UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
CURSO DE CIÊNCIA DA COMPUTAÇÃO

VICTOR LUIZ BERNARDES

UMA PROPOSTA DE PERSISTÊNCIA DE DADOS DESCENTRALIZADA BASEADA EM BLOCKCHAIN E *SIDECHAIN* UTILIZANDO A REDE IPFS PARA DADOS MÉDICOS HETEROGÊNEOS

JOINVILLE
2024

VICTOR LUIZ BERNARDES

UMA PROPOSTA DE PERSISTÊNCIA DE DADOS DESCENTRALIZADA BASEADA EM BLOCKCHAIN E *SIDECHAIN* UTILIZANDO A REDE IPFS PARA DADOS MÉDICOS HETEROGÊNEOS

Trabalho de conclusão de curso apresentado ao curso de Ciência da Computação da Universidade do Estado de Santa Catarina (UDESC), como requisito parcial para obtenção do grau de bacharel em Ciência da Computação.

Orientador: Dr. Adriano Fiorese

JOINVILLE
2024

Para gerar a ficha catalográfica de teses e
dissertações acessar o link:
https://www.udesc.br/bu/manuais/ficha

Bernardes, Victor Luiz
Uma proposta de persistência de dados descentralizada baseada em Blockchain e *Sidechain* utilizando a rede IPFS para dados médicos heterogêneos / Victor Luiz Bernardes. - Joinville, 2024.
66 p. : il. ; 30 cm.

Orientador: Dr. Adriano Fiorese.
.
Monografia (Graduação) - Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Bacharelado em Ciência da Computação, Joinville, 2024.

1. Blockchain. 2. *InterPlanetary File System*. 3. Dados médicos. 4. Análise de desempenho. I. Fiorese, Dr. Adriano . II. , . III. Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Bacharelado em Ciência da Computação. IV. Título.

VICTOR LUIZ BERNARDES

UMA PROPOSTA DE PERSISTÊNCIA DE DADOS DESCENTRALIZADA BASEADA EM BLOCKCHAIN E *SIDECHAIN* UTILIZANDO A REDE IPFS PARA DADOS MÉDICOS HETEROGÊNEOS

Trabalho de conclusão de curso apresentado ao curso de Ciência da Computação da Universidade do Estado de Santa Catarina (UDESC), como requisito parcial para obtenção do grau de bacharel em Ciência da Computação.

Orientador: Dr. Adriano Fiorese

BANCA EXAMINADORA:

Dr. Adriano Fiorese
UDESC

Membros:

Dra. Débora Cabral Nazário
UDESC

Dr. Rafael Rodrigues Obelheiro
UDESC

Joinville, 14 de junho de 2024

RESUMO

O uso da tecnologia *Blockchain* no setor de saúde tem o potencial de revolucionar a maneira como os dados do paciente são armazenados, acessados e compartilhados, dadas as características de imutabilidade e maior proteção dos dados disponibilizadas. No entanto, as preocupações com escalabilidade, privacidade e interoperabilidade permanecem. As *sidechains*, que são uma das soluções propostas para esses desafios, oferecem uma maneira de melhorar a escalabilidade da *Blockchain* principal, melhorando tempo de busca, inserção e aumentando a capacidade de armazenamento de arquivos, permitindo transações mais rápidas e privadas, mantendo a integridade das informações e a transparência.

Dados médicos são sensíveis e críticos, sendo necessário estar em conformidade com normas e regulamentações para realizar o armazenamento e uso de dados pessoais. O modelo de persistência para dados médicos *Electronic Health Record* (EHR) permite a transferência de dados entre instituições médicas de forma a entrar em conformidade com todas as instituições pertencentes a rede.

Neste sentido, este trabalho propõe a utilização de uma rede *Blockchain* em conjunto com uma rede *sidechain peer-to-peer* para aprimorar o modelo de persistência de dados, a fim de aumentar o número de transações por segundo de uma rede *Blockchain* e permitir que arquivos de grande volume gerados por consultas médicas, por exemplo, exames sofisticados de imagem que apresentam grandes volumes, também possam ser armazenados. Este trabalho apresenta uma solução de arquitetura em três camadas, sendo elas: interação, armazenamento e monitoramento.

A partir da implementação da arquitetura proposta, este trabalho tem como objetivo realizar um conjunto de experimentos para analisar o desempenho da rede em operações de envio e recuperação de arquivos, observando latências, transações por segundo (tps) e uso de recursos computacionais em diferentes cenários comparativos.

Palavras-chave: Blockchain. *InterPlanetary File System*. Dados médicos. Análise de desempenho.

ABSTRACT

The use of Blockchain technology in the healthcare sector has the potential to revolutionize the way patient data is stored, accessed, and shared, given its characteristics of immutability and enhanced data protection. However, concerns regarding scalability, privacy, and interoperability remain. Sidechains, which are one of the proposed solutions to these challenges, offer a way to improve the scalability of the main Blockchain, enhancing search and insertion times, and increasing the capacity for file storage, thereby enabling faster and more private transactions while maintaining the integrity and transparency of information.

Medical data is sensitive and critical, requiring compliance with regulations and standards for storing and using personal data. The persistence model for Electronic Health Record (EHR) data allows the transfer of data between medical institutions in a manner that ensures compliance across all institutions in the network.

In this context, this work proposes the use of a Blockchain network in conjunction with a peer-to-peer sidechain network to enhance the data persistence model, aiming to increase the number of transactions per second (tps) of a Blockchain network and enable the storage of large-volume files generated by medical consultations, such as sophisticated imaging exams. This work presents a three-layered architectural solution comprising interaction, storage, and monitoring layers.

By implementing the proposed architecture, this work aims to conduct a series of experiments to analyze the network's performance in file sending and retrieval operations, observing latencies, transactions per second (tps), and computational resource usage in different comparative scenarios.

keywords: Blockchain. InterPlanetary File System. Healthcare Data. Performance Analysis.

LISTA DE FIGURAS

Figura 1 – Exemplo de sequência de blocos em uma rede *Blockchain*. 21
Figura 2 – Exemplo de uma verificação de transação na rede *Blockchain* 21
Figura 3 – Exemplo de uma estrutura do tipo Merkle DAG, na sua configuração em árvore. 27
Figura 4 – Módulos da camada IPFS 28
Figura 5 – Diagrama de interação entre médicos e a rede. 33
Figura 6 – Diagrama de interação entre paciente e a rede. 33
Figura 7 – Abordagem inicial da arquitetura proposta. 34
Figura 8 – Camadas do sistema 35
Figura 9 – Grafo exibindo as conexões entre IPFS *daemon* e IPFS *Cluster* em uma rede com 8 nós. 37
Figura 10 – Arquitetura utilizada para realizar os experimentos 39
Figura 11 – Latências de recuperação e envio de arquivos para a rede *Blockchain*. 44
Figura 12 – Latências após removidas as transações mal sucedidas. 44
Figura 13 – Transações por segundo relativo ao número de nós na rede para rede *Blockchain (onchain)*. 46
Figura 14 – Transações por segundo relativo ao número de nós na rede para modelo híbrido em *(sidechain)* 47
Figura 15 – Transações por segundo utilizando somente *Blockchain* comparado ao armazenamento híbrido em *sidechain* 48
Figura 16 – Comparativo entre a média de utilização de CPU e memória RAM por conjunto de nós entre armazenamento *onchain* e armazenamento híbrido em *sidechain* 49
Figura 17 – Gráfico de dispersão das latências para ambas as redes em sobrecarga (envio) 50
Figura 18 – Latências para operações de recuperação entre 8 e 16 nós para diferentes tamanhos de arquivos (1MB, 10MB e 100MB) 52
Figura 19 – Latências para operações de envio entre 8 e 16 nós para diferentes tamanhos de arquivos (1MB, 10MB e 100MB) 53
Figura 20 – Dispersão das latências entre os tamanhos avaliados para a rede com 8 nós . 54
Figura 21 – Comparação de latências para o experimento com mudanças nos fatores de replicação para as operações de recuperação e envio em relação ao número de nós. 56
Figura 22 – Comparação entre a média de transações por segundo para cada conjunto de nós. 57

LISTA DE TABELAS

Tabela 1 – Tabela de comparação entre modelos de *Blockchain* 23
Tabela 2 – Tabela de comparação entre métodos de consenso. 24
Tabela 3 – Comparativo entre os trabalhos relacionados. 31
Tabela 4 – Configuração da máquina de experimentos. 43
Tabela 5 – Resumo da análise ANOVA e métrica de latência para o cenário *onchain* variando o número de nós da rede. 45
Tabela 6 – Conjunto de métricas coletadas para o cenário de sobrecarga utilizando apenas a operação de envio de arquivos. 53
Tabela 7 – Latências para o envio e recuperação de arquivos de 1 MB, 10 MB e 100 MB em 8 e 16 nós. 66
Tabela 8 – Média de recursos computacionais utilizados no intervalo de execução dos experimentos para arquivos de diferentes volumes. 66

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
CID	*Content Identifier*
EHR	*Electronic Health Record*
GDPR	*General Data Protection Regulation*
HIPAA	*Health Insurance Portability and Accountability Act*
IPFS	*InterPlanetary File System*
LGPD	Lei Geral de Proteção de Dados Pessoais
P2P	*Peer-to-Peer*
PBFT	*Practical Byzantine Fault Tolerance*

SUMÁRIO

1 INTRODUÇÃO 11
1.1 ESTRUTURA DO TRABALHO 15
2 REFERENCIAL TEÓRICO 16
2.1 SEGURANÇA EM *HEALTHCARE* 16
2.1.1 ELECTRONIC HEALTH RECORDS (EHR) 17
2.1.2 REGULAMENTAÇÕES 17
2.2 BLOCKCHAIN 19
2.2.1 MODELOS DE NEGÓCIO 21
2.2.2 CONSENSO 22
2.2.3 CONTRATOS INTELIGENTES 24
2.2.4 *FRAMEWORKS* 25
2.3 SIDECHAINS 25
2.4 IPFS 26
2.4.1 MERKLE DAG 27
2.4.2 IPFS CLUSTER 27
2.5 CONSIDERAÇÕES DO CAPÍTULO 28
3 TRABALHOS RELACIONADOS 29
3.1 CONSIDERAÇÕES DO CAPÍTULO 31
4 PROPOSTA 32
4.1 ATORES PARTICIPANTES 32
4.2 ARQUITETURA 34
4.2.1 BLOCKCHAIN 35
4.2.2 IPFS 36
4.2.3 INTERAÇÃO 36
4.2.4 MONITORAMENTO 38
4.3 PLANO DE TESTES 39
4.4 CONSIDERAÇÕES DO CAPÍTULO 41
5 EXPERIMENTOS 42
5.1 DESCRIÇÃO DOS EXPERIMENTOS 42
5.2 COMPARAÇÃO ENTRE ARMAZENAMENTO *ONCHAIN* E *SIDECHAIN* 43
5.2.1 CENÁRIO COM ALTA DEMANDA 46
5.3 COMPARAÇÃO ENTRE ARQUIVOS COM VOLUMES DIFERENTES PARA ARMAZENAMENTO HÍBRIDO EM *SIDECHAIN* 51
5.3.1 CENÁRIO COM ALTA DEMANDA 51

5.4 COMPARAÇÃO ENTRE OS FATORES DE REPLICAÇÃO PARA O ARMAZENAMENTO EM *SIDECHAIN* **55**
5.5 DISCUSSÃO DOS RESULTADOS **57**
5.6 CONSIDERAÇÕES DO CAPÍTULO **58**
6 CONCLUSÃO E CONSIDERAÇÕES FINAIS 59
REFERÊNCIAS 61
APÊNDICE A – CÓDIGO FONTE E INSTRUÇÕES 65
APÊNDICE B – RESULTADOS OBTIDOS EM EXPERIMENTOS COM ARQUIVOS DE MÚLTIPLOS TAMANHOS 66

1 INTRODUÇÃO

Com o aumento do número de informações geradas por diferentes processos e atores produtivos na última década, diferentes áreas da economia foram forçadas a se adaptar neste novo paradigma de ”sociedade da informação”. A flexibilidade deste novo paradigma permite uma maior adaptação de trabalhadores, consumidores, produtores e usuários, devido ao seu contínuo aperfeiçoamento (WERTHEIN, 2000). O conceito de paradigma da informação evoluiu ao longo do tempo, abrangendo a ideia de que os sistemas vivos são fundamentalmente entidades processadoras de informação, sendo a vida não apenas baseada em química, mas também em informação (BARBIERI, 2016). Nesse paradigma, o grande volume de informações gerado e circulante proporciona mais eficiência, flexibilidade, escalabilidade e maior lucratividade nos processos que as aproveitam (COHEN, 2002).

Alguns setores da economia foram rapidamente se adaptando a este novo modelo, como a indústria automobilística e financeira. Porém, existem setores onde a adaptação ao modelo de grandes volumes de informação acontece de maneira lenta e cuidadosa. É o caso do setor da saúde. Nesse cenário, as informações oriundas dos pacientes e clínicas são extremamente críticas e possuem restrições sobre quem possui acesso e como elas serão utilizadas (CHANG; CHEN, 2020).

Segundo (SINGH et al., 2021), alguns dos principais problemas relacionados a sistemas *healthcare* podem ser listados como sendo: controle de acesso, identificação única, confidencialidade, integridade e métodos para transmissão de dados. Desta forma, é um desafio fundamental elaborar sistemas complexos que atendem a todas as necessidades e que sejam adaptados a diferentes leis regulatórias.

Pensando em formas de conduzir essa nova era da informação de maneira segura, países e blocos econômicos começaram a implementar leis que fornecem amparo jurídico, a fim de aumentar a privacidade dos dados que trafegam e são armazenados por empresas e governos. A regulamentação mais conhecida e amplamente utilizada é a *General Data Protection Regulation (GDPR)*, implementada pela União Europeia em 2016.

A lei geral de proteção de dados (LGPD) no Brasil estende os conceitos da GDPR para o âmbito nacional e abrange uma vasta categoria de tópicos e níveis de sensibilidade de informação. A LGPD estabelece normas e regras para a coleta, armazenamento, processamento e o compartilhamento de dados pessoais, incluindo informações de saúde. Assim, os sistemas de saúde devem se adequar às exigências da lei para garantir a proteção e privacidade das informações dos pacientes. Segundo (SIQUEIRA et al., 2022), as seguintes medidas devem ser consideradas para se ter conformidade com a LGPD:

- Coletar apenas as informações necessárias para o tratamento do paciente, evitando a coleta excessiva de dados;
- Garantir a segurança e confidencialidade dos dados pessoais, por meio de medidas

técnicas e organizacionais adequadas;

- Obter o consentimento do paciente para coleta e tratamento de seus dados, informando-o sobre as finalidades do processamento dos dados; e
- Assegurar o direito do paciente de acessar, corrigir, excluir seus dados pessoais.

Para sistemas de informação no âmbito da saúde, as informações dos pacientes representam elemento central para aplicações e também de proteção conforme a LGPD. Tais informações constituem o conceito de registro eletrônico de saúde, do inglês *electronic health record*. O conceito de EHR *(Electronic Health Record)* foi introduzido ao existir a preocupação com os meios tecnológicos utilizados na área da saúde. Como as informações são sensíveis, em vários países existem leis que protegem essas informações (ALASSAF; ALKAZEMI; GUTUB, 2017). O grande volume de dados sensíveis gerenciados por sistemas de EHR ao redor do mundo tem sido alvo de ataques cibernéticos, onde os atacantes podem utilizar desta base de dados de forma maliciosa e vender para outras pessoas mal intencionadas no mercado clandestino. Segundo (PILARES et al., 2022), durante a pandemia do COVID-19 houve um aumento exponencial do número de ataques cibernéticos a hospitais e clínicas, justificado pela grande quantidade de dados sensíveis que estava em mãos dos hospitais e clínicas e a falta de cuidado com o controle de acesso interno dos sistemas. O principal ataque realizado durante este período foi o sequestro de dados, no qual o atacante bloqueava o acesso à base de dados e requisitava uma recompensa, geralmente em moedas digitais como Bitcoin.

O setor da saúde tem sido particularmente vulnerável a ataques de *ransomware*, com uma alta em número de incidentes envolvendo *malwares* em sistemas médicos, em grande parte atribuídos a um tipo específico chamado de ransomware (MAIMó et al., 2019). Os hospitais são alvos fáceis devido à natureza crítica dos seus serviços e aos dados sensíveis dos pacientes que detêm. Estes ataques levam a perdas financeiras significativas e a perturbações nos serviços de saúde, com alguns hospitais a terem de pagar resgates substanciais para recuperar o acesso aos seus dados (ZIMBA; CHISHIMBA, 2019).

A utilização de *Blockchain* tem permitido criar sistemas financeiros descentralizados, oferecendo uma alternativa segura, íntegra e transparente para proteção dos dados transacionados sem a necessidade de uma entidade intermediária entre as partes. Devido a suas características, *Blockchain* começou a chamar a atenção de empresas fora do setor financeiro, tendo aplicações em áreas como: Internet das coisas, gerenciamento de cadeias de suprimento e *healthcare* (WUST; GERVAIS, 2018).

Nesse sentido, *Blockchain* possui um grande potencial de aplicação na área médica, de forma que podem ser citadas ferramentas que utilizam dessa tecnologia para atender as demandas de interoperabilidade dos dados (ESPOSITO et al., 2018). Imutabilidade, rastreabilidade e a eliminação de um terceiro ator que gerencie os dados salvos são as principais características de uma *Blockchain* (NAKAMOTO, 2009). Portanto, com a *Blockchain* é possível criar aplicações

médicas descentralizadas com a garantia de imutabilidade e privacidade entre os dados compartilhados por diferentes instituições. Além disso, há a eliminação de um terceiro órgão para controlar a distribuição e o acesso aos dados, diminuindo um ponto de falha e atribuindo essa responsabilidade a rede *Blockchain* privada ou até mesmo pública.

De fato, já existem aplicações na área médica que utilizam da *Blockchain* para registrar dados de pacientes, realizar acompanhamentos e consultas. De acordo com (HÖLBL et al., 2018), as principais propostas na área de *healthcare* para a utilização da *Blockchain* são: facilitar o acesso dos pacientes aos seus próprios dados, registro imutável, confiança e também propostas para a utilização dos dados de forma anônima por institutos de pesquisa.

Os dados que podem ser salvos por transação em uma *Blockchain* são limitados, sendo geralmente salvos apenas dados essenciais. Assim, para aplicações de *healthcare*, dados como médico que atendeu determinado paciente, dados relacionados a condição de saúde do paciente e a instituição que realizou o atendimento, devem ser armazenados na cadeia principal da *Blockchain*, de forma a serem facilmente recuperáveis. Porém, existem dados como imagens médicas e resultados de exames que não podem ser gravados em uma transação na *Blockchain* por conta da limitação no tamanho do bloco e da consequente latência de recuperação dos mesmos.

Um dos métodos para se contornar as limitações de escalabilidade em uma *Blockchain* é a utilização de uma rede ou *Blockchain* externa, com uma conexão com a *Blockchain* principal. Segundo (SINGH et al., 2020), uma *sidechain* pode ser definida como uma comunicação bidirecional que permite a troca de informações entre a cadeia de blocos principal (*blockchain* principal) e uma cadeia de blocos secundária (*blockchain* secundária). *Sidechains* podem ter seus próprios mecanismos de consenso, implementações de novas funcionalidades e outras características diferentes da rede principal. Ou seja, uma *sidechain* permite que a rede principal *Blockchain* possa efetuar mais transações por segundo, consequentemente aumentando a escalabilidade da mesma, armazenar arquivos de tamanhos variados sem comprometer a rede principal, entre outras características a depender de como e qual *sidechain* será implementada.

O protocolo *InterPlanetary File System* (IPFS) tem aumentado sua popularidade gradativamente durante os últimos anos. Essa popularidade se dá por suas características de armazenamento e acesso a arquivos de forma distribuída e descentralizada, sendo um protocolo *peer-to-peer (P2P)* modular, que pode ser integrado com outras tecnologias como *Blockchain*. O protocolo IPFS oferece vantagens significantes em termos de integridade e persistência dos dados ao comparar-se com protocolos como *Distributed File System (DFS)*, utilizando-se em seu mecanismo principal uma estrutura de dados chamada *Merkle Directed Acyclic Graph (Merkle DAG)* (CRISTEA et al., 2020). Nesse sentido, o endereçamento de arquivos no IPFS é feito através de um identificador baseado no *hash* dos arquivos salvos, ou seja, para realizar buscas na rede basta utilizar o *content identifier* (CID) e o arquivo será recuperado. Com esse método de armazenar os CID de cada arquivo em uma Merkle DAG, o tempo de busca dos arquivos pela rede é drasticamente reduzido.

Assim, o problema da utilização de *sidechains* para armazenamento de dados em *blockchains* tem sido pesquisado. Entre outros, especificamente para a área da saúde, o trabalho realizado por (AZBEG; OUCHETTO; Jai Andaloussi, 2022) contempla uma proposta de sistema chamado BlockMedCare, onde foi utilizada uma combinação de dispositivos médicos para Internet das Coisas (IoT), a *Blockchain* Ethereum na sua versão *permissioned* utilizando como consenso *Proof of Authority (PoA)* e o protocolo IPFS para armazenamento de dados. As principais características deste trabalho foram a implementação da rede principal e sua *sidechain*, a criação de uma arquitetura de interação entre IPFS (*Sidechain*) e a *Blockchain* e a especificação pública do *smart contract* utilizado pela *Blockchain*. Porém, a falta de testes na interação entre *sidechain* e a rede principal, deixou o trabalho sem resultados para perguntas como qual o impacto em tempos de busca e inserção ao armazenar arquivos de tamanhos variados na *sidechain*, o tempo para criptografar e descriptografar os arquivos salvos na *sidechain* e como lidar com novas versões de arquivos salvos na *sidechain*.

Portanto, a proposta deste trabalho é desenvolver a arquitetura de armazenamento híbrido em *sidechain*, implementando bem como realizar experimentos com base em uma abordagem integradora entre *Blockchain* e uma *sidechain* baseada no protocolo IPFS para aumentar a escalabilidade de uma *Blockchain* no cenário de dados médicos heterogêneos. Para realizar este objetivo, será realizada uma integração entre uma *Blockchain* em um modelo consorciado e o protocolo IPFS como alternativa de persistência de dados de alto volume em *sidechain*. Após implementada a arquitetura proposta, serão realizados os experimentos sobre a rede para comparar o seu desempenho em operações de envio e recuperação de arquivos em diferentes cenários.

Na abordagem proposta, a inserção dos dados na *sidechain* se dará através da utilização do *Application Programming Interface (API)* do módulo IPFS através da linguagem Golang. Será utilizado também um contrato inteligente para fazer as operações de envio das informações (endereço de paciente, médico e instituição) para a *Blockchain* junto ao CID do arquivo salvo após o envio do arquivo para a *sidechain*.

Com o envio dos arquivos no IPFS, será retornado um identificador do conteúdo chamado CID. Após a transferência do arquivo para a rede IPFS, o CID do conteúdo salvo na *sidechain* será registrado em uma transação na rede principal *Blockchain* junto aos dados de identificação de médico, paciente e instituição médica. A *sidechain* permitirá o armazenamento de arquivos de imagens, como raio-x e modelos 3D, além de outros dados tipos de dados na Blockchain, como token do médico, token da instituição onde foi realizada a transação e o token do paciente. Parte do trabalho envolve a avaliação do protocolo IPFS para verificar se ele pode fornecer o desempenho necessário em termos de latência, vazão e tamanho de arquivos suportados sem comprometer o desempenho da rede. comparando cenários onde há sobrecarga de arquivos, diferentes fatores de replicação e também comparando o desempenho da proposta em relação ao armazenamento de arquivos somente na rede *Blockchain*.

1.1 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado da seguinte forma. No Capítulo 2 são introduzidos os conceitos básicos sobre dados médicos e suas regulamentações, *Blockchain* e sua estrutura, *Sidechains* e por fim os aspectos básicos do IPFS. No Capítulo 3 é realizada uma análise sobre os trabalhos que propõem o uso de *Blockchain* e *sidechain*, voltado ao cenário de armazenamento de dados médicos. O Capítulo 4 define a arquitetura inicial e a proposta de experimentação a ser utilizada. O Capítulo 5 apresenta os experimentos realizados e os resultados obtidos, bem como sua análise. Por fim, o Capítulo 6 aborda a conclusão do trabalho e as considerações finais, bem como possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico necessário para compreensão do trabalho, de forma a envolver o leitor em relação aos principais tópicos, tecnologias e ferramentas utilizadas ao decorrer do trabalho. Primeiramente, na Seção 2.1 é realizada uma análise sobre dados médicos, regulamentações e modelos de armazenamento de dados médicos. Na sequência, na Seção 2.2 são definidos os conceitos sobre *Blockchain* e suas ferramentas. Na Seção 2.3, é abordada uma breve contextualização sobre *sidechains*. Por fim, na Seção 2.4 são apresentados os conceitos envolvendo a rede IPFS e seus principais componentes a serem utilizados neste trabalho.

2.1 SEGURANÇA EM *HEALTHCARE*

Os sistemas de saúde estão evoluindo rapidamente e se tornando cada vez mais complexos. Com o avanço da tecnologia e da digitalização, os prestadores de serviços de saúde conseguiram melhorar seus serviços, oferecendo melhor atendimento ao paciente, diagnósticos mais rápidos e planos de tratamento mais eficientes. No entanto, com esses avanços, surge um novo conjunto de desafios para manter a segurança dos dados confidenciais dos pacientes.

Uma grande preocupação é o risco de ataques cibernéticos aos sistemas de saúde que podem comprometer os dados do paciente. Segundo (FERNÁNDEZ-ALEMÁN et al., 2013), nos últimos anos a privacidade em sistemas médicos tem sido ameaçada por hackers, vírus e outros tipos de *malware*. Em muitos casos, são reportados perda acidental de dados ou roubo de informações sensíveis, o que pode gerar multas e punições para as empresas detentoras dos dados. O setor da saúde tem sido particularmente vulnerável a ataques de ransomware, justificado pela grande quantidade de dados sensíveis que estava em mãos dessas instituições e a falta de cuidado com a segurança interna para controlar o acesso aos sistemas. Segundo (MAIMó et al., 2019), ataques utilizando ransomware constituem uma grande parcela dos incidentes envolvendo *malware* (programa malicioso) no setor da saúde, com mais de 70% dos ataques resultando em vazamento de dados.

Portanto, é essencial proteger a integridade das informações de saúde para garantir a segurança dos pacientes. Um elemento crucial dessa proteção é assegurar que todo o ciclo de vida das informações seja completamente auditável. A disponibilidade das informações de saúde também é fundamental para garantir uma prestação eficaz de cuidados de saúde. Os sistemas de saúde devem continuar funcionando mesmo diante de desastres naturais, falhas de sistema e ataques de negação de serviço (DoS). Além disso, a segurança também implica responsabilidade, refletindo-se no direito das pessoas de questionar ou criticar a ocorrência de eventos.

Desta forma, a segurança no armazenamento de dados médicos é de extrema importância para a organização, gerando uma evolução lenta dos sistemas médicos, visto que é necessário avaliar de maneira cautelosa o uso de tecnologias emergentes no mercado como Internet das Coisas, *Blockchain*, computação em nuvem entre outras. Além disso, os sistemas de gerenciamento

de dados médicos necessitam estar em conformidade com as leis de armazenamento e uso de dados sensíveis no país.

Com a necessidade de gerenciar os dados provenientes dos pacientes e suas consultas, foram desenvolvidos modelos que armazenam e gerenciam o acesso a estas informações sensíveis. Segundo (HEART; BEN-ASSULI; SHABTAI, 2017), os principais modelos de armazenamento para dados voltados a saúde são *Electronic Medical Record* (EMR), *Personal Medical Record* (PMR) e *Electronic Health Record* (EHR). Cada modelo possui um nível de acesso e exposição a informações do paciente. Neste trabalho abordaremos o modelo interoperável entre instituições médicas, chamado de EHR.

2.1.1 ELECTRONIC HEALTH RECORDS (EHR)

O conceito de *(Electronic Health Record)* (EHR) foi introduzido ao existir a preocupação com os meios tecnológicos utilizados na área da saúde. Como as informações são sensíveis, em vários países existem leis que protegem essas informações (ALASSAF; ALKAZEMI; GUTUB, 2017). Segundo (ONC, 2019), EHR pode ser definido como um registro em versão eletrônica do prontuário físico do paciente, e um sistema EHR é construído para ir além dos dados clínicos coletados no consultório de um provedor, e pode incluir uma visão ampla dos cuidados de um paciente. A principal característica de um sistema que possui EHR é ser interoperável entre instituições clinicas.

Outros modelos como EMR *(Electronic medical records)*, são limitados em alguns aspectos. Segundo (ONC, 2011), um EMR pode ser descrito como um resumo geral da consulta realizada, contendo dados como contagem de plaquetas, contagem de glóbulos vermelhos, entre outros dados relacionados ao exame realizado. Porém em um EMR, as informações contidas não são compartilhadas entre laboratórios ou instituições médicas. O prontuário do paciente pode até ser impresso e entregue por serviços de transporte de cartas até um especialista. Portanto, um EMR pode ser caracterizado como uma versão da consulta atual de forma digital que agrupa os dados de uma consulta singular, não compartilhado entre instituições de saúde.

Com o uso de EHRs, todos os membros da equipe têm acesso imediato às informações mais recentes, permitindo um atendimento mais coordenado e centrado no paciente. Alguns dos pontos fortes da utilização de EHR segundo (ONC, 2011) são: o paciente consegue acompanhar a sua evolução nos exames e tratamentos médicos, especialistas conseguem visualizar a saúde geral do paciente, evitando em alguns casos duplicação de consultas.

2.1.2 REGULAMENTAÇÕES

No Brasil, segundo (SANTOS, 2020), a Lei Geral de Proteção de Dados Pessoais (LGPD) foi promulgada para proteger os direitos fundamentais de liberdade e de privacidade e a livre formação da personalidade de cada indivíduo. A LGPD discorre sobre o tratamento de dados pessoais, dispostos em meio físico ou digital, feito por pessoa física ou jurídica de direito público

ou privado, englobando um amplo conjunto de operações que podem ocorrer em meios manuais ou digitais. No âmbito da LGPD, o tratamento dos dados pessoais pode ser realizado por dois agentes de tratamento, o Controlador e o Operador. Além deles, há a figura do Encarregado, que atua como canal de comunicação entre o Controlador, o Operador, os titulares dos dados e a Autoridade Nacional de Proteção de Dados (ANPD) (ANPD, 2024). Segundo (BRASIL, 2018), o tratamento de dados diz respeito a qualquer atividade que utiliza um dado pessoal na execução da sua operação, como, por exemplo: coleta, produção, recepção, classificação, utilização, acesso, reprodução, transmissão, distribuição, processamento, arquivamento, armazenamento, eliminação, avaliação ou controle da informação, modificação, comunicação, transferência, difusão ou extração. No caso do setor público, a principal finalidade do tratamento está relacionada à execução de políticas públicas. A LGPD estabelece uma estrutura legal de direitos dos titulares de dados pessoais. Esses direitos devem ser garantidos durante toda a existência do tratamento dos dados pessoais realizado pelo órgão ou entidade. Para o exercício destes direitos, a LGPD prevê um conjunto de ferramentas que aprofundam obrigações de transparência ativa e passiva e criam meios processuais para mobilizar a Administração Pública.

De acordo com (BRASIL, 2018), o titular possui os seguintes direitos referentes aos seus dados:

- Confirmação da existência de tratamento;
- Acesso aos dados;
- Correção de dados incompletos, inexatos ou desatualizados;
- Anonimização, bloqueio ou eliminação de dados desnecessários, excessivos ou tratados em desconformidade com o disposto na LGPD;
- Portabilidade dos dados a outro fornecedor de serviço ou produto, mediante requisição expressa;
- Informação das entidades públicas e privadas com as quais o Controlador realizou uso compartilhado de dados;
- Informação sobre a possibilidade de não fornecer consentimento e sobre consequências da negativa;
- Revogação do consentimento.

Outra regulamentação amplamente discutida é a *General Data Protection Regulation (GDPR)*, implementada pela União Europeia em 2018 (PADRÃO; LOPES, 2023). Assim como a LGPD no Brasil, ela abrange o uso de dados de maneira mais geral, não focando apenas em dados médicos, mas ao uso generalizado. Ela garante direitos ao titular por meio de aplicações de multas e penalidades para as instituições que descumprirem as normas pré-estabelecidas.

Por outro lado, nos Estados Unidos é aplicado a regulamentação chamada de *Health Insurance Portability and Accountability Act (HIPAA)*, criado em 1996. Conforme (CONTROL; (CDC), 1996), a HIPAA é uma lei federal que definiu padrões nacionais para proteger informações sensíveis de saúde do paciente de serem divulgadas sem o consentimento ou conhecimento do paciente. A HIPAA, diferentemente da LGPD e GDPR, é uma regulamentação focada somente no âmbito de dados médicos, sendo uma complementação de outras normas de privacidade já existentes no país.

Todas as regulamentações citadas possuem penalidades e multas estipuladas pela quantidade de dados vazados e a sensibilidade dos mesmos. No Brasil, segundo (BRASIL, 2018), a ANPD é o órgão que fiscaliza e é responsável por aplicar as multas e penalidades. De acordo com (BRASIL, 2018), as sanções administrativas que podem ser impostas a uma instituição que não cumprir as normas da LGPD incluem:

- Multas de até R$50 milhões ou até 2% do faturamento da empresa;
- Bloqueio dos dados;
- Publicação da infração;
- Multa diária até a adequação conforme as normas;
- Advertência.

2.2 BLOCKCHAIN

A tecnologia *Blockchain* foi concebida como a base que possibilita o funcionamento do Bitcoin, destacando-se por suas propriedades descentralizadas (NAKAMOTO, 2009). A incorporação de uma rede *Blockchain* tem dado origem a sistemas financeiros descentralizados, proporcionando uma maneira segura, íntegra e transparente de proteger os dados das transações, sem a necessidade de intermediários.

O potencial da tecnologia *Blockchain* na Indústria 4.0 é ainda enfatizado pela sua capacidade de melhorar processos em vários setores, incluindo saúde, construção e transporte marítimo. Na área da saúde, a *Blockchain* oferece soluções para gestão de identidade, gestão de contratos e integridade dos dados armazenados (AGBO; MAHMOUD; EKLUND, 2019).

O objetivo principal de uma *Blockchain* pode ser definido como um método capaz de garantir que haja consenso entre as transações realizadas na rede, dentro de um contexto de participantes, utilizando-se da sua arquitetura de armazenamento descentralizada. Segundo (ZHENG et al., 2018), as principais características de uma rede *Blockchain* podem ser listadas a seguir:

- **Descentralização**: Em um sistema centralizado há a necessidade de uma instituição confiável fazendo a intermediação entre as instituições envolvidas; no caso da *Blockchain*,

o sistema distribuído se encarrega de entrar em consenso e atualizar todos os nós da rede, garantindo via algoritmos criptográficos a validade das transações realizadas.

- **Transparência**: Cada transação validada pela rede é replicada para todos os nós participantes.
- **Imutabilidade**: Os registros salvos na rede não podem ser alterados ou corrompidos, apenas em casos onde o método de consenso escolhido pode apresentar falhas. O chamado ataque de 51%, quando 51% dos nós da rede são maliciosos e orquestrados, no caso do método de consenso *Proof of Work (PoW)*, é um exemplo onde os registros podem ser comprometidos.
- **Auditabilidade**: Todos os blocos são interconectados e possuem a lista de transações realizadas, o monitoramento do fluxo das transações torna-se uma tarefa simples, basta percorrer a cadeia de transações realizadas.
- **Anonimato**: As chaves utilizadas para assinar as transações não contêm informações que podem levar a revelar a identidade do usuário, e podem ser geradas novas chaves para realizar transações na rede, evitando que uma determinada chave de assinatura possa ser vinculada a uma pessoa em específico.

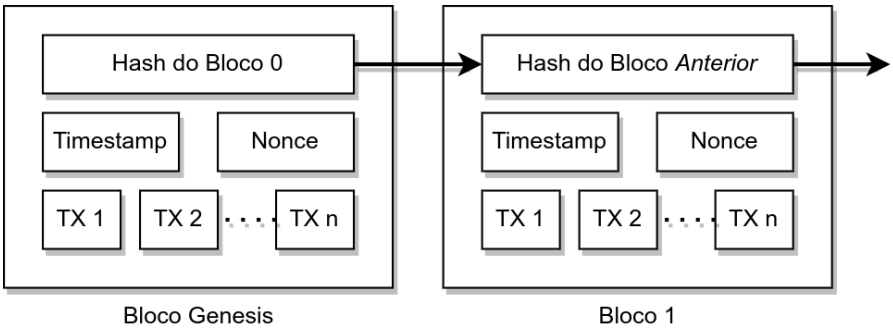
A cadeia de blocos na rede *Blockchain* pode ser representada como na Figura 1, onde existe uma ligação entre cada bloco, garantido que os blocos anteriores não sofreram alterações devido ao seu *hash* único gerado a partir das informações armazenadas. Cada bloco possui uma série de campos e metadados necessários para identificar o bloco na rede. De acordo com (NAKAMOTO, 2009), os campos que compõem um bloco, sendo os principais, mas não se limitando a estes, são:

- *Hash* do bloco anterior - Referência ao bloco anterior.
- *Timestamp* - Data e hora de criação do bloco.
- *Nonce* - Campo referente a um valor inteiro incrementado a cada novo bloco criado, sendo necessário para a geração de um *hash* distinto.
- Transações - *Hash* de todas as transações feitas no bloco.

O bloco número 0, ou também chamado de gênesis na Figura 1, é o primeiro bloco a ser formado na rede e, portanto, não possui *hash* anterior e neste caso é geralmente atribuído um valor nulo. Além disso, o bloco gênesis pode conter informações diferentes de acordo com a implementação da *Blockchain*, por exemplo, dados de configurações.

Transações são parte fundamental da rede *Blockchain*. Elas são responsáveis por transferir valores e informações, validando por meio de assinaturas digitais os conteúdos enviados e

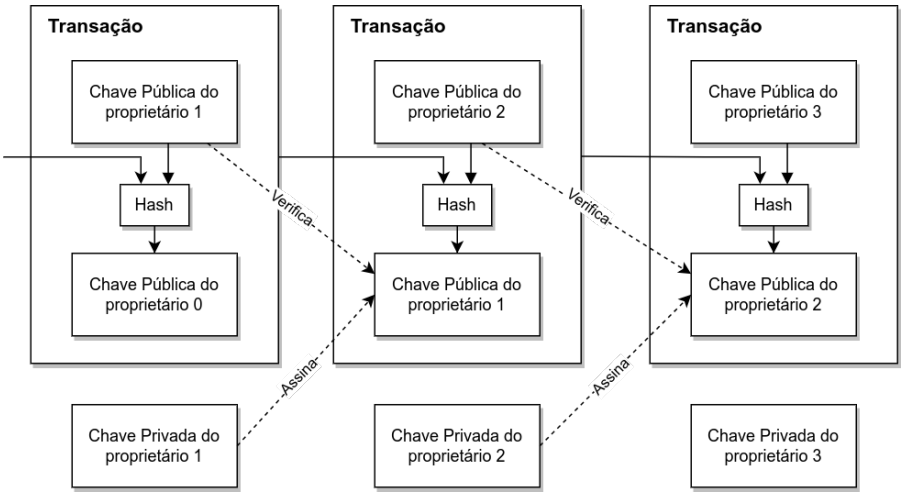
Figura 1 – Exemplo de sequência de blocos em uma rede *Blockchain*.



Fonte: Elaborado pelo autor (2024).

recebidos em cada uma delas. Uma transação ocorre de um endereço (identificador) para o outro, sendo necessária a chave pública do destinatário e a chave privada do remetente da transação, que a assina. Na Figura 2, temos um exemplo de verificação de assinatura de uma transação, onde o remetente, ou seja o proprietário, assina digitalmente a transação junto ao *hash* da transação anterior, utilizando-se também da chave pública do destinatário. O destinatário pode verificar as assinaturas da transação realizada, demonstrando a transparência entre transações, uma das propriedades de uma rede *Blockchain*.

Figura 2 – Exemplo de uma verificação de transação na rede *Blockchain*



Fonte: Adaptado de (NAKAMOTO, 2009).

2.2.1 MODELOS DE NEGÓCIO

Atualmente, existem três modelos de redes *Blockchain*, sendo eles: Público, Privado e Consórcio. Cada um destes modelos possui características e indicações de uso diferentes dependendo do objetivo da rede. Segundo (BUTERIN, 2015), podemos caracterizar os modelos da seguinte forma:

- *Blockchain* Pública: Todos ao redor do mundo podem ler e enviar transações na rede e participar do processo de consenso. Segundo (BUTERIN, 2015), *Blockchains* públicas são protegidas através de sua economia interna, como é o caso da *Blockchain* Ethereum e o *token* Ether, onde existe uma representação de valor monetário na forma de *tokens* na *Blockchain*. Neste caso, os incentivos a participação e proteção da rede são proporcionais a força computacional ou a quantidade de *tokens* que um participante possui. Por este motivo, são consideradas totalmente descentralizadas.

- *Blockchain* em Consórcio: Neste modelo o processo de consenso é controlado por um conjunto pré-selecionado de nós. Por exemplo, entre 15 instituições financeiras, dos 15 nós da rede (i.e., um para cada instituição), apenas 10 se comprometeriam a validar os blocos da rede. Podem ser redes públicas ou restritas. Também podem ser redes híbridas onde parte da rede é disponível para o público geral e outras camadas apenas para os integrantes do consórcio.

- *Blockchain* privada: Apenas uma instituição é responsável pela validação, sendo considerado um modelo centralizado.

Na Tabela 1 foram resumidos os principais aspectos dos modelos de rede *Blockchain*, separando-os pelas seguintes propriedades:

- Consenso: Nós que podem participar do processo de consenso e tomar a decisão final de validação.

- Permissão de leitura: Quem, ou seja, quais nós podem realizar a leitura das transações feitas na *Blockchain*.

- Imutabilidade: Se a rede pode ser adulterada, ou seja, se os dados podem ser adulterados com ataques ao método de consenso.

- Eficiência: Quantidade de transações validadas e propagadas pela rede por unidade de tempo.

- Centralização: Se a rede é centralizada, parcialmente centralizada ou descentralizada em termos das instituições responsáveis pela validação das transações.

- Processo de consenso: Nós pertencentes à rede *Blockchain* que podem participar das etapas de consenso.

2.2.2 CONSENSO

O bom funcionamento de uma rede *Blockchain* passa principalmente pela escolha e implementação de um método de consenso. Como a *Blockchain* utiliza de uma arquitetura

Tabela 1 – Tabela de comparação entre modelos de *Blockchain*

Propriedade	**Modelo de rede *Blockchain***		
	Pública	*Consórcio*	*Privada*
Consenso	Todos	Selecionados	Uma instituição
Permissão de leitura	pública	Pode ser pública ou restrita	Pode ser pública ou restrita
Imutabilidade	Dificuldade alta de adulterar	Pode ser adulterado	Pode ser adulterado
Eficiência	Baixa	Alta	Alta
Centralização	Não	Parcialmente	Sim
Processo de consenso	Todos	Selecionados	Selecionados

Fonte: Adaptado de (ZHENG et al., 2018).

peer-to-peer (P2P) para distribuir e validar as transações, é essencial que haja um consenso entre os nós que validam as mesmas. Um algoritmo de consenso é responsável por assegurar que a maioria dos nós da rede entrem em acordo, mesmo que alguns nós da rede falhem ou sejam maliciosos, ou seja, possuem a intenção de corromper ou danificar a rede.

Cada método de consenso possui características que diferem entre si e muitos estão atrelados ao modelo da rede escolhida, seja pública, privada ou em consórcio. Alguns exemplos de algoritmos de consenso:

- *Proof of Work* (PoW): Introduzido na *Blockchain* do Bitcoin (NAKAMOTO, 2009), neste método de consenso, os nós validadores da rede usam a força computacional para garantir a integridade da rede. Nesse caso, é utilizada CPU para resolver uma função de *hash*. Os nós que resolvem essa função recebem uma recompensa pelo esforço computacional. Segundo (NAKAMOTO, 2009), uma vez que o esforço computacional tenha sido realizado para satisfazer a prova de trabalho, alterar o bloco corresponde a alterar o *hash* final gerado, sendo necessário refazer todos as operações sobre o *nonce* e dados de cada bloco para gerar um novo *hash*. Sendo uma operação com custo computacional intenso, acaba por dificultar ataques a rede.

- *Proof of Stake* (PoS): O algoritmo PoS foi implementado pela rede Ethereum recentemente a fim de resolver alguns dos principais problemas causados pelo algoritmo PoW, ou seja, o custo computacional da validação das transações e o desperdício energético causado (ETHEREUM, 2023). Neste algoritmo os validadores da rede são escolhidos com base em sua participação na rede, ou seja, quanto maior a participação em *tokens* em uma rede pública como Ethereum, maiores são as chances de ser escolhido para propor e validar um bloco na rede.

- *Practical Byzantine Fault Tolerance* (PBFT): O algoritmo PBFT foi criado inicialmente para resolver o problema de consenso chamado problema dos generais bizantinos em sistemas distribuídos. Segundo (CASTRO; LISKOV et al., 1999), o algoritmo estipula um mínimo de $3f + 1$ (f sendo o número de nós maliciosos) nós que são necessários para manter o sistema confiável. O nível de tolerância do algoritmo é calculado por $f < n/3$, onde f é o número de nós bizantinos (maliciosos) e n o número total de nós da rede,

ou seja, é esperado que mais de $2/3$ da rede concordem que a informação é confiável. Por não utilizar-se de força computacional, o algoritmo ganha destaque principalmente em relação ao PoW. Porém, é necessário conhecer os nós da rede que são confiáveis, não sendo um método de consenso indicado para redes *Blockchain* públicas.

- Raft: O algoritmo se baseia em ser *crash fault-tolerant*, ou seja, pode tolerar nós que parem de funcionar na rede de forma repentina (FERDOUS et al., 2020). Raft se baseia em uma estratégia de eleger um nó líder, onde as transações serão replicadas a partir dele. Também é um método de consenso que possui baixo consumo de energia, já que não utiliza de poder computacional para calcular uma função de *hash*. Sendo assim, recomenda-se a utilização do método de consenso Raft em *Blockchains* privadas e de consórcio.

A Tabela 2 apresenta as principais diferenças entre os métodos de consenso previamente discutidos. A escolha do método de consenso ideal se baseia principalmente no modelo da rede a ser utilizada, seja pública, privada ou em consórcio. Após definido o método de consenso a ser utilizado, é necessário verificar quais os *frameworks* de desenvolvimento voltado para redes *Blockchain* estão disponíveis no mercado e que suprem os requisitos da rede. Neste trabalho, o *framework* GoQuorum supre as demandas da proposta de arquitetura, ter suporte a redes em modelo de consórcio e a opção de utilizar o método de consenso RAFT para validação dos blocos da rede *Blockchain*.

Tabela 2 – Tabela de comparação entre métodos de consenso.

Propriedade	**PoW**	**PoS**	**PBFT**	**Raft**
Modelo de rede	Pública	Pública	Privada/Consórcio	Privada/Consórcio
Consumo de energia	Alto	Baixo	Baixo	Baixo
Tolerância	Poder computacional	Participação	Nós maliciosos	Falhas
Exemplo	Bitcoin	Ethereum	GoQuorum	GoQuorum

Fonte: Elaborado pelo autor (2024).

2.2.3 CONTRATOS INTELIGENTES

De acordo com (ZHENG et al., 2018), um contrato inteligente ou *smart contract* do inglês, pode ser definido como um fragmento de código que pode ser executado automaticamente pelos nós validadores de transações. Nas plataformas que implementam sua *Blockchain* seguindo as especificações Ethereum, os contratos inteligentes são a peça fundamental para a geração de *tokens*, moedas e várias funções que agregam funcionalidades ao sistema. Segundo (KHAN et al., 2021), a plataforma Ethereum foi a primeira a dar suporte ao desenvolvimento de contratos inteligentes, utilizando-se de uma máquina virtual Turing-Completa (consegue ser descrita em termos de uma máquina de Turing (SIPSER, 2012)) chamada de Ethereum Virtual Machine (EVM)). EVM é o ambiente que executa todos os contratos inteligentes, implementada em cada nó da rede Ethereum.

Segundo (BUTERIN et al., 2014), Solidity foi a linguagem de alto nível escolhida para escrever contratos inteligentes na rede Ethereum, por se tratar de uma linguagem parecida em sintaxe a outras linguagens conhecidas e utilizadas no desenvolvimento de *software* para a web. A linguagem é apenas uma camada para o *bytecode* que irá executar na *Blockchain*.

2.2.4 *FRAMEWORKS*

A revisão de *frameworks Blockchain* realizado por (CAPOCASALE; GOTTA; PERBOLI, 2023), realizou-se com os framworks Hyperledger Fabric, Hyperledger Sawtooth, Hyperledger Besu e GoQuorum. Embora as ferramentas testadas estejam obsoletas em questão de ajustes de segurança e novas funcionalidades, conforme a data da última versão dos *frameworks*[1], ainda é possível utilizar dos resultados para escolher qual *framework* se encaixa melhor na necessidade do problema, baseando-se na análise de desempenho e no número de transações por segundo (tps).

Assim, conforme analisado por (CAPOCASALE; GOTTA; PERBOLI, 2023), pode-se observar que o Hyperledger Fabric demonstra eficiência, embora careça de um algoritmo de consenso BFT. O Hyperledger Sawtooth, por outro lado, oferece flexibilidade, embora sua eficiência possa não ser tão alta. Por fim, o *framework* Quorum destaca-se por seu desempenho sólido, a inclusão de um algoritmo de consenso BFT e o suporte a transações privadas utilizando-se de um módulo adicional. Uma das vantagens evidentes em relação à durabilidade e atualizações de segurança do *framework* Quorum é a adoção do GoQuorum, uma versão personalizada do cliente da *Blockchain* pública Ethereum, conhecido como Go-Ethereum (Geth). Este cliente é amplamente utilizado e mantido pela comunidade, contribuindo significativamente para a manutenção e segurança contínua do *framework* Quorum.

2.3 SIDECHAINS

Uma *sidechain* é uma rede secundária conectada à blockchain principal por meio de um mecanismo de bidirecional. Esta conexão permite que informações sejam transferidas bidirecionalmente entre a cadeia principal e a cadeia secundária. As *sidechains* podem ter seus próprios protocolos de consenso, que podem diferir daqueles da *Blockchain* (rede principal), permitindo-lhes adicionar novas funcionalidades, aumentar a privacidade e melhorar a segurança das *Blockchains* tradicionais.

Sidechains são usadas para enfrentar desafios em sistemas baseados em *Blockchain*, como escalabilidade, funcionalidade limitada e privacidade. Ao fornecer alternativas para ampliar as capacidades da *Blockchain* principal sem alterá-la diretamente, as *sidechains* oferecem uma solução para essas limitações. Dessa forma, elas permitem a inovação e a experimentação sem

[1] A versão do Hyperledger Sawtooth utilizada é de 2019 e os outros *frameworks* são versões com datas entre janeiro e fevereiro de 2021.

afetar a segurança e a estabilidade da cadeia principal, permitindo o desenvolvimento de novos recursos e aplicações (SINGH et al., 2020).

2.4 IPFS

O *InterPlanetary File System* (IPFS) tem gradualmente ganhado popularidade nos últimos anos. Essa crescente adoção se deve às suas características de armazenamento e acesso distribuído e descentralizado. O IPFS é um sistema de armazenamento modular *peer-to-peer* (P2P) que pode ser facilmente integrado com outras tecnologias, incluindo *Blockchain*. Uma das principais vantagens do IPFS em comparação a outros protocolos P2P está na integridade e persistência dos dados, que são mantidos através de uma estrutura de dados chamada *Merkle Directed Acyclic Graph* (Merkle DAG) (CRISTEA et al., 2020).

Segundo (BENET, 2014), IPFS é um sistema de arquivos distribuído que sintetiza ideias bem-sucedidas de sistemas *peer-to-peer* já consolidados, incluindo DHTs, BitTorrent, Git e *Self-Certified Filesystems* (SFS). A contribuição do IPFS é simplificar, evoluir e conectar técnicas comprovadas em um único sistema coeso, maior que a soma de suas partes. O IPFS apresenta uma nova plataforma para escrever e implantar aplicativos e um novo sistema para distribuição e versionamento de grandes volumes de dados.

Sendo um sistema P2P, os nós estabelecem conexão, descobrindo novos pares de nós e realizando as transferências de objeto entre eles. Os objetos representam arquivos e outras estruturas de dados, como, metadados dos arquivos e ligações para outros objetos na rede.

O IPFS pode ser dividido em partes, as quais possuem diferentes funcionalidades (BENET, 2014):

- **Identificação**: Nós da rede possuem identificação baseada em chave pública e privada;
- **Rede**: Gerencia a conexão com outros nós da rede;
- **Roteamento**: Mantém informações para localizar nós e objetos específicos. Responde a consultas locais e remotas;
- **Comutação/Troca**: Troca de blocos entre nós da rede, uma maneira de incentivar a persistência de dados;
- **Objetos**: Merkle DAG de CIDs, usado para representar estruturas de dados arbitrárias, como hierarquias de arquivos e sistemas de comunicação;
- **Arquivos**: Conjunto de objetos que modelam um sistema de versionamento usando Merkle DAG;
- **Nomenclatura (CID)**: *Hash* único gerado para cada arquivo inserido na rede, ou seja, se o arquivo for modificado outro *hash* precisa ser gerado.

2.4.1 MERKLE DAG

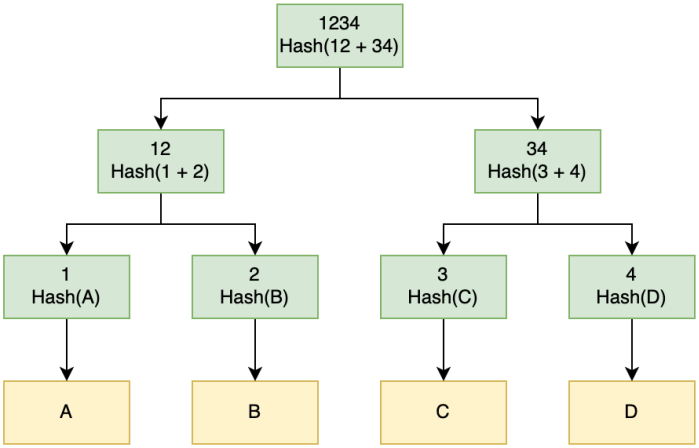
Merkle Directed Acyclic Graph (Merkle DAG) pode ser considerado como o coração do IPFS. Essa estrutura de dados é responsável por endereçar o conteúdo salvo na rede IPFS. (CRISTEA et al., 2020) define uma estrutura Merkle DAG como uma representação em árvore de objetos criptografados e suas referências no sistema. Como cada objeto possui seu *hash* único (CID) na rede, existe uma propriedade de imutabilidade entre as referências de objeto.

Segundo (BENET, 2014), a utilização dessa estrutura possibilita algumas propriedades interessantes em uma rede descentralizada, por exemplo:

- Endereçamento do conteúdo: Baseado em *hash*, cada arquivo é único no sistema;
- Resistência a fraude: Todo conteúdo pode ser verificado pelo seu *checksum*;
- *Deduplication*: Todos os objetos que possuem o mesmo conteúdo, possuem o mesmo *hash*, portanto o mesmo *hash* é armazenado apenas uma vez.

A Figura 3 apresenta um exemplo da estrutura Merkle DAG na sua configuração em árvore, onde o nó principal (*root*) guarda as informações dos nós filhos, mantendo uma cadeia de *hashes* até chegar no conteúdo (folhas da árvore). Sendo assim, caso o conteúdo do bloco A fosse modificado, seu *hash* também sofreria mudanças e não seria mais válido.

Figura 3 – Exemplo de uma estrutura do tipo Merkle DAG, na sua configuração em árvore.



Fonte: (COX, 2023)

2.4.2 IPFS CLUSTER

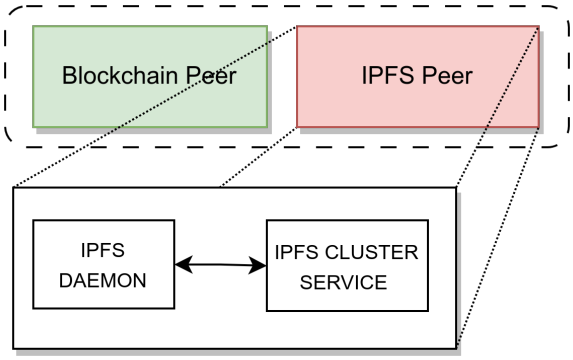
O IPFS Cluster fornece orquestração de dados em um grupo de *daemons* IPFS, alocando, replicando e rastreando um conjunto de conteúdos globalmente distribuído entre vários nós. Desta forma é possível configurar o comportamento dos arquivos salvos na rede, utilizando-se de configurações como número de réplicas minímas, máximo de conteúdos fixados localmente

em cada nó e outras configurações possíveis usando o módulo adicional IPFS Cluster junto ao *daemon* do IPFS. Na Figura 4 podemos observar a comunicação entre os módulos IPFS de um nó (IPFS Daemon e IPFS Cluster Service) e que também compõe a rede *Blockchain*.

Segundo (LABS, 2023), o sistema IPFS constitui-se de três principais módulos:

- ipfs-cluster-service: Gerencia a fixação de conteúdo localmente, usando fatores como replicação e coordenação de nós em um ipfs-cluster;
- ipfs-cluster-ctl: Ferramenta em linha de comando para auxiliar o administrador do *cluster* a gerenciar os conteúdos e coletar métricas;
- ipfs-cluster-follow: É uma versão mais básica do ipfs-cluster-service, permitindo apenas a replicação dos arquivos na rede, ou seja, não é possível editar ou modificar a forma como os conteúdos fixados localmente são gerenciados pelo ipfs-cluster-service. Ele é utilizado para permitir nós não confiáveis na rede, limitando suas configurações no ipfs-cluster.

Figura 4 – Módulos da camada IPFS



Fonte: Elaborado pelo autor (2024).

2.5 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo foram abordados os principais tópicos que envolvem o armazenamento de dados médicos, passando por uma revisão bibliografia sobre as principais regulamentações existentes no cenário atual. Também foi realizado um levantamento sobre as tecnologias e conceitos relacionados a *Blockchain*, sendo realizado um comparativo entre os diferentes modelos de redes e métodos de consenso disponíveis para redes públicas e em consórcio.

Nesse capítulo também decidiu-se acerca de qual método de consenso seria adequado utilizar, após feita uma análise dos requisitos da rede híbrida proposta. Ainda, após análise dos componentes da rede *Blockchain*, foi realizado uma pesquisa bibliográfica para conceituar o funcionamento da rede IPFS e seu módulo adicional IPFS *Cluster*.

3 TRABALHOS RELACIONADOS

O trabalho realizado por (AZBEG; OUCHETTO; Jai Andaloussi, 2022) contempla uma proposta de sistema chamado BlockMedCare, onde foi utilizado uma combinação de dispositivos médicos para Internet das Coisas (IoT), a *Blockchain* Ethereum na sua versão *permissioned* utilizando como consenso *Proof of Authority (PoA)* e o protocolo IPFS para armazenamento de dados. As principais características deste trabalho foram a implementação da rede principal e sua *sidechain*, a criação de uma arquitetura de interação entre IPFS (*Sidechain*) e a *Blockchain* e a especificação pública do *smart contract* utilizado pela *Blockchain*. Porém, a falta de testes na interação entre *sidechain* e a rede principal, deixou o trabalho sem resultados para perguntas como qual o impacto em tempos de busca e inserção ao armazenar arquivos de tamanhos variados na *sidechain* e como lidar com novas versões de arquivos salvos na *sidechain*.

(JAYABALAN; JEYANTHI, 2022) propõem uma arquitetura voltada ao paciente, usando *Blockchain* e IPFS para armazenar os dados médicos. O principal fator diferencial do trabalho é o uso de diferentes métodos de criptografia para o armazenamento dos dados na rede IPFS. São utilizados algoritmos de chave simétrica e assimétrica para garantir a integridade do arquivo na rede, bem como a restrição de acesso aos mesmos, usando o modelo de chave publica e privada. Por outro lado, o uso do método de consenso *Proof of Work* (PoW) restringe o uso da arquitetura em alguns cenários e também não foram realizados testes com arquivos maiores que 300KB. Segundo (DINOV, 2016) podemos ter dados médicos que ocupam megabytes e até mesmo gigabytes de espaço a depender do tipo de exame ou diagnostico escolhido. Portanto, neste cenário haveria uma limitação dos testes realizados não considerando o comportamento da rede IPFS ao armazenar e recuperar arquivos com tamanho superior a 300KB.

A arquitetura proposta por (KUMAR; MARCHANG; TRIPATHI, 2020), foca na persistência de dados médicos como diagnósticos produzidos por exames como tomografia computadorizada e radiografia. O armazenamento é realizado em uma *Blockchain* consorciada utilizando como consenso *Proof of Work (PoW)* e o protocolo IPFS como *sidechain*. Não foi especificado se o IPFS foi utilizado em sua versão pública ou utilizando nós privados, também não foram divulgados os números de nós da rede *Blockchain* utilizados nos testes. Por fim, os resultados foram baseados em testes realizados com vários tipos de arquivos partindo de 1MB até 128MB, analisando o tempo para *download* e *upload* dos arquivos, bem como tempo de execução para determinados números de nós no intervalo de 3,6,9,12,15 e 18 nós.

(NUNES; MA; FILHO, 2021) apresentam proposta para armazenamento usando *Blockchain* e IPFS, usando OpenPGP como meio de garantir a privacidade entre os arquivos da rede IPFS. Não foi especificado o modelo de consenso a ser utilizado na *Blockchain*, o número de nós do IPFS e da rede principal, bem como seria realizada a interação e a persistência das chaves por meio de contratos inteligentes.

(ALI; DOLUI; ANTONELLI, 2017) realizam uma proposta voltada para Internet das Coisas, utilizando-se da rede *Blockchain*, utilizando como mecanismo de consenso o algoritmo

Proof of Stake (PoS) e da *sidechain* com IPFS para armazenar dados gerados por sensores e atuadores. Foi realizado um comparativo entre os *frameworks* Monax (*Proof of Stake*) e Ethereum (*Proof of Work*) em um ambiente de estresse voltado as aplicações IoT, no qual o método de consenso mais adequado para as aplicações IoT foi o *Proof of Stake* (PoS). Foram realizados vários testes analisando o *overhead* sobre as redes *Blockchain*, e até mesmo o tempo de processamento de cada bloco inserido na cadeia. Por outro lado, faltaram métricas sobre a rede IPFS, não considerando os tempos de inserção e recuperação de arquivos na rede IPFS.

O trabalho de (GOMES; COUTINHO, 2022), apresenta uma proposta de armazenamento em *Blockchain* utilizando-se da plataforma Hyperledger Fabric e do algoritmo de consenso Raft, diferente de outros trabalhos que utilizam *Proof of Stake* ou *Proof of Work* como método de consenso. O trabalho possui uma arquitetura completa para o monitoramento de dispositivos IoT em um ambiente hospitalar, desde a coleta até a visualização dos dados em um *dashboard*. Os testes com múltiplos dispositivos em horas do dia diferentes mostrou que a quantidade de transações suportados pela *Blockchain* pode ser um fator limitante, adicionando uma maior latência conforme aumenta o número de dispositivos na rede. Por fim, a utilização de diagramas e recursos visuais, bem como o grande acervo de trabalhos relacionados deixa o trabalho bem definido, ajudando a entender como cada parte foi implementada.

Na Tabela 3 estão agrupadas as principais características dos trabalhos relacionados, analisadas com base nos tópicos abaixo. Os campos contendo o caractere "-" definem que não houve informações suficientes no trabalho analisado a fim de realizar o comparativo.

Lista de itens analisados nos trabalhos relacionados:

1. Plataforma utilizada para criar a *Blockchain*;
2. Método de consenso utilizado;
3. Tipo de *Blockchain* (*Permissioned* ou *Permissionless*);
4. Se possui armazenamento em *sidechain*;
5. Se foi ou não realizada implementação da proposta;
6. Se foi ou não realizada análise de desempenho da rede proposta;
7. Se foi ou não utilizado módulo IPFS *Cluster* para replicação de arquivos.

A Tabela 3 apresenta o comparativo dos trabalhos avaliados. Com base nos artigos avaliados, é perceptível que existe uma falta de informações sobre as arquiteturas propostas e análise de desempenho da rede híbrida. Além disso, muitos artigos não explicaram qual modelo de rede foi utilizado para a *sidechain* IPFS, sendo modelo público, privado ou em consórcio. Neste sentido, este trabalho propõe uma arquitetura baseada no modelo de consórcio, utilizando-se também do módulo de replicação IPFS *Cluster* para garantir a disponibilidade dos arquivos na rede *sidechain*.

Tabela 3 – Comparativo entre os trabalhos relacionados.

Trabalho	**1**	**2**	**3**	**4**	**5**	**6**	**7**
(AZBEG; OUCHETTO; Jai Andaloussi, 2022)	Ethereum	PoA	Permissioned	IPFS	✓	-	-
(KUMAR; MARCHANG; TRIPATHI, 2020)	-	PoW	Permissioned	IPFS	✓	✓	-
(NUNES; MA; FILHO, 2021)	-	-	-	IPFS	-	-	-
(ALI; DOLUI; ANTONELLI, 2017)	Tendermint	PoS	Permissioned	IPFS	✓	✓	-
(GOMES; COUTINHO, 2022)	Hyperledger	Raft	Permissioned	-	✓	✓	-
(JAYABALAN; JEYANTHI, 2022)	-	PoW	-	IPFS	✓	✓	-
Esta proposta	GoQuorum	PoA (RAFT)	Permissioned	IPFS	✓	✓	✓

Fonte: Elaborado pelo autor (2024).

3.1 CONSIDERAÇÕES DO CAPÍTULO

Neste Capítulo foi apresentada a revisão bibliografia dos trabalhos atuais que utilizam de *Blockchain* e *sidechain* para armazenamento de arquivos de grande volume, buscando diferenciar as principais características e aplicações de cada arquitetura proposta, bem como seus pontos positivos e negativos em relação ao uso da tecnologia IPFS.

4 PROPOSTA

Como já mencionado, neste trabalho é proposta uma arquitetura, implementação e análise de desempenho da rede IPFS como *sidechain* para armazenamento de dados médicos de grande volume em conjunto com uma *Blockchain*. Primeiramente são definidos os requisitos do sistema e quais tecnologias e *frameworks* serão utilizados com base nos requisitos levantados. A seguir, é implementada uma prova de conceito para coletar métricas de desempenho para balizamento de comparação, seguindo o plano de testes proposto.

4.1 ATORES PARTICIPANTES

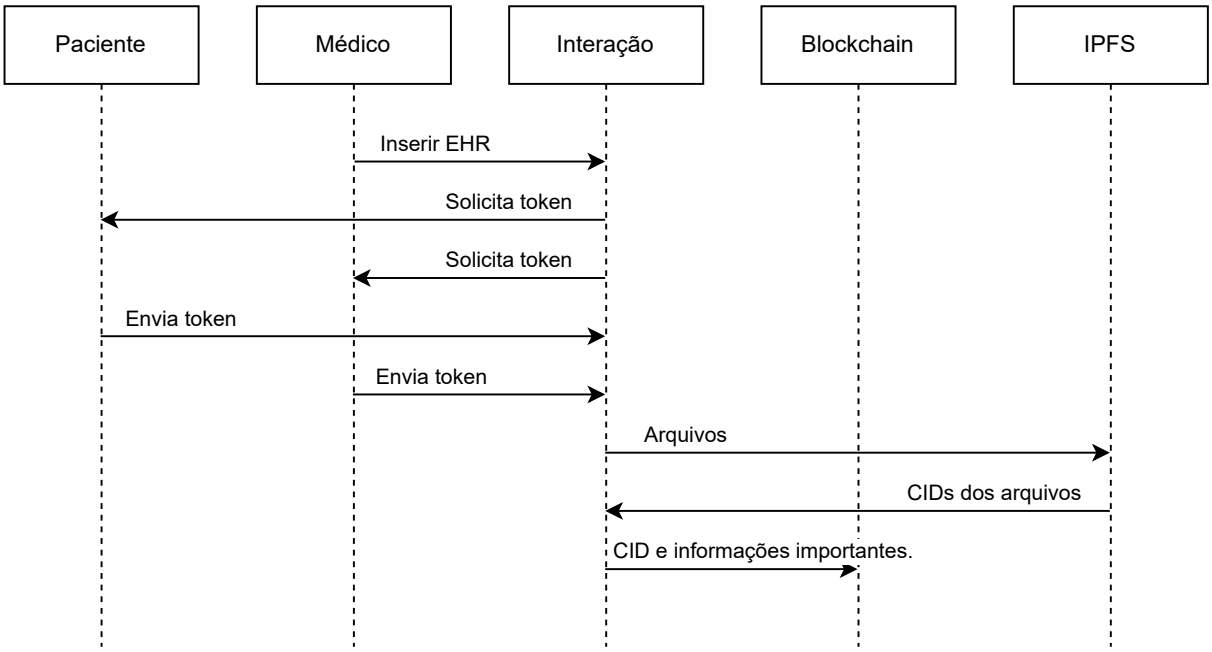
Diferentes atores participantes interagem com a rede *Blockchain* e *sidechain* para armazenamento de dados médicos de grande volume, conforme proposto nesse trabalho. Atores diferentes executam ações diferentes e possuem diferentes poderes de escrita e leitura na *Blockchain* e na *sidechain*. Assim, temos os seguintes atores:

- Instituições médicas: São responsáveis por manter o funcionamento de validadores e nós da rede;
- Médicos: Incluem relatórios gerados pelo EHR e seus arquivos gerados na consulta, por exemplo, arquivos gerados por uma tomografia computacional. Também podem acessar os dados do paciente para visualizar seu histórico de consultas e arquivos relacionados;
- Pacientes: Não interagem diretamente com o sistema, porém participam no processo de envio e recuperação com suas credenciais (tokens).

Um médico é responsável por avaliar o paciente e inserir os dados resultantes da consulta na rede. A Figura 5 demonstra o diagrama de sequência e a troca de mensagens entre os sistemas para o caso de uma inserção de dados médicos. Primeiramente são necessários os *tokens* de identificação do médico e paciente. Após coletados os *tokens*, é necessário utilizar a camada de interação, a qual é responsável por comunicar-se com a camada de armazenamento. Na camada de interação serão primeiramente inseridos os arquivos de grande volume na rede IPFS. Caso o trâmite de transferência ocorra da forma esperada, o IPFS retornará o CID dos arquivos salvos. Após a obtenção dos CIDs, a camada de interação irá se comunicar com os contratos inteligentes da *blockchain* para salvar as informações gerais (EHR) associadas aos CIDs dos arquivos que estão na rede IPFS, utilizando os dois *tokens* de identificação. O médico também pode recuperar informações do paciente, utilizando o *token* do paciente.

O paciente pode recuperar seu histórico de consultas através da camada de interação. Seguindo o diagrama de sequência da Figura 6, temos algo parecido com a interação de inserção de arquivos. Porém, agora a camada de interação realizará a chamada de API para a *blockchain* buscando executar o contrato inteligente. O contrato poderá retornar uma ou mais transações

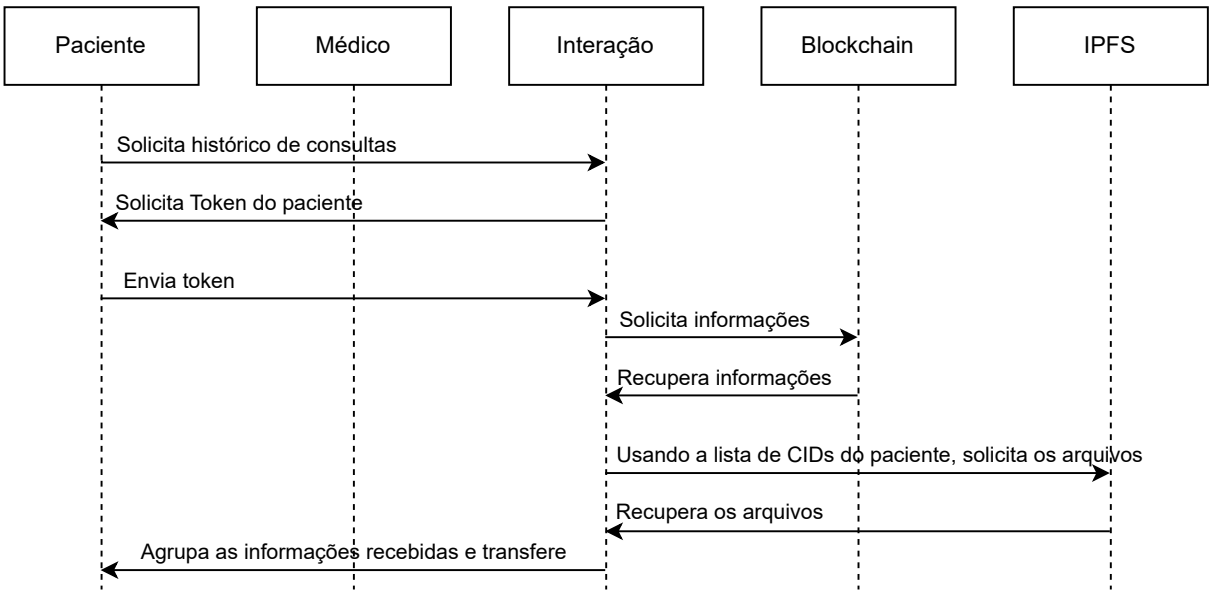
Figura 5 – Diagrama de interação entre médicos e a rede.



Fonte: Elaborado pelo autor (2024).

que estão salvas na *Blockchain* e que pertencem ao *token* do paciente que foi informado. Após recuperadas as informações da *Blockchain*, a camada de interação irá buscar os arquivos na rede IPFS. Para cada CID listado pelas transações recuperadas, será feito uma busca na rede para identificar quais nós possuem o arquivo e realizar posteriormente o *dowload* do mesmo. Após todos os arquivos necessários serem recuperados, a camada de interação fica responsável por disponibilizar as informações das consultas e seus arquivos correspondentes.

Figura 6 – Diagrama de interação entre paciente e a rede.

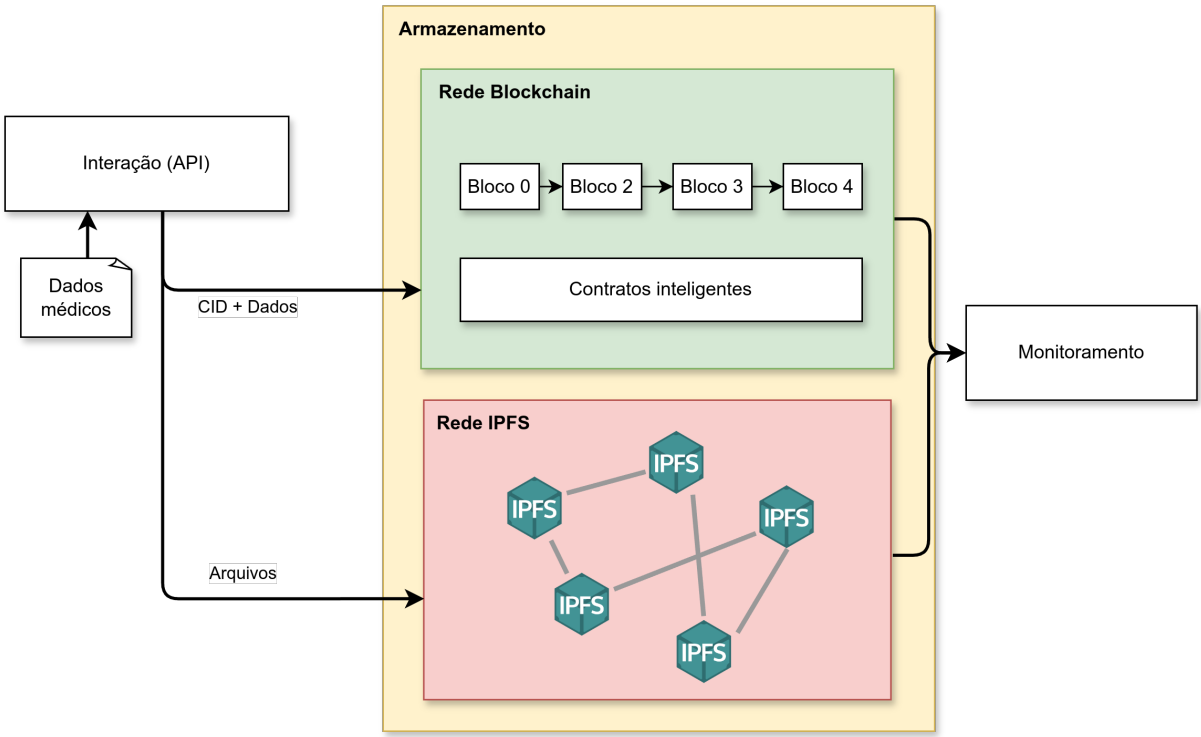


Fonte: Elaborado pelo autor (2024).

4.2 ARQUITETURA

A arquitetura de avaliação de desempenho de *sidechain* em rede *Blockchain* para grandes volumes de dados na área da saúde está dividida em camadas, facilitando a modularização caso seja necessário mudar tecnologias ou fazer ajustes. As três principais camadas do sistema, serão definidas conforme a Figura 7, sendo elas a camada de interação, armazenamento e monitoramento. Após a criação das duas camadas principais, será adicionada uma camada de monitoramento, para obter métricas sobre a rede de armazenamento, a fim de monitorar o comportamento da *sidechain* com IPFS e da rede *Blockchain*. Uma outra visão da interação entre as camadas, incluindo a camada de monitoramento e interação, pode ser observada na Figura 8.

Figura 7 – Abordagem inicial da arquitetura proposta.



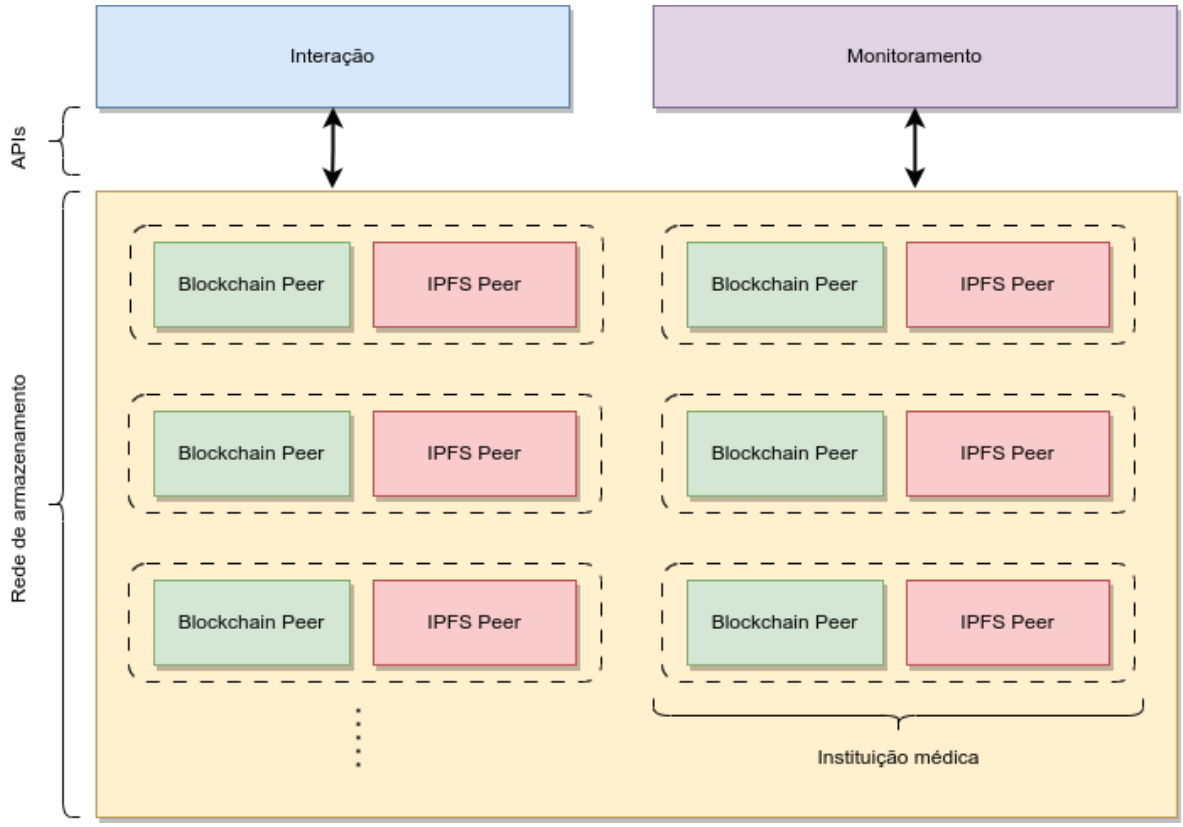
Fonte: Elaborado pelo autor (2024).

A Figura 7 exemplifica o fluxo das informações entre as camadas de forma mais genérica que os diagramas de sequência propostos pelas Figuras 5 e 6. Contudo, as interações entre as camadas da arquitetura seguem as interações apresentadas conforme os atores as utilizam. Em especial, a camada de interação é onde os pacientes, instituições médicas e os próprios médicos têm acesso à entrada e busca de dados por meio da *Blockchain* e da *sidechain* de maneira indireta. A camada de armazenamento representa uma rede unificada, porém cada instituição participante da rede deve manter nós IPFS e da rede *Blockchain*. A camada de monitoramento será responsável pela observação das métricas, integrada diretamente à camada de armazenamento, junto às redes *Blockchain* e IPFS.

Na Figura 8, temos a camada de armazenamento de forma explícita. Nela, cada instituição

médica que faz parte da rede manterá um conjunto de nós *Blockchain* e IPFS. Assim, cada par de nós será monitorado por uma camada externa. Ambas as camadas de interação e monitoramento são ligadas a cada instituição médica. Sendo assim, é feita a troca de informações via APIs, seja a interação com os contratos inteligentes e a rede IPFS, ou o monitoramento utilizando as ferramentas Prometheus e Jaeger.

Figura 8 – Camadas do sistema



Fonte: Elaborado pelo autor (2024).

Dentro da camada de armazenamento teremos um conjunto de instituições que serão responsáveis pela manutenção e execução da rede *Blockchain* e IPFS. Cada instituição deve manter ao menos um nó da *Blockchain* e um nó IPFS na forma apresentada pela Figura 4.

4.2.1 BLOCKCHAIN

Para a rede *Blockchain*, a escolha do método de consenso foi baseada principalmente pelo modelo da rede. Neste caso, conhece-se todos os nós da rede e a adição de novas integrantes é controlada, resultando em uma rede em consórcio entre instituições médicas.

Visando atingir altas capacidades de transações por segundo (tps) e economia de recursos computacionais, a utilização de um método de consenso adequado que proporcione tais características se faz necessário. No contexto de uso de uma rede em consórcio, a utilização de um método de consenso como RAFT agrega características como tolerância a nós falhos na rede, baixo consumo de energia e uma alta taxa de transações por segundo.

Portanto, a escolha do *framework* quorum se encaixa dentre os seguintes requisitos: suportar uma rede em consórcio e suportar métodos de consenso baseados em RAFT. Segundo (CAPOCASALE; GOTTA; PERBOLI, 2023), o *framework* escolhido se mostrou eficiente em relação a tps e ao uso de recursos computacionais. O *framework* quorum também possui integração com ferramentas de monitoramento, como o Prometheus, facilitando a obtenção de métricas sobre a rede.

4.2.2 IPFS

Cada nó IPFS será composto de dois serviços internamente, visto que somente o IPFS *daemon* é responsável pela rede IPFS, ou seja, nomenclaturas (CIDs), transmissão de blocos e arquivos pela rede. Porém, para gerenciar como os arquivos serão salvos, é necessário um módulo adicional, o IPFS-Cluster. Ele será responsável por replicar os arquivos pela rede e gerenciar o escalonamento da mesma. Inicialmente, será necessário usar um fator de réplica miníma de três arquivos na rede, ou seja, o IPFS-Cluster será responsável por alocar esses arquivos em ao menos três nós da rede. O fator de ao menos três arquivos foi escolhido para ser a base de testes de forma empírica, visto que se um dos 3 nós que possuem o arquivo falharem, ainda restam ao menos dois nós.

A comunicação entre os serviços IPFS *daemon* e IPFS-Cluster acontece através da disponibilização de informações do nó a partir de uma porta HTTP (5001), utilizando uma API que disponibiliza os dados para outros serviços utilizarem. Neste caso, o IPFS-Cluster irá consumir as informações do ipfs *daemon* para monitorar a rede e usar os dados como base para gerenciar e replicar os arquivos.

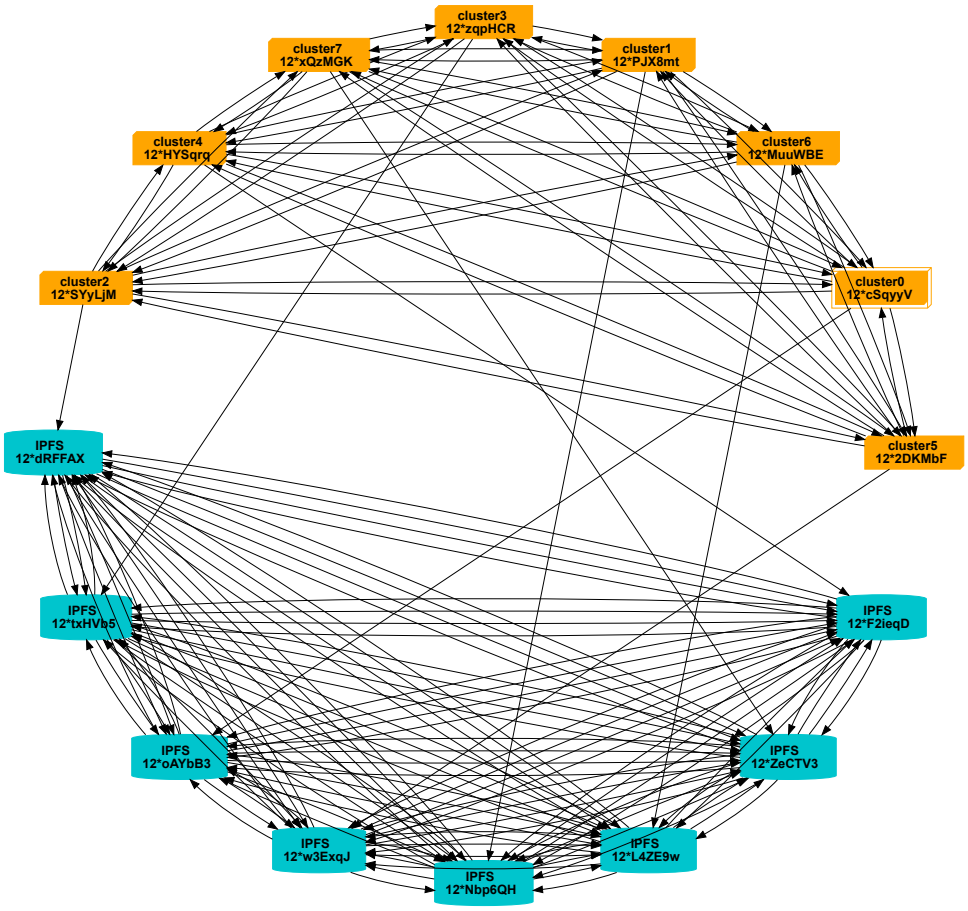
A Figura 9 expõe as conexões entre nós IPFS *daemon* marcados com o nome IPFS (parte inferior do grafo) e os nós IPFS *Cluster* com nome cluster (parte superior). Na Figura 9 pode-se observar que cada nó IPFS *cluster* é ligado a um nó IPFS *daemon*. Porém, todos os nós IPFS *cluster* e IPFS *daemon* possuem ligações entre seus próprios pares. Por meio destas, ocorre a troca de mensagens para balanceamento de arquivos e outras trocas de informações. Este cenário representa o pior caso de replicação de arquivos no sistema, em termos de uso de recursos, especialmente tráfego de rede e armazenamento de dados, onde todos possuem uma cópia do arquivo. Outras configurações podem ser aplicadas para manter a disponibilidade do arquivo sem sobrecarregar o armazenamento de todos os nós.

4.2.3 INTERAÇÃO

A camada de interação será responsável por intermediar a conexão entre a *Blockchain* e a rede IPFS na camada de armazenamento. Para acessar os dados presentes na rede *Blockchain* e IPFS, será necessário fazer a utilização de APIs disponibilizadas por ambas as redes.

A camada de interação proposta neste trabalho, foi desenvolvida baseada no modelo de API REST, disponibilizando assim os *endpoints* para envio e recuperação de arquivos.

Figura 9 – Grafo exibindo as conexões entre IPFS *daemon* e IPFS *Cluster* em uma rede com 8 nós.



Fonte: Elaborado pelo autor (2024).

Neste cenário de experimentação não foi utilizada nenhuma verificação de usuário, apenas o funcionamento básico da aplicação com os métodos GET e POST.

Envio de arquivos e informações através da camada de interação foram efetuados da seguinte forma:

1. Médico realiza a requisição HTTP utilizando método POST para o *endpoint* "/api/v1/record", passando as informações de endereço (*Blockchain*) do paciente, médico e da instituição através do corpo da requisição junto ao arquivo. Para enviar arquivo é necessário adicionar ao cabeçalho da requisição o parâmetro "Content-Type : multipart/form-data";

2. Envio dos arquivos de grande volume para o módulo IPFS *Daemon*, sendo retornado seu CID após inserção com sucesso;

3. Com o CID do arquivo é realizada uma requisição HTTP POST para o módulo IPFS *Cluster*, enviando apenas o CID do arquivo enviado anteriormente. Esse procedimento sinaliza para o módulo IPFS *Cluster* que o arquivo com determinado CID está pronto para replicação;

4. Por fim, será realizada a requisição para a rede *Blockchain*, enviando o CID junto aos endereços do paciente, médico e instituição;

5. Será retornado para o médico apenas o id da requisição, sendo este necessário para recuperação da transação e os arquivos.

A recuperação de arquivos possui um fluxo mais simples se comparado ao envio, bastando ter o id da transação realizada (envio) e seguir os passos:

1. Realizar a requisição HTTP GET para o *endpoint* "/api/v1/record/id", passando o id como um parâmetro na URL;

2. A camada de interação irá realizar as requisições para a rede *Blockchain* afim de recuperar a transação e seus dados;

3. Com os dados da transação é possível recuperar o arquivo na rede IPFS com o CID salvo na *Blockchain*;

4. Será retornado os dados da transação (endereços, CID e data de realização da transação) junto ao arquivo.

4.2.4 MONITORAMENTO

Para realizar a coleta e observação do comportamento de ambas as redes, será utilizada a ferramenta de monitoramento Prometheus. O motivo da escolha pela ferramenta é a fácil integração, sendo suportado pelo *framework* goquorum e IPFS. Por seu modelo de código aberto e ser mantida pela *Native Computing Foundation* (PROMETHEUS, 2012), se torna uma ferramenta de monitoramento sólida e confiável. Também será utilizada a ferramenta Jaeger, a qual também é mantida pela *Native Computing Foundation* (JAEGER, 2023). Jaeger será responsável por monitorar todo o fluxo de requisições na rede IPFS, ou seja, ao inserir um arquivo na rede teremos todo o rastro de solicitações entre nós e requisições feitas pela rede. Apenas a rede IPFS possui integração para coleta do fluxo de requisições usando a ferramenta Jaeger.

Com a camada de monitoramento pode-se coletar métricas da camada de armazenamento. Alguns dados importantes a serem coletados podem ser descritos como:

- **Blockchain**
  - Tempo de processamento de cada bloco;
  - Tamanho da *Blockchain*;
  - Métricas sobre uso de CPU, memória e disco por nó.
- **IPFS (Daemon e Cluster)**

- Métricas gerais sobre os conteúdos fixados por cada nó, por exemplo, fila de espera e erros;
- Tamanho do armazenamento utilizado por nó;
- Métricas sobre uso de CPU, memória e disco por nó.

4.3 PLANO DE TESTES

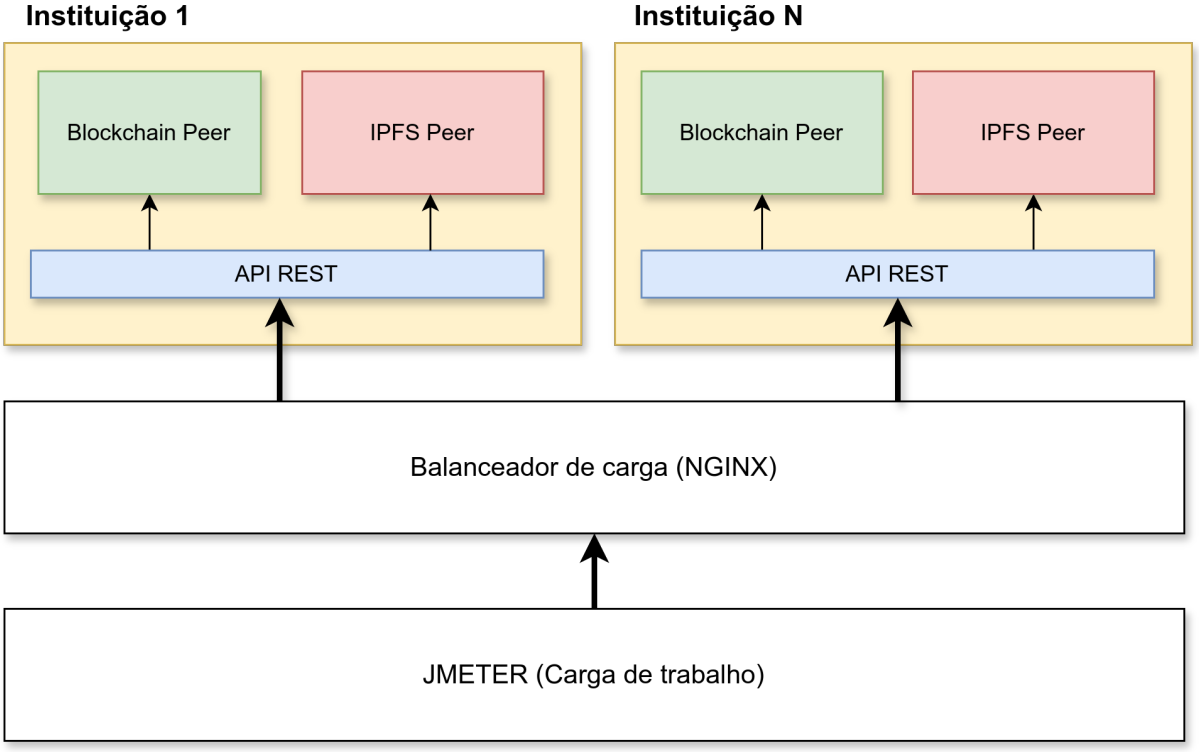
Por fins de regulamentação e ética, neste trabalho serão utilizados apenas dados fictícios gerados randomicamente sem nenhuma conexão com pacientes ou médicos. Portanto, para estipularmos o tamanho dos dados médicos de forma geral, será utilizado o artigo (VARMA, 2008), o qual reúne um conjunto de informações a respeito de consultas provenientes de equipamentos médicos.

Para realizar os testes foi inserido um balanceador de carga antes da camada de interação, conforme a Figura 10. Essa adição à arquitetura foi necessária para distribuir a carga de trabalho realizada nos experimentos, provocando uma sobrecarga sobre todos os nós da rede.

A análise do conjunto de interação entre *Blockchain* e IPFS será realizada observando os seguintes aspectos:

- Latências de envio e recuperação de arquivos;

Figura 10 – Arquitetura utilizada para realizar os experimentos



Fonte: Elaborado pelo autor (2024).

- Transações por segundo;
- Utilização de recursos computacionais (CPU, memória RAM e alguns casos utilização de disco).

A avaliação de desempenho será dividida em cenários de estudo, onde em cada cenário será gerada uma carga de trabalho (arquivos) específica baseada nos requisitos do experimento. Serão avaliados os seguintes cenários: Armazenamento de arquivos somente em *Blockchain (onchain)* versus armazenamento híbrido em *sidechain*; Sobrecarga de arquivos em *Blockchain (onchain)* versus armazenamento híbrido em *sidechain*; Armazenamento híbrido em *sidechain* com diferentes tamanhos de arquivos; e Mudança no fator de replicação do módulo IPFS *Cluster* para armazenamento híbrido em *sidechain*.

Nesse sentido serão executados vários experimentos. O primeiro experimento será uma avaliação entre as redes *Blockchain* e *sidechain* (armazenamento híbrido). Serão enviados e recuperados uma amostra de 1.000 arquivos de tamanho igual a 63KB de forma sequencial. O tamanho escolhido do arquivo deve-se a limitações do tamanho máximo de uma transação na rede *Blockchain* (128KB no total, contando os dados da transação e os arquivos). O propósito deste experimento é estressar ambas as redes em um cenário típico de utilização e comparar as métricas coletadas, realizando um teste de significância entre o número de nós em ambas as redes e explorando os dados obtidos observando o comportamento delas (possíveis requisições que falharam). Para o cenário de sobrecarga (alta demanda), foi realizada a inserção de 50.000 arquivos diferentes na rede, buscando explorar o limite do armazenamento na rede *Blockchain (onchain)* e comparar com a proposta deste trabalho (modelo híbrido).

O experimento seguinte será uma avaliação entre os diferentes tamanhos de arquivos (1MB, 10MB e 100MB) inseridos e recuperados na rede com armazenamento híbrido (*sidechain*). O experimento será realizado com 100 amostras, número inferior ao experimento anterior devido ao espaço de armazenamento limitado no servidor de testes. Este experimento tem o propósito de comparar o comportamento da rede com um intervalo de tamanho de arquivos distintos, observando as latências, transações por segundo (tps) e uso de recursos computacionais. Ao final do experimento é avaliado o impacto na variação em tamanho de arquivos e seu impacto na rede.

Por fim, o último experimento avaliará diferentes configurações para o fator de replicação miníma e máxima do módulo IPFS *Cluster* na rede de armazenamento híbrida (*sidechain*), utilizando-se da Equação 1 para determinar o fator de replicação através do número de nós da rede. Neste experimento será utilizada uma amostra de 1.000 arquivos com tamanho igual a 63KB, a fim de comparar com o primeiro cenário de experimentos. Portanto, este experimento tem como propósito comparar o fator de replicação entre nós, avaliando as métricas coletas para realizar o comparativo com a rede *sidechain* em replicação total de arquivos (todos os nós da rede possuem o arquivo). Ao final do experimento é esperado obter uma diferença estatisticamente significativa nas métricas relacionadas ao uso de recursos computacionais (CPU, RAM) e nas métricas relacionadas às requisições (latências e tps).

4.4 CONSIDERAÇÕES DO CAPÍTULO

Neste Capítulo foram definidas as camadas da arquitetura proposta, suas configurações e como elas se conectam entre si, realizando o armazenamento de arquivos de forma híbrida, utilizando *Blockchain* e *sidechain* com o protocolo IPFS. Também foram definidas as especificações sobre o envio e recuperação de arquivos na rede proposta.

Após realizado o levantamento das configurações sobre a arquitetura proposta, foram levantados os principais cenários de avaliação de desempenho. Os experimentos englobam diferentes características, configurações e tamanhos de arquivos diferentes, buscando realizar um comparativo entre a utilização de uma rede de armazenamento somente em *Blockchain* em relação à utilização de uma arquitetura híbrida (*sidechain*). Também foram realizados experimentos somente na rede de armazenamento híbrida, considerando aspectos como fator de replicação para o módulo IPFS *Cluster* e tamanho de arquivos diferentes.

Por fim, espera-se que na avaliação de desempenho da arquitetura para armazenamento híbrido, esta tenha menor latência, mais transações por segundo e menor consumo de recursos computacionais, ao comparar o modelo de armazenamento de arquivos somente na rede *Blockchain (onchain)*.

5 EXPERIMENTOS

Este capítulo apresenta os experimentos realizados e resultados obtidos, tendo em vista as métricas de monitoramento e plano de testes apresentadas no Capítulo 4. A Seção 5.1 descreve as configurações e ferramentas utilizadas nos experimentos. A Seção 5.2 compara o desempenho de envio e recuperação de arquivos entre as redes *onchain* e *sidechain*. A Seção 5.3 compara o desempenho da rede *sidechain* em diferentes volumes de arquivos. A Seção 5.4 compara o desempenho da rede *sidechain* utilizando diferentes fatores de replicação. Por fim, a Seção 5.5 traz uma discussão geral dos resultados obtidos nos experimentos.

5.1 DESCRIÇÃO DOS EXPERIMENTOS

Os experimentos foram realizados utilizando o modelo de arquitetura proposto no Capítulo 4, com o adicional de um balanceador de carga antes da camada de interação das instituições médicas. Desta forma é possível distribuir a carga de trabalho para envio e recuperação de arquivos de forma homogênea e estressar o sistema globalmente. A ferramenta escolhida como balanceador de carga foi o NGINX[1], utilizando sua configuração padrão de escalonamento junto as seguintes configurações: worker_connections = 8196 e worker_processes = auto.

Para envio e recuperação das cargas de trabalho foi utilizada a ferramenta de *benchmark* HTTP JMETER[2]. As configurações utilizadas variam conforme o cenário de teste, porém algumas configurações permaneceram inalteradas. As seguintes configurações não se alteram conforme o cenário: “Number of Threads” = 1 e “Ramp-up Period (seconds)” = 100. Além disso, todos os experimentos foram feitos sequencialmente, isto é, cada requisição enviada para o balanceador de carga envia seu arquivo e recebe seu código de resposta, continuando até finalizar todas as iterações. Outras configurações foram mantidas com seus valores de forma padrão da ferramenta.

Para evitar o consumo excessivo de CPU e disco durante a execução das requisições pela ferramenta JMETER, os arquivos utilizados como carga de trabalho foram gerados previamente, antes da execução de todos os cenários de teste. Por exemplo, 1.000 arquivos de 63KB foram gerados antecipadamente usando a ferramenta DD, que copia bytes aleatórios do diretório “/dev/urandom” para os arquivos usados para os experimentos.

Os experimentos sobre a rede *Blockchain (onchain)* e *sidechain* foram realizados em um mesmo servidor, utilizando-se de contêineres Docker para emulação de vários nós da rede, e automatizando o processo de configuração destes utilizando a ferramenta Docker Compose. A configuração da máquina utilizada no experimento é apresentada pela Tabela 4. Cada contêiner possui total acesso aos recursos da máquina, não ficando limitado a um determinado valor de CPU/RAM.

[1] A versão do NGINX utilizada foi a 1.27.0
[2] A versão do JMETER utilizada foi a 5.6.0

Para agrupar as métricas de CPU e memória RAM foram capturadas os dados de todos os nós da rede e calculada a média de consumo no período de realização do *benchmark*. Desta forma obtém-se o uso médio de recurso computacional por nó. Para o cenário em *sidechain*, foi necessário somar o uso de recursos computacionais dos módulos IPFS (*daemon* e *cluster*) junto ao consumo da *Blockchain* por nó, e depois foi calculada a média aritmética no período de realização do *benchmark*. As métricas de utilização de memória RAM estão padronizadas em megabytes (MB), em CPU as métricas estão em porcentagem (%). A ferramenta Prometheus coleta as métricas de CPU observando seu uso por núcleo, ou seja, quando se observa 300% de utilização de CPU, significa que aquele determinado nó da rede está usando 3 threads no total.

Tabela 4 – Configuração da máquina de experimentos.

Tipo	**Configuração**
CPU	Intel(R) Xeon(R) Silver 4216 (64 threads total)
RAM	96 GB DDR4
Armazenamento	930 GB SSD

Fonte: Elaborado pelo autor (2024).

5.2 COMPARAÇÃO ENTRE ARMAZENAMENTO *ONCHAIN* E *SIDECHAIN*

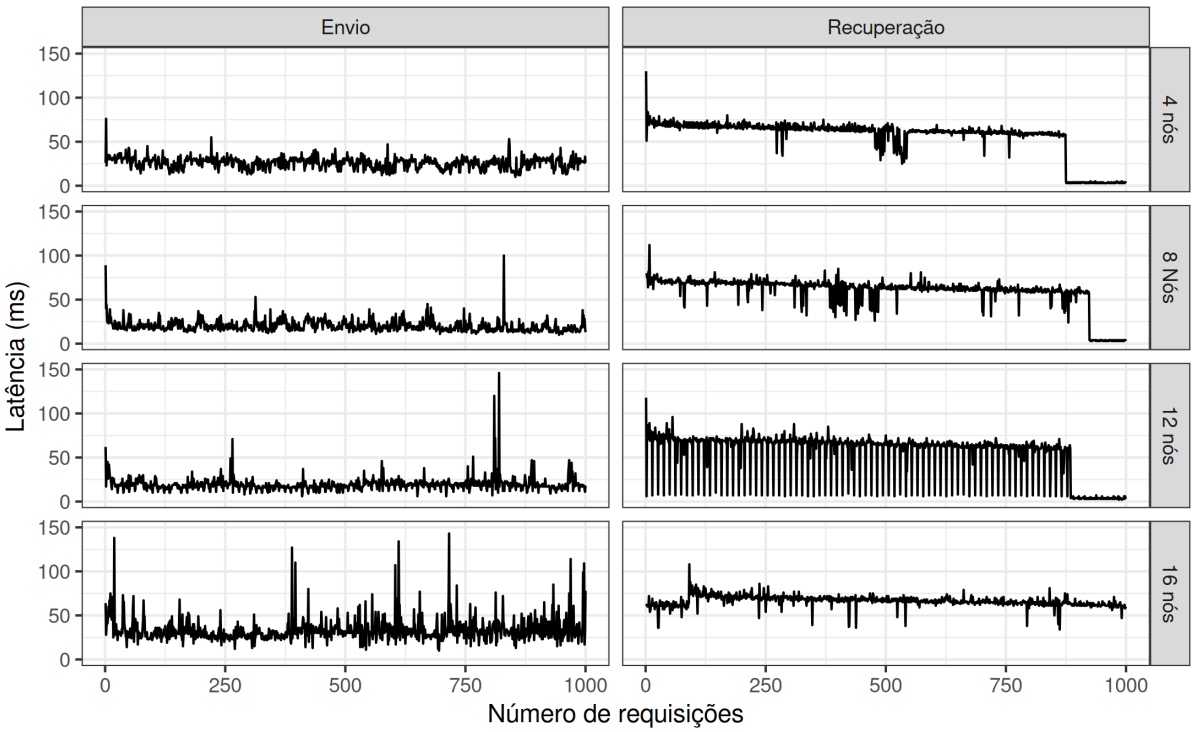
Neste cenário foram gerados 1.000 arquivos de tamanho igual a 63KB, utilizando a ferramenta DD. Este tamanho foi propositalmente selecionado por ser o tamanho máximo que a *Blockchain* suporta armazenar por transação. Sendo assim, temos uma base comum de comparação sobre latência das requisições e o uso de recursos computacionais de cada rede.

Foram realizados testes utilizando 4, 8, 12 e 16 nós para a rede *Blockchain (onchain)* e para o armazenamento hibrido *sidechain*, o qual agrega *Blockchain* + IPFS (*Daemon e Cluster*). Porém, os dados referentes ao cenário de 16 nós em *sidechain* foram corrompidos na escrita devido a algum fator externo, e por este motivo as comparações entre *onchain* e *sidechain* terão apenas 4, 8 e 12 nós como referência.

As Figuras 11, 12, 13, 14, 15 e 16 apresentam os resultados obtidos. Na Figura 11, pode-se observar a latência de cada requisição de envio e recuperação de arquivos na rede *Blockchain (onchain)*. Através da análise exploratória, percebe-se que para envios há um padrão das latências, aumentando os casos de picos máximos com o aumento do número de nós na rede. Porém, nos casos de recuperação nota-se um comportamento inconsistente. Isso se deve ao fato de as recuperações serem mal sucessivas, ou seja, retornam um erro (com código HTTP 500 ou 304) através da API na camada de interação. Este fato acaba por deixar os tempos de recuperação extremamente baixos especialmente ao final do experimento nos cenários envolvendo 4, 8 e 12 nós para a rede *blockchain*.

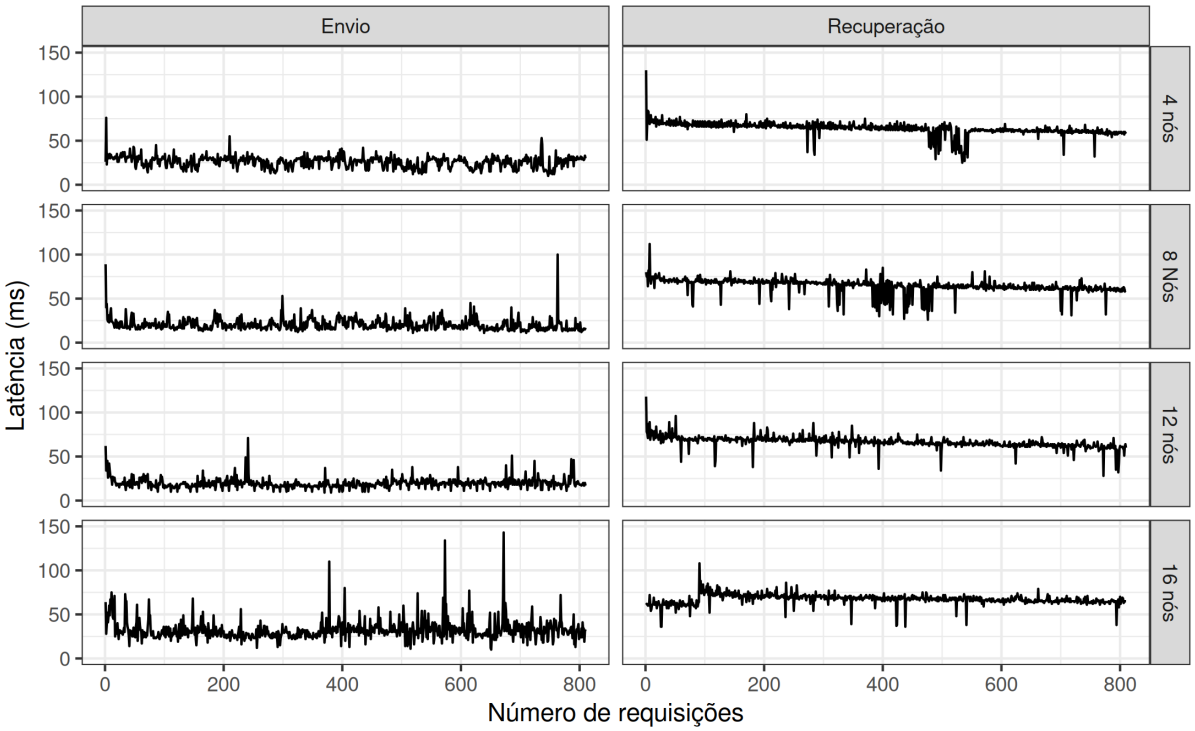
Na Figura 12 pode-se observar as latências das requisições para a camada de interação sem as transações que retornaram erro. Para os testes e análises a seguir serão consideradas apenas as transações bem sucedidas, evitando comprometer os dados.

Figura 11 – Latências de recuperação e envio de arquivos para a rede *Blockchain*.



Fonte: Elaborado pelo autor (2024).

Figura 12 – Latências após removidas as transações mal sucedidas.



Fonte: Elaborado pelo autor (2024).

Tabela 5 – Resumo da análise ANOVA e métrica de latência para o cenário *onchain* variando o número de nós da rede.

		Envio				**Recuperação**			
	Métrica	**Nós da rede**							
		4	**8**	**12**	**16**	**4**	**8**	**12**	**16**
Latência (ms)	Média	26	19.9	19.5	32.6	63.2	63.6	66.3	66.4
	C.V.	0.24	0.33	0.29	0.35	0.11	0.13	0.09	0.09
ANOVA	$F_{calculado}$	552.1				53.1			
	$F_{critico}$	3.79				3.79			

Fonte: Elaborado pelo autor (2024).

Aplicando o teste de variância ANOVA para a latência de Envios e Recuperações por conjuntos de nós, temos que para o envio e recuperação há diferença estatisticamente significativa entre as médias dos grupos ($p < 0,001$) utilizando um intervalo de confiança de 99%. A Tabela 5 expõe os valores calculados para o teste de variância utilizando $\alpha = 0.01$ e também um resumo geral das métricas de latência. Observa-se que para envio e recuperação o valor $F_{calculado} > F_{critico}$, ou seja, existe diferença estatisticamente significativa entre o número de nós da rede para ambas as operações de envio e recuperação.

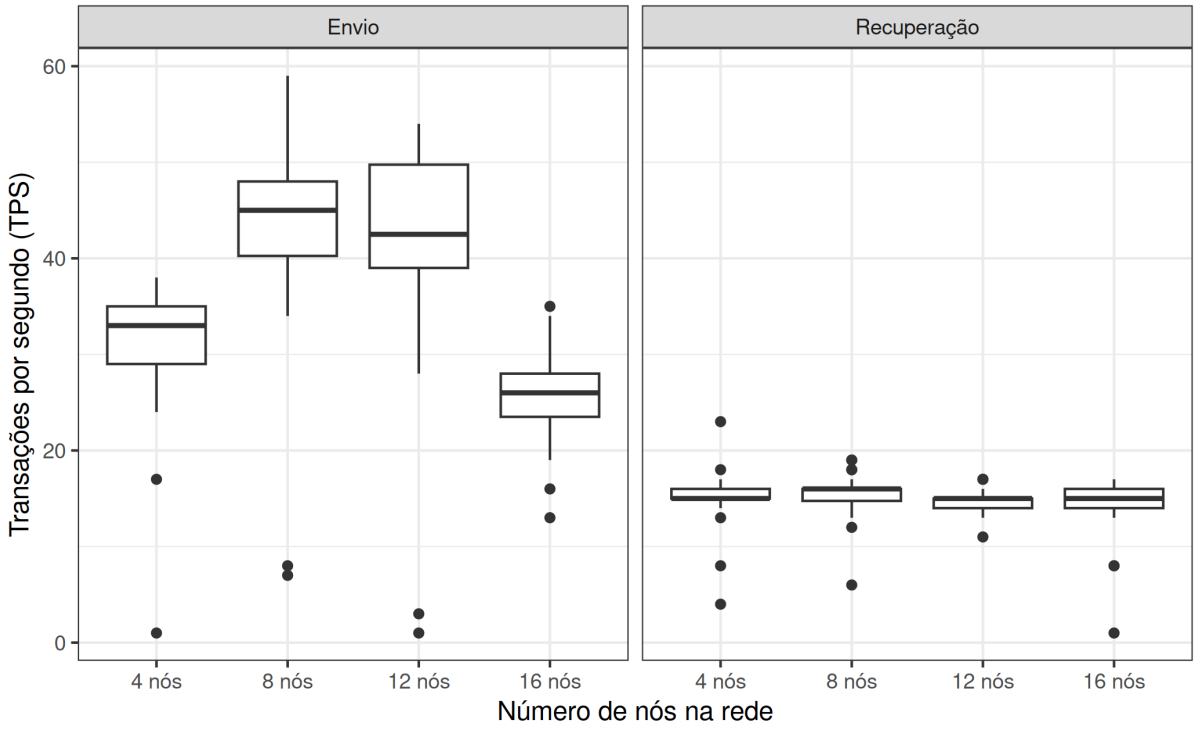
Realizando o teste Post-hoc (Tukey) para as alternativas (4, 8,12 e 16 nós) em envio, não há diferença estatisticamente significativa apenas entre os nós 8 e 12, com um valor de $p = 0,73$. Para a operação de recuperação de arquivos, os cenários com 4 e 8 nós para a rede *blockchain (onchain)*, não apresentaram diferença estatisticamente significativa, com um valor de $p = 0,64$. Também para a operação de recuperação entre 16 e 12 nós, não há diferença estatisticamente significativa, com um valor de $p = 0,99909$. Para os demais casos comparados houve diferenças significativas em relação ao número de nós na rede.

Observando a quantidade de transações realizadas por segundo, por meio da Figura 13, pode-se observar que há um limite nas latências em questão de recuperação na *Blockchain*, não importando a quantidade de nós utilizados. Este fato pode ser explicado devido a limitações na transferência de arquivos com volume grande na *Blockchain*. Essas limitações são principalmente atribuídas ao tamanho máximo dos blocos na *Blockchain*. Quando grandes volumes de dados são armazenados, eles ocupam mais espaço, o que significa que os blocos da rede ficam preenchidos mais rapidamente (menos transações por bloco). Isso leva a um aumento no número de blocos que precisam ser verificados e lidos para recuperar informações específicas.

Realizando análise de variância ANOVA da quantidade de transações por segundo para a operação de recuperação na *Blockchain (onchain)* com um nível de significância $\alpha = 0,05$, obtemos o valor $p = 0,0253$, ou seja, há diferença estatisticamente significativa entre as alternativas (nós). Rejeitando a hipótese de não haver diferença em alterar o número de nós em uma rede *onchain* para as operações de recuperação.

Na Figura 14, pode-se observar um aumento significativo em relação ao número de

Figura 13 – Transações por segundo relativo ao número de nós na rede para rede *Blockchain (onchain)*.



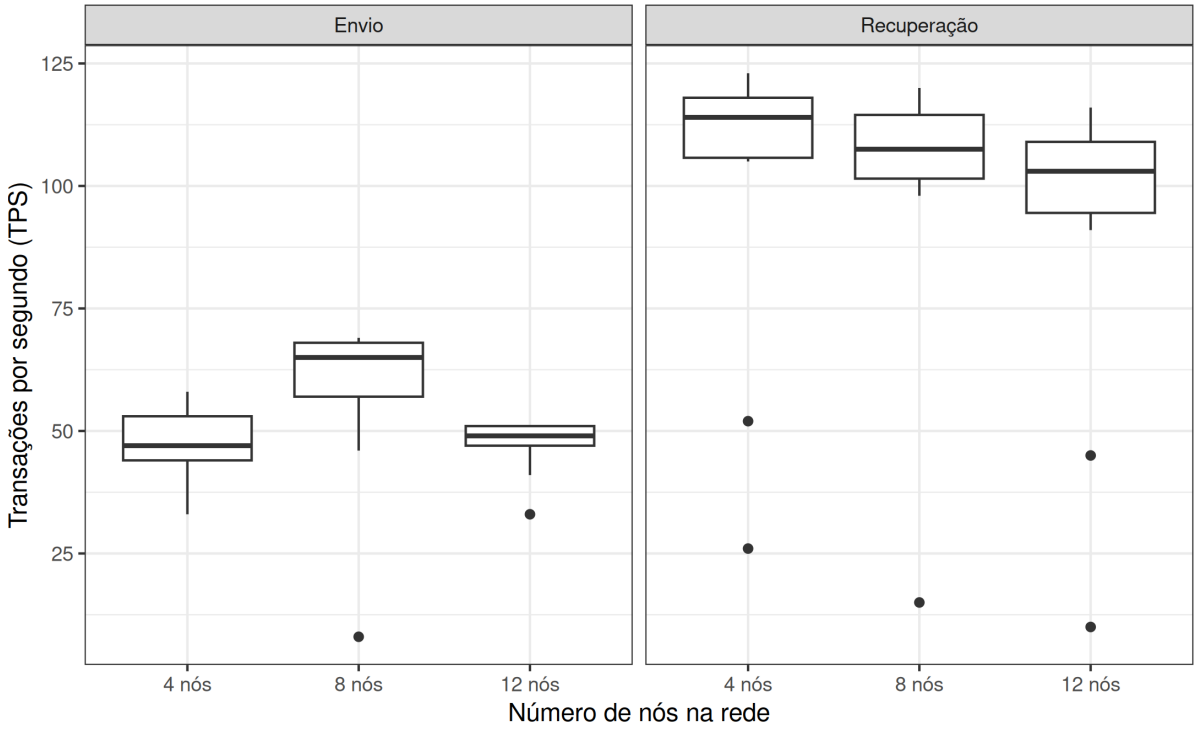
Fonte: Elaborado pelo autor (2024).

transações por segundo entre envio e recuperação de arquivos. Na Figura 15 foram comparados os cenários *onchain* e *sidechain* apenas utilizando TPS (transações por segundo) como base. Nesse caso, pode-se observar no cenário de recuperação que há diferença estatisticamente significativa entre todos os conjuntos de nós testados. Observa-se também na Figura 16 a comparação entre consumo de memória RAM e CPU em cada cenário (*onchain e sidechain*), por quantidade de nós usados e em ambas operações de envio e recuperação de arquivos. Nota-se que para a operação de envio de arquivos em ambas as redes o uso de CPU foi intenso, justificado pela mineração dos blocos e validação das transações no cenário *onchain* e também a replicação dos arquivos no cenário *sidechain*. Porém, o consumo de memória RAM se mostrou significativamente inferior no cenário em *sidechain* ao comparar-se com *onchain*, para envio e recuperação, uma vez que a *Blockchain* armazena parte das transações em memória principal (RAM).

5.2.1 CENÁRIO COM ALTA DEMANDA

Para o cenário com alta demanda foram realizadas 50.000 transações, onde, portanto, foram enviados 50.000 arquivos de 63KB, a fim de mensurar as latências e o uso de recursos computacionais em um ambiente com alta demanda de transações por segundo. Por limitação da própria *Blockchain*, não há como aumentar o tamanho do arquivo salvo por transação, porém com este teste de sobrecarga serão armazenadas mais transações em um único bloco, de modo a estressar a rede em um cenário adverso.

Figura 14 – Transações por segundo relativo ao número de nós na rede para modelo híbrido em *(sidechain)*



Fonte: Elaborado pelo autor (2024).

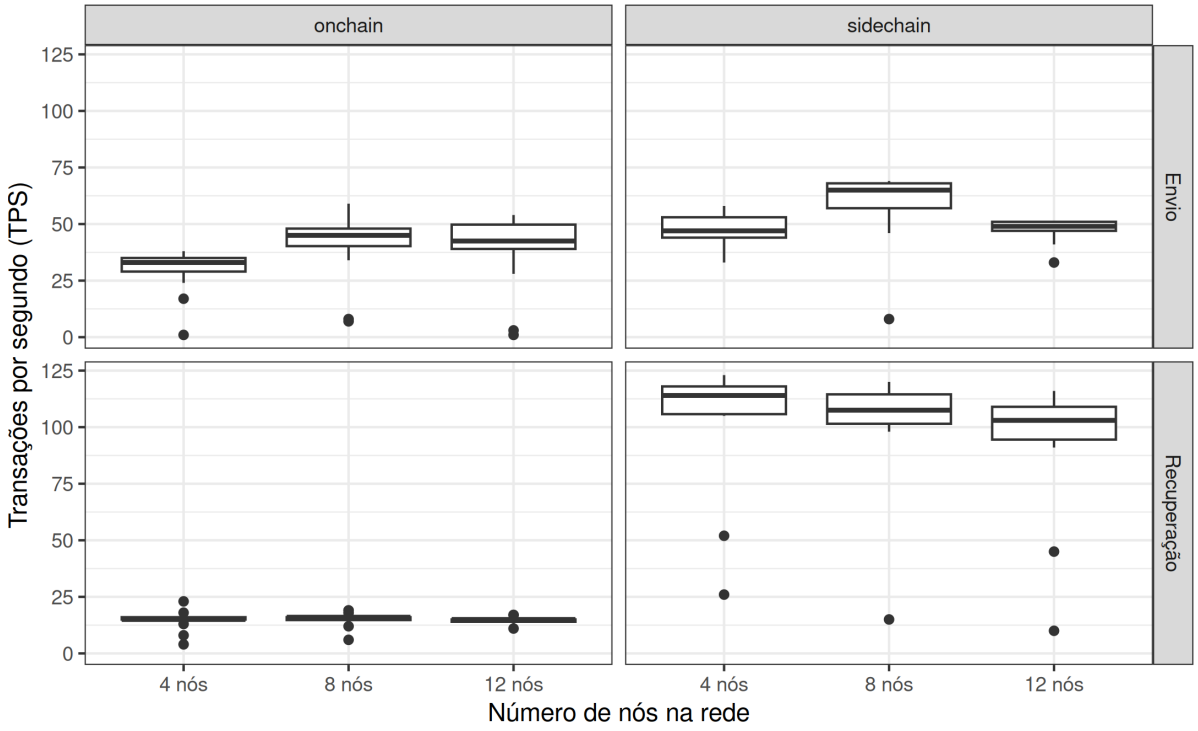
Todos os arquivos utilizados foram gerados previamente antes da execução do *benchmark* com a ferramenta JMETER. Para este cenário de teste foram realizados apenas envios de arquivos para ambas as redes (sem recuperação), sendo configurados 12 nós para ambas. Para o armazenamento hibrido em *sidechain* foi utilizado fator de replicação mínima e máxima igual ao total do número de nós na rede, ou seja, 12 (todos os nós possuem uma cópia do arquivo).

A Figura 17 demonstra a dispersão de latências de envio para ambas as redes. Para o cenário *onchain* foi obtida uma mediana de 38ms, média de aproximadamente 44ms e desvio padrão igual a 28,3ms. O cenário *onchain* apresentou uma taxa de erros nas requisições de aproximadamente 22% no experimento, esta taxa de erro ocorreu em diferentes períodos durante a execução do experimento, porém na Figura 17 é apenas representada as transações com sucesso no intervalo. Ou seja, não significa que aproximadamente as últimas 10.000 transações foram mal sucessivas, mas que houve 10.000 transações mal sucessivas na rede em determinados períodos durante o experimento.

Na Figura 17 fica evidente a lacuna de transações que faltam se compararmos com o cenário em *sidechain*, o qual realizou todos os envios. Já no cenário *sidechain* foi obtida uma mediana de 19ms, média de 20,6ms e um desvio padrão de 9,57ms.

Portanto, nota-se que para o cenário *onchain* a latência de envio é duas vezes maior em relação à mediana e média, ao se comparar com a rede *sidechain*. Além das latências serem significativamente maiores, o cenário *onchain* em sobrecarga não conseguiu realizar o envio

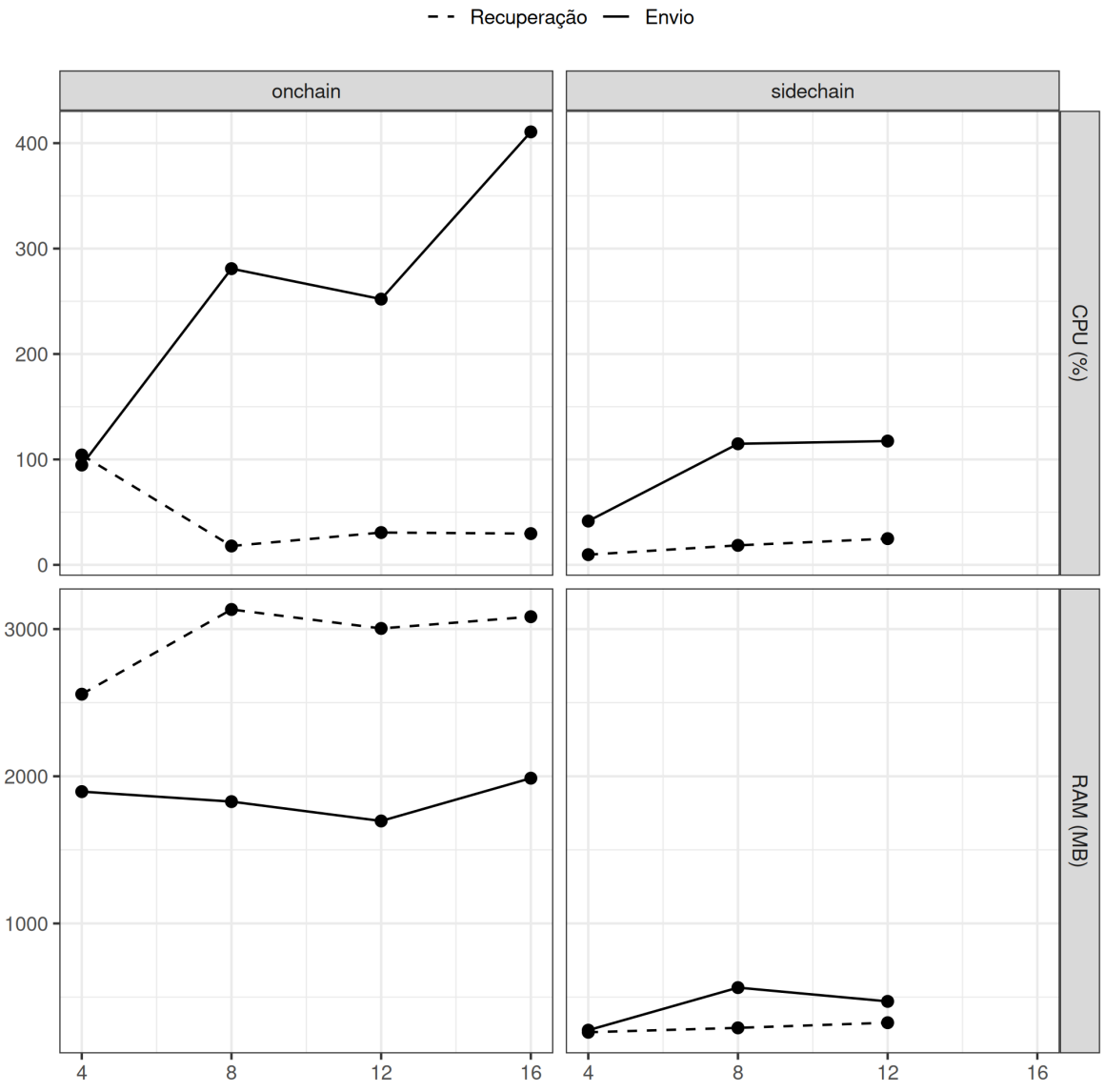
Figura 15 – Transações por segundo utilizando somente *Blockchain* comparado ao armazenamento híbrido em *sidechain*



Fonte: Elaborado pelo autor (2024).

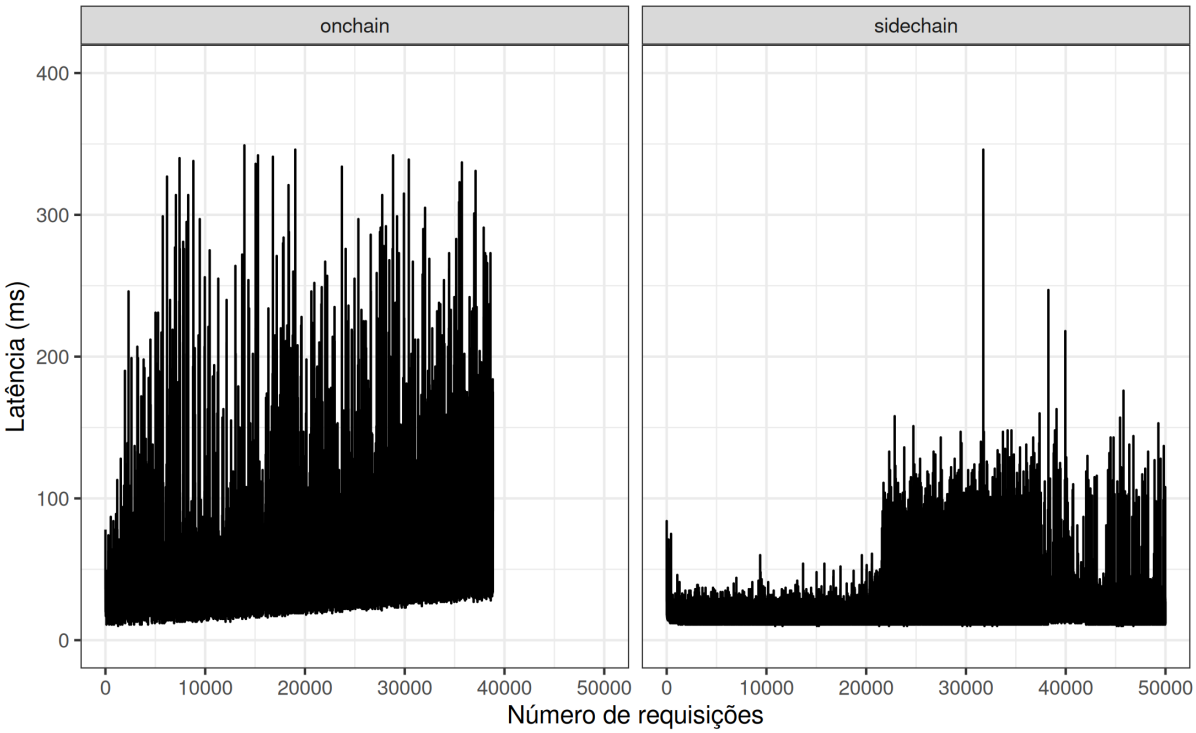
de todos os 50.000 arquivos, recusando 22% das requisições realizadas. As requisições mal sucessivas podem ser explicadas pelo elevado uso de CPU em cada nó da rede *Blockchain (onchain)*, onde as conexões são finalizadas por falta de recurso da máquina de teste.

Figura 16 – Comparativo entre a média de utilização de CPU e memória RAM por conjunto de nós entre armazenamento *onchain* e armazenamento híbrido em *sidechain*



Fonte: Elaborado pelo autor (2024).

Figura 17 – Gráfico de dispersão das latências para ambas as redes em sobrecarga (envio)



5.3 COMPARAÇÃO ENTRE ARQUIVOS COM VOLUMES DIFERENTES PARA ARMAZENAMENTO HÍBRIDO EM *SIDECHAIN*

Neste cenário foram comparados três diferentes tamanhos de arquivo (1MB, 10MB, 100MB) para recuperação e envio, baseados nos tamanhos listados pelo artigo (VARMA, 2008). Por limitações de armazenamento do servidor utilizado, realizar 1000 inserções de cada tipo de volume de arquivo se tornaria inviável, visto que em um cenário com 16 nós e replicação total de cópias na rede híbrida proposta, o volume total seria de aproximadamente 1,63 TB, passando a cota de 930GB disponíveis em disco. Portanto, sabendo desta limitação, foram feitas amostras de 100 requisições para envio e recuperação em todos os tamanhos de arquivo. Desta maneira ainda é possível observar o comportamento da rede mantendo o limite de utilização de disco longe do limite.

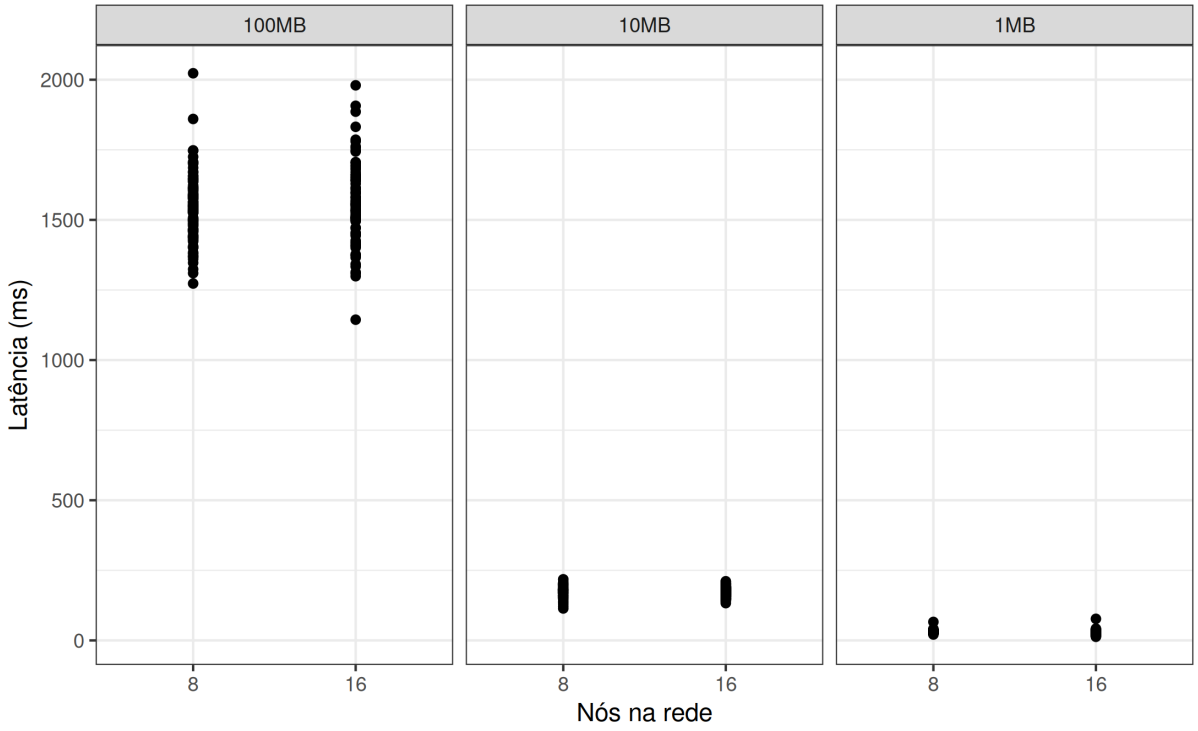
Os resultados das requisições podem ser observadas por meio das Figuras 18 e 19, representando a distribuição das latências em relação ao número de nós da rede de armazenamento híbrido em *sidechain*. Observa-se, conforme a Figura 18 e 19 que a distribuição de latências das requisições para os arquivos com tamanho igual a 100MB foi significativamente superior aos demais casos. Na operação de recuperação utilizando arquivos de tamanho igual a 100MB, é observada uma semelhança entre as latências para 8 e 16 nós. Realizando o teste de variância ANOVA entre 8 e 16 nós para recuperação em arquivos de 100MB, foi constato que $p < 2e-16$, ou seja, adotando um nível de significância igual a $0,05$, existe diferença estatisticamente significativa entre os grupos observados. Portanto, o aumento de número de nós na rede em operações de recuperação com arquivos de 100MB resulta em latências diferentes.

A Tabela B.7, disponível no Apêndice B resume os dados obtidos no experimento para envio e recuperação, com a média, mediana, desvio padrão, mínimo, máximo e o percentil 99% (representado por P99) para cada tamanho de arquivo e número de nós na rede. As métricas capturadas sobre o consumo de CPU e memória RAM não apresentaram mudanças significativas comparadas ao cenário de envio e recuperação de 1.000 arquivos com volume igual a 63KB para modelo de armazenamento híbrido (*sidechain*). Na Tabela B.8 do Apêndice B é possível visualizar um resumo das médias das métricas coletadas, podemos observar que a utilização de CPU por nó aumenta significativamente conforme o volume dos arquivos e o volume de arquivo enviado/recuperado. Este fator está relacionado com as replicações dos arquivos pela rede IPFS (*Daemon* + *Cluster*) e pela validação das transações na *Blockchain*. Para a utilização de memória RAM não foi observada uma mudança estatisticamente significativa entre o número de nós e tamanhos dos arquivos na rede.

5.3.1 CENÁRIO COM ALTA DEMANDA

Este experimento visa observar o comportamento da rede *sidechain* em um cenário de sobrecarga, ou seja, ao realizar a inserção (envio) de novos arquivos, com arquivos já armazenados na rede. Neste cenário foram inseridos previamente 35.000 arquivos de tamanho igual a

Figura 18 – Latências para operações de recuperação entre 8 e 16 nós para diferentes tamanhos de arquivos (1MB, 10MB e 100MB)

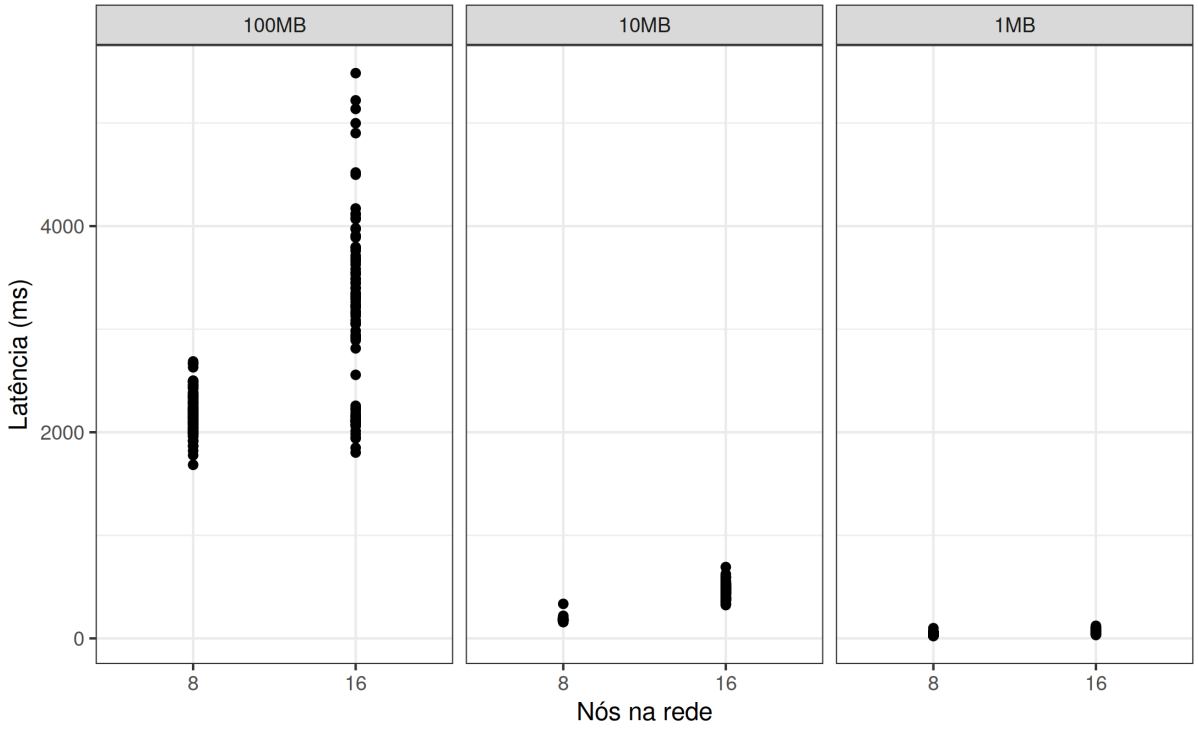


Fonte: Elaborado pelo autor (2024).

63KB (utilizado para o experimento da Seção 5.2), de forma a popular a rede disponibilizando capacidade de sobrecarga com novas inserções de arquivos com diferentes tamanhos. Além disso, o modo de replicação para o IPFS *Cluster* foi definido para que todos os nós tenham uma cópia do arquivo. Também foi utilizado um número de nós fixos em 8, simulando 8 instituições médicas na rede.

O número de amostras enviadas foi estabelecido em 100 para fins de comparação com o cenário de experimentos anterior. Portanto, para cada tamanho de arquivo (1MB, 10MB e 100MB) foram enviados 100 arquivos e obtidas suas respectivas métricas através da camada de monitoramento. A Figura 20 resume a dispersão das latências entre os arquivos enviados. É notável a diferença de latência entre o tamanho de 100MB em relação aos demais; um padrão que já foi observado no cenário de avaliação de de armazenamento híbrido em *sidechain* com tamanhos de arquivos de diferentes volumes sem sobrecarga e se repetiu para o mesmo cenário com sobrecarga de armazenamento de arquivos. Na Tabela 6 estão resumidos os tempos de latência para os arquivos enviados, bem como a média das métricas coletadas no período do experimento. Observa-se que a utilização de memória RAM e CPU por nó manteve o mesmo padrão dos outros experimentos, deixando o uso de RAM com pouca variação entre os arquivos enviados, mas em compensação o uso de CPU por nó para os arquivos de 100MB sofreu um aumento de 77% em relação aos arquivos de 1MB e de 53% em relação aos arquivos de 10MB.

Figura 19 – Latências para operações de envio entre 8 e 16 nós para diferentes tamanhos de arquivos (1MB, 10MB e 100MB)



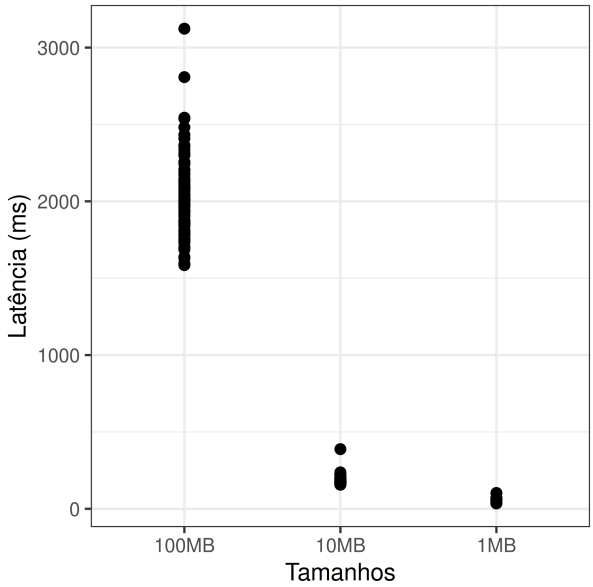
Fonte: Elaborado pelo autor (2024).

Tabela 6 – Conjunto de métricas coletadas para o cenário de sobrecarga utilizando apenas a operação de envio de arquivos.

Métrica	**Tamanhos de Arquivos**		
	1 MB	**10 MB**	**100 MB**
Média (ms)	53.43	185.5	2005
Mediana (ms)	54.00	179.5	1977
Desvio padrão (ms)	8.47	26.53	256.34
P99 (ms)	71.32	238.51	2811.15
Máximo (ms)	103.00	388.0	3123
Mínimo (ms)	35.00	157.0	1585
CPU (%)	711.49	817.68	1257.62
RAM (MB)	788.94	792.99	743.8

Fonte: Elaborado pelo autor (2024).

Figura 20 – Dispersão das latências entre os tamanhos avaliados para a rede com 8 nós



Fonte: Elaborado pelo autor (2024).

5.4 COMPARAÇÃO ENTRE OS FATORES DE REPLICAÇÃO PARA O ARMAZENAMENTO EM *SIDECHAIN*

Neste experimento foram realizados testes modificando o fator de replicação do módulo IPFS *Cluster* para o armazenamento híbrido em *sidechain*, a fim de detectar possíveis variações na latência, número de transações por segundo e uso de recursos computacionais. O objetivo deste experimento é encontrar qual fator de replicação é o mais adequado para a rede híbrida *sidechain* em diferentes cenários. Especificamente, buscava-se determinar como diferentes fatores de replicação impactam a eficiência e o desempenho da rede em termos de latência, transações por segundo e uso de recursos computacionais.

Para calcular o fator de replicação mínimo e máximo foi utilizada a Equação 1, onde n é o número de nós da rede. A relação entre número de nós e a divisão por três se mostrou coerente com a proposta de ter ao menos 2 nós da rede que possuem o arquivo, sendo o número de nós maior que 3. É uma abordagem similar ao problema dos generais bizantinos, porém aplicado a replicação de arquivos e não para nós maliciosos da rede. Por exemplo, para uma rede com 8 instituições, teremos que $\frac{n}{3} = 2,7$, ou seja, é necessário arrendondar para o próximo número inteiro que será igual a 3. Desta forma a replicação mínima e máxima será calculada conforme a Equação 1, onde o mesmo valor obtido será atribuído para replicação mínima e máxima nos experimentos a seguir.

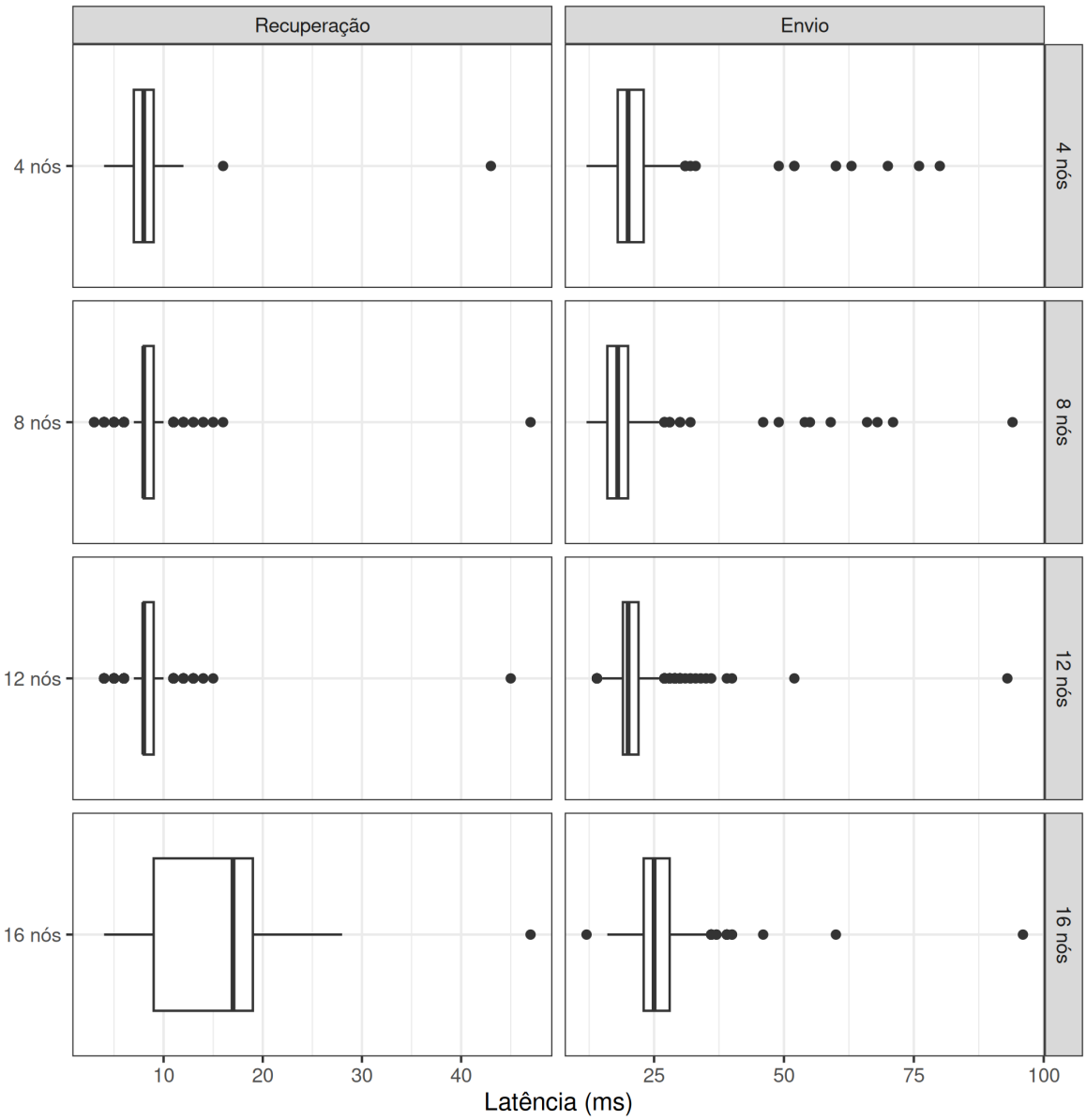
$$f(n) = \begin{cases} \frac{n}{3} & \text{se } \frac{n}{3} \in \mathbb{Z}, \\ \lceil \frac{n}{3} \rceil & \text{se } \frac{n}{3} \notin \mathbb{Z}. \end{cases} \tag{1}$$

A escolha de um fator de replicação divisível por 3 foi proposital para que ao menos 2 nós da rede tenham o arquivo com as configurações do experimento (4, 8, 12 e 16 nós). O menor fator de replicação do experimento será com 4 nós na rede, sendo $\frac{4}{3} = 1,33$ que será um fator de replicação igual a 2. O maior fator será dado pelo conjunto de 16 nós da rede, sendo $\frac{16}{3} = 5,33$ produzindo um fator de replicação igual a 6.

As cargas de trabalho, ou seja, tamanhos de arquivos enviados e recuperados, foram arquivos de 63KB com 1.000 amostras para cada cenário (diferentes fatores de replicação), variando o número de nós e replicações seguindo a Equação 1 proposta. A Figura 21 apresenta os *boxplots* para as latências para recuperação e inserção de cada cenário avaliado. O cenário de 16 nós em recuperação possuiu 4 casos extremos onde a recuperação demorou acima de 1.000ms para ser completada. Uma explicação para esta ocorrência pode ser o fato de todos os nós estarem executando sobre uma mesma máquina, resultando em alguns casos extremos que não condizem com o restante da amostra coletada.

Na Figura 22 podem ser observadas as médias de transações por segundo (tps) para cada conjunto de nós na rede hibrida em *sidechain*. Dentre os experimentos realizados, o conjunto com 16 nós e fator de replicação máxima e mínima igual a 6 obteve o pior desempenho, seja em

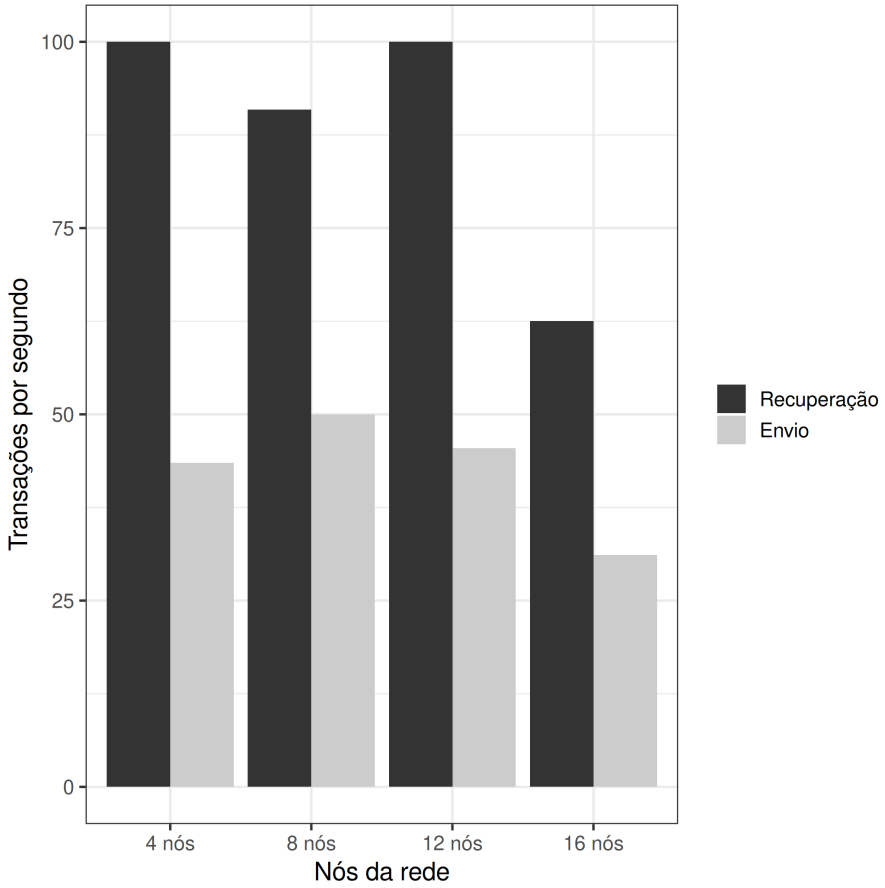
Figura 21 – Comparação de latências para o experimento com mudanças nos fatores de replicação para as operações de recuperação e envio em relação ao número de nós.



Fonte: Elaborado pelo autor (2024).

latências e número de transações por segundo, algo esperado devido a maior complexidade da rede.

Figura 22 – Comparação entre a média de transações por segundo para cada conjunto de nós.



Fonte: Elaborado pelo autor (2024).

5.5 DISCUSSÃO DOS RESULTADOS

Com base nos resultados apresentados, pode-se observar um comportamento recorrente em ambas as redes (*blockchain* e na proposta de armazenamento híbrido em *sidechain*. Na Seção 5.2, torna-se claro a falta de eficiência da rede *Blockchain (onchain)* para salvar arquivos, mesmo que sendo de tamanhos reduzidos (63KB). Nos primeiros experimentos realizados somente na rede *onchain*, pode-se observar através da Figura 13 que existe algo próximo de um limite nas transações recuperadas, algo que certamente afeta o uso em larga escala. Neste sentido, ao comparar ambas as redes na Figura 15 em termos de transações por segundo, fica claro a dificuldade da rede *Blockchain (onchain)* em lidar com a recuperação de arquivos. Ainda comparando ambas as redes, com uma carga razoavelmente baixa (1.000 amostras) é perceptível a alta utilização média de CPU e memória RAM por nó, passando de 3GB em muitos momentos, junto ao uso elevado de CPU para envio dos arquivos. No cenário de sobrecarga, essa diferença

entre as redes se acentua, onde nesse caso, a rede *onchain* não conseguiu enviar todos os 50.000 arquivos, afetando 22% das operações de envio feitas no experimento.

Portanto, é perceptível através dos experimentos realizados que a rede *onchain* não é adequada para armazenamento de arquivos, devido ao seu alto uso de recursos computacionais em envio e recuperação de arquivos, afetando diretamente as latências das requisições. Porém, com o auxílio de uma rede de armazenamento *sidechain* pode tolerar um alto volume de transações sem afetar o desempenho do sistema. Os experimentos realizados somente para a camada hibrida *sidechain* demonstram que o número de nós (instituições), os tamanhos de arquivos e o fator de replicação afetam diretamente as métricas de transações por segundo, latência e uso de recursos computacionais.

5.6 CONSIDERAÇÕES DO CAPÍTULO

A Seção 5.2 demonstra os experimentos comparativos entre as redes, sendo observados grandes variações de latência, transações por segundo e uso de recursos computacionais por conjuntos de nós (tamanho da rede) entre as redes. Neste sentido, somente o uso da *Blockchain* como método para armazenamento de arquivos é altamente não recomendado, sendo necessária uma adequação de um sistema de armazenamento paralelo como o protocolo IPFS.

Já na Seção 5.3 foi realizado um conjunto de experimentos para mensurar o comportamento da rede com diferentes tamanhos de arquivos, variando entre 1MB, 10MB e 100MB para um cenário com replicação completa dos arquivos na rede *sidechain*, ou seja, todos os nós possuem o arquivo armazenado. Também foi realizado um experimento com a rede em sobrecarga, buscando identificar os possíveis gargalos na rede com 35.000 arquivos previamente inseridos.

Por fim, na Seção 5.4 foram realizados experimentos para identificar a possível relação entre o desempenho da rede e o número de replicações de arquivo utilizado.

6 CONCLUSÃO E CONSIDERAÇÕES FINAIS

Este trabalho apresenta uma abordagem ao problema de armazenamento decentralizado para arquivos de grande volume utilizando *Blockchain*. São abordados os principais tópicos de discussão sobre privacidade, integridade e armazenamento de dados médicos de forma decentralizada. A arquitetura desenvolvida para atacar o problema utilizou como base para experimentação o cenário de dados médicos com grande volume.

Para desenvolvedor a arquitetura proposta no Capítulo 4, foi necessário o estudo dos principais conceitos sobre redes decentralizadas, tanto para *Blockchain* como para os módulos IPFS (*daemon* e *cluster*) avaliando as possibilidades de configurações e qual método de consenso adequado para utilização baseada nos requisitos de um cenário com grande volume de dados em uma rede *Blockchain* permissionada (*permissioned*).

A revisão de trabalhos realizada no Capítulo 3 possibilitou a identificação e compreensão de quais são os principais fatores a considerar ao escolher uma rede decentralizada para armazenamento de arquivos, bem como seus pontos positivos e negativos em relação à utilização e implementação. Porém, os trabalhos realizados deixaram uma lacuna sobre o uso do protocolo IPFS como método de armazenamento *sidechain* em conjunto com uma *Blockchain* para arquivos com grande volume, faltando experimentos extensivos sobre tal modelo de armazenamento.

Neste sentido, com base nas revisões e análise dos trabalhos relacionados, foi apresentada uma proposta de arquitetura híbrida de armazenamento decentralizada utilizando ambas as redes *Blockchain* e IPFS. A arquitetura proposta foi desenvolvida em três camadas, sendo elas: armazenamento, monitoramento e interação. O principal destaque da arquitetura fica para a rede IPFS, onde seu módulo adicional chamado de *IPFS Cluster* permitiu a replicação de arquivos na rede considerando um fator de replicação configurado previamente, ou por padrão com replicação para todos os nós da rede, algo que os trabalhos analisados não abordaram como uma alternativa que amplifica o protocolo IPFS.

Após a concepção da implementação foi realizada uma bateria de experimentos sobre a rede *Blockchain (onchain)* e o modelo proposto de armazenamento híbrido *sidechain*. Com os resultados dos experimentos foi possível verificar os principais problemas ao armazenar arquivos em uma rede *Blockchain (onchain)*, sendo eles alta latência das transações e o consumo excessivo de recursos computacionais por nó pertencente à rede. Também foram realizados experimentos considerando diversos fatores na rede híbrida *sidechain*, como variação no volume dos arquivos inseridos e recuperados na rede, modificações no fator de replicação e o comportamento com a rede em sobrecarga (já populada previamente ao experimento).

Os resultados obtidos através dos experimentos demonstram que utilizar uma rede *sidechain* como alternativa à rede *Blockchain*, para o armazenamento decentralizado de arquivos de grande volume é eficiente em diversos fatores. Os principais fatores em que a eficiência é observada foram: taxa de transações por segundo que foi estatisticamente superior no experimento de comparação entre armazenamento somente em *Blockchain (onchain)* em relação ao

armazenamento no modelo híbrido proposto; baixa latência das transações ao compararmos com o armazenamento em *Blockchain (onchain)*; e o uso significativamente menor de recursos computacionais como memória RAM e CPU por nó.

Portanto, é notável que utilizar uma rede *peer-to-peer (P2P)* decentralizada que possui características de integridade e imutabilidade semelhantes a uma *Blockchain* desenvolverá ganhos significativos em desempenho para o sistema. Porém, é necessário avaliar a real necessidade da utilização de uma rede externa *sidechain*, visto que não há garantia de que os arquivos estejam disponíveis, sendo dependente do fator de replicação utilizado na rede *sidechain*.

Para trabalhos futuros, sugere-se a comparação de uma *Blockchain* utilizando o *framework* Hyperledger Fabric nesta arquitetura proposta, observando se a mudança de *framework* impacta nas latências e no número de transações por segundo. Sugere-se também a implementação de mecanismo de privacidade para os arquivos na rede, aplicando criptografias e gerenciamento de chaves utilizando os contratos inteligentes.

REFERÊNCIAS

AGBO, Cornelius C.; MAHMOUD, Qusay H.; EKLUND, J. Mikael. Blockchain technology in healthcare: A systematic review. **Healthcare**, v. 7, n. 2, 2019. ISSN 2227-9032. Disponível em: <https://www.mdpi.com/2227-9032/7/2/56>. Citado na página 19.

ALASSAF, Norah; ALKAZEMI, Basem; GUTUB, Adnan. Applicable light-weight cryptography to secure medical data in IoT systems. **Journal of Research in Engineering and Applied Sciences (JREAS)**, v. 2, p. 50–58, 04 2017. Citado 2 vezes nas páginas 12 e 17.

ALI, Muhammad Salek; DOLUI, Koustabh; ANTONELLI, Fabio. IoT data privacy via blockchains and IPFS. In: **Proceedings of the Seventh International Conference on the Internet of Things**. New York, NY, USA: Association for Computing Machinery, 2017. (IoT '17), p. 1–7. ISBN 978-1-4503-5318-2. Disponível em: <https://doi.org/10.1145/3131542.3131563>. Citado 2 vezes nas páginas 29 e 31.

ANPD. **Autoridade Nacional de Proteção de Dados**. 2024. <https://www.gov.br/anpd/pt-br>. [Accessed 14-Jun-2024]. Citado na página 18.

AZBEG, Kebira; OUCHETTO, Ouail; Jai Andaloussi, Said. Blockmedcare: A healthcare system based on IoT, blockchain and IPFS for data management security. **Egyptian Informatics Journal**, v. 23, n. 2, p. 329–343, 2022. ISSN 1110-8665. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1110866522000160>. Citado 3 vezes nas páginas 14, 29 e 31.

BARBIERI, M. What is information? **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 374, p. 20150060, 2016. Citado na página 11.

BENET, Juan. **IPFS - Content Addressed, Versioned, P2P File System**. arXiv, 2014. ArXiv:1407.3561 [cs]. Disponível em: <http://arxiv.org/abs/1407.3561>. Citado 2 vezes nas páginas 26 e 27.

BRASIL, Ministério da Saúde. **Lei Geral de Proteção de Dados Pessoais**. 2018. Disponível em: <https://www.gov.br/saude/pt-br/acesso-a-informacao/lgpd>. Citado 2 vezes nas páginas 18 e 19.

BUTERIN, Vitalik. **On Public and Private Blockchains**. 2015. Disponível em: <https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains>. Citado 2 vezes nas páginas 21 e 22.

BUTERIN, Vitalik et al. A next-generation smart contract and decentralized application platform. **white paper**, v. 3, n. 37, p. 2–1, 2014. Disponível em: <https://finpedia.vn/wp-content/uploads/2022/02/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf>. Citado na página 25.

CAPOCASALE, Vittorio; GOTTA, Danilo; PERBOLI, Guido. Comparative analysis of permissioned blockchain frameworks for industrial applications. **Blockchain: Research and Applications**, v. 4, n. 1, p. 100113, 2023. ISSN 2096-7209. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2096720922000549>. Citado 2 vezes nas páginas 25 e 36.

CASTRO, Miguel; LISKOV, Barbara et al. Practical byzantine fault tolerance. In: **OsDI**. [s.n.], 1999. v. 99, n. 1999, p. 173–186. Disponível em: <https://pmg.csail.mit.edu/papers/osdi99.pdf>. Citado na página 23.

CHANG, Shuchih E.; CHEN, Yichian. When Blockchain Meets Supply Chain: A Systematic Literature Review on Current Development and Potential Applications. **IEEE Access**, v. 8, 2020. ISSN 2169-3536. Citado na página 11.

COHEN, Max F. Alguns aspectos do uso da informação na economia da informação. **Ciência da Informação**, v. 31, n. 3, p. 26–36, set. 2002. ISSN 0100-1965. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652002000300003&lng=pt&tlng=pt>. Citado na página 11.

CONTROL, Centers for Disease; (CDC), Prevention. **Health Insurance Portability and Accountability Act of 1996 (HIPAA)**. 1996. Disponível em: <https://www.cdc.gov/phlp/publications/topic/hipaa.html>. Citado na página 19.

COX, Graham. **How Do Merkle Trees Work?** 2023. Disponível em: <https://www.baeldung.com/cs/merkle-trees>. Citado na página 27.

CRISTEA, Alexandru-Gabriel et al. Offline but still connected with IPFS based communication. **Procedia Computer Science**, v. 176, p. 1606–1612, 2020. ISSN 1877-0509. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050920320858>. Citado 3 vezes nas páginas 13, 26 e 27.

DINOV, Ivo D. Volume and value of big healthcare data. **J. Med. Stat. Inform.**, Herbert Publications PVT LTD, v. 4, n. 1, p. 3, 2016. Citado na página 29.

ESPOSITO, Christian et al. Blockchain: A panacea for healthcare cloud-based data security and privacy? **IEEE Cloud Computing**, v. 5, n. 1, p. 31–37, 2018. Citado na página 12.

ETHEREUM. **PROOF OF STAKE (POS)**. 2023. Disponível em: <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/>. Citado na página 23.

FERDOUS, Md Sadek et al. Blockchain Consensus Algorithms: A Survey. arXiv, fev. 2020. ArXiv:2001.07091 [cs]. Disponível em: <http://arxiv.org/abs/2001.07091>. Citado na página 24.

FERNÁNDEZ-ALEMÁN, José Luis et al. Security and privacy in electronic health records: A systematic literature review. **Journal of Biomedical Informatics**, v. 46, n. 3, p. 541–562, 2013. ISSN 1532-0464. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1532046412001864>. Citado na página 16.

GOMES, Alan Nascimento; COUTINHO, Emanuel Ferreira. Uma Solução para Compartilhamento de Dados de Saúde Baseada em Blockchain Permissionada e Internet das Coisas para Hospitais Inteligentes. In: **V Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain 2022)**. Brasil: Sociedade Brasileira de Computação - SBC, 2022. p. 1–14. Disponível em: <https://sol.sbc.org.br/index.php/wblockchain/article/view/21467>. Citado 2 vezes nas páginas 30 e 31.

HEART, Tsipi; BEN-ASSULI, Ofir; SHABTAI, Itamar. A review of PHR, EMR and EHR integration: A more personalized healthcare and public health policy. **Health Policy and Technology**, v. 6, n. 1, p. 20–25, mar. 2017. ISSN 2211-8837. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2211883716300624>. Citado na página 17.

HÖLBL, Marko et al. A systematic review of the use of blockchain in healthcare. **Symmetry**, MDPI AG, v. 10, n. 10, p. 470, Oct 2018. ISSN 2073-8994. Disponível em: <http://dx.doi.org/10.3390/sym10100470>. Citado na página 13.

JAEGER. **Jaeger's documentation**. 2023. Disponível em: <https://www.jaegertracing.io/>. Citado na página 38.

JAYABALAN, Jayapriya; JEYANTHI, N. Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy. **Journal of Parallel and Distributed Computing**, v. 164, p. 152–167, jun. 2022. ISSN 0743-7315. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0743731522000648>. Citado 2 vezes nas páginas 29 e 31.

KHAN, Shafaq Naheed et al. Blockchain smart contracts: Applications, challenges, and future trends. **Peer-to-Peer Networking and Applications**, v. 14, n. 5, p. 2901–2925, set. 2021. Citado na página 24.

KUMAR, Randhir; MARCHANG, Ningrinla; TRIPATHI, Rakesh. Distributed Off-Chain Storage of Patient Diagnostic Reports in Healthcare System Using IPFS and Blockchain. In: **2020 International Conference on Communication Systems & NETworkS (COMSNETS)**. Bengaluru, India: IEEE, 2020. p. 1–5. ISBN 978-1-72813-187-0. Disponível em: <https://ieeexplore.ieee.org/document/9027313/>. Citado 2 vezes nas páginas 29 e 31.

LABS, Protocol. **Documentation**. 2023. Disponível em: <https://ipfscluster.io/documentation/deployment/architecture/>. Citado na página 28.

MAIMó, L. F. et al. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. **Sensors**, v. 19, p. 1114, 2019. Citado 2 vezes nas páginas 12 e 16.

NAKAMOTO, Satoshi. Bitcoin: A Peer-to-Peer Electronic Cash System. p. 9, 2009. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Citado 5 vezes nas páginas 12, 19, 20, 21 e 23.

NUNES, Caroline Castro; MA, Stephane; FILHO, Marcelo Silveira Teixeira. Armazenamento descentralizado no Sistema Único de Saúde brasileiro (SUS) usando Interplanetary File System (IPFS) e Blockchain. **Revista de Direito**, v. 13, n. 01, p. 01–25, abr. 2021. ISSN 2527-0389. Number: 01. Disponível em: <https://periodicos.ufv.br/revistadir/article/view/11695>. Citado 2 vezes nas páginas 29 e 31.

ONC, National Coordinator for Health Information Technology. **Electronic health records (EHR)**. 2011. Disponível em: <https://www.healthit.gov/buzz-blog/electronic-health-and-medical-records/emr-vs-ehr-difference>. Citado na página 17.

ONC, National Coordinator for Health Information Technology. **Electronic health records (EHR)**. 2019. Disponível em: <https://www.healthit.gov/faq/what-electronic-health-record-ehr>. Citado na página 17.

PADRÃO, Pascoal; LOPES, Isabel. Implementation of the general regulation on data protection – in the intermunicipal community of alto tâmega and barroso, portugal. In: ABRAHAM, Ajith et al. (Ed.). **Innovations in Bio-Inspired Computing and Applications**. Cham: Springer Nature Switzerland, 2023. p. 836–844. ISBN 978-3-031-27499-2. Citado na página 18.

PILARES, Iris Cathrina Abacan et al. Addressing the challenges of electronic health records using blockchain and IPFS. **Sensors**, MDPI AG, v. 22, n. 11, p. 4032, May 2022. ISSN 1424-8220. Disponível em: <http://dx.doi.org/10.3390/s22114032>. Citado na página 12.

PROMETHEUS. **What is Prometheus?** 2012. Disponível em: <https://prometheus.io/docs/introduction/overview/>. Citado na página 38.

SANTOS, Flávia Alcassa dos. A lei geral de proteção de dados pessoais (LGPD) e a exposição de dados sensíveis nas relações de trabalho. **Revista do Tribunal Regional do Trabalho da 10ª Região**, v. 24, n. 2, p. 145–151, 2020. Number: 2. Disponível em: <https://revista.trt10.jus.br/index.php/revista10/article/view/419>. Citado na página 17.

SINGH, Amritraj et al. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. **Journal of Network and Computer Applications**, v. 149, p. 102471, 2020. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804519303315>. Citado 2 vezes nas páginas 13 e 26.

SINGH, A. K. et al. A survey on healthcare data: A security perspective. **ACM Trans. Multimedia Comput. Commun. Appl.**, Association for Computing Machinery, New York, NY, USA, v. 17, n. 2s, may 2021. ISSN 1551-6857. Disponível em: <https://doi.org/10.1145/3422816>. Citado na página 11.

SIPSER, M. **Introduction to the Theory of Computation**. Cengage Learning, 2012. ISBN 9781133187790. Disponível em: <https://books.google.com.br/books?id=H94JzgEACAAJ>. Citado na página 24.

SIQUEIRA, Lorrana et al. Adoção da LGPD ao armazenamento de dados médicos confidenciais. **JTnI**, v. 2, n. 2, jul. 2022. Citado na página 11.

VARMA, Ravi. Storage media for computers in radiology. **The Indian journal of radiology & imaging**, v. 18, p. 287–9, 11 2008. Citado 2 vezes nas páginas 39 e 51.

WERTHEIN, Jorge. A sociedade da informação e seus desafios. **Ciência da Informação**, v. 29, n. 2, p. 71–77, ago. 2000. ISSN 0100-1965. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652000000200009&lng=pt&tlng=pt>. Citado na página 11.

WUST, Karl; GERVAIS, Arthur. Do you Need a Blockchain? In: **2018 Crypto Valley Conference on Blockchain Technology (CVCBT)**. IEEE, 2018. p. 45–54. ISBN 978-1-5386-7204-4. Disponível em: <https://ieeexplore.ieee.org/document/8525392/>. Citado na página 12.

ZHENG, Zibin et al. Blockchain challenges and opportunities: A survey. **International Journal of Web and Grid Services**, v. 14, p. 352, out. 2018. Citado 3 vezes nas páginas 19, 23 e 24.

ZIMBA, A.; CHISHIMBA, M. Understanding the evolution of ransomware: paradigm shifts in attack structures. **International Journal of Computer Network and Information Security**, v. 11, p. 26–39, 2019. Citado na página 12.

APÊNDICE A – CÓDIGO FONTE E INSTRUÇÕES

As informações referentes a execução da arquitetura e dos experimentos podem ser encontrada na plataforma Github. Todos os dados do repositório estão sob a licença GNU GPL v3.0, deixando os arquivos de configurações e dados obtidos livres para modificações e possíveis melhorias. Os *scripts* contendo os experimentos, arquivos de execução e as instruções de uso da implementação podem ser encontradas através dos diretórios no repositório. Em cada pasta há um arquivo README.md contendo instruções de execução da arquitetura ou dos experimentos realizados. O conteúdo está disponível através do endereço:
<https://github.com/victor99z/tcc-implementacao-experimentos>

Além disso, é necessário ter instalado na máquina os seguintes *softwares* para executar a aplicação e visualizar os experimentos:

- **Docker (Client e Server)** Versão 24.0.7
- **Jmeter** Versão 5.6.2
- **RStudio** Versão 2023.12.1 Build 402
- **R** Versão 4.3.2
- **Python3** Versão 3.11.6

Outras configurações como versão da linguagem Golang, versão da plataforma GoQuorum são independentes visto que o processo de automação utiliza a versão mais recente disponível através das imagens Docker.

Existem dois cenários disponíveis para execução dos experimentos, *onchain* e *sidechain*, onde os componentes são os nós da camada de armazenamento junto a camada de monitoramento e interação para cada nó participante da rede. O passo a passo para execução da rede é descrito a seguir:

1. Escolher o diretório do cenário que planeja ser testado (*onchain* ou *sidechain*);
2. Executar o *script* Python para montagem da rede com determinado número de nós;
3. Caso o cenário testado seja *sidechain*, é necessário executar um *script* adicional para configurar o fator de replicação da rede IPFS;

As métricas podem ser coletadas em formato CSV através do endereço local do Grafana mapeado para a porta 3000, ou podem ser coletas executando o *script* disponível. Instruções para executar o *script* coletor de métricas estão disponíveis no repositório.

APÊNDICE B – RESULTADOS OBTIDOS EM EXPERIMENTOS COM ARQUIVOS DE MÚLTIPLOS TAMANHOS

Tabela 7 – Latências para o envio e recuperação de arquivos de 1 MB, 10 MB e 100 MB em 8 e 16 nós.

?	Métrica	Envio (ms)						Recuperação (ms)					
		1 MB		10 MB		100 MB		1 MB		10 MB		100 MB	
		8 nós	16 nós	8 nós	16 nós	8 nós	16 nós	8 nós	16 nós	8 nós	16 nós	8 nós	16 nós
Latência	Média	46.8	72.44	182.7	462.4	2193	3112	30.98	30.22	175.5	172.9	1541	1558
	Mediana	50.0	71.00	178.5	469.5	2181	3214	30.50	30.00	177.5	173.5	1538	1550
	Desvio padrão	11.43	16.30	19.87	67.3	196.84	867.15	5.02	6.57	19.42	15.3	114.24	141.64
	P99	71.28	109.13	220.16	624.68	2666.21	5233.2	41.25	42.35	217.01	209.02	1861.63	1913.57
	Máximo	99.0	122.00	335.0	692.0	2687	5484	66.00	77.00	218.0	211.0	2023	1980
	Mínimo	24.0	33.00	158.0	323.0	1684	1803	21.00	13.00	114.0	133.0	1273	1144

Tabela 8 – Média de recursos computacionais utilizados no intervalo de execução dos experimentos para arquivos de diferentes volumes.

Métrica	Envio						Recuperação					
	1 MB		10 MB		100 MB		1 MB		10 MB		100 MB	
	8 nós	16 nós	8 nós	16 nós	8 nós	16 nós	8 nós	16 nós	8 nós	16 nós	8 nós	16 nós
CPU (%)	21.9	66.99	107.02	249.43	380.26	561.71	50.9	55.8	56.73	117.89	387.79	448.82
RAM (MB)	572.76	571.42	659.61	489.02	463.01	505.66	400.41	453.47	389.42	474.26	396.44	474.64