surgem problemas de segurança dos dados (CHINNASAMY *et al.*, 2021);

- *Identidade digital:* estima-se que 2,4 bilhões de pessoas no mundo não possuem documentos para comprovação de sua identidade. Uma das soluções para isso é o conceito de Identidade Digital baseada em *blockchain*, a qual pode ser utilizada como meio de autenticação para serviços digitais e proporciona ao cidadão o controle sobre os seus dados (RIVERA *et al.*, 2017). O *Hyperledger Indy* pode ser utilizado de forma integrada a outras tecnologias com esta finalidade. Trata-se de uma plataforma de *blockchain* para gerenciamento de Identidade Digital Descentralizada (SOPEK *et al.*, 2019);

- *Histórico médico:* atualmente existe uma grande fragmentação do Registro do Histórico Médico (RHM) de um paciente, já que os dados estão espalhados pelos hospitais, clínicas e centros de imagens onde o mesmo realizou consultas e exames (AZARIA *et al.*, 2016). Neste contexto é possível criar aplicações baseadas em *blockchain* que mantenham todos os registros do RHM e, sob autorização do paciente, realizem o compartilhamento destes dados entre médicos e outros profissionais da saúde (ALLADI *et al.*, 2019);

- *Sistemas forenses para investigação:* de acordo com Lone e Mir (2019), a evidência digital é volátil e complexa, sendo de fácil transmissão e suscetível à adulteração e/ou remoção. A manutenção da integridade da evidência é necessária durante todo o período de uma investigação. Isso se torna complexo quando é necessário documentar toda a cronologia de manipulação da evidência – processo denominado Cadeia de Custódia.

Em contextos como este, onde é necessário garantir a integridade e a autenticidade dos registros, as aplicações baseadas em *blockchain* trazem grandes vantagens (LE *et al.*, 2019);

- *Votação eletrônica e sistemas de opinião popular:* tendo em vista a imutabilidade de seus registros, a *blockchain* se torna uma escolha sensata e pode garantir que um plebiscito não seja manipulado (MUTH *et al.*, 2019).

- *Cadeia de abastecimento e logística:* de acordo com Alladi *et al.* (2019), são mecanismos de rastreamento que mantêm os registros dos produtos transportados, deste a origem até a chegada na prateleira do comerciante. A *blockchain* pode ser utilizada com esta finalidade, resolvendo com eficiência os problemas de rastreabilidade em cadeias de abastecimento de produtos. Mecanismos como este podem evitar que problemas como a contaminação de alimentos ou falsificação de remédios prejudiquem em grande escala a população;

- *Serviços financeiros:* inicialmente a *blockchain* foi utilizada para transações financeiras na rede do *Bitcoin*, onde transações são efetivadas através da confiança distribuída e sem a necessidade de um terceiro confiável (NAKAMOTO, 2008). Outros sistemas desta natureza já foram implementados, os quais proporcionam redução de custos e maior confiabilidade (SARAF; SABADRA, 2018; ALLADI *et al.*, 2019).

3 TRABALHOS RELACIONADOS

Com base na literatura pesquisada, esta seção apresenta informações a respeito de aplicações relacionadas aos temas deste trabalho: *blockchain*, oráculos, computação em névoa e resiliência de cidades inteligentes.

3.1 Oráculos em *Blockchain*

Ao implantar a *blockchain* como parte de um sistema maior, Xu *et al.* (2016) abrem novas possibilidades de aplicações. Neste estudo existe interação entre a *blockchain* e os outros componentes do sistema através de um oráculo de validação de dados. No cenário proposto, qualquer conjunto muito grande de dados é mantido fora da *blockchain* devido a limitações de desempenho. Algumas bases de dados auxiliares armazenam estes arquivos e trocam dados com a *blockchain*. Em outro trabalho, Adler *et al.* (2018) utilizaram o formato de votação gamificada com oráculo descentralizado. Neste oráculo, denominado de ASTRAEA, uma entidade que faz parte do sistema pode desempenhar uma ou mais funções que podem ser de: *emissor*, o qual envia as proposições; *votante* vota nas proposições disponíveis; ou *certificador*, que realiza a validação do processo.

Breidenbach *et al.* (2021) retratam a *Chainlink*[1] como um provedor de recursos computacionais para contratos inteligentes. O sistema funciona como um *framework* de propósito geral, o qual possui nós executados por diversos tipos de entidades que juntas formam um oráculo distribuído. Este oráculo possui um processo de consenso e consegue entregar conjuntos de dados externos para um contrato inteligente da *blockchain*. Outra solução comercial é o oráculo descentralizado *Town Crier*[2] introduzido por Zhang *et al.* (2016). Este oráculo utiliza o protocolo *Hypertext Transfer Protocol Secure* (HTTPS) para buscar conjuntos de dados denominados como datagramas. Estes são validados por um enclave SGX e posteriormente repassados para um contrato inteligente na *blockchain*.

Kolinko (2014) descreve um *framework* de código aberto denominado *Orisi*. Trata-se de um oráculo de entrada descentralizado para transferências de *Bitcoin*. Através de um canal comum vários oráculos recebem os dados que serão analisados e o mesmo canal é usado para informar sobre a validação destes dados. Cada oráculo possui um endereço de assinatura e para que os dados sejam considerados válidos, são necessárias 50%+1 das assinaturas do total de oráculos ativos. Além disso são necessárias as assinaturas do remetente e do favorecido pela transferência do valor. O *Tiny Oracle*[3], apresentado por Beregszaszi (2016), é outro projeto de código aberto. Trata-se de um modelo que pode ser utilizado como referência para imple-

[1] https://chain.link/
[2] https://www.town-crier.org/
[3] https://github.com/axic/tinyoracle

mentação de um oráculo para redes *Ethereum*. Ele utiliza a chamada remota de processos para executar várias funcionalidades, inclusive a validação dos dados repassados pelas fontes.

Hess, Malahov e Pettersson (2017) apresentam a *Aeternity*[4], uma plataforma de *blockchain* com oráculos nativos e incorporados em forma de módulos. A principal ideia que os autores apresentam é que *não* se deve armazenar o estado e o código dos contratos inteligentes na *blockchain*, pois isso limita a comunicação com o mundo externo. Desta forma a *Aeternity* utiliza oráculos implementados dentro de sua rede durante o processo de consenso, os quais podem requisitar dados externos através de APIs. O site oficial da plataforma destaca também que a implementação de oráculos nativos é um dos caminhos para se chegar a uma *hyperchain*, ou seja, a integração de duas ou mais *blockchain*. Em sua dissertação conceitual Egberts (2017) aborda a questão conhecida como *O Problema do Oráculo*, explicando as características e desvantagens de um oráculo centralizado. Em seu texto o autor destaca que a utilização de oráculos na *blockchain* reintroduz um ponto único de falha em um sistema descentralizado. Isso significa, por exemplo, que em determinadas situações a *blockchain* pode ter que aguardar uma resposta do oráculo para finalizar o processamento de uma transação. O autor também traz um compilado de várias soluções já propostas para resolver os problemas apresentados, como: sistema de reputação, oráculos humanos e Intel SGX.

3.2 Resiliência de computação em Névoa

Ribeiro Junior e Kamienski (2021b) propõe a criação do sistema *Fog-DaRe (Fog-based IoT Data Resilience)*, que traz resiliência para o fluxo de dados entre a névoa e a nuvem, mesmo em momentos de desconexão da rede. Isso é alcançado através de mecanismos de filtragem e persistência de dados. Em um trabalho anterior, Ribeiro Junior e Kamienski (2021a) propuseram o *TW-IoT (Trustworthiness for IoT Framework)*, que foca em aspectos teóricos e destaca conceitos relacionados à confiabilidade e resiliência em névoa e IoT. Também é proposto um *framework* conceitual aplicável na comunicação entre os dispositivos de IoT e a camada de névoa.

Em seu trabalho Jeong *et al.* (2017) propõe o *framework* distribuído *Crystal*, no qual a aplicação em névoa executa uma ou mais instâncias deste *framework*. A principal característica é que ao detectar a falha de um nó, o sistema realiza automaticamente a alocação de um nó substituto, o qual recebe a cópia dos dados do nó que falhou. Outro *framework* foi proposto por Al-Khafajiy *et al.* (2019) e busca melhorar a QoS (Qualidade do Software). A estratégia adotada é o processamento compartilhado entre os nós da névoa, o qual é controlado por um mecanismo de balanceamento de carga. Desta forma é possível tratar uma grande quantidade de dados recebidos dos dispositivos da borda.

O trabalho de Luzuriaga *et al.* (2017) utilizou os conceitos de Redes Tolerantes a Atrasos (DTN) e implementou melhorias no protocolo *Message Queue Telemetric Transport (MQTT)*,

[4] https://aeternity.com/

utilizado para comunicação entre dispositivos de IoT. Os autores destacam que a principal limitação do MQTT é a baixa resiliência. Focando nesta limitação, os pesquisadores realizaram a integração do MQTT com uma DTN e conduziram experimentos para avaliar e melhorar a comunicação entre uma infraestrutura de sensores sem fio e dois *Raspberry Pi*, que atuam como nós da DTN. Os dados foram armazenados em memória RAM. Ainda utilizando conceitos de DTN os dois trabalhos abaixo foram aplicados em ambientes rurais nos quais existem muitas áreas sem infraestrutura de rede. Este tipo de situação é chamada de *Challenged Networks (CN)* e nela os dispositivos de borda passam muito tempo desconectados da camada de névoa. Em uma CN para que os dados cheguem à névoa, os dispositivos utilizam uma estratégia de *store-carry-and-foward*, onde eles replicam seus dados para outro dispositivo sempre que um está ao alcance do outro. Assim, sempre que um dispositivo consegue conectar-se à névoa ele repassa para a rede todos os seus dados e também os dados replicados por outros dispositivos. No trabalho de Castellano, Risso e Loti (2018) foram utilizados dispositivos conectados a máquinas agrícolas. Já no estudo conduzido por Kulatunga *et al.* (2017) os dispositivos estavam acoplados a vacas que eram manejadas pela propriedade.

3.3 Soluções para Resiliência de Cidades Inteligentes

Um dos trabalhos foi produzido por Babar, Tariq e Jan (2020) e propõe um sistema confiável para gerenciamento da distribuição de energia (DiE). Os autores consideram que setores importantes, como segurança nacional e economia, dependem fortemente da eletricidade. Para evitar problemas na DiE, o sistema utiliza IoT e Aprendizado de Máquina para administrar de forma segura uma rede inteligente de distribuição de energia. Este é um fator de grande importância no campo de resiliência das cidades, já que a energia é parte fundamental para o funcionamento de sistemas que mantém o comércio e os serviços funcionando.

É necessário destacar também as soluções teóricas apontadas no trabalho de Bolin e Stanford (1998) que não trata de tecnologia, porém aponta a coalizão que ocorreu entre vários órgãos, entidades e empresas para reerguer as comunidades afetadas pelo terremoto de *Northridge* em 1994. O autor destaca o conceito de Kent (1987), no qual os desastres não podem ser dissociados da vida cotidiana já que são um reflexo dela e, desta forma, quaisquer práticas que tentam realizar esta dissociação não conseguem traçar ações efetivas para prevenção de calamidades.

Também no campo teórico, Godschalk (2003) tratou da mitigação de riscos urbanos relacionados a perigos naturais ou terroristas. Ele destaca que é necessário desenvolver cidades resilientes que sejam capazes de resistir a choques severos e que estes não sejam sucedidos pelo caos imediato ou dano permanente. Batty *et al.* (2012) propõe um modelo teórico para cidades inteligentes com aplicações que podem trazer mais eficiência na administração da cidade. O autor propõe aplicações ligadas à mobilidade, transações financeiras e cadeia de suprimentos. Com os dados destes sistemas destaca que é possível melhorar a tomada de decisões. Por

fim, o compilado de artigos *Public Space Development in the Context of Urban and Regional Resilience* (POLKO, 2013) trata principalmente da resiliência econômica das cidades citando alguns casos reais que ocorreram ao longo da história.

3.4 Contribuição desta pesquisa

Alguns dos trabalhos citados anteriormente que utilizam *blockchain*, computação em névoa ou oráculos abordam esferas relacionadas à resiliência de redes, reconexão segura em névoa, replicação de dados entre os nós da névoa, criação de oráculos descentralizados ou ainda *frameworks* que buscam canalizar o fluxo de dados de várias fontes. Todas estas aplicações dependem de leitura/escrita de dados em alguma fonte que pode ser uma *blockchain* ou banco de dados tradicional. Os problemas aparecem nos momentos de desconexão, quando um nó (ou grupo de nós) não consegue mais se comunicar com as fontes de dados. Este trabalho foca na resiliência para os sistemas de informação de cidades inteligentes em momentos de desconexão da rede. Integrando os conceitos de *blockchain*, computação em névoa, oráculo e cidades inteligentes, a arquitetura proposta busca disponibilizar dados para os clientes conectados aos segmentos temporariamente isolados da rede.

A tabela 2 mostra o resumo dos trabalhos relacionados e o presente estudo. A coluna *Reconexão Segura* indica se o estudo utiliza alguma forma segura para se reconectar com a sua fonte de dados principal. O agrupamento de colunas indicado por *Dados Durante a Desconexão* mostra as funcionalidades disponíveis mesmo durante um período de desconexão. Neste grupo, a coluna *Consulta de dados* indica se as consultas continuam ativas para todos os dados existentes até o momento da desconexão. *Procedência da consulta* indica se o estudo garante que os dados consultados durante a desconexão são provenientes da fonte de dados principal. Já a *Segurança em novos dados* indica se existe algum mecanismo que fornece algum nível de segurança para a inclusão de novos dados, evitando que estes sejam adulterados. A coluna FSA (*Framework, Sistema ou Arquitetura*) informa se o estudo propõe a criação de uma estrutura mínima para tratamento dos dados e a coluna *Armazenamento na névoa* mostra qual tipo de estrutura é usada para armazenar os dados na camada de névoa. A última coluna, *Aplicação proposta*, informa qual é a principal aplicação do estudo conforme indicado pelos autores.

Os trabalhos relacionados possuem mecanismos que incrementam a resiliência dos SI, porém em momentos de desconexão da Internet os mesmos não possuem uma abordagem eficaz para manter o sistema em funcionamento, já que dependem de fontes de dados *online*. Este trabalho se difere dos demais apresentados pois, mesmo durante os períodos de desconexão, consegue manter as consultas a uma fonte de dados e lidar com as requisições dos usuários. Desta forma é possível manter um SI funcionando nestes períodos sem grandes prejuízos para os usuários. Dentre as características levantadas com base nos trabalhos relacionados, indicadas na tabela 2, o presente estudo possui todas e consegue preencher uma parte da lacuna relacionada a resiliência de sistemas de informação.

Tabela 2 – Compilação dos trabalhos relacionados

		DADOS DURANTE A DESCONEXÃO					
	Reconexão Segura	**Consulta de dados**	**Procedência da consulta**	**Segurança novos dados**	**FSA**	**Armazenamento na névoa**	**Aplicação proposta**
Kolinko (2014)	*				*	N/A	Oráculos
Zhang et al. (2016)	*				*	N/A	Oráculos
Xu *et al.* (2016)	*				*	N/A	Oráculos
Beregszaszi (2016)	*				*	N/A	Oráculos
Jeong *et al.* (2017)	*				*	Banco de dados	Névoa
Luzuriaga *et al.* (2017)	N/A	*	*	Protocolo MQTT	*	Memória RAM	Névoa DTN
Kulatunga *et al.* (2017)	N/A	*	N/A	N/A	*	Memórias flash e RAM	Névoa DTN
Hess *et al.* (2017)	*				*	N/A	Oráculos
Adler *et al.* (2018)	*				*	N/A	Oráculos
Castellano *et al.* (2018)		*			*	Memórias flash e RAM	Névoa DTN
Al-Khafajiy et al. (2019)	*	*			*	Banco de dados	Névoa
Babar et al. (2020)	*		*		*	N/A	Distr. Energia
Breidenbach et al. (2021)	*				*	N/A	Oráculos
Ribeiro Junior e Kamienski (2021a)	*			Criptografia		Banco de dados	Névoa
Ribeiro Junior e Kamienski (2021b)	*	*		Persistência BD	*	Banco de dados	Névoa
Resiliência Sist. Inform. (proposto)	*	*	*	*Blockchain*	*	Banco de dados	Resiliência de sistemas

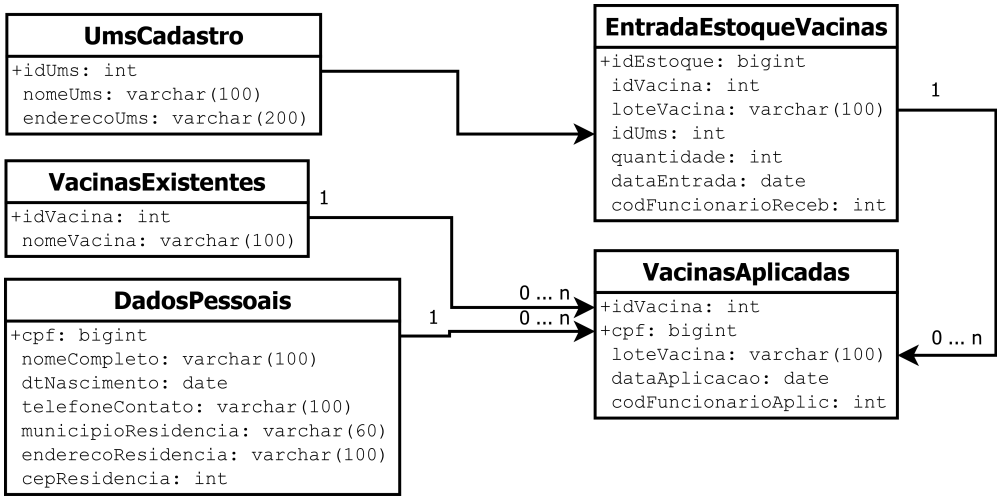
Fonte: O autor.

4 ARQUITETURA PROPOSTA

4.1 Caso de uso

Para ilustrar a arquitetura proposta será utilizado o caso de uso das Unidades Municipais de Saúde (UMS), que possuem uma aplicação de Carteira Digital de Vacinação (CDV). Trata-se de uma aplicação web que faz requisições de escrita e leitura dos dados pessoais dos cidadãos, das vacinas aplicadas e das vacinas existentes. As requisições da CDV são enviadas para uma camada de névoa que as executa utilizando uma rede *blockchain* como base de dados distribuída. O diagrama simplificado de relacionamento dos dados pode ser visualizado na figura 8, na qual, ao lado esquerdo superior, está a *UmsCadastro*, que possui como identificador o campo *idUms* e mantém o cadastro das UMSs. Ao lado direito superior está a *EntradaEstoque-Vacinas*, que possui os dados de entradas do estoque, sendo o identificador único o *idEstoque*. No centro ao lado esquerdo estão as *VacinasExistentes*, que possuem como identificador único o campo *idVacina*. Logo abaixo estão os *DadosPessoais* do cidadão, cujo identificador único é o *cpf*; já as *VacinasAplicadas* têm como identificador composto os campos *cpf* e *idVacina*.

Figura 8 – Diagrama simplificado de relacionamento de dados da CDV



Fonte: O autor.

Esta aplicação deve permanecer funcional em uma determinada UMS, mesmo durante um período de desconexão do segmento de rede onde a mesma está localizada. Considera-se a desconexão (ou isolamento) como um período no qual a rede da UMS permanece sem Internet e não possui qualquer forma de comunicação com nós externos à sua rede local. Neste cenário, manter a CDV funcionando evita que os funcionários da unidade sejam obrigados a realizar anotações manuais de vacinas aplicadas e posteriormente transportar as mesmas para o sistema. Com isso é possível mitigar erros de transcrição, aplicação de doses de vacinas em duplicidade e até mesmo fraudes no estoque daquela UMS. Para uma situação como esta existem alguns requisitos de funcionamento da CDV:

- Acesso aos dados de cadastro dos cidadãos, vacinas disponíveis na unidade e vacinas já aplicadas;
- Manter as funcionalidades de cadastramento e consulta de vacinas aplicadas e dos dados pessoais dos cidadãos;
- Armazenar os novos registros realizados durante o período de desconexão, permitindo que os mesmos sejam incluídos na *blockchain* posteriormente e com possibilidade de rastreamento/auditoria destes.

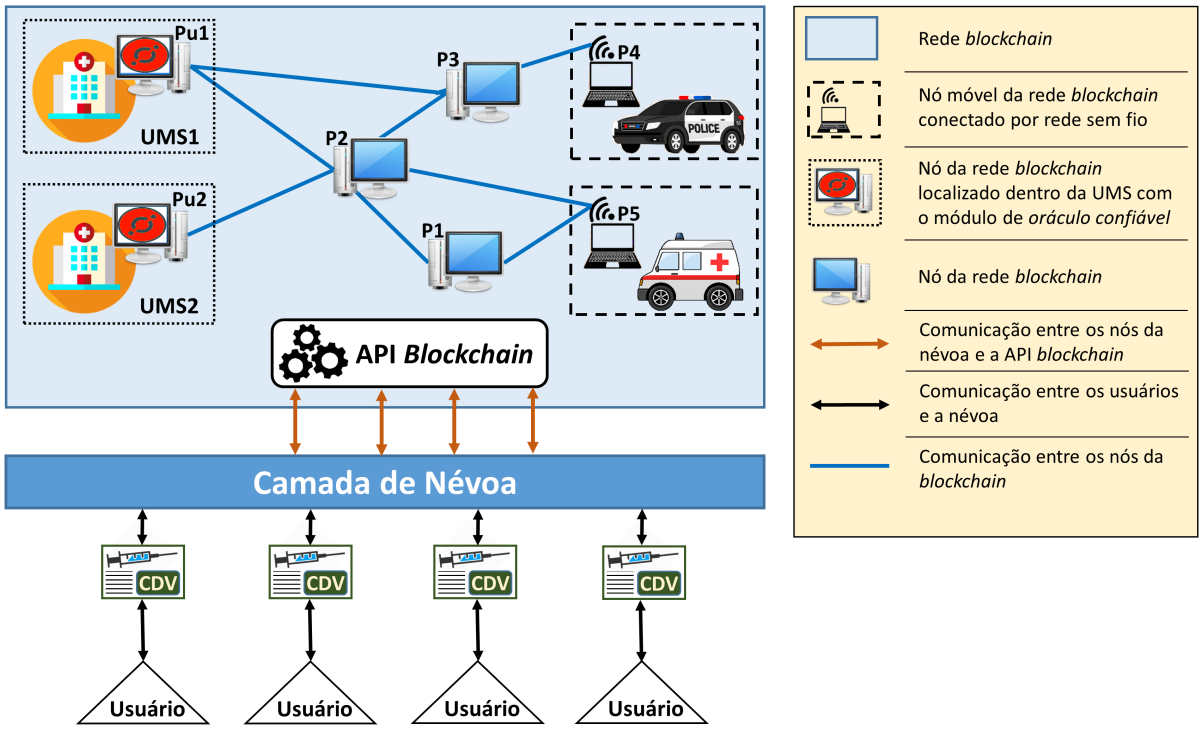
4.2 Arquitetura proposta

Para atender aos requisitos mencionados na seção 4.1 esta pesquisa utiliza os conceitos de *blockchain*, computação em névoa e a lógica de um *oráculo centralizado*. A figura 9 exibe os principais componentes da arquitetura proposta. Na parte superior, o retângulo azul demarca a estrutura da rede *blockchain* dentro da qual é possível visualizar os *peers* da rede. Nesta figura, *P1, P2 e P3* representam *peers* que estão geograficamente localizados em qualquer lugar do território nacional e são ligados por rede cabeada. Já *P4* e *P5* representam *peers* conectados à *blockchain* por rede sem fio. Estes dois *peers* podem ser desconectados intencionalmente a qualquer momento da *blockchain*, levados para lugares remotos onde não existe estrutura de rede e prover acesso local para dispositivos de borda. Após um determinado período, quando possuírem novamente conexão com a Internet, *P4* e *P5* podem se reconectar novamente à rede *blockchain*.

O foco do estudo está nos *peers* fisicamente localizados dentro das UMSs, os quais foram chamados de *Peer Ums (Pu)*. Na figura 9 estes são representados por *Pu1* e *Pu2*. Na situação apresentada nesta figura, *Pu1* está fisicamente localizado dentro da UMS1 e *Pu2* dentro da UMS2. Ambos possuem o módulo de oráculo e um Banco de Dados Local (BDL) instalados, bem como uma cópia do livro-razão da *blockchain* com os dados pessoais dos cidadãos e os dados das vacinas. No nível intermediário está a camada de névoa, responsável pelo armazenamento temporário e tratamento das requisições de leitura e escrita de dados. Logo abaixo desta camada está a representação das instâncias do aplicativo web CDV que estão sendo executadas pelos usuários, os quais estão representados no nível inferior. Os usuários utilizam dispositivos de borda para solicitar leitura e escrita de dados através da CDV.

Com esta arquitetura, durante períodos de desconexão da Internet em uma UMS, é possível que os usuários com dispositivos autenticados na LAN da UMS enviem requisições de leitura de dados para um nó *Pu* através do endereço local (IP e porta) do mesmo. Desta forma os dados pré-existentes na *blockchain* permanecem disponíveis na rede local da UMS. Durante a desconexão também é possível gerar novos dados que, por estarem fora da *blockchain*, são chamados de dados *off-chain* (fora da cadeia). Para garantir que os novos dados não sejam adulterados será utilizado um módulo de oráculo, o qual realiza a criptografia dos dados com

Figura 9 – Arquitetura proposta para incremento da resiliência dos sistemas de informação



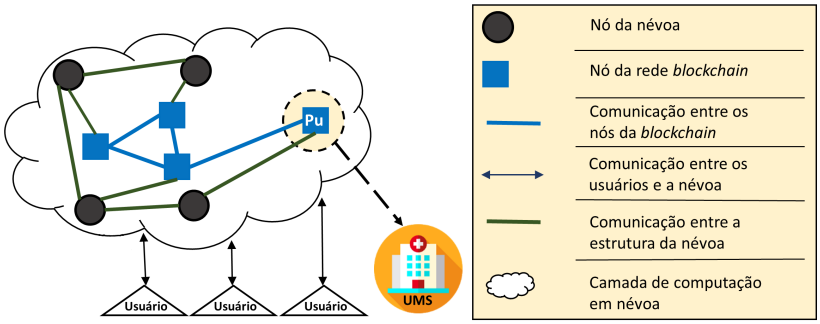
Fonte: O autor.

sua chave privada e o armazenamento destes em um BDL. Posteriormente, o próprio oráculo verifica a sua assinatura nestes dados (utilizando sua chave pública) e envia os mesmos para gravação na *blockchain*. A escolha das chaves assimétricas se deve à possibilidade de utilização de uma assinatura digital, a qual simplifica a dinâmica de verificação dos dados salvos no BDL, já que a conferência da mesma garante que os dados foram criptografados pelo oráculo.

A escolha do oráculo utilizado considerou os recursos computacionais limitados disponíveis na rede local, desta forma o oráculo é de *confiança centralizada*, o qual consome menos recursos. A sua *fonte de dados* é de *software*, tendo em vista que o mesmo poderá consultar apenas dados do nó *Pu*. Com relação ao *fluxo de dados* o mesmo será *bidirecional*, pois em momentos de consulta ele busca dados na *blockchain* e disponibiliza estes para a rede *(saída)* e em outros momentos ele busca dados no BDL e envia para gravação na *blockchain (entrada)*. Finalmente com relação ao padrão de arquitetura, nos momentos em que possui conexão com a Internet, o mesmo pode ser classificado como *publish-subscribe*.

O funcionamento da arquitetura pode ser visualizado na figura 10, a qual mostra uma estrutura de computação em névoa operando juntamente com uma rede *blockchain*. Aqui considera-se que cada nó (da névoa e da *blockchain*) está localizado em um local geograficamente distinto e que o nó *Pu* está fisicamente localizado dentro da UMS. Neste primeiro momento, todos os usuários utilizam seus dispositivos de borda para se conectar à Internet e fazer requisições de dados para os nós da camada de névoa. Os nós da névoa comunicam-se com a *blockchain* através de uma API e retornam os dados recebidos para os usuários.

Figura 10 – Camada de névoa funcionando com a *blockchain*



Fonte: O autor.

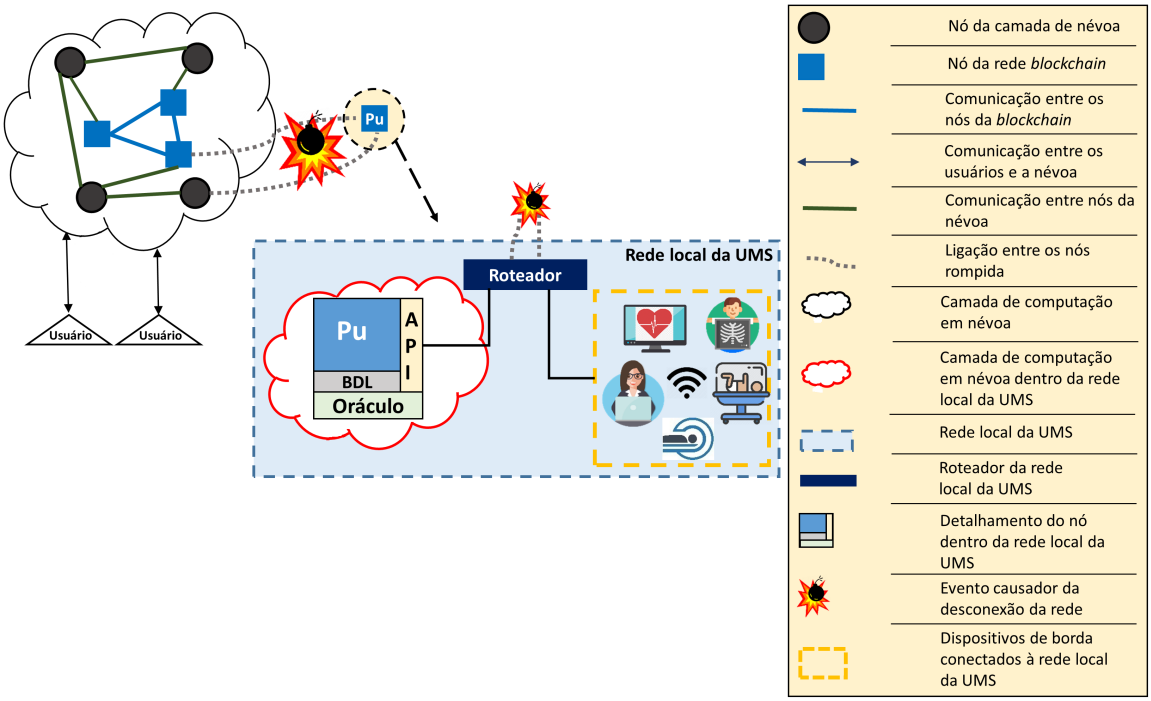
4.3 Mecanismo de Resiliência

Na figura 11 é exibido o momento em que um evento qualquer causa a desconexão do segmento de rede onde está localizada a UMS. Agora é possível observar que alguns usuários permanecem conectados na parte da camada de névoa que continua funcionando normalmente (lado esquerdo da figura). Porém, o segmento de rede onde se encontra o nó *Pu* está desconectado do restante da rede. O retângulo pontilhado na cor azul da figura 11 representa o detalhamento da rede local (LAN) da UMS na qual o nó *Pu* está inserido. Esta LAN é formada por um roteador ao qual estão conectados alguns dispositivos de borda (usados pelos usuários para fazer requisições de leitura/escrita) e o nó *Pu* da *blockchain*. Em cenários semelhantes a este é possível utilizar a arquitetura proposta onde o nó *Pu* passa a se comportar como névoa, por este motivo o mesmo foi representado dentro da nuvem vermelha.

O objetivo principal é manter os dados disponíveis para os usuários da CDV mesmo após a desconexão. Para isso, sempre que a desconexão for constatada a dinâmica de funcionamento da CDV passa a ser conforme exemplificado na figura 12. Nela, a usuária Alice deseja consultar os dados pessoais e de vacinas aplicadas a um cidadão. Para isso, ela faz a solicitação de consulta 1, que é enviada para o endereço local da API do nó *Pu* conforme mostrado em 2. Alice recebe a resposta da API e realiza o cadastramento de uma nova vacina aplicada ao cidadão 3. A solicitação de cadastramento é enviada para o endereço local do oráculo 4, o qual utiliza sua chave privada para criptografar os dados e solicita a gravação dos mesmos no BDL de *Pu* 5.

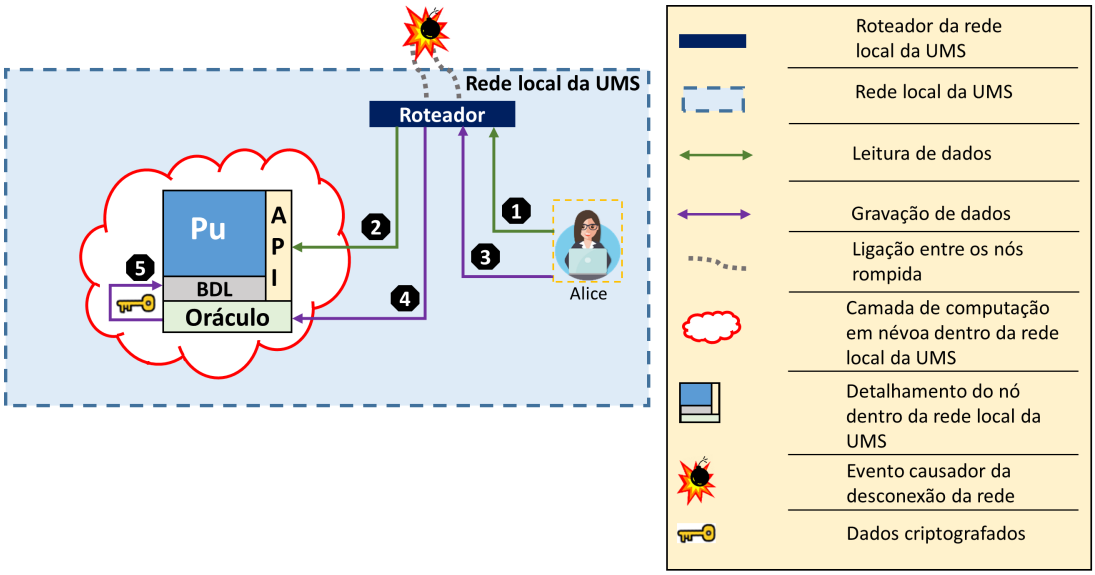
Passado algum período de tempo verifica-se que a conexão com a internet da LAN da UMS foi restabelecida, conforme está ilustrado na figura 13. Na indicação de 6 podemos observar que *Pu* agora consegue se comunicar com a *blockchain* e está novamente disponível para receber requisições de dados dos nós da névoa. Nesta situação o *oráculo* inicia a conciliação das informações, ou seja: o envio dos novos dados guardados e criptografados no BDL para a *blockchain*. Para isso o oráculo realiza a leitura dos dados do BDL 7, verifica a sua assinatura digital nos mesmos usando sua *chave pública* e envia uma solicitação de gravação destes dados para a API do nó *Pu* 8. Após isso a solicitação é enviada para a *blockchain* 9

Figura 11 – Evento causa a desconexão da rede onde está localizada a UMS



Fonte: O autor.

Figura 12 – Usuária realiza requisições durante o período de desconexão

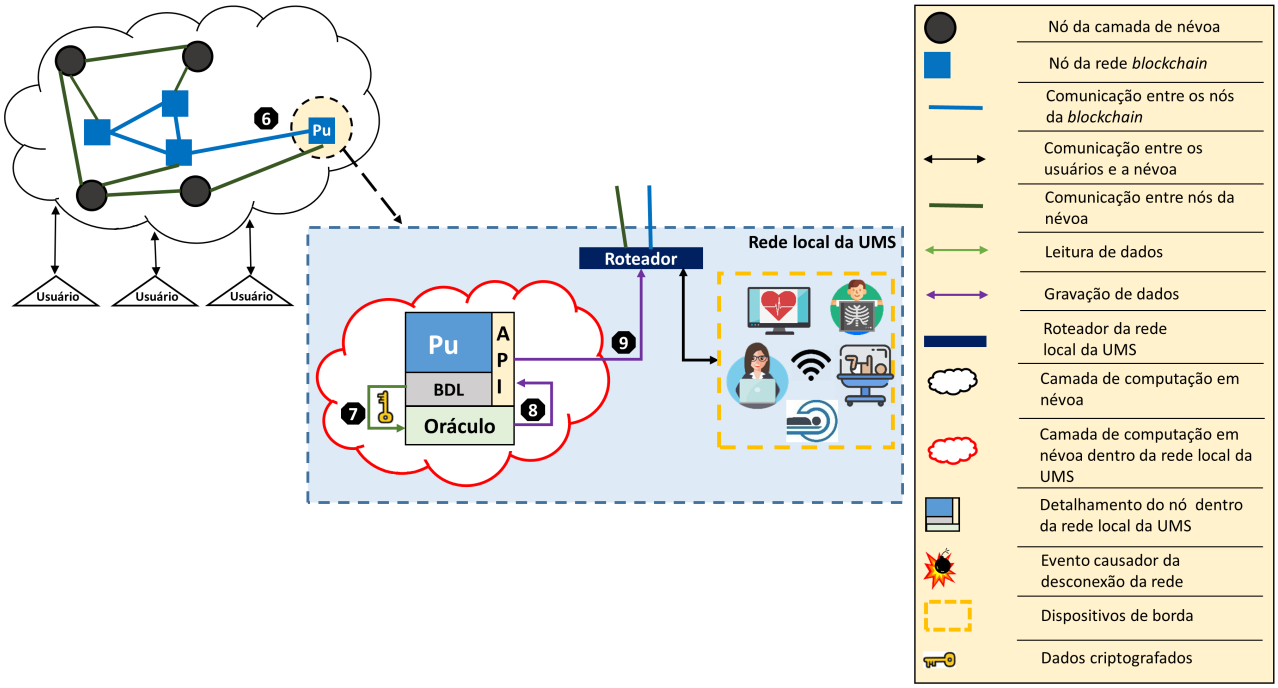


Fonte: O autor.

e a resposta recebida pela API é repassada ao oráculo que solicita a gravação desta no BDL. As respostas recebidas pelo oráculo podem ser a *chave de confirmação de registro* indicando sucesso na gravação ou uma mensagem de erro que deverá ser tratada. No caso de mensagens de erro as mesmas podem ser principalmente de indisponibilidade/erro da *blockchain* ou ainda de duplicidade de registros.

Como o foco deste estudo está na disponibilidade dos dados, não houve implementação de protocolo que verifique a desconexão da névoa. Para esta funcionalidade foi convencionado que ao realizar duas tentativas seguidas sem resposta de comunicação com a névoa o sistema considera que existe uma desconexão temporária. Para constatar a reconexão também foi convencionado que o sistema envia uma requisição de consulta de dados a cada intervalo de tempo e, caso receba resposta da névoa, considera que houve a reconexão. No que diz respeito a *reconexão segura* a mesma se refere sempre ao nó da *blockchain* que ficou isolado. Os nós do *Hyperledger Fabric* já possuem protocolo de reconexão segura e com atualização dos dados da rede sempre que ocorre alguma desconexão.

Figura 13 – Reconexão da rede e conciliação dos dados fora da cadeia



Fonte: O autor.

4.4 Pontos de Proteção e de Falha da Arquitetura Proposta

Com a dinâmica de funcionamento da arquitetura proposta, durante os períodos de desconexão os seus componentes são capazes de disponibilizar os dados da *blockchain* para consultas e ainda manter a funcionalidade de escrita de dados fora da cadeia. Desta forma, mesmo durante a desconexão, o sistema de informação continua funcionando sem interrupções significativas para os usuários da UMS. Em decorrência disso, a necessidade de realizar anotações manuais dos dados pessoais ou de vacinas aplicadas aos cidadãos é evitada, já que os funcionários da UMS continuam utilizando o sistema. Evitar este tipo de anotação também implica na mitigação de erros de transcrição dos dados para o sistema. Desta forma é possível manter o

controle da aplicação de doses das vacinas evitando duplicidades, bem como manter o controle do estoque de vacinas da UMS mitigando possíveis fraudes.

Além disso, a arquitetura utiliza dois tipos de dados para facilitar o rastreamento e/ou auditoria dos registros: dados de *originação* e dados de *resposta da blockchain*. A *originação* são os dados brutos e as informações de data e hora dos novos dados gerados fora da cadeia com a utilização do oráculo. A *resposta da blockchain*, como mencionado na seção 4.3, indica as respostas que o oráculo pode receber da *blockchain* ao enviar dados para gravação. Uma destas respostas é a *duplicidade de registros*, que pode indicar tentativas de um cidadão em receber mais de uma dose da mesma vacina. Seja a resposta recebida pelo oráculo de sucesso ou erro, é possível confrontar a mesma com o livro-razão para rastrear e/ou auditar os registros, viabilizando a verificação de adulteração das informações.

Com base nos conceitos sobre oráculo e considerando que o foco da arquitetura proposta está na disponibilidade dos dados, é possível identificar algumas oportunidades de ataque à arquitetura. Uma destas está relacionada às chaves utilizadas pelo oráculo. Caso estas sejam roubadas, podem ser utilizadas por atacantes para incluir, excluir e alterar dados no BDL, os quais posteriormente seriam conciliados com os dados da *blockchain*. Para dificultar a ação de atacantes nestas situações, é possível criar rotinas que realizam a verificação dos intervalos de tempo nos quais ocorreu a desconexão da Internet na UMS e fazer com que a etapa de *conciliação* dos dados do oráculo apenas busque os dados no BDL que tenham sido gerados dentro destes intervalos de desconexão. Os dados gerados fora deste intervalo seriam indicados para auditoria.

Um ponto de falha para a qual o sistema não está preparado é a situação em que o cidadão recebe a *vacina X* na UMS-1 que está passando por desconexão. Após isso, ele se dirige à UMS-2, que está com sua rede em funcionamento normal, e recebe a mesma *vacina X*. Nesta situação, como a rede da UMS-1 ainda está isolada, a *blockchain* ainda não recebeu as atualizações do oráculo. Aqui, portanto, não haverá sinalização de que o cidadão já tomou a referida vacina.

Apesar das falhas mencionadas, os demais aspectos apresentados nesta seção criam um mecanismo de resiliência que ajuda a manter o funcionamento de um sistema de informação em momentos de desconexão. O resultado disso é a continuidade da prestação do serviço para a comunidade, evitando prejuízos e transtornos para a população.

5 EXPERIMENTOS REALIZADOS

Para entender se a arquitetura proposta é viável, inicialmente foram realizados alguns experimentos com o emprego de uma arquitetura tradicional de *blockchain* como fonte para escrita e leitura. Os *peers* utilizados foram alocados em forma de *máquinas virtuais* em provedores de serviço. O padrão adotado foram máquinas com processadores de 1 ou 2 núcleos, memória RAM de 1GB ou 2GB e HD SSD de 15GB até 60GB - sempre de acordo com a disponibilidade existente. Para gravação de dados nas *máquinas virtuais* foi utilizado o banco de dados *MySQL Server* e para desenvolvimento das ferramentas de testes foi utilizada a linguagem *Node.js*.

Para realizar a preparação das máquinas (instalação dos *softwares* e pacotes necessários para rodar a *Hyperledger Fabric*) e também realizar a implantação da rede *blockchain*, foi utilizado o *GoFabric*[1], um *orquestrador* de redes *blockchain* permissionadas baseadas na tecnologia *Hyperledger Fabric*. Esta ferramenta possui opções de criação e alteração das redes (nós, organizações, *chaincodes* e API) bem como backup dos dados incluídos. O *chaincode* utilizado foi o *GoShare*, que é um padrão já existente para o *GoFabric*. Além disso foi usado o *Grafana Monitor*, com o módulo *Hyperledger Fabric Monitor 1.4*, para observar o comportamento do hardware e da rede durante os momentos de ociosidade e de utilização.

5.1 Rede *Blockchain* Implantada para os Testes

Para a execução dos testes realizados foi implantada uma *blockchain* com duas organizações: a *Org1* representando o SUS (Sistema Único de Saúde) e a *Org2* representando outra instituição qualquer. Com relação às configurações adotadas para a *blockchain*, a regra de endosso definida permite que apenas a organização SUS realize a inclusão de dados. Esta mesma organização recebeu acesso de leitura e escrita em todos os dados, possibilitando que os seus nós realizem operações através da API. Outra configuração realizada foi habilitar cada nó localizado dentro de uma UMS como API. Desta forma os mesmos podem receber e responder requisições de leitura dos dados de uma forma padronizada. Neste trabalho assume-se que para utilizar os sistemas de uma UMS, os usuários devem realizar a autenticação em um dispositivo que esteja concectado à LAN da UMS. Assume-se também que o oráculo é confiável e sua chave privada está guardada de forma segura.

A razão de se utilizar uma *blockchain* permissionada com duas organizações é a adoção de um cenário realista, no qual o SUS possui acesso para leitura e escrita de qualquer dado na *blockchain*. Já a *Org2* representa uma organização que apenas pode ter acesso a um conjunto limitado dos dados existentes nesta mesma *blockchain*. Uma situação como esta pode ocorrer quando, por exemplo, uma empresa ganha uma licitação para realizar um estudo para o governo. Este estudo está relacionado com alguns perfis dos cidadãos que receberam doses de alguma vacina e/ou medicamento. Desta forma a empresa apenas pode ter acesso a um

[1] http://gofabric.io/ da empresa GoLedger

conjunto específico de dados necessários para realização do referido estudo. Para a criação deste tipo de controle de acesso em uma *blockchain Fabric* é possível utilizar uma *Coleção de Dados Privados (CDP)*. Ademais, cada uma das organizações possui a seguinte configuração:

- ✓ um nó que cumpre o papel de *ordenador* (fazendo parte do *Serviço de Pedidos*), *peer* da rede e de *API*;
- ✓ um nó que cumpre o papel de *Autoridade Certificadora* e também de *peer* da rede.

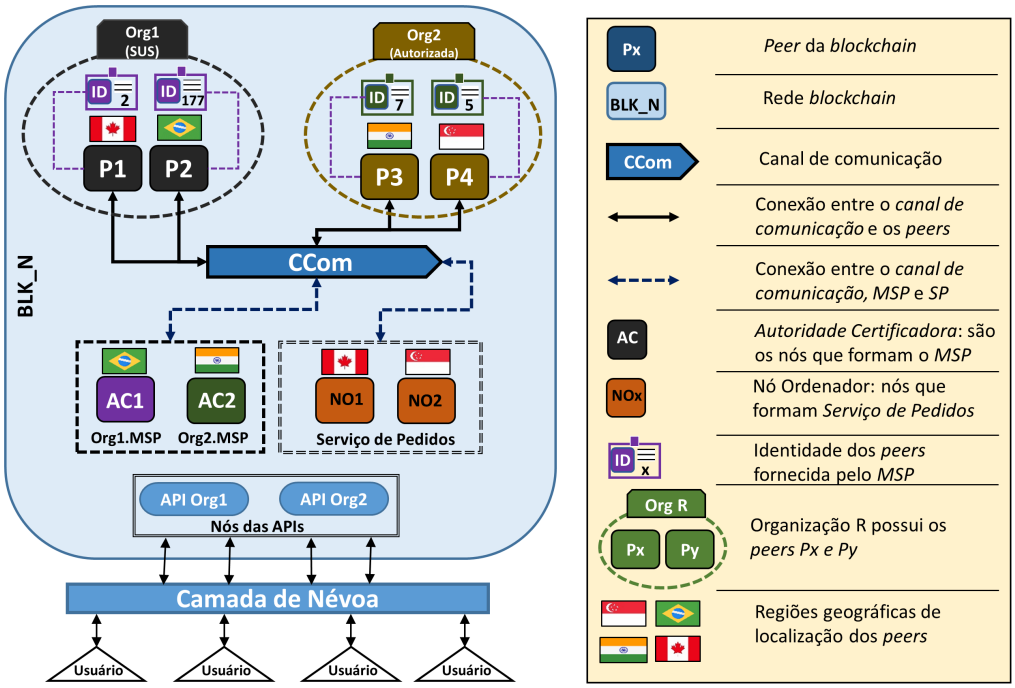
A figura 14 exibe a rede implantada. Dentro de cada organização foram alocados nós em diferentes locais geográficos, representados pelas bandeiras dos países. A escolha destes locais geograficamente distintos e distantes, simula uma situação real onde existiriam nós da *blockchain* espalhados por diversos estados do Brasil e também mitiga o risco de paralização da rede devido a indisponibilidade conjunta dos nós em uma determinada região geográfica. Nesta mesma figura 14 também são exibidos separadamente alguns serviços nativos da *Hyperledger Fabric*, como a AC, o MSP e o SP. Também são exibidos os nós responsáveis pela API *(API Org1 e API Org2)*, através dos quais é possível acessar todas as funcionalidades descritas no Anexo A, para atuar sobre os dados da *blockchain* (Hyperledger Fabric, 2020).

Todos os testes realizados para verificação de disponibilidade das *Orgs*, *stress* de acesso, leitura/escrita de dados foram realizados através da API. Durante a simulação da desconexão as requisições para a API são realizadas através da rota */query/search* (Anexo A), a qual realiza buscas sem necessidade de registro na *blockchain*. No Apêndice A estão listadas as portas configuradas para os nós da *blockchain*. Por fim, também utilizando a *GoFabric* foram criados os ativos da *blockchain* exibidos anteriormente na figura 8 e abaixo descritos:

- *DadosPessoais*: dados cadastrais básicos do cidadão;
- *VacinasAplicadas*: relação das vacinas aplicadas aos cidadãos;
- *VacinasExistentes*: vacinas cadastradas no sistema;
- *EntradasEstoqueVacinas*: dados para controle do estoque de doses das vacinas;
- *UmsCadastro*: cadastro das Unidades Municipais de Saúde;

Uma camada de névoa foi implementada com dois nós e entrega as funcionalidades da nuvem para os usuários. Esta camada é responsável por receber e armazenar as requisições, enviar os dados destas para as APIs da *blockchain*, aguardar e devolver a resposta para os usuários. Esta camada também tem a função de verificar a disponibilidade da *blockchain* comunicando-se com os nós das APIs. Por fim, na parte inferior da figura 14 estão os usuários. Estes acessam a camada de névoa através de vários tipos de dispositivos, realizam requisições de leitura/escrita de dados na *blockchain* e aguardam o resultado de suas solicitações.

Figura 14 – *Blockchain* implantada para realização dos testes



Fonte: o autor

5.2 Resultados Obtidos

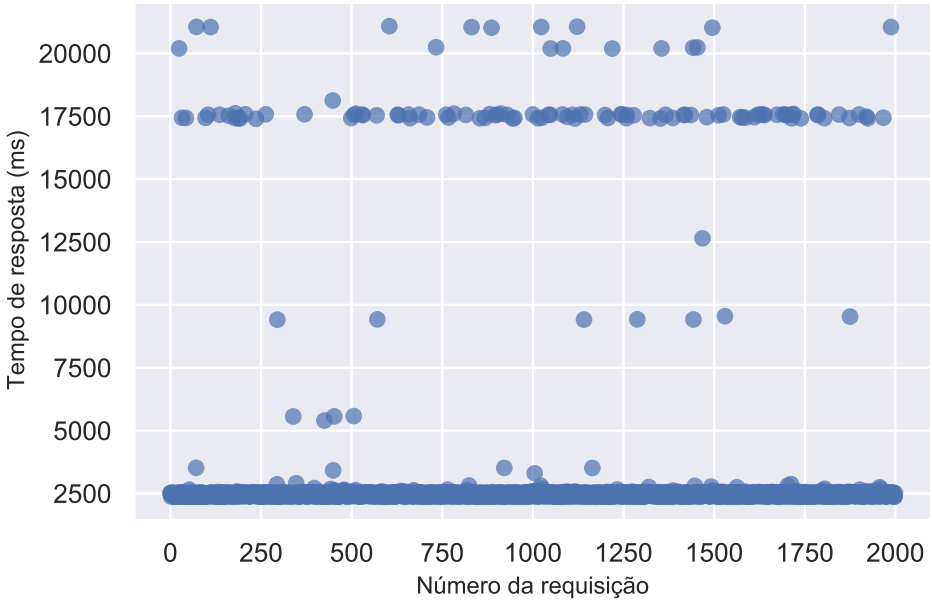
Os testes de escrita consistiram principalmente em realizar o cadastramento de lotes de 2.000 novos registros (escrita) de dados dos cidadãos através das APIs da organização SUS. Em um primeiro cenário todos os nós da rede foram mantidos *online*, ou seja, a *blockchain* funcionando normalmente. A dispersão do tempo de resposta em meio às 2.000 requisições de escrita pode ser observada na figura 15. Nela constata-se a existência de uma linha base abaixo dos 5.000ms formada por 1.878 requisições, ou seja, a maior parte das requisições teve duração inferior a este tempo. Entre os tempos de 5.001ms e 10.000ms existem 11 requisições e ainda 111 requisições com tempo de resposta superior a 10.001ms.

Os tempos observados são aceitáveis do ponto de vista da experiência do usuário (MILLER, 1968), já considerando a configuração das máquinas virtuais utilizadas, a distância geográfica entre os *peers*, o tempo de processamento da névoa e do banco de dados. O gráfico da figura 16 foi gerado através do *Grafana* [2], colhendo informações diretamente da *blockchain*. Nela é possível observar que os maiores tempos de processamento do banco de dados estão próximos de 1 segundo, ou seja, em torno 30% do tempo da maior parte das respostas.

Ao realizar os testes simulando a rede local desconectada da internet, os *peers* P2, P3 e P4 foram desligados, permanecendo apenas P1. Ao se realizar requisições locais de leitura, algumas das vezes o nó P1 retornou a mensagem *"Error: Failed to connect before the deadline URL:grpcs://[ip do peer]:7051"* . Na ocorrência deste erro é necessário repetir a consulta. Este

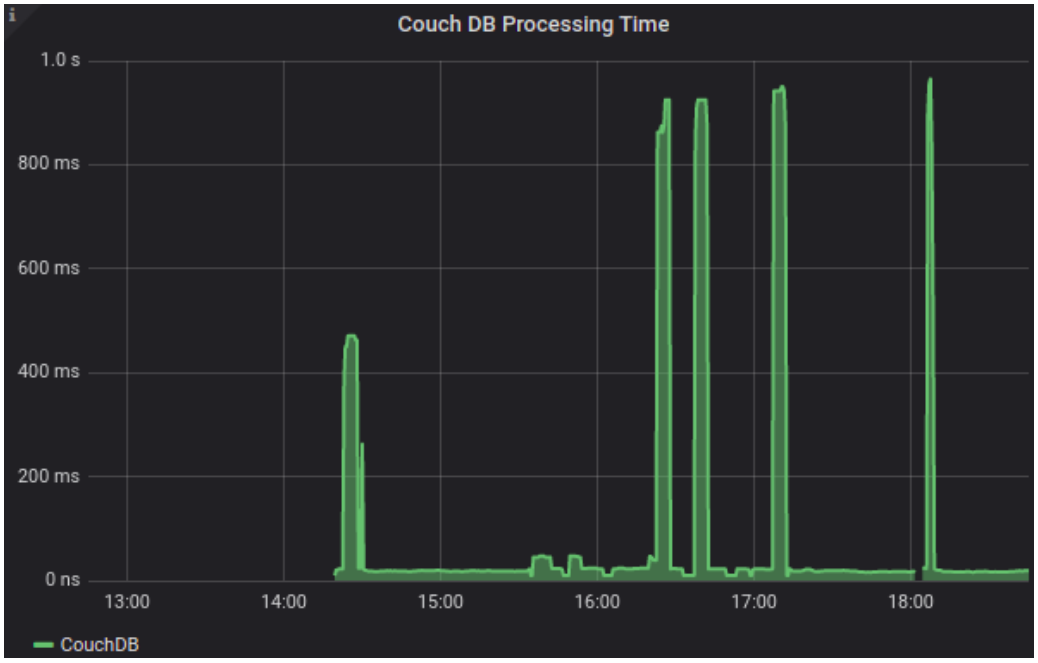
[2] https://grafana.com/

Figura 15 – Tempo de resposta das requisições de escrita na *blockchain*



Fonte: o autor

Figura 16 – Tempo de processamento do banco de dados utilizado no *Fabric*

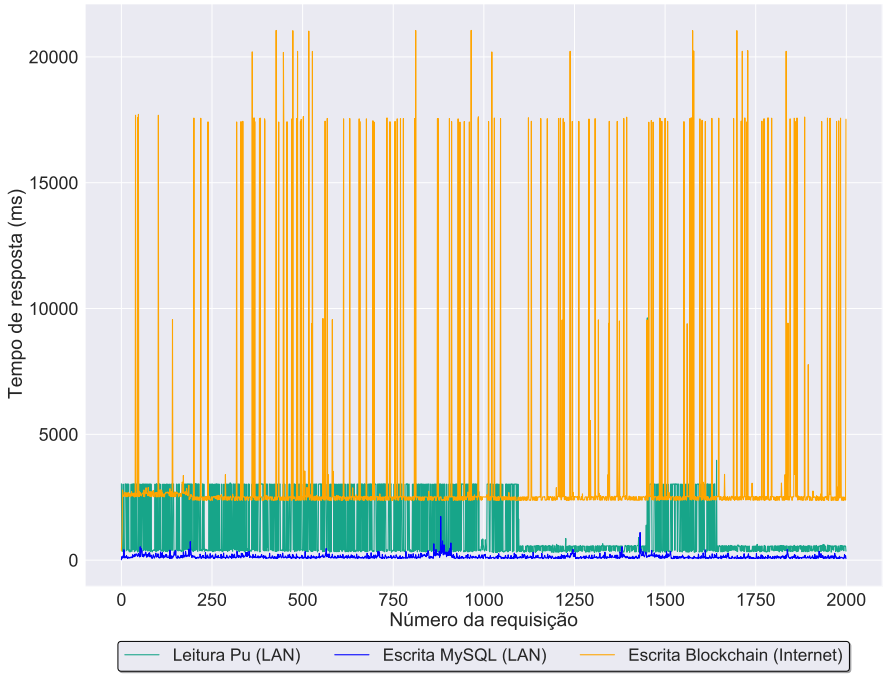


Fonte: o autor

erro ocorre pois em uma rede *Hyperledger Fabric* existe uma lista dos *peers* que executam a API (Hyperledger Fabric, 2020). Desta forma, quando chega uma requisição de consulta ela é direcionada para um dos *peers* da lista. Caso o mesmo esteja fora do ar, a mensagem de erro é retornada. Ao se reenviar a requisição de consulta logo em seguida, outro *peer* da lista será

responsável por responder. Nos testes realizados, dois *peers* foram configurados para executar a API. Neste contexto, 70,4% das requisições retornaram com sucesso e 29,6% com erro.

Figura 17 – Tempo de processamento das requisições



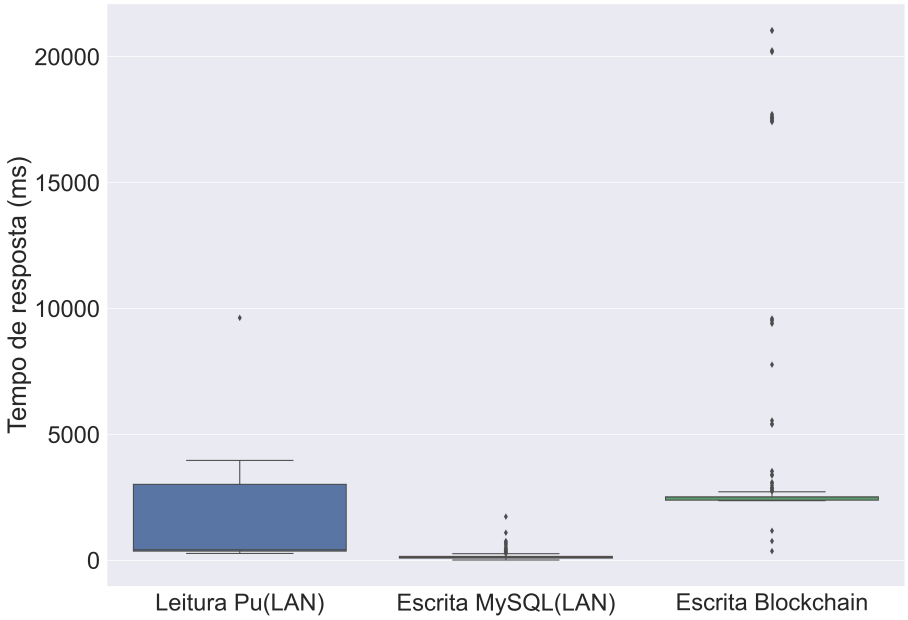
Fonte: o autor

O gráfico exibido na figura 17 segmenta os tempos de resposta para *leitura dos dados do nó Pu* e da *escrita dos dados no MySQL* (através do oráculo confiável), ambos na rede local simulando a desconexão. Além disso exibe os tempos das requisições de *escrita na blockchain* através da Internet, já considerando a rede funcionando normalmente. Considerando o desvio padrão destes dados constata-se que a escrita na *blockchain*, representada pela cor laranja, consome cerca de 74,6% do tempo de todo o processo realizado. Os 25,4% do tempo restante se refere ao processo do oráculo (leitura do nó *Pu* e escrita no banco de dados MySQL).

A figura 18 exibe uma *box plot* para cada um dos tempos apresentados anteriormente. Nesta figura é possível observar que, na rede local, as operações de *Leitura dos dados de Pu* são representadas por um retângulo mais longo o qual alcança valores mais altos do que as operações de *Escrita MySQL*. Isso significa que aquelas possuem tempos mais variados e maiores do que estas, principalmente devido à necessidade de se refazer as requisições de *leitura dos dados de Pu* que retornam erro. Já as operações de *Escrita de dados na Blockchain* realizadas pela Internet mostram uma variação menor, concentrando-se em tempos abaixo dos 5.000ms, com vários *outliers* maiores e apenas três menores.

Os resultados obtidos nas medições dos tempos mostram que 93,9% das requisições tiveram um tempo de resposta adequado, permanecendo dentro dos limites apropriados para manter a atenção dos usuários (NIELSEN, 1993). Desta forma, além da arquitetura proposta

Figura 18 – Variação dos tempos de processamento



Fonte: o autor

atingir o objetivo de manter um sistema de informação funcionando durante um período de desconexão, os resultados demonstram que a mesma é viável do ponto de vista da usabilidade relacionada ao tempo de resposta para o usuário.

6 CONCLUSÕES E PERSPECTIVAS

A necessidade dos sistemas de informação possuírem recursos que os tornem cada vez mais resilientes é perceptível em cenários relacionados às cidades inteligentes. Nestes cenários existe uma grande demanda por recursos computacionais de borda e, para se manter os serviços prestados aos cidadão, as aplicações devem permanecer em funcionamento mesmo nas situações de precariedade ou falha da rede de dados.

Esta dissertação apresentou uma arquitetura que fornece resiliência para um sistema de informação em momentos de desconexão de sua rede. Para isso foi utilizada a *blockchain* como banco de dados distribuído, um oráculo confiável para manipulação de dados locais e os conceitos de computação em névoa para prover funcionalidades aos usuários. Apesar das limitações mencionadas na seção 4.4, a arquitetura atingiu seu objetivo principal. Em cenários de desconexão, ela mantém um sistema de informação funcionando através da disponibilização dos dados no nó da *blockchain* e possibilita a escrita de dados através do oráculo e do banco de dados local.

Os testes realizados demonstram que a arquitetura é viável. A introdução de uma camada de névoa facilitou o desenvolvimento da ferramenta de testes, já que a mesma é responsável pelo tratamento das requisições. Além disso, os estudos anteriores (seção 2.2) demonstram que esta camada reduz a latência de comunicação e reduz os problemas de comunicação entre os usuários e serviços. A utilização de um nó local da *blockchain*, do oráculo e um banco de dados se mostra de grande utilidade, já que os usuários de um sistema conseguem consultar e gravar dados em situações adversas.

Outra grande vantagem é a reconexão segura e automática do nó da *blockchain*. Este processo é padrão para os nós da *Hyperledger Fabric* e é realizado sempre que o nó se recupera de uma desconexão. Desta forma não é necessário realizar qualquer procedimento de atualização manual dos dados do nó. A reconexão segura é um grande facilitador, pois a responsabilidade de verificação e atualização dos dados recai apenas sobre a rede da *Fabric*.

Em estudos futuros é possível utilizar um segmento maior de rede, no qual duas ou mais entidades possam realizar as operações de leitura e escrita de dados em momentos de desconexão da rede. Além disso, um nó móvel da *blockchain* provido de conexão via rede móvel e infraestrutura *wireless* também pode ser adicionado à rede. Este nó poderia se deslocar até alguma das unidades de saúde que estivesse passando por desconexão e prover acesso para os seus dispositivos de borda. Algo que se torna muito interessante é a utilização do conceito de Redes Tolerantes a Atrasos e Desconexões em conjunto com o nó móvel para prover resiliência aos sistemas que corriqueiramente passam por períodos de desconexão em locais remotos. Por fim, outra alternativa de aprimoramento seria substituir o banco de dados local, que armazena os registros durante a desconexão, por uma *blockchain* local. Como esta permite apenas inclusões de dados, haveria um incremento na rastreabilidade dos mesmos.

REFERÊNCIAS

ADLER, J. *et al.* Astraea: A Decentralized Blockchain Oracle. *In*: **Proceedings - IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/Gree**. [*S.l.*: *s.n.*], 2018. p. 1145–1152. ISBN 9781538679753.

AL-BREIKI, H. *et al.* Decentralized access control for IoT data using blockchain and trusted oracles. **Proceedings - IEEE International Conference on Industrial Internet Cloud, ICII 2019**, n. Icii, p. 248–257, 2019.

AL-BREIKI, H. *et al.* Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges. **IEEE Access**, v. 8, p. 85675–85685, 2020. ISSN 21693536.

AL-KHAFAJIY, M. *et al.* Iot-fog optimal workload via fog offloading. *In*: **Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion 2018**. [*S.l.*]: IEEE, 2019. p. 349–352. ISBN 9781728103594.

ALGHAMDI, A.; ALZAHRANI, A.; THAYANANTHAN, V. Fog Network Area Management Model for Managing Fog-cloud Resources in IoT Environment. **International Journal of Advanced Computer Science and Applications**, v. 12, n. 3, p. 482–489, 2021. ISSN 21565570.

ALLADI, T. *et al.* Blockchain Applications for Industry 4.0 and Industrial IoT: A Review. **IEEE Access**, IEEE, v. 7, p. 176935–176951, 2019. ISSN 21693536.

ANDROULAKI, E. *et al.* Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. **Proceedings of the 13th EuroSys Conference, EuroSys 2018**, v. 2018-Janua, 2018.

ANMULWAR, S.; GUPTA, A. K.; DERAWI, M. Challenges of IoT in Healthcare. *In*: GUPTA, N.; PAIVA, S. (Ed.). **IoT and ICT for Healthcare Applications**. 1st. ed. [*S.l.*]: EAI/Springer Innovations in Communication and Computing, 2020. p. 175–189. ISBN 9783030429331.

ANTHOPOULOS, L.; FITSILIS, P. From digital to ubiquitous cities: Defining a common architecture for urban development. **Proceedings - 2010 6th International Conference on Intelligent Environments, IE 2010**, p. 301–306, 2010.

AZARIA, A. *et al.* MedRec: Using blockchain for medical data access and permission management. **Proceedings - 2016 2nd International Conference on Open and Big Data, OBD 2016**, p. 25–30, 2016.

BABAR, M.; TARIQ, M. U.; JAN, M. A. Secure and resilient demand side management engine using machine learning for IoT-enabled smart grid. **Sustainable Cities and Society**, Elsevier, v. 62, n. June, p. 102370, 2020. ISSN 22106707. Disponível em: https://doi.org/10.1016/j.scs.2020.102370.

BABU, R.; Hyperledger Fabric. **APIs - CLI, REST, and Node.js**. 2017. Data de acesso: 05/09/2021. Disponível em: https://hyperledger-fabric.readthedocs.io/en/v0.6/API/CoreAPI.html.

BACH, L. M.; MIHALJEVIĆ, B.; ZAGAR, M. Comparative Analysis of Blockchain Consensus Algorithms. *In*: **MIPRO**. Opatija, Croatia: [*s.n.*], 2018. p. 1545–1550.

BARON, M. DO WE NEED SMART CITIES FOR RESILIENCE. **Journal of Economics And Management**, v. 10, p. 32–46, 2012.

BATTY, M. *et al.* Smart cities of the future. **European Physical Journal: Special Topics**, v. 214, n. 1, p. 481–518, 2012. ISSN 19516355.

BEN-ARI, M.; MONDADA, F. Elements of Robotics. **Finite State Machines; In: Elements of Robotics**, p. 1–308, 2017. Data de acesso: 03/02/2022. Disponível em: http://mtc-m16d.sid.inpe.br/col/sid.inpe.br/mtc-m19@80/2010/02.06.20.39/doc/publicacao.pdfhttps://doi.org/10.1007/978-3-319-62533-1_4.

BENIICHE, A. A Study of Blockchain Oracles. p. 1–9, 2020. Data de acesso: 22/03/2022. Disponível em: http://arxiv.org/abs/2004.07140.

BENTOV, I.; GABIZON, A.; MIZRAHI, A. Cryptocurrencies without proof of work. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 9604 LNCS, n. 240258, p. 142–157, 2016. ISSN 16113349.

BEREGSZASZI, A. **TinyOracle**. 2016. Disponível em: https://github.com/axic/tinyoracle.

BOLIN, R.; STANFORD, L. The Northridge earthquake: Community-based approaches to unmet recovery needs. **Disasters**, v. 22, n. 1, p. 21–38, 1998. ISSN 03613666.

BONOMI, F. *et al.* Fog computing and its role in the internet of things. **MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop**, p. 13–15, 2012.

BRANDENBURGER, M. *et al.* Blockchain and trusted computing: Problems, pitfalls, and a solution for hyperledger fabric. **arXiv**, 2018. ISSN 23318422.

BRANDENBURGER, M. *et al.* Rollback and Forking Detection for Trusted Execution Environments Using Lightweight Collective Memory. **Proceedings - 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2017**, p. 157–168, 2017. Disponível em: https://cachin.com/cc/papers/lcm.pdf.

BREIDENBACH, L. *et al.* Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks. p. 1–136, 2021. Disponível em: https://research.chain.link/whitepaper-v2.pdf.

BROTSIS, S. *et al.* On the Security and Privacy of Hyperledger Fabric: Challenges and Open Issues. **Proceedings - 2020 IEEE World Congress on Services, SERVICES 2020**, p. 197–204, 2020.

CACHIN, C.; VUKOLIĆ, M. Blockchain consensus protocols in thewild. **arXiv**, 2017. ISSN 23318422.

CALDARELLI, G. Understanding the blockchain oracle problem: A call for action. **Information (Switzerland)**, v. 11, n. 11, p. 1–19, 2020. ISSN 20782489.

California Institute for Smart Communities. **Smart Communities Guidebook**. 1st. ed. San Diego, California: [*s.n.*], 1997.

CASINO, F.; DASAKLIS, T. K.; PATSAKIS, C. A systematic literature review of blockchain-based applications: Current status, classification and open issues. **Telematics and Informatics**, Elsevier, v. 36, n. May 2018, p. 55–81, 2019. ISSN 07365853. Disponível em: https://doi.org/10.1016/j.tele.2018.11.006.

CASTELLANO, G.; RISSO, F.; LOTI, R. Fog Computing over Challenged Networks: A Real Case Evaluation. **Proceedings of the 2018 IEEE 7th International Conference on Cloud Networking, CloudNet 2018**, IEEE, p. 1–7, 2018.

CHANDRA, S. *et al.* A study and analysis on symmetric cryptography. **2014 International Conference on Science Engineering and Management Research, ICSEMR 2014**, 2014.

CHEN, S.; ZHANG, T.; SHI, W. Fog Computing. **IEEE Internet Computing**, v. 21, n. 2, p. 4–6, 2017. ISSN 10897801.

CHINNASAMY, P. *et al.* Blockchain technology in smart-cities. **Intelligent Systems Reference Library**, v. 203, n. May, p. 179–200, 2021. ISSN 18684408.

CHOURABI, H. *et al.* Understanding smart cities: An integrative framework. **Proceedings of the Annual Hawaii International Conference on System Sciences**, p. 2289–2297, 2012. ISSN 15301605.

CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and Smart Contracts for the Internet of Things. **IEEE Access**, IEEE, v. 4, p. 2292–2303, 2016. ISSN 21693536.

CISCO. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. **White Paper**, p. 1–6, 2015.

COCCHIA, A. Smart and Digital City: A Systematic Literature Review. *In*: DAMERI, R. P.; ROSENTHAL-SABROUX, C. (Ed.). **Smart City - How to Create Public and Economic Value with High Technology in Urban Space**. Springer International Publishing Switzerland, 2014. cap. 2, p. 13–44. ISBN 978-3-319-06160-3. Disponível em: http://www.springer.com/series/10440.

COUCLELIS, H. The construction of the digital city. **Environment and Planning B: Planning and Design**, v. 31, n. 1, p. 5–19, 2004. ISSN 02658135.

COULOURIS, G. *et al.* **Sistemas Distribuídos - Conceitos e Projetos**. Porto Alegre/RS: Bookman, 2013. ISBN 978-0-13-214301-1.

DAVIS, K. **The urbanization of the human population**. 1965. 309–344 p.

DU, M. *et al.* A review on consensus algorithm of blockchain. **2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017**, v. 2017-Jan, p. 2567–2572, 2017.

DWORK, C.; NAOR, M. Pricing via processing or combatting junk mail. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 740 LNCS, p. 139–147, 1993. ISSN 16113349. Disponível em: https://link.springer.com/content/pdf/10.1007/3-540-48071-4_10.pdf.

EGBERTS, A. **The Oracle Problem - An Analysis of how Blockchain Oracles Undermine the Advantages of Decentralized Ledger Systems**. 2017. Tese (Doutorado) — EBS Business School, Rheingaustraße - Alemanha, 2017.

ERGAZAKIS, K.; METAXIOTIS, K.; PSARRAS, J. Towards knowledge cities: Conceptual analysis and success stories. **Journal of Knowledge Management**, v. 8, n. 5, p. 5–15, 2004. ISSN 17587484.

Ethereum Foundation. **Ethereum**. 2022. Disponível em: https://ethereum.org/.

FERGUSON, D.; SAIRAMESH, J.; FELDMAN, S. Open Frameworks for Information Cities. **Communications of the ACM**, v. 47, n. 2, p. 45–49, 2004. Disponível em: 10.1145/966389.966414.

FREY, D. *et al.* Dietcoin: Shortcutting the Bitcoin verification process for your smartphone. **arXiv**, n. March, 2018. ISSN 23318422.

GIFFINGER, R. *et al.* Smart cities: ranking of European mid-sized cities. **Digital Agenda for Europe**, n. October, p. 28, 2007. ISSN 02642751. Disponível em: https://ec.europa.eu/digital-agenda/en/smart-citieswww.smart-cities.eu.

GODSCHALK, D. R. Urban Hazard Mitigation: Creating Resilient Cities. **Natural Hazards Review**, v. 4, n. 3, p. 136–143, 2003. ISSN 1527-6988.

GORENFLO, C. *et al.* FastFabric: Scaling hyperledger fabric to 20 000 transactions per second. **International Journal of Network Management**, v. 30, n. 5, p. 1–18, 2020. ISSN 10991190.

GORI, P.; PARCU, P. L.; STASI, M. L. Smart Cities and Sharing Economy. **EUI Working Papers**, v. 96, 2015. Disponível em: http://ssrn.com/abstract=2706603https://cadmus.eui.eu/bitstream/handle/1814/38264/RSCAS_2015_96.pdf.

HAJIBABA, M.; GORGIN, S. A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing. **Journal of Computing and Information Technology**, v. 22, n. 2, p. 69–84, 2014. ISSN 13301136.

HAMED, B. Design & Implementation of Smart House Control Using LabVIEW. **International Journal of Soft Computing and Engineering (IJSCE)**, v. 1, n. 6, p. 2231–2307, 2012.

HASSEN, H. B.; DGHAIS, W.; HAMDI, B. An E-health system for monitoring elderly health based on Internet of Things and Fog computing. **Health Inf Sci Syst.**, v. 7, n. 24, 2019. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6811806/.

HESS, Z.; MALAHOV, Y.; PETTERSSON, J. Æternity Blockchain Whitepaper. v. 01, p. 1–10, 2017. Disponível em: https://blockchain.aeternity.com/{\T1\ae}ternity-blockchain-whitepaper.

Hyperledger Fabric. **Hyperledger Fabric Docs**. 2020. Data de acesso: 06/03/2022. Disponível em: https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html.

Hyperledger Foundation. **Hyperledger**. 2016. Disponível em: https://www.hyperledger.org/.

IBM. **Blockchain: Behind the Architecture of Hyperledger Fabric**. 2018. Data de acesso: 20/04/2022. Disponível em: https://www.ibm.com/blogs/research/2018/02/architecture-hyperledger-fabric/.

IORGA, M. *et al.* Fog Computing Conceptual Model Recommendations of the National Institute of Standards and Technology. **NIST Special Publication 500-325**, v. 1, n. 2, p. 169–178, 2018. ISSN 22106774.

JENKINS, N.; LONG, C.; WU, J. An Overview of the Smart Grid in Great Britain. **ORCA Online Research**, 2015. Disponível em: https://orca.cardiff.ac.uk/83965/1/AnOverviewoftheSmartGridinGreatBritain.pdf.

JEONG, T. *et al.* Towards a Distributed Computing Framework for Fog. *In*: **IEEE Fog World Congress (FWC)**. [*S.l.*: *s.n.*], 2017. ISBN 9781538636664.

KENT, R. C. **Anatomy of Disaster Relief**. London: [*s.n.*], 1987.

KOLB, J. *et al.* Core concepts, challenges, and future directions in blockchain: A centralized tutorial. **ACM Computing Surveys**, v. 53, n. 1, p. 1–39, 2020. ISSN 15577341.

KOLINKO, T. **Orisi White Paper**. 2014. Disponível em: https://github.com/orisi/wiki/wiki/Orisi-White-Paper.

KOMNINOS, N. The architecture of intelligent cities: Integrating human, collective and artificial intelligence to enhance knowledge and innovation. **IET Conference Publications**, n. 518, p. 13–20, 2006.

KULATUNGA, C. *et al.* Opportunistic Wireless Networking for Smart Dairy Farming. **IT Professional**, v. 19, n. 2, p. 16–23, 2017. ISSN 15209202.

LAMPORT, L.; SHOSTAK, R.; PEASE, M. The Byzantine Generals Problem. **ACM Transactions on Programming Languages and Systems (TOPLAS)**, v. 4, n. 3, p. 382–401, 1982. ISSN 15584593.

LE, D. P. *et al.* BIFF: A Blockchain-based IoT Forensics Framework with Identity Privacy. **IEEE Region 10 Annual International Conference, Proceedings/TENCON**, v. 2018-Octob, n. May 2019, p. 2372–2377, 2019. ISSN 21593450.

LO, S. K. *et al.* Reliability analysis for blockchain oracles. **Computers and Electrical Engineering**, Elsevier Ltd, v. 83, p. 1–10, 2020. ISSN 00457906.

LONE, A. H.; MIR, R. N. Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. **Digital Investigation**, Elsevier Ltd, v. 28, p. 44–55, 2019. ISSN 17422876. Disponível em: https://doi.org/10.1016/j.diin.2019.01.002.

LUZURIAGA, J. E. *et al.* A disruption tolerant architecture based on MQTT for IoT applications. **2017 14th IEEE Annual Consumer Communications and Networking Conference, CCNC 2017**, p. 71–76, 2017.

MAYER, A. H. **FogChain - A Fog computing architecture integrating Blockchain and Internet of things for personal heath records**. 2020. Tese (Dissertação) — UNISINOS, 2020. Disponível em: http://www.repositorio.jesuita.org.br/handle/UNISINOS/9212.

MCKEEN, F. *et al.* Intel® Software Guard Extensions (Intel® SGX) support for dynamic memory management inside an enclave. **ACM International Conference Proceeding Series**, v. 18-June-20, 2016.

MICROSOFT. **Understanding Public Key Cryptography**. 2014. Data de acesso: 23/08/2021. Disponível em: https://docs.microsoft.com/en-us/previous-versions/tn-archive/aa998077(v=exchg.65)?redirectedfrom=MSDN.

MILETI, D. **Disasters by Design: A Reassessment of Natural Hazards in the United States**. Washington, D.C. - USA: Joseph Henry Press, 1999. ISBN 0309518490. Disponível em: http://www.nap.edu/catalog/5782.html.

MILLER, R. B. Response time in man-computer conversational transactions. Introductions and major concepts. **Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I**, p. 267–277, 1968.

MOINDROT, O.; BOURNHONESQUE, C. Proof of Stake Made Simple with Casper. p. 1–7, 2017.

MOURA, L. M. F. de; BRAUNER, D. F.; JANISSEK-MUNIZ, R. Blockchain e a Perspectiva Tecnológica para a Administração Pública: Uma Revisão Sistemática. **Revista de Administração Contemporânea**, v. 24, n. 3, p. 259–274, 2020. ISSN 1415-6555.

MÜHLBERGER, R. *et al.* Foundational Oracle Patterns: Connecting Blockchain to the Off-Chain World. **Lecture Notes in Business Information Processing**, v. 393 LNBIP, p. 35–51, 2020. ISSN 18651356.

Munich Reinsurance Company. **Munich Reinsurance Company Annual Report 2012**. Munich, Germany, 2001. 148 p. Disponível em: http://www.munichre.com/publications/302-07807_en.pdf.

MUTH, R. *et al.* BBBlockchain: Blockchain-based participation in urban development. **Proceedings - IEEE 15th International Conference on eScience, eScience 2019**, IEEE, p. 321–330, 2019.

NAKAMOTO, S. A peer-to-peer Electronic Cash System. **Nakamoto Institute**, 2008. Disponível em: https://nakamotoinstitute.org/bitcoin/.

NAVEEN, S.; KOUNTE, M. R. Distributing the cloud into fog and edge: New weather in iot based deep learning. *In*: GUNJAN, V. K.; ZURADA, J. M. (Ed.). **Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications**. Singapore: Springer Nature Singapore, 2022. p. 749–758. ISBN 978-981-16-6407-6.

NIELSEN, J. **Usability Engineering**. 1st. ed. [*S.l.*]: Morgan Kaufmann Publishers Inc., 1993.

OUSTERHOUT, J.; ONGARO, D. In search of an understandable consensus algorithm. **2014 USENIX Annual Technical Conference**, p. 305–19, 2014. Disponível em: https://www.usenix.org/system/files/conference/atc14/atc14-paper-ongaro.pdf.

PASDAR, A.; DONG, Z.; LEE, Y. C. Blockchain Oracle Design Patterns. p. 1–25, 2021. Data de acesso: 26/11/2021. Disponível em: http://arxiv.org/abs/2106.09349.

PIZZOLATO, N. D. **O Veículo Conectado: perspectivas sobre a aplicação da internet das coisas no transporte de carga rodoviária**. 2019. Tese (Dissertação) — Pontíficia Universidade Católica do Rio de Janeiro (PUC-RJ), 2019. Disponível em: https://www.maxwell.vrac.puc-rio.br/42207/42207.PDF.

POLKO, A. **Public Space Development in the Context of Urban and Regional Resilience**. 2013. 1–164 p. Disponível em: http://wydawnictwo.ue.katowice.pl/uploads/media/Journal{_}010.pdf{\#}p.

RABAH, K.; RESEARCH, M.; NAIROBI, K. Convergence of AI, IoT, Big Data and Blockchain: A Review. **The Lake Institute Journal**, v. 1, n. 1, p. 1–18, 2018. Disponível em: www.thelakeinstitute.org.

Ribeiro Junior, F. M.; KAMIENSKI, C. A. A Survey on Trustworthiness for the Internet of Things. **IEEE Access**, v. 9, p. 42493–42514, 2021. ISSN 21693536.

Ribeiro Junior, F. M.; KAMIENSKI, C. A. Data resilience system for fog computing. **Computer Networks**, Elsevier B.V., v. 195, n. March, p. 108218, 2021. ISSN 13891286. Disponível em: https://doi.org/10.1016/j.comnet.2021.108218.

RIVERA, R. *et al.* How Digital Identity on Blockchain can contribute in a smart city environment. **2020 IEEE International Smart Cities Conference, ISC2017**, p. 1–4, 2017.

SANTANA, Í. J. S. *et al.* Utilizando a Máquina de Turing para o Desenvolvimento de um Projeto Interdisciplinar. **ADS - IFBA**, n. April 2019, 2015.

SARAF, C.; SABADRA, S. Blockchain platforms: A compendium. **2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018**, IEEE, n. May, p. 1–6, 2018.

SAURABH; DHANARAJ, R. K. A Review Paper on Fog Computing Paradigm to solve Problems and Challenges during Integration of Cloud with IoT. **Journal of Physics: Conference Series**, v. 2007, n. 1, 2021. ISSN 17426596.

SCHULER, D. Digital cities and Digital citizens. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 2362, p. 71–85, 2002. ISSN 16113349. Disponível em: https://link.springer.com/chapter/10.1007/3-540-45636-8_6.

SINGH, A. *et al.* A survey and taxonomy of consensus protocols for blockchains. **Journal of Systems Architecture**, Elsevier B.V., v. 127, n. March, p. 102503, 2022. ISSN 1383-7621. Disponível em: https://doi.org/10.1016/j.sysarc.2022.102503.

SOPEK, M. *et al.* Legal Entity Identifier Blockchained by a Hyperledger Indy Implementation of GraphChain. **Communications in Computer and Information Science**, v. 846, p. 26–36, 2019. ISSN 18650929.

SOUZA, E. F. D. Geração de casos de teste para sistemas da área espacial usando critérios de teste para máquinas de estados finitos. **Ministério da Ciência e Tecnologia - Instituto Nacional de Pesquisas Espaciais (INPE)**, 2010. Data de acesso: 22/07/2021. Disponível em: http://mtc-m16d.sid.inpe.br/col/sid.inpe.br/mtc-m19@80/2010/02.06.20.39/doc/publicacao.pdfhttp://urlib.net/sid.inpe.br/mtc-m19@80/2010/02.06.20.39.

SUGAYAMA, R.; NEGRELLI, E. Veículo conectado na rota da indústria 4.0. *In*: **XXIV Simpósio Internacional de Engenharia Automotiva - Blucher Engineering Proceedings**. [*S.l.*: *s.n.*], 2016. v. 3, n. 1, p. 48–63.

SURWASE, V. REST API Modeling Languages -A Developer ' s Perspective Related papers REST API Modeling Languages - A Developer ' s Perspective. **IJSTE - International Journal of Science Technology and Engineering**, v. 2, n. 10, p. 634–637, 2016.

SWAN, M. **Blockchain: Blueprint for a new economy**. [*S.l.*]: O'Reilly Media Inc, 2015.

TULI, S. *et al.* HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. **Future Generation Computer Systems**, v. 104, p. 187–200, 2020. ISSN 0167739X.

United Nations. **World Urbanization Prospects - The 2018 Revision**. New York, 2018. v. 12, 197–236 p. Data de acesso: 17/04/2022. Disponível em: https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf.

VALENTA, M.; SANDNER, P. Comparison of Ethereum, Hyperledger Fabric and Corda. **Frankfurt School Blockchain Center**, n. June, p. 8, 2017. Disponível em: www.fs-blockchain.decontact@fs-blockchain.dewww.twitter.com/fsblockchainwww.facebook.de/fsblockchain\%0Ahttps://medium.com/@philippsandner/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6.

VRIES, A. de. Bitcoin's Growing Energy Problem. **Joule**, Elsevier Inc., v. 2, n. 5, p. 801–805, 2018. ISSN 25424351. Disponível em: https://doi.org/10.1016/j.joule.2018.04.016.

WANG, S. *et al.* On private data collection of hyperledger fabric. **Proceedings - International Conference on Distributed Computing Systems**, v. 2021-July, p. 819–829, 2021.

WOOD, G. Ethereum: a secure decentralised generalised transaction ledger. **Ethereum Project Yellow Paper**, p. 1–32, 2014. ISSN 1098-6596.

XIE, J. *et al.* A Survey of Blockchain Technology Applied to Smart Cities: Research Issues and Challenges. **IEEE Communications Surveys and Tutorials**, IEEE, v. 21, n. 3, p. 2794–2830, 2019. ISSN 1553877X.

XU, X. *et al.* The blockchain as a software connector. **Proceedings - 2016 13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016**, p. 182–191, 2016.

XU, X. *et al.* A pattern collection for blockchain-based applications. **ACM International Conference Proceeding Series**, n. May, 2018.

YAGA, D. *et al.* **Blockchain Technology Overview**. [*S.l.*], 2018. Disponível em: http://arxiv.org/abs/1906.11078%0Ahttp://dx.doi.org/10.6028/NIST.IR.8202.

ZHANG, F. *et al.* Town crier: An authenticated data feed for smart contracts. **Proceedings of the ACM Conference on Computer and Communications Security**, v. 24-28-Octo, p. 270–282, 2016. ISSN 15437221.

ZHANG, L. *et al.* Ethereum Transaction Performance Evaluation Using Test-Nets. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 11997 LNCS, n. May, p. 179–190, 2020. ISSN 16113349.

ZHOU, J.; LAM, K. Y. Securing digital signatures for non-repudiation. **Computer Communications**, v. 22, n. 8, p. 710–716, 1999. ISSN 01403664.

APÊNDICE A – Portas Utilizadas na Configuração do *Fabric*

Quadro 1 – Portas de configuração dos nós *Hyperledger Fabric*.

Portas utilizadas na configuração dos nós da *Hyperledger Fabric*	
Porta	**Função**
22	SSH (porta padrão para este tipo de comunicação)
7054	Conexão para geração de certificados na Autoridade Certificadora (CA)
7050	Conexão com *Orderers*
80, 443, 7053	Conexão com a *Rest API (Event Hub)*
2376	*Docker* SDK para configurações através do *GoFabric*
3000	*Proxy de administração de redes GoFabric (Hyperledger Fabric)*
8443, 9090	Conexão com *Prometheus* , utilizado para coleta de dados e relatórios
1695	Conexão com *Grafana* utilizado para coleta de dados e relatórios
8080	Conexão com a interface gráfica básica do *GoFabric*

Fonte: O autor.

ANEXO A – Api Hyperledger Fabric

Quadro 2 – Comandos básicos da API Hyperledger Fabric.

Fonte: Babu e Hyperledger Fabric (2017).