



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Instituto de Matemática e Estatística

Wallace Vinicius Silva de Oliveira


**Uma proposta de blockchain utilizando emparelhamento entre
médicos residentes e hospitais como prova de trabalho útil**

Rio de Janeiro

2024

Wallace Vinicius Silva de Oliveira

Uma proposta de blockchain utilizando emparelhamento entre médicos residentes e hospitais como prova de trabalho útil



Projeto Final apresentado ao Instituto de Matemática e Estatística da Universidade do Estado do Rio de Janeiro, para obtenção do grau de bacharel em Ciência da Computação.

Orientadora: Prof.^a Dra. Lucila Maria de Souza Bento

Rio de Janeiro

2024

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC-A

SXXX Oliveira, Wallace Vinicius Silva de.
Uma proposta de blockchain utilizando
emparelhamento entre médicos residentes e hospitais
como prova de trabalho útil/ Wallace Vinicius Silva de
Oliveira. – 2024.
?? f. : il. (se tiver ilustrações)
Orientador: Lucila Maria de Souza Bento.
Projeto final (Bacharel em Ciência da Computação)
- Universidade do Estado do Rio de Janeiro, Instituto
de Matemática e Estatística.
1. XXXXXXXXXXXX 2. XXXXXXXX. I. Bento, Lucila
Maria de Souza. II. Universidade do Estado do Rio de
Janeiro. Instituto de Matemática e Estatística. III. Uma
proposta de blockchain utilizando emparelhamento
entre médicos residentes e hospitais como prova de
trabalho útil.

CDU XXX.XX

Autorizo para fins acadêmicos e científicos, a reprodução total ou parcial
deste projeto final.

Assinatura

Data

Wallace Vinicius Silva de Oliveira

Uma proposta de blockchain utilizando emparelhamento entre médicos residentes e hospitais como prova de trabalho útil

Projeto Final apresentado ao Instituto de Matemática e Estatística da Universidade do Estado do Rio de Janeiro, para obtenção do grau de bacharel em Ciência da Computação.

Aprovada em 01 de julho de 2024.

Banca Examinadora:

Prof.^a Dra. Lucila Maria de Souza Bento - Orientadora
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Fabiano de Souza Oliveira
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Raphael Melo Guedes
Instituto de Matemática e Estatística - UERJ

Rio de Janeiro

2024

AGRADECIMENTOS

Inicialmente, gostaria de agradecer à minha família pelo apoio e suporte ao longo dos anos de estudo. Um agradecimento especial à minha irmã Kelly, por seus incentivos fundamentais no início da minha jornada acadêmica.

À UERJ, pela oportunidade de uma formação de qualidade, proporcionada por um excelente corpo docente e funcionários dedicados.

Ao corpo docente da Universidade, em especial ao departamento de computação, por compartilhar seus conhecimentos e experiências que foram essenciais para minha formação.

Aos colegas que fiz ao longo da graduação, por tornarem essa jornada mais divertida e prazerosa.

À minha orientadora, Lucila Bento, por todas as conversas e sugestões valiosas que me permitiram concluir este trabalho com êxito.

Sempre tive interesse em usar a matemática para fazer o mundo funcionar melhor.

Alvin Roth

RESUMO

OLIVEIRA, Wallace Vinicius Silva de. **Uma proposta de blockchain utilizando emparelhamento entre médicos residentes e hospitais como prova de trabalho útil**. 2024. 65 f. Projeto final (Bacharelado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2024.

A escolha de um programa de residência médica é um processo essencial na formação de um estudante de medicina. Em todo o mundo, diversos processos seletivos são responsáveis por facilitar a correspondência entre médicos residentes e hospitais, com destaque para o *National Resident Matching Program* - NRMP dos Estados Unidos, amplamente reconhecido e com milhares de participantes anualmente. No entanto, a seleção conduzida pelo NRMP tem sido alvo de críticas recorrentes devido à sua falta de transparência e possíveis vieses algorítmicos, levando os estudantes a buscar alternativas, algumas das quais implicam custos adicionais e podem dificultar uma correspondência assertiva entre residentes e hospitais.

A proposta deste trabalho é integrar a tecnologia blockchain ao processo de seleção do NRMP, aproveitando sua capacidade de oferecer um registro imutável e verificável de transações. Essa abordagem visa mitigar as críticas frequentemente direcionadas ao NRMP, ao mesmo tempo em que se busca aumentar a transparência e eficácia do processo de seleção. Além disso, o cálculo de emparelhamento entre médicos residentes e hospitais é empregado como prova de trabalho, tornando a solução apresentada completa, já que o cálculo do emparelhamento é utilizado para produzir os resultados armazenados na blockchain e aumentar a segurança do sistema. Os resultados obtidos demonstraram a aplicabilidade efetiva desta solução proposta com prova de trabalho útil, destacando sua eficácia e viabilidade no contexto do processo de seleção do NRMP.

Palavras-chave: Emparelhamento estável. NRMP. Problema dos hospitais residentes. Algoritmo de Gale-Shapley. Blockchain. Prova de trabalho útil. Aplicações da blockchain.

ABSTRACT

OLIVEIRA, Wallace Vinicius Silva de. A blockchain proposal using matching between medical residents and hospitals as a proof of useful work. 2024. 65 p. Final project (Bachelor's in Computer Science) - Institute of Mathematics and Statistics, Rio de Janeiro State University, Rio de Janeiro, 2024.

Choosing a medical residency program is an essential process in the education of a medical student. Worldwide, various selection processes are responsible for facilitating the matching between medical residents and hospitals, with the NRMP of the United States being widely recognized and hosting thousands of participants annually. However, the selection conducted by the NRMP has faced recurring criticism due to its lack of transparency and potential algorithmic biases, leading students to seek alternatives, some of which incur additional costs and may hinder accurate matching between residents and hospitals.

This study proposes integrating blockchain technology into the NRMP selection process, leveraging its capability to provide an immutable and verifiable record of transactions. This aims to address the frequent criticisms directed at the NRMP while seeking to enhance the transparency and effectiveness of the selection process. Additionally, the complex matching calculation between medical residents and hospitals is employed as a proof of work. Thus, the solution is comprehensive, as the matching calculation is used to produce the results stored on the blockchain and to enhance the system's security. The results obtained demonstrated the effective applicability of this proposed solution with proof of useful work, highlighting its efficiency and feasibility in the context of the NRMP selection process.

Keywords: Stable matching. NRMP. Hospitals/Residents problem. Gale-Shapley algorithm. Blockchain. Proof of useful work. Blockchain applications.

LISTA DE FIGURAS

Figura 2.1 – Representação do SMP	20
Figura 2.2 – Representação do RGS	23
Figura 2.3 – Representação do HGS	25
Figura 2.4 – Representação de um bloco	28
Figura 2.5 – Representação de uma cadeia de blocos	28
Figura 2.6 – Compartilhamento do <i>ledger</i>	32
Figura 5.1 – Estrutura do JSON com os hospitais e suas preferências	47
Figura 5.2 – Estrutura do JSON com os residentes e suas preferências	48
Figura 5.3 – Estrutura da blockchain construída	50
Figura 5.4 – Estrutura dos blocos	51
Figura 6.1 – <i>Bucket</i> no S3	53
Figura 6.2 – Repositório no ECR	54
Figura 6.3 – Cluster com os nós da blockchain	54
Figura 6.4 – Comparação entre os resultados do 1º bloco	57
Figura 6.5 – Comparação entre os resultados do 2º bloco	57
Figura 6.6 – Comparação entre os resultados do 3º bloco	57
Figura 6.7 – Comparação entre os resultados do 4º bloco	58
Figura 6.8 – Comparação entre os resultados do 5º bloco	58

LISTA DE TABELAS

Tabela 6.1 –	Configuração de cada nó	55
Tabela 6.2 –	Tempo de resposta de cada bloco minerado	56

LISTA DE ABREVIATURAS E SIGLAS

API –	<i>Application Programming Interface</i>
AWS –	<i>Amazon Web Services</i>
CaRMS –	<i>Canadian Resident Matching Service</i>
CPU –	<i>Central Process Unit</i>
DPoS –	<i>Delegated Proof of Stake</i>
ECR –	<i>Amazon Elastic Container Registry</i>
ECS –	<i>Amazon Elastic Container Service</i>
HTTP –	<i>Hypertext Transfer Protocol</i>
IP –	<i>Internet Protocol</i>
JRMP –	<i>Japan Residency Matching Program</i>
JSON –	<i>JavaScript Object Notation</i>
NFT –	<i>Non-Fungible Tokens</i>
NRMP –	<i>National Resident Matching Program</i>
P2P –	<i>Peer-to-Peer</i>
PoS –	<i>Proof of Stake</i>
PoUW –	<i>Proof of Useful Work</i>
SMP –	<i>Stable Marriage Problem</i>
S3 –	<i>Amazon Simple Storage Service</i>
vCPU –	<i>Virtual Central Processing Unit</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	16
1.2	Justificativa	16
1.3	Escopo	17
1.4	Organização do documento	18
2	REFERENCIAL TEÓRICO	19
2.1	NRMP	19
2.1.1	Problema dos Hospitais/Residentes	20
2.1.2	Algoritmo de Gale-Shapley	21
2.1.2.1	Algoritmo de Gale-Shapley Orientado aos Residentes	22
2.1.2.2	Algoritmo de Gale-Shapley Orientado aos Hospitais	23
2.1.2.3	Complexidade	25
2.2	Blockchain	26
2.2.1	Componentes da Blockchain	27
2.2.2	Algoritmo de Consenso	29
2.2.2.1	PoW - Proof of Work	29
2.2.2.2	PoS - Proof of Stake	30
2.2.2.3	DPoS - Delegated Proof of Stake	30
2.2.2.4	PoUW - Proof of Useful Work	31
2.2.3	Funcionamento	32
2.2.4	Tipos de Blockchain	33
2.2.5	Aplicações da Blockchain	33
3	TRABALHOS RELACIONADOS	35
3.1	<i>Proposal for a fully decentralized blockchain and proof-of-work algorithm for solving NP-complete problems</i>	35
3.2	<i>Coin.AI: A Proof-of-Useful-Work Scheme for Blockchain-Based Distributed Deep Learning</i>	35
3.3	<i>A novel algorithm for peer-to-peer ridesharing match problem</i>	36
3.4	<i>Exploring Arbitrary Real-Life Problems in Proof-of-Useful-Work: Myth Busting?</i>	36
3.5	<i>A novel proof of useful work for a blockchain storing transportation</i>	

	<i>transactions</i>	36
3.6	Síntese e Perspectivas Futuras	37
4	MÉTODOS E FERRAMENTAS	39
4.1	Tipo de blockchain escolhido	39
4.2	A variante do problema solucionado	39
4.3	Ferramentas utilizadas	40
4.3.1	Linguagem de programação	40
4.3.2	Bibliotecas e Frameworks	41
4.3.3	Softwares	42
4.3.4	Infraestrutura	42
4.4	Testes	44
4.4.1	Validação da blockchain	44
4.4.2	Verificação da consistência dos dados	44
5	IMPLEMENTAÇÃO	46
5.1	Conjunto de dados	46
5.2	Blockchain	48
5.2.1	Adição de novos nós da Blockchain	50
5.2.2	Mineração de blocos	50
5.2.3	Propagação de blocos e consenso	51
6	TESTES	53
6.1	Preparação dos dados	53
6.2	Infraestrutura	53
6.3	Execução da blockchain	55
6.4	Análise dos resultados	56
7	CONSIDERAÇÕES FINAIS	59
7.1	Avaliação da solução	59
7.2	Trabalhos futuros	59
7.3	Limitações	61
	REFERÊNCIAS	62

INTRODUÇÃO

Todos os anos, milhares de estudantes de medicina buscam uma vaga de residência em hospitais de sua preferência, enquanto os hospitais buscam residentes de acordo com suas próprias necessidades. Sem uma coordenação eficiente, esse processo de preenchimento de vagas pode se tornar caótico para ambas as partes. Nos Estados Unidos, o principal programa de seleção de residentes é o *National Resident Matching Program* (NRMP), também conhecido como *The Match*, surgiu com o propósito de trazer ordem a esse complexo processo de seleção. Antes do NRMP, os candidatos à residência se candidatavam diretamente aos hospitais, os quais ofereciam vagas sem levar em conta as ofertas de outros hospitais. Isso resultava numa competição entre os hospitais pelos residentes, devido ao excesso de vagas disponíveis em relação ao número de candidatos [15].

Essa falta de coordenação gerava uma série de problemas, como pressão sobre os estudantes, que precisavam escolher uma vaga de residência até mesmo antes de estarem formados ou de terem a oportunidade de explorar todas as opções disponíveis. Além disso, o processo era desigual, pois alguns hospitais preenchiam todas as suas vagas enquanto outros lutavam para encontrar candidatos adequados. Mesmo com mais vagas do que candidatos, muitos ficavam sem colocação, pois priorizavam os hospitais localizados em grandes centros urbanos.

Desde sua criação em 1952, o NRMP mitigou esses problemas, proporcionando um processo mais organizado e equilibrado. Desde então, milhares de estudantes participaram da seleção anual do NRMP, conhecida como *Annual Main Residency Match*. No ano de 2024, foram 44.853 candidatos para um total de 41.503 posições oferecidas [2]. Embora o NRMP seja o programa mais reconhecido para a seleção de médicos residentes, existem outros programas semelhantes que se destacam, como o *Canadian Resident Matching Service* (CaRMS) [4] e o *Japan Residency Matching Program* (JRMP) [17].

No funcionamento do NRMP, candidatos e programas de residência enviam listas ranqueadas com suas preferências. A solução empregada pelo NRMP consiste em realizar emparelhamentos baseada nessas preferências, em um processo conhecido como emparelhamento estável. Gale e Shapley [1] formalizaram matematicamente o problema do emparelhamento estável em 1962, desenvolvendo

um algoritmo para solucioná-lo, o qual já era similarmente empregado pelo NRMP antes desta formalização. Ao longo dos anos, diversos autores contribuíram para o campo do emparelhamento estável, culminando com o reconhecimento de Lloyd Shapley e Alvin Roth com o Prêmio Nobel de Economia em 2012, por suas contribuições nessa área.

Apesar da sua eficiência e longevidade, o processo de seleção de residência do NRMP é frequentemente criticado por sua centralização e controle por uma única organização, o que gera preocupações com a transparência e a confiança no sistema. Médicos e hospitais não têm acesso ao processo de emparelhamento – isso pode levar à busca por outros métodos de seleção, os quais podem ser custosos. Porém, é importante ressaltar que uma das características do emparelhamento estável é que, dada uma lista de preferências, não há um emparelhamento melhor do que o proposto. Portanto, a busca por outros programas de seleção fora do NRMP só ocorre quando não há confiança no processo de seleção.

Embora implemente um algoritmo similar ao de Gale-Shapley, os detalhes exatos da implementação do NRMP não são divulgados publicamente, o que levanta questões sobre equidade, imparcialidade e possíveis vieses algorítmicos. Além disso, o algoritmo empregado pelo NRMP passou por poucas alterações ao longo dos anos, com a alteração mais significativa conhecida tendo ocorrido em 1998 [15]. Portanto, há uma necessidade contínua de revisão e aprimoramento do NRMP para assegurar um processo justo, seguro, transparente e eficaz para todas as partes envolvidas.

Uma solução potencial para mitigar essas críticas é a integração da tecnologia de blockchain no processo de seleção, que poderia aumentar a segurança e a transparência. Blockchain (que em uma tradução literal significa "cadeia de blocos") é um sistema descentralizado que registra transações em uma rede distribuída sem a necessidade de uma autoridade central; a validação dessas transações é feita por meio de algoritmos de consenso. Entre eles, destaca-se a prova de trabalho (PoW), que, apesar de sua alta demanda por recursos computacionais e energia, garante a segurança da rede ao dificultar a produção de registros falsos.

A prova de trabalho é o algoritmo mais utilizado nas implementações de blockchain, tendo sido inicialmente projetado para mitigar o envio de spams [14]. No

entanto, ganhou bastante visibilidade ao ser implementado no blockchain do Bitcoin e de outras criptomoedas. No PoW, o alto consumo de recursos se deve a busca por um valor de hash que satisfaça o desafio proposto pelo algoritmo, um processo conhecido como mineração, o que levanta preocupações e questionamentos quanto ao uso eficiente de recursos computacionais num problema que carece de aplicação prática.

Recentemente, surgiu a prova de trabalho útil (do inglês *Proof of Useful Work* - PoUW), que une a segurança do PoW a tarefas com aplicação prática. Para ser considerado para PoUW, um problema deve ter:

- Aplicação prática: o problema deve estar relacionado a uma tarefa ou cálculo que tenha uma aplicação prática real e útil fora do contexto da blockchain. Isso significa que a resolução do problema deve contribuir de alguma forma para resolver desafios do mundo real ou gerar resultados benéficos em áreas de interesses comuns.
- Complexidade Computacional: o problema deve ser suficientemente desafiador do ponto de vista computacional, exigindo o uso de recursos de processamento e tempo para ser resolvido. Essa complexidade garante que a realização da prova de trabalho represente um custo computacional significativo, o que é essencial para manter a segurança da rede, prevenindo ataques e manipulações por meio da dificuldade inerente à tarefa.
- Verificabilidade: a solução para o problema deve ser facilmente verificável por outros nós participantes da rede. Isso significa que a verificação da correção da solução proposta deve ser realizável de maneira rápida e eficiente, permitindo que a rede valide e aceite a solução sem atrasos significativos. Essa propriedade é crucial para garantir a integridade e a confiabilidade do sistema de consenso, assegurando que todas as contribuições sejam adequadamente verificadas e incorporadas ao blockchain.

Note que o processo de emparelhamento do NRMP se enquadra nesses requisitos, visto que é essencial para a alocação de médicos residentes, envolve cálculos complexos e produz resultados verificáveis por outras partes da rede, assegurando a transparência e a integridade do processo.

1.1 Objetivos

Este trabalho tem como objetivo geral oferecer uma alternativa ao processo de seleção atualmente adotado pelo NRMP, visando aprimorar sua eficiência, segurança e transparência.

Os objetivos específicos incluem

1. Implementação de uma solução baseada em blockchain que utilize o PoUW para gerenciar e facilitar o processo de emparelhamento entre médicos residentes e hospitais.
2. Avaliação da eficácia da resolução baseada em blockchain em superar os desafios enfrentados pelo NRMP, focando em aspectos como equidade, imparcialidade e acessibilidade.
3. Uso da blockchain para o processamento e armazenamento dos dados de emparelhamento de forma segura e transparente, garantindo que os resultados possam ser verificados e validados por terceiros com poder computacional adequado.

1.2 Justificativa

A alocação eficiente de médicos residentes a hospitais é um desafio crítico no setor de saúde, influenciando diretamente a qualidade do treinamento médico e a distribuição de serviços de saúde. Tradicionalmente, esse processo tem sido complicado por questões de transparência e eficiência, principalmente devido à complexidade inerente na coordenação de preferências de uma grande quantidade de candidatos e instituições.

A introdução de tecnologias como a blockchain tem o potencial de revolucionar este processo ao oferecer uma plataforma que não só melhora a transparência e a segurança dos dados, mas também aumenta a eficiência da alocação através de algoritmos conhecidos pela comunidade e descentralizados de emparelhamento. A tecnologia blockchain, com sua capacidade de fornecer um registro imutável e verificável de transações, apresenta uma solução promissora

para muitos dos problemas enfrentados pelos sistemas tradicionais de alocação de residentes.

Neste cenário, o presente estudo é justificado pela necessidade de explorar alternativas tecnológicas que possam mitigar os problemas associados aos métodos convencionais de alocação de residentes, como a falta de transparência nas decisões de emparelhamento e a possibilidade de manipulação de resultados. Ao utilizar a blockchain para gerenciar o processo de emparelhamento do NRMP, propomos uma abordagem que não apenas fortalece a integridade dos dados, mas também proporciona uma plataforma onde tanto hospitais quanto residentes podem ter uma visão clara e inequívoca dos critérios e resultados de emparelhamento.

Além disso, a implementação de uma solução baseada em blockchain para o NRMP pode servir como um modelo para outros contextos de emparelhamento e alocação em diferentes indústrias e setores, destacando a versatilidade e a capacidade de adaptação desta tecnologia. Por essas razões, este estudo busca não apenas testar a viabilidade técnica de tal aplicação, mas também contribuir para o debate mais amplo sobre a modernização de processos críticos em setores essenciais para a sociedade.

1.3 Escopo

O presente estudo foca especificamente na implementação de uma solução blockchain para o processo de emparelhamento do *National Resident Matching Program* (NRMP). O principal objetivo é avaliar a aplicabilidade da tecnologia blockchain como uma ferramenta para aumentar a transparência, segurança e eficiência na alocação de residentes médicos a hospitais. O escopo deste trabalho inclui o desenvolvimento e teste de uma aplicação blockchain prototípica que simula o ambiente de emparelhamento do NRMP, utilizando dados fictícios gerados para representar tanto os residentes quanto os hospitais.

Este estudo não aborda a comparação direta com sistemas atuais em uso, uma vez que os dados reais dos candidatos e listas de preferências não são publicados, e não entra em detalhes sobre os potenciais desafios legais ou

regulatórios que poderiam ser enfrentados ao implementar uma solução blockchain em um contexto real de alocação de residentes médicos.

Com isso, o presente estudo visa proporcionar uma análise focada em como a tecnologia blockchain pode ser adaptada e utilizada em um contexto específico, sem pretender ser uma solução completa para todos os aspectos do processo de emparelhamento de residência médica.

1.4 Organização do documento

No Capítulo 2, será apresentado o referencial teórico, discutindo os conceitos fundamentais utilizados ao longo deste documento. O Capítulo 3 consistirá na análise dos trabalhos relacionados já publicados. No Capítulo 4, será detalhada a metodologia e as ferramentas empregadas no desenvolvimento da blockchain proposta. Em seguida, no Capítulo 5, serão expostos todos os detalhes relativos à implementação. No Capítulo 6, serão apresentadas as etapas necessárias para executar a aplicação e os resultados dos testes realizados.

Por fim, o Capítulo 7 abordará as conclusões deste trabalho, onde serão discutidas as limitações encontradas e sugeridos possíveis trabalhos futuros.

REFERENCIAL TEÓRICO

Neste capítulo, serão apresentados os fundamentos teóricos que serviram de base para este trabalho. Inicialmente, exploraremos o NRMP, que é uma das implementações mais notáveis do problema de casamento estável, essencial para entender as complexidades da alocação de residentes médicos. Posteriormente, discutiremos a tecnologia de blockchain, que favorece potenciais melhorias em termos de segurança e transparência para a solução do NRMP

2.1 NRMP

O NRMP é uma das aplicações práticas mais conhecidas do problema de emparelhamento estável, também conhecido como problema do casamento estável (do inglês *Stable Marriage Problem* - SMP). Basicamente esse problema consiste em encontrar uma alocação de pares entre dois conjuntos de elementos, de modo que nenhum par preferiria se separar para formar um novo par com outro elemento.

A Figura 2.1 ilustra uma instância do problema do casamento estável, mostrando pares de homens e mulheres, onde cada pessoa é representada exatamente uma vez. Na solução do exemplo, todas as mulheres conseguem sua primeira escolha e as listas de preferências estão indicadas à direita, em ordem decrescente. No problema, se o par (a, b) está no conjunto I , então a e b são parceiros em I , denotados por $b = I(a)$ e $a = I(b)$. O conjunto I é considerado estável se não existir um homem a e uma mulher b tal que a prefira b ao invés de $I(b)$ e b prefira a ao invés de $I(a)$.

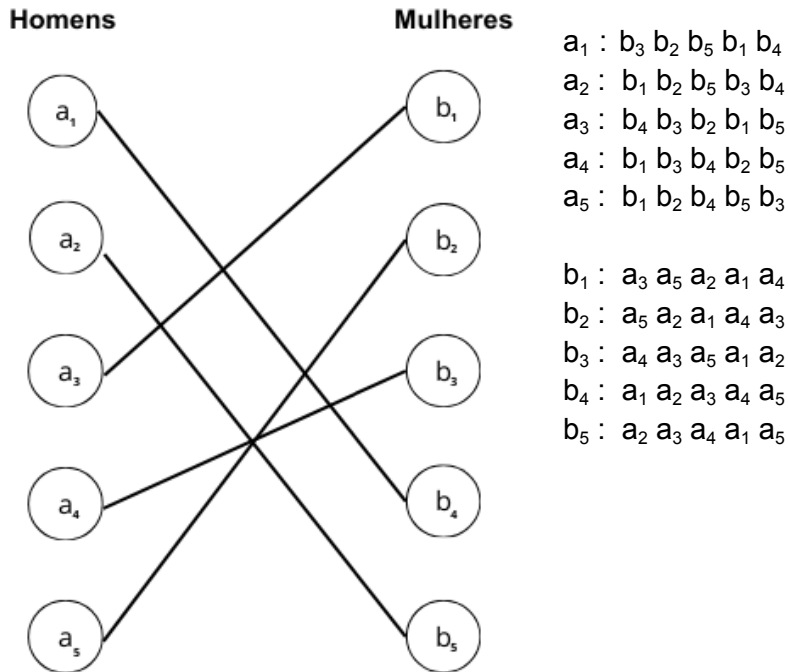


Figura 2.1 – Representação do SMP.

Adaptado: MIT [5].

O problema do casamento estável tem aplicações em diversas áreas, incluindo processos seletivos de admissão em escolas e universidades, seleção de residentes por hospitais, alocação de doadores de órgãos a pacientes em fila de transplante, alocação de tarefas em sistemas de gerenciamento de projetos, entre outros.

2.1.1 Problema dos Hospitais/Residentes

Descrito por Gale e Shapley como o problema da admissão em faculdades (do inglês *College Admissions*) [1]. O problema dos hospitais residentes (HR) é uma generalização do casamento estável. No HR, o problema consiste em encontrar uma alocação estável de pares hospital-residente, onde nenhum hospital e residente não pareado prefeririam estar juntos do que com seus parceiros atribuídos.

Uma instância típica I do HR, é composta por:

- Um conjunto de hospitais $H = \{h_1, h_2, \dots, h_n\}$, onde cada hospital h_i possui uma capacidade c_{hi} , representando o número máximo de residentes que o hospital pode aceitar.
- Um conjunto de residentes $R = \{r_1, r_2, \dots, r_n\}$.
- Para cada hospital h_i , uma lista de preferências ordenada L_{hi} que contém todos os residentes em ordem de preferência.
- Para cada residente r_j , uma lista de preferências ordenada L_{rj} que contém todos os hospitais em ordem de preferência.

Podemos definir M como um emparelhamento do HR quando as seguintes condições são atendidas:

- Cada residente r_j está emparelhado a, no máximo, um hospital h_i .
- Cada hospital h_i possui, no máximo, c_{hi} residentes emparelhados.

Este emparelhamento M é uma relação do tipo 1:N entre residentes e hospitais. Além disso, é importante destacar que podemos ter hospitais e residentes os quais não encontrem um par correspondente numa instância / do emparelhamento estável.

2.1.2 Algoritmo de Gale-Shapley

O Algoritmo de Gale-Shapley é uma solução eficaz para resolver o HR [1]. Este algoritmo busca encontrar uma alocação estável de residentes para hospitais (ou vice-versa) numa instância do problema.

Uma característica relevante do algoritmo é sua capacidade de ser orientado tanto aos residentes quanto aos hospitais, dependendo dos requisitos específicos do contexto. Na versão orientada aos residentes, os residentes fazem propostas aos hospitais, enquanto na versão orientada aos hospitais, os hospitais fazem propostas aos residentes.

2.1.2.1 Algoritmo de Gale-Shapley Orientado aos Residentes

No algoritmo de Gale-Shapley Orientado aos Residentes (RGS), inicialmente, é estabelecido um conjunto vazio de emparelhamentos e, em seguida, é feita uma iteração sobre cada residente não emparelhado, buscando o hospital melhor classificado em sua lista de preferências que ainda não o rejeitou. Se um hospital estiver cheio, o algoritmo verifica se o residente atual é preferido ao pior residente já emparelhado com o hospital; se sim, realiza uma realocação apropriada. Se nenhum hospital rejeitar o residente, ele é emparelhado com o hospital e essa alocação é adicionada ao conjunto de emparelhamentos. O algoritmo continua iterando até que todos os residentes estejam emparelhados ou rejeitados por todos os hospitais em suas listas de preferências, resultando em um emparelhamento estável que atende às preferências de residentes e hospitais. O algoritmo 1 apresenta o pseudocódigo da solução descrita anteriormente.

Algoritmo 1 Algoritmo de Gale-Shapley Orientado aos Residentes

Entrada: Uma instância I do HR
Saída: Um emparelhamento estável M de I

```

1 início
2    $M := \emptyset$ 
3   Enquanto existir um residente  $r$  não emparelhado e não rejeitado por todos os
    hospitais em sua lista de preferências Faça
4     Seja  $h$  o próximo hospital melhor classificado na lista de preferências de  $r$  que
    ainda não o rejeitou
5     Se  $h$  está cheio Então
6       Seja  $r'$  o pior residente emparelhado a  $h$ 
7       Se  $r$  está mais bem ranqueado que  $r'$  na lista de preferências de  $h$  Então
8          $M := M \setminus \{(r', h)\}$            # É desfeito o emparelhamento entre  $r'$  e  $h$ 
9          $M := M \cup \{(r, h)\}$            # Inicia-se o emparelhamento entre  $r$  e  $h$ 
10      Fim Se
11    Senão
12       $M := M \cup \{(r, h)\}$ 
13    Fim Se
14  Fim Enquanto
15  Retorna  $M$ 
```

Adaptado: SAMBINELLI, M. [32].

A Figura 2.2 apresenta uma solução para uma instância I do problema de alocação de residências em hospitais (HR), com a solução sendo orientada aos residentes e utilizando o algoritmo 1. Nesta solução, todos os residentes são alocados em sua primeira escolha, com as listas de preferências indicadas à direita em ordem decrescente.

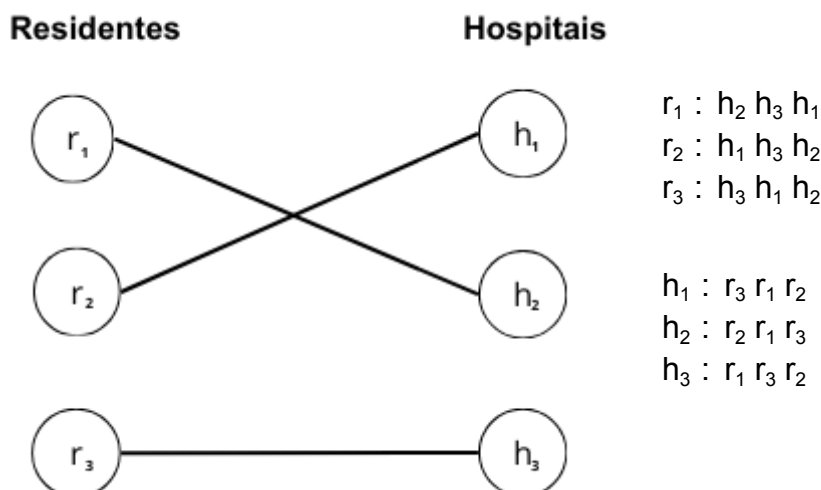


Figura 2.2 – Representação do RGS.

2.1.2.2 Algoritmo de Gale-Shapley Orientado aos Hospitais

O Algoritmo de Gale-Shapley Orientado aos Hospitais (HGS) funciona de maneira semelhante ao RGS, mas inverte a dinâmica das propostas. No HGS, são realizadas iterações enquanto houver um hospital que ainda não está cheio e que tenha um residente não emparelhado em sua lista de preferências. Se um residente estiver emparelhado, então é verificado se o hospital que fez a proposta de emparelhamento atual está mais bem colocado na lista de preferência do residente. Se estiver, o emparelhamento atual é desfeito e um novo emparelhamento é iniciado com o hospital preferido. Esse processo continua até que todos os hospitais estejam cheios ou não tenham mais residentes em suas listas de preferências. O algoritmo 2 apresenta a solução em pseudocódigo.

Algoritmo 2 Algoritmo de Gale-Shapley Orientado aos Hospitais

Entrada: Uma instância I do HR

Saída: Um emparelhamento estável M de I

```

1 início
2    $M := \emptyset$ 
3   Enquanto existir hospital  $h$  não preenchido e não rejeitado por todos os
    residentes em sua lista de preferências Faça
4     Seja  $r$  o próximo residente não emparelhado com  $h$  melhor classificado na lista
    de preferências de  $h$  que ainda não o rejeitou
5     Se  $r$  está emparelhado Então
6       Seja  $h'$  o hospital emparelhado a  $r$ 
7       Se  $h$  está mais bem ranqueado que  $h'$  na lista de preferências de  $r$  Então
8          $M := M \setminus \{(h, r)\}$ 
9          $M := M \cup \{(h, r)\}$ 
10      Fim Se
11    Senão
12       $M := M \cup \{(h, r)\}$ 
13    Fim Se
14  Fim Enquanto
15  Retorna  $M$ 
  
```

Adaptado: SAMBINELLI, M. [32].

A Figura 2.3 apresenta uma solução para a mesma instância do problema de alocação de residências em hospitais (HR) representada na Figura 2.2. No entanto, neste caso, o problema é resolvido com a orientação aos hospitais e utilizando o algoritmo 2. É importante observar que o emparelhamento resultante é distinto para a mesma instância, uma vez que a orientação do algoritmo é diferente.

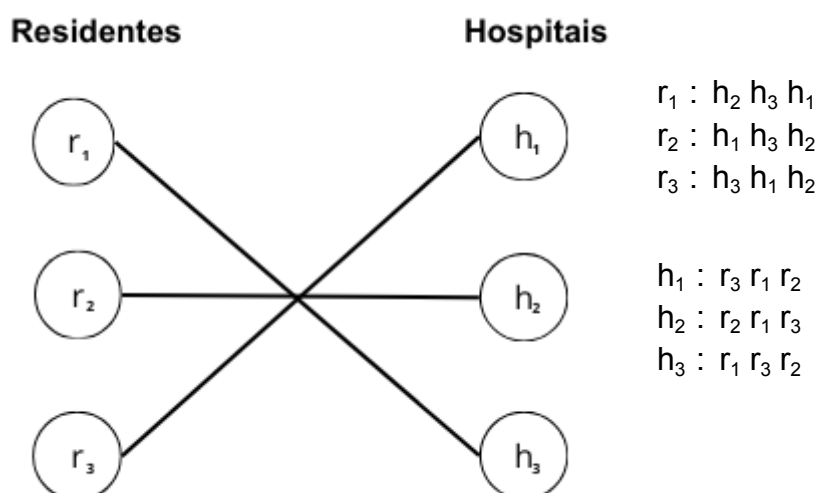


Figura 2.3 – Representação do HGS.

2.1.2.3 Complexidade

A complexidade computacional de tempo do algoritmo de Gale e Shapley é $O(n^2)$ [16], onde n é o número de elementos em cada conjunto (hospitais e residentes). Essa complexidade decorre da natureza iterativa do algoritmo, que é executado em duas fases:

1. Fase de proposta, na qual cada residente (ou hospital, dependendo da orientação do algoritmo) faz propostas aos hospitais (ou residentes), progressivamente, conforme sua lista de preferências.
2. Fase de aceitação, na qual os hospitais (ou residentes) aceitam ou rejeitam as propostas dos residentes (ou hospitais) com base em quem já está emparelhado e nas preferências estabelecidas.

Cada uma dessas fases pode exigir um número de iterações, no máximo, igual ao número de hospitais ou ao número de residentes, o que resulta em uma complexidade total de $O(n^2)$ para o algoritmo completo. Vale destacar que a complexidade é a mesma tanto para o algoritmo orientado aos hospitais quanto na versão orientada aos residentes.

2.2 Blockchain

Uma blockchain é um tipo específico de banco de dados distribuído que mantém um registro contínuo e crescente de transações, organizadas em blocos, que são interligados e protegidos por criptografia. Cada bloco contém um conjunto de transações confirmadas e um hash criptográfico que o conecta ao bloco anterior, formando uma cadeia cronológica de blocos - daí o nome "blockchain".

A primeira aplicação conhecida da blockchain foi proposta por Stuart Haber e W. Scott Stornetta na década de 1990. Eles conceberam um sistema no qual os registros de data e hora dos documentos seriam imutáveis [6]. Posteriormente, essa ideia foi aprimorada com o uso de árvores de Merkle [7], uma estrutura hierárquica de hashes na qual cada nó interno é um hash das combinações de seus filhos [8], melhorando a eficiência e segurança dos dados através desses hashes conectados.

Até o surgimento do Bitcoin, a tecnologia blockchain não era utilizada em larga escala. O Bitcoin foi criado por uma pessoa (ou grupo de pessoas) que usava o pseudônimo Satoshi Nakamoto. O conceito inicial foi descrito em um artigo publicado em 2008, intitulado "Bitcoin: A Peer-to-Peer Electronic Cash System" [9]. Nesse artigo, Nakamoto propôs um sistema de dinheiro eletrônico descentralizado, baseado em criptografia, que eliminaria a necessidade de uma autoridade central, como um banco, para validar as transações financeiras. No ano seguinte, foi lançada a primeira implementação do software Bitcoin como código aberto, marcando o início da rede Bitcoin. Devido à relevância que o Bitcoin alcançou, sua implementação de blockchain tem sido estudada e servido de modelo para outras implementações.

Além das criptomoedas, a tecnologia blockchain encontrou aplicações em vários outros campos. Suas características principais incluem:

- **Descentralização:** A blockchain opera em uma rede descentralizada de nós, onde cada nó da rede possui uma cópia completa das transações. Com isso, é possível verificar e validar as transações independentemente. Isso a torna mais resistente a fraudes, uma vez que não há um único ponto de falha.
- **Imutabilidade:** Uma vez registradas, as transações em um bloco não podem ser alteradas ou excluídas. Isso garante a integridade do histórico de transações na rede.

- **Consenso:** A validação das transações é alcançada por meio de um processo de consenso entre os participantes da rede. Isso garante que apenas transações válidas sejam registradas.
- **Transparência:** Todas as transações registradas são visíveis para todos os participantes da rede, proporcionando transparência e visibilidade.
- **Segurança:** A blockchain utiliza criptografia avançada para garantir a segurança das transações. Os dados são protegidos por meio de algoritmos criptográficos, tornando extremamente difícil alterar registros de transações existentes.

2.2.1 Componentes da Blockchain

Os principais componentes de uma blockchain são:

- **Blocos:** unidades de dados que armazenam informações sobre transações. Cada bloco contém um cabeçalho e um conjunto de transações. O cabeçalho do bloco geralmente inclui um hash do bloco anterior, um *timestamp* e um nonce (número usado apenas uma vez) para garantir a segurança e integridade da cadeia de blocos.

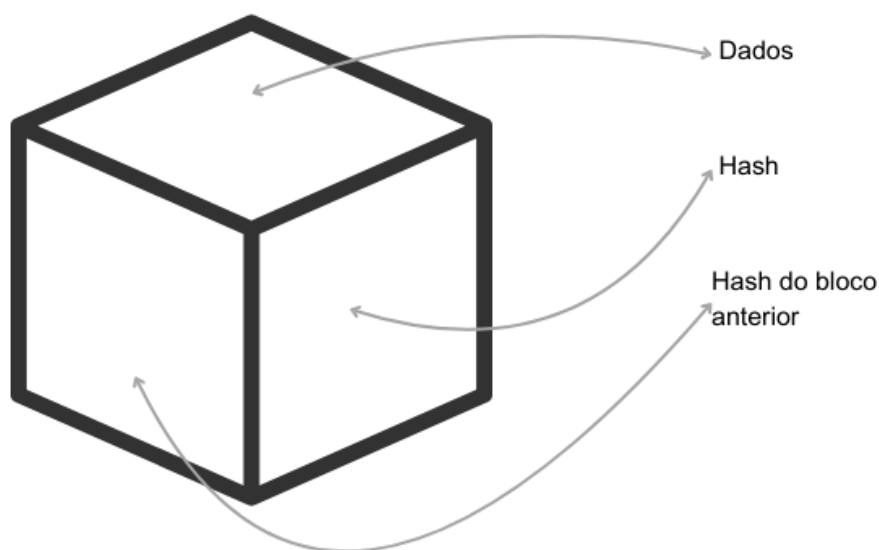


Figura 2.4 – Representação de um bloco.

Adaptado: Simply Explained [10].

- **Cadeia de Blocos:** é formada pela conexão sequencial de blocos, onde cada bloco faz referência ao bloco anterior por meio de um hash. Isso cria uma cadeia imutável e cronológica de transações, permitindo que qualquer alteração em um bloco afete toda a cadeia, tornando-a resistente à adulteração.

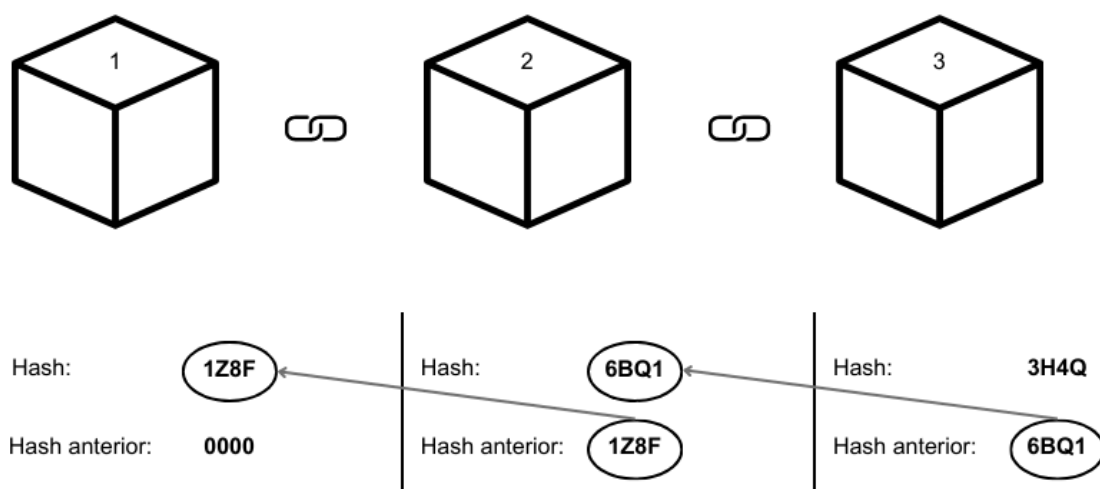


Figura 2.5 – Representação de uma cadeia de blocos.

Adaptado: Simply Explained [10].

- **Rede ponto a ponto** (do inglês *Peer-to-Peer* - P2P): Uma blockchain opera em uma rede descentralizada de pares (ou nós) interconectados. Cada nó na rede mantém uma cópia completa do registro de transações (o livro-razão) e participa do processo de validação e consenso.
- **Livro razão** (do inglês *ledger*): é essencialmente o banco de dados que registra todas as transações e atividades ocorridas na rede. É uma parte fundamental da infraestrutura de uma blockchain, pois mantém um registro completo e imutável de todas as transações que ocorreram desde o seu início. Todos os nós participantes da rede possuem uma cópia completa do *ledger*.

2.2.2 Algoritmo de Consenso

Os algoritmos de consenso definem as regras para a validação de transações e inclusão de blocos na blockchain. Os principais algoritmos de consenso são descritos a seguir.

2.2.2.1 PoW - *Proof of Work*

A premissa do algoritmo é bastante simples: o usuário deve executar uma tarefa que demande poder computacional. Essa abordagem visa reduzir a incidência de usuários mal-intencionados, uma vez que o custo e a complexidade associados à realização dessa tarefa desencorajam comportamentos maliciosos.

O uso mais conhecido da prova de trabalho é na blockchain do Bitcoin. Nele a prova de trabalho é usada para encontrar um valor chamado nonce que, quando adicionado ao cabeçalho de um bloco, produz um hash que atenda a determinados critérios de dificuldade. Os mineradores competem para encontrar esse nonce realizando modificações constantes nele e recalculando o hash do bloco até que um adequado seja encontrado. O objetivo é encontrar um nonce que, quando combinado com o restante do cabeçalho do bloco, gere um hash que comece com

um número específico de zeros. Este processo requer uma quantidade significativa de poder computacional e é intensivo em recursos, o que garante a segurança da rede Bitcoin. Uma vez que um minerador encontra um nonce o qual atenda aos critérios de dificuldade, ele propaga o bloco para a rede e é recompensado com bitcoins recém-criados, além das taxas de transação incluídas no bloco.

Embora a prova de trabalho seja eficaz em garantir a segurança do Bitcoin e outras blockchains, ela também é criticada por seu alto consumo de energia e ineficiência em termos de recursos computacionais. Em 2009, durante o primeiro ano do Bitcoin, a mineração de uma unidade da criptomoeda exigia apenas alguns segundos de consumo de energia, juntamente com um computador doméstico básico [11]. No entanto, em 2023, o processo de mineração teve um consumo anual similar ao de toda a Austrália [38]. Como resultado, outros algoritmos de consenso estão sendo explorados como alternativas mais sustentáveis.

2.2.2.2 PoS - *Proof of Stake*

Diferente da PoW, que depende do poder computacional para validar transações e criar novos blocos na cadeia, a PoS atribui a responsabilidade com base na participação de moedas mantidas por cada participante.

Neste algoritmo, a probabilidade de um participante ser escolhido para criar um novo bloco e validar transações é determinada pela quantidade de moedas que ele possui e está disposto a bloquear como garantia. Em vez de competir para resolver problemas computacionalmente complexos como na PoW, os participantes da PoS são incentivados a agir de forma honesta, pois a penalidade por comportamento malicioso pode resultar na perda das moedas dadas como garantia.

A PoS é considerada mais eficiente em termos de consumo de energia do que a PoW, pois não requer o mesmo nível de poder computacional para operar.

2.2.2.3 DPoS - *Delegated Proof of Stake*

DPOS é um algoritmo de consenso distribuído que introduz a ideia de delegação de poder de voto para participantes da rede assim como nos sistemas democráticos. Os detentores de qualquer quantidade de tokens na carteira têm a opção de votar em representantes os quais serão responsáveis por validar transações e produzir blocos na blockchain. Além disso, o número de representantes é limitado para garantir uma distribuição justa e equilibrada do poder de voto.

Os representantes eleitos no DPOS têm a responsabilidade de garantir a segurança e o bom funcionamento da rede, e são incentivados a agir de forma honesta e transparente, pois podem ser retirados de sua posição se não cumprirem suas obrigações adequadamente. No geral, é um algoritmo elogiado por sua eficiência energética e escalabilidade, mas também enfrenta críticas e desafios, incluindo preocupações com a centralização do poder de voto e a possibilidade de manipulação por grandes detentores de tokens.

2.2.2.4 PoUW - *Proof of Useful Work*

É um algoritmo que combina os princípios da prova de trabalho com a execução de tarefas que possuem aplicação prática no mundo real. A ideia por trás da PoUW é abordar algumas das críticas a PoW tradicional, como o alto consumo de energia e a falta de utilidade além da segurança da rede. Ao introduzir uma dimensão de utilidade no trabalho realizado pelos mineradores, espera-se que o processo de mineração seja mais sustentável e beneficie a sociedade de alguma forma.

Enquanto o PoW convencional requer que os participantes resolvam problemas computacionais difíceis (como o algoritmo hashcash [14] usado no Bitcoin), a PoUW adiciona uma camada extra exigindo que o trabalho realizado tenha algum valor útil fora do contexto da blockchain.

A implementação da PoUW pode variar de acordo com a blockchain específica e os requisitos de aplicação. No entanto, o objetivo principal é incentivar os participantes a realizar trabalho que não só garanta a segurança da rede, mas também tenha um impacto positivo no mundo real.

2.2.3 Funcionamento

No geral, as blockchains que operam numa rede P2P possuem cinco camadas [12]:

- **Camada de consenso:** é a responsável por garantir que todos os nós da rede cheguem a um acordo sobre a validade das transações e a ordem dos blocos na blockchain. Ela implementa algoritmos de consenso, valida transações, seleciona blocos, resolve conflitos e garante a segurança da rede
- **Camada de mineração:** parte essencial de uma blockchain, especialmente em sistemas baseados em PoW. Ela é responsável por criar novos blocos na cadeia, validando e registrando transações.
- **Camada de propagação:** é a responsável por decidir qual livro razão e bloco será compartilhado na rede.
- **Camada semântica:** cuida de como os novos blocos se relacionam com os blocos anteriores e fornece o protocolo para verificar as regras de consenso.
- **Camada de aplicação:** é o nível mais alto da arquitetura de uma blockchain, responsável por definir as regras de negócio e implementar as funcionalidades específicas da aplicação que está sendo desenvolvida sobre a blockchain.

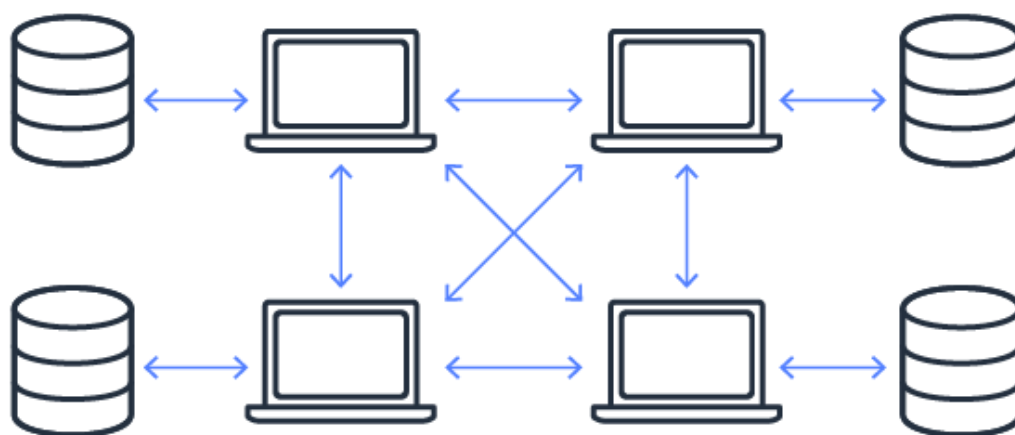


Figura 2.6 – Compartilhamento do *ledger* [13].

2.2.4 Tipos de Blockchain

Os principais tipos de blockchain são:

- **Blockchain Pública:** é aberta e descentralizada, o que significa que qualquer pessoa pode participar da rede, enviar transações e validar blocos. Exemplos conhecidos de blockchains públicas incluem Bitcoin, Ethereum, entre outras.
- **Blockchain Privada:** é controlada por uma única entidade ou organização. Ela não é aberta ao público e geralmente requer permissão para participar. Os participantes da rede são selecionados e autorizados pela entidade controladora. Essas blockchains são usadas principalmente em casos de uso empresarial, onde a privacidade e a governança são prioridades.
- **Blockchain híbrida:** Como o nome sugere, possui características tanto de blockchains públicas quanto privadas. Essa flexibilidade é alcançada por meio de um modelo de permissões, onde alguns nós têm permissão para operar de forma pública e outros de forma privada.
- **Blockchain de Consórcio:** é compartilhada entre várias organizações ou entidades que concordam em cooperar e compartilhar responsabilidades na operação da rede. Ela é semi-aberta, com permissões concedidas a um grupo restrito de participantes que representam os membros do consórcio. Essas blockchains são usadas em setores onde várias partes interessadas precisam colaborar, como no setor bancário, logística e saúde.

2.2.5 Aplicações da Blockchain

A tecnologia blockchain tem mostrado um potencial significativo para além das transações financeiras. Sua natureza descentralizada, segura e transparente a torna adequada para uma variedade de aplicações em diferentes setores. Algumas das principais áreas de aplicação incluem:

- **Mercado Financeiro:** Inicialmente, as criptomoedas, como Bitcoin e Ethereum, foram as primeiras e mais conhecidas aplicações da blockchain. Atualmente, as instituições financeiras estão utilizando a blockchain para uma variedade

de funcionalidades, incluindo transações seguras e transparentes, tokenização de ativos, gestão de identidade financeira, entre outros.

- Tokens Não Fungíveis (do inglês *Non-Fungible Tokens* - NFT): são ativos digitais únicos e indivisíveis que são registrados numa blockchain. Ao contrário das criptomoedas tradicionais, onde cada unidade é indistinguível e pode ser trocada por outra de igual valor, os NFTs possuem propriedades únicas que os tornam distintos e não intercambiáveis.
- Contratos Inteligentes (do inglês *smart contracts*): são programas auto executáveis que executam os termos de um contrato quando as condições pré-definidas são atendidas. Eles são executados na blockchain, o que garante sua imutabilidade e execução confiável. Os contratos inteligentes têm aplicação em uma variedade de setores, incluindo finanças, imobiliário, logística e saúde.
- Votação Eletrônica: baseada em blockchain oferece uma forma segura e transparente de conduzir eleições, garantindo a integridade e a confiabilidade do processo eleitoral. A blockchain pode ser usada para registrar e contar os votos de forma segura e imutável, protegendo contra fraudes e manipulações.

Essas são apenas algumas das muitas aplicações da blockchain em diferentes setores. À medida que a tecnologia continua a evoluir, é provável que surjam novas aplicações que aproveitem os seus benefícios.

TRABALHOS RELACIONADOS

Neste capítulo, serão apresentados trabalhos que contribuíram para o contexto e o desenvolvimento deste estudo. Por meio desta análise, busca-se fornecer uma compreensão abrangente do estado atual e das contribuições mais significativas nos assuntos abordados neste trabalho.

3.1 Proposal for a fully decentralized blockchain and proof-of-work algorithm for solving NP-complete problems

Este trabalho propõe um algoritmo de prova de trabalho que recompensa os mineradores de blockchain por usar recursos computacionais para resolver problemas NP-completos. A ideia é desenvolver uma blockchain que não só mantenha um registro seguro de transações, mas também aprimore soluções para problemas com aplicações práticas. Dentre os problemas NP-completos sugeridos estão alinhamento múltiplo de sequências, dobramento e design de proteínas/biomoléculas e modelo de Ising. O funcionamento da blockchain é descrito em [18].

3.2 Coin.AI: A Proof-of-Useful-Work Scheme for Blockchain-Based Distributed Deep Learning

O artigo apresenta uma proposta teórica para um esquema de PoUW que apoia uma criptomoeda operando em uma blockchain, chamada *Coin.AI*. Nesse sistema, o esquema de mineração requer o treinamento de modelos de aprendizado profundo (do inglês *deep learning*), e um bloco só é minerado quando o desempenho desse modelo excede um limite. O sistema distribuído permite que os nós verifiquem os modelos entregues pelos mineradores de maneira fácil, determinando quando um bloco deve ser gerado. Além disso, este trabalho

apresenta um esquema de prova de armazenamento (do inglês *proof-of-storage*) para recompensar os usuários que fornecem armazenamento para os modelos de aprendizagem profunda. A descrição completa da *Coin.AI* é encontrada em [19].

3.3 A novel algorithm for peer-to-peer ridesharing match problem

Este artigo aborda a correspondência bilateral na construção de um problema de compartilhamento de viagens com um único motorista e vários passageiros numa rede P2P, levando em consideração uma correspondência ideal e estável em todo o sistema. Através de alguns experimentos, os autores demonstram a eficiência computacional do algoritmo proposto e sua aplicabilidade prática. Os resultados mostraram que o algoritmo fornece alta estabilidade nas soluções de correspondência de carona, embora com um pequeno sacrifício em termos de desempenho em todo o sistema. Mais detalhes são fornecidos em [20].

3.4 Exploring Arbitrary Real-Life Problems in Proof-of-Useful-Work: Myth Busting?

O artigo discute o protocolo de consenso PoUW no contexto de sistemas em Blockchain, destacando sua eficiência e segurança. Os autores refletem sobre os desafios de implementação ao definir e resolver problemas de otimização combinatória da vida real como trabalho útil dentro desses sistemas. Eles propõem uma estrutura para incluir novos problemas de combinatória e métodos de solução correspondentes, garantindo que as instâncias sejam bem definidas e que algoritmos de otimização apropriados estejam disponíveis. O trabalho completo pode ser encontrado em [21].

3.5 A novel proof of useful work for a blockchain storing transportation transactions

Este trabalho propõe uma PoUW para resolver um problema de otimização NP-difícil. O foco é na otimização de solicitações de transporte, onde o protocolo de consenso não apenas rastreia e valida transações, mas também otimiza custos de transporte, gerando economia para o ecossistema. O artigo apresenta o framework do PoUW, o modelo de otimização associado e o mecanismo de recompensa dos mineradores, argumentando que essa abordagem pode reduzir o desperdício de energia e oferecer resultados úteis para a gestão da cadeia de suprimentos marítima. O artigo completo é encontrado em [22].

3.6 Síntese e Perspectivas Futuras

Os trabalhos discutidos neste capítulo ilustram o potencial diversificado e a aplicabilidade da blockchain, especialmente no desenvolvimento de soluções de PoUW. Em geral, os problemas de otimização e treinamento de modelos de inteligência artificial têm sido os mais empregados como formas de PoW. O desenvolvimento de algoritmos de consenso mais eficientes e sustentáveis tem sido objeto de amplo debate, e protocolos como PoS e DPoS têm se destacado nesse contexto. No entanto, embora diversos pesquisadores tenham proposto ideias para a utilização de PoUW nos últimos anos, ainda não há a adoção desses algoritmos em blockchains de grande escala.

Além disso, foi apresentado um estudo que demonstra a utilização do emparelhamento estável em um ambiente P2P, o qual exibiu resultados consistentes e eficientes [20]. Vale destacar que este campo de pesquisa é vasto e tem sido alvo de diversos estudos que também revelaram resultados promissores.

Em outras palavras, embora os algoritmos como PoS e DPoS se destaquem por sua eficiência, a exploração de PoUW demonstra um interesse crescente em abordagens que combinam segurança com benefícios tangíveis para a sociedade. Este campo continua a expandir-se, com novas pesquisas desafiando os limites da tecnologia e propondo soluções inovadoras, como os trabalhos apresentados neste

capítulo que forneceram uma base sólida para a concepção e desenvolvimento do presente trabalho.

MÉTODO E FERRAMENTAS

Neste capítulo, serão detalhados os procedimentos e ferramentas adotados para a realização deste trabalho, incluindo a definição do tipo de blockchain escolhido, a especificação da variante do problema solucionado, as ferramentas utilizadas para a solução do problema e a descrição dos testes realizados.

4.1 Tipo de blockchain escolhido

A blockchain proposta neste trabalho é do tipo híbrida devido à ausência de bonificações para os participantes responsáveis pela mineração, à baixa quantidade de transações (NRMP ocorre uma vez por ano) e à demanda significativa de poder computacional. Neste modelo, o gerenciamento e processamento da blockchain são realizados pelo administrador, enquanto terceiros têm a opção de validar os resultados de emparelhamentos executando o mesmo algoritmo e lista de preferências, o que garante que os resultados sejam reproduzíveis. Além disso, todo o algoritmo utilizado na blockchain será público, simplificando a identificação de possíveis falhas.

4.2 A variante do problema solucionado

O NRMP não divulga o algoritmo utilizado para realizar os emparelhamentos entre residentes e hospitais, apenas o conceito geral [3]. Sabe-se que o algoritmo de Gale-Shapley é a base para o emparelhamento realizado pelo NRMP até os dias atuais, só que o algoritmo usado atualmente é mais complexo, pois deve lidar com diversas variações de emparelhamento, incluindo casais de residentes [15]. Como apresentado anteriormente, existem duas formas do algoritmo – uma orientada aos hospitais (HGS) e outra aos estudantes (RGS). Neste trabalho, será utilizado o RGS, considerando que o NRMP atualmente utiliza uma versão que prioriza as

preferências dos estudantes, desconsiderando outras variações de emparelhamento, como a de casais.

4.3 Ferramentas utilizadas

A seleção das ferramentas e tecnologias é crucial para o sucesso de qualquer projeto de blockchain, devido a necessidade de segurança, eficiência e adaptação às necessidades específicas de cada projeto. A seguir são fornecidas informações sobre as linguagens de programação, bibliotecas, frameworks, softwares e a infraestrutura de suporte que foram escolhidas para implementar e testar a blockchain híbrida proposta. Cada escolha foi feita com o objetivo de maximizar a eficácia da solução e garantir que todos os aspectos do sistema sejam robustos e confiáveis.

4.3.1 Linguagem de programação

Para o desenvolvimento deste trabalho, foram utilizadas duas linguagens de programação: C# e Python 3.

Como a relação dos candidatos e hospitais participantes do *The Match* e suas respectivas preferências não são públicas, foi necessário criar participantes e listas de preferências artificiais baseados nos números que são divulgados pelo NRMP a cada edição. Para atender a tal necessidade, dentre as diversas opções disponíveis, C# foi selecionada devido à sua integração nativa com a plataforma .NET, que fornece uma gama de ferramentas e recursos para desenvolvimento, além de ser uma linguagem robusta e orientada a objetos, ideal para lidar com manipulação de dados e operações em arquivos.

Para a implementação da blockchain, Python foi escolhida devido a sua facilidade de uso, versatilidade, ampla comunidade de desenvolvedores e vasta coleção de bibliotecas, o que facilita o desenvolvimento de sistemas complexos. Sua sintaxe limpa e legibilidade também facilitam o desenvolvimento e a manutenção.

4.3.2 Bibliotecas e Frameworks

Para desenvolver e implementar efetivamente a blockchain híbrida e as funcionalidades associadas a este projeto, foram utilizadas as bibliotecas e frameworks detalhados a seguir.

- *Matching*: é um pacote disponível em Python utilizado para resolver problemas de emparelhamento. Com esta biblioteca, é possível definir conjuntos de agentes com preferências sobre outros conjuntos e encontrar emparelhamentos estáveis com base nessas preferências [23]. No contexto deste projeto, a biblioteca facilita a implementação do algoritmo de Gale-Shapley, permitindo definir conjuntos de residentes e hospitais, bem como suas preferências mútuas. Ela auxilia na busca de emparelhamentos que sejam estáveis e otimizados, conforme as necessidades do sistema NRMP, além de verificar se o emparelhamento resultante é válido e estável.
- Boto3: é uma biblioteca desenvolvida pela *Amazon Web Services* (AWS) que permite interagir e gerenciar serviços da AWS, fornecendo uma interface simples e intuitiva para acessar os serviços da AWS diretamente do código Python, facilitando o desenvolvimento de aplicativos e scripts que interagem com a nuvem da AWS [24]. No contexto deste projeto, o Boto3 desempenha um papel essencial ao facilitar o acesso eficiente aos recursos da AWS utilizados na blockchain, permitindo uma integração fluida e confiável.
- Python-dateutil: é uma biblioteca utilizada para manipulação de datas e horários de forma simplificada e eficiente. Ela fornece funcionalidades extras em relação às bibliotecas padrão do Python, como análise de datas em strings e cálculos de intervalos de tempo [25], o que é vital para o registro correto e a verificação das transações dentro da blockchain.
- *Requests*: é uma biblioteca amplamente utilizada em Python para enviar e receber solicitações HTTP (*Hypertext Transfer Protocol*) de forma simples e eficiente via código, permitindo que o desenvolvimento se concentre na lógica de aplicação, em vez de lidar com detalhes de baixo nível de comunicação HTTP [26]. Neste projeto, *Requests* é usada para facilitar a comunicação

interna da nossa blockchain, além de possibilitar interações com outras aplicações externas.

- *Flask*: é um framework Python leve e flexível para construção de aplicativos web. Ele oferece simplicidade e facilidade de uso, permitindo a rápida criação de aplicativos web robustos e escaláveis [27]. Ele suporta o desenvolvimento da aplicação através do qual os usuários podem interagir com o sistema de emparelhamento, visualizar resultados e validar transações.
- *Newtonsoft.Json*: é uma das bibliotecas mais populares para manipulação de JSON em C#, oferecendo uma variedade de recursos para serialização e deserialização de objetos para JSON e vice-versa [33], o que é fundamental para manipulação dos dados no presente projeto.

4.3.3 Softwares

A implementação e o teste de uma blockchain requerem o uso de softwares para facilitar a depuração e a avaliação do sistema. A seguir são apresentados dois softwares essenciais utilizados durante o processo de teste da blockchain desenvolvida neste trabalho: KDiff3 e Postman.

- KDiff3: é uma ferramenta de comparação e mesclagem de arquivos [36], que desempenha um papel fundamental na verificação da integridade e consistência dos dados produzidos pelos nós da blockchain.
- Postman: é uma plataforma de colaboração que simplifica o processo de teste de interface de programação de aplicação (do inglês *Application Programming Interface* - API) [37]. Utilizado neste trabalho para enviar solicitações HTTP às interfaces da blockchain, o Postman oferece uma interface intuitiva e poderosa para explorar, testar e depurar as funcionalidades da blockchain desenvolvida.

4.3.4 Infraestrutura

Para suportar a infraestrutura exigida pela blockchain, optamos por utilizar um serviço de computação em nuvem, escolhendo a AWS devido a sua liderança como provedor de serviços em nuvem, sua ampla gama de recursos que atendiam às necessidades específicas para a execução da blockchain desenvolvida neste trabalho, sua política de custos acessíveis, a flexibilidade de recursos sob demanda e a facilidade de escalabilidade. Os serviços da AWS empregados neste projeto foram:

- *Amazon Simple Storage Service (S3)*: é um serviço de armazenamento altamente escalável, projetado para armazenar e recuperar grandes quantidades de dados de forma simples e rápida [28], permitindo o armazenamento dos arquivos utilizados na blockchain com segurança e alta disponibilidade.
- *Amazon Elastic Container Registry (ECR)*: é um registro de contêineres totalmente gerenciado, que facilita o armazenamento, gerenciamento e implantação de imagens de contêineres do Docker [29], essencial para o gerenciamento eficiente das várias versões e configurações da nossa aplicação blockchain.
- *Amazon Elastic Container Service (ECS)*: é um serviço de orquestração de contêineres que simplifica o processo de execução, escalonamento e gerenciamento de aplicações em contêineres na AWS [30], proporcionando a capacidade computacional necessária para processar e validar transações na blockchain.
- *Amazon Fargate*: é um serviço de computação *serverless* (sem servidor), projetado para simplificar a execução de contêineres Docker na nuvem. Com ele, é possível executar aplicações em contêineres sem a necessidade de provisionar e gerenciar servidores [35]. A utilização do Fargate na blockchain proporciona uma infraestrutura altamente escalável e flexível, permitindo uma implantação mais rápida e eficiente dos nós da rede.

Adicionalmente, para facilitar a implantação e gerenciamento da aplicação, recorreremos ao Docker [31], uma plataforma de containerização que permite empacotar, distribuir e executar aplicativos de forma consistente em qualquer ambiente. Com o Docker, foi possível encapsular a aplicação e suas dependências em contêineres, garantindo a portabilidade e a padronização do ambiente de

execução. Esta abordagem minimiza os conflitos de dependências e simplifica o processo de implantação em diferentes ambientes de produção ou testes.

4.4 Testes

Para garantir a robustez e a confiabilidade da blockchain, implementamos dois tipos principais de testes: validação da blockchain e verificação da consistência dos dados. Não foi utilizada nenhuma ferramenta de teste especializada, mas adotamos abordagens práticas para avaliar cada aspecto do sistema.

Estes testes são fundamentais não apenas para garantir a operação técnica correta da blockchain, mas também para verificar se o sistema atende aos requisitos de confiabilidade, transparência e segurança necessários para aplicações críticas como o emparelhamento NRMP.

4.4.1 Validação da blockchain

Como afirmado anteriormente, a blockchain desenvolvida utiliza o protocolo de consenso PoUW, no qual o participante da rede com maior poder computacional consegue o resultado do emparelhamento mais rapidamente do que os outros. Para testar a eficácia da blockchain em diferentes cenários, simulamos variados níveis de poder computacional por meio da configuração de múltiplas instâncias do tipo AWS Fargate no ECS, cada uma com diferentes capacidades de processamento. Isso nos permitiu observar como a blockchain se comporta sob diferentes condições de carga e verificar a eficiência do algoritmo de mineração em ambientes computacionais variados.

4.4.2 Verificação da consistência dos dados

Um dos principais atributos do emparelhamento NRMP é que resultados consistentes são gerados para entradas idênticas, independentemente do nó que executa o cálculo. Para validar essa propriedade, realizamos testes onde múltiplos nós na rede processaram o mesmo conjunto de dados de emparelhamento. Observamos se todos exibiam resultados idênticos, garantindo assim a integridade e a confiabilidade do emparelhamento. Assim, um bloco só é adicionado ao *ledger* se o resultado for igual ao de todos os participantes, confirmando a consistência e a correteude das operações realizadas pela blockchain.

IMPLEMENTAÇÃO

Neste capítulo, será detalhado a implementação da blockchain, abordando desde a geração do conjunto de dados utilizados até a estruturação da própria blockchain.

5.1 Conjunto de dados

Devido à natureza confidencial dos dados utilizados no NRMP, os detalhes específicos dos participantes e suas listas de preferências não são divulgados publicamente. Portanto, para simular de forma realista as condições do NRMP e testar nossa blockchain híbrida, foi necessário desenvolver uma aplicação, denominada *ResidentsHospitals-Matching-Generator*, que cria um conjunto de dados fictícios. Essa aplicação, construída em C# .NET 6, gera participantes e listas de preferências de maneira pseudo-aleatória, permitindo-nos modelar e analisar o comportamento da blockchain sob condições controladas, mas representativas. A aplicação é dividida em dois módulos executáveis: *ResidentsDataset* e *PreferenceLists*.

ResidentsDataset é o projeto responsável por gerar os candidatos participantes de uma determinada edição do NRMP. Ele lê um arquivo contendo nomes fictícios de residentes e seleciona, de maneira pseudo-aleatória, residentes desse arquivo um por um até preencher o arquivo de saída com a quantidade de participantes requerida para a edição do NRMP. Essa quantidade é obtida a partir dos dados divulgados pelo NRMP.

Já o *PreferenceLists* é o responsável por criar as listas de preferências tanto para os residentes quanto para os hospitais. Ele começa lendo o arquivo contendo os residentes fictícios gerados pelo *ResidentsDataset*. Em seguida, são criados hospitais fictícios, cada um com uma capacidade arbitrária que varia entre 5 a 61 vagas de residência. Esse intervalo foi definido de forma que a quantidade de hospitais e residentes seja balanceada.

Durante a criação de cada hospital, também é gerada a sua lista de preferência. Essa lista pode ser maior do que a sua capacidade, mas nunca menor. Quando um residente é adicionado à lista de preferência de um hospital, o hospital também é adicionado à lista de preferência do residente, embora não necessariamente na mesma ordem de preferência. Isso se deve ao fato de que um emparelhamento só pode ocorrer se ambas as partes desejarem se emparelhar.

Esse processo de criação de hospitais continua até que a quantidade total de vagas de residência para a edição seja alcançada. Ao final, utilizando a biblioteca *Newtonsoft.Json*, o *PreferenceLists* gera dois arquivos no formato JSON (*JavaScript Object Notation*): um contendo as preferências dos hospitais (ver exemplo na Figura 5.1) e outro com as preferências dos residentes (ver exemplo na Figura 5.2).

```
{
  "Preferences": [
    {
      "Program": "Hospital_1",
      "Capacity": 5,
      "Preferences": [
        "Gabriel Correa",
        "João Almeida",
        "Lucas Oliveira",
        "Sofia Santos",
        "Manuela Costa"
      ]
    },
    {
      "Program": "Hospital_2",
      "Capacity": 6,
      "Preferences": [
        "João Almeida",
        "Sofia Santos",
        "Gabriel Correa",
        "Laura Pereira",
        "Manuela Costa",
        "Lucas Oliveira"
      ]
    },
    {
      "Program": "Hospital_3",
      "Capacity": 5,
      "Preferences": [
        "Manuela Costa",
        "Laura Pereira",
        "Lucas Oliveira",
        "Sofia Santos",
        "Gabriel Correa"
      ]
    }
  ]
}
```

Figura 5.1 – Estrutura do JSON com os hospitais e suas preferências


```

{
  "Preferences": [
    {
      "Name": "Sofia Santos",
      "Preferences": [
        "Hospital_3",
        "Hospital_1",
        "Hospital_2"
      ]
    },
    {
      "Name": "Lucas Oliveira",
      "Preferences": [
        "Hospital_1",
        "Hospital_3",
        "Hospital_2"
      ]
    },
    {
      "Name": "Manuela Costa",
      "Preferences": [
        "Hospital_3",
        "Hospital_2",
        "Hospital_1"
      ]
    },
    {
      "Name": "Gabriel Correa",
      "Preferences": [
        "Hospital_2",
        "Hospital_1",
        "Hospital_3"
      ]
    },
    {
      "Name": "Laura Pereira",
      "Preferences": [
        "Hospital_3",
        "Hospital_2"
      ]
    },
    {
      "Name": "João Almeida",
      "Preferences": [
        "Hospital_1",
        "Hospital_2"
      ]
    }
  ]
}

```

Figura 5.2 – Estrutura do JSON com os residentes e suas preferências

O código da aplicação *ResidentsHospitals-Matching-Generator* está disponível no repositório do *GitHub*:

<https://github.com/wallacevncs/ResidentsHospitals-Matching-Generator>

5.2 Blockchain

A aplicação blockchain foi construída utilizando uma API *Flask* em Python (descrita na Seção 4.3.2), a qual consiste em quatro rotas: *mine_block*, *get_chain*, *update_chain* e *connect_node*. A comunicação entre essas rotas é realizada por meio das operações padrão do HTTP, com as três primeiras rotas utilizando o método GET e a última utilizando o POST. Todas as interações geram respostas em formato JSON, e a aplicação opera dentro de um container Docker para garantir consistência e eficiência.

A Figura 5.3 mostra a estrutura da blockchain construída. Inicialmente, uma transação é solicitada a um nó da rede, que atualiza sua cópia da blockchain para sincronizá-la com as dos demais nós. Em seguida, verifica-se a existência de arquivos contendo as preferências dos hospitais e residentes no S3. No caso de ausência desses arquivos, a transação é abortada. Contudo, se os arquivos estiverem disponíveis, procede-se com o processamento do emparelhamento. Após a conclusão da PoUW, é verificado a validade e estabilidade do resultado gerado. Caso a verificação não seja positiva, a transação é imediatamente interrompida. Por outro lado, caso seja confirmada a validade do resultado, um novo bloco é gerado e incorporado à blockchain do respectivo nó. Posteriormente, os arquivos processados são removidos do S3, marcando assim o encerramento da transação.

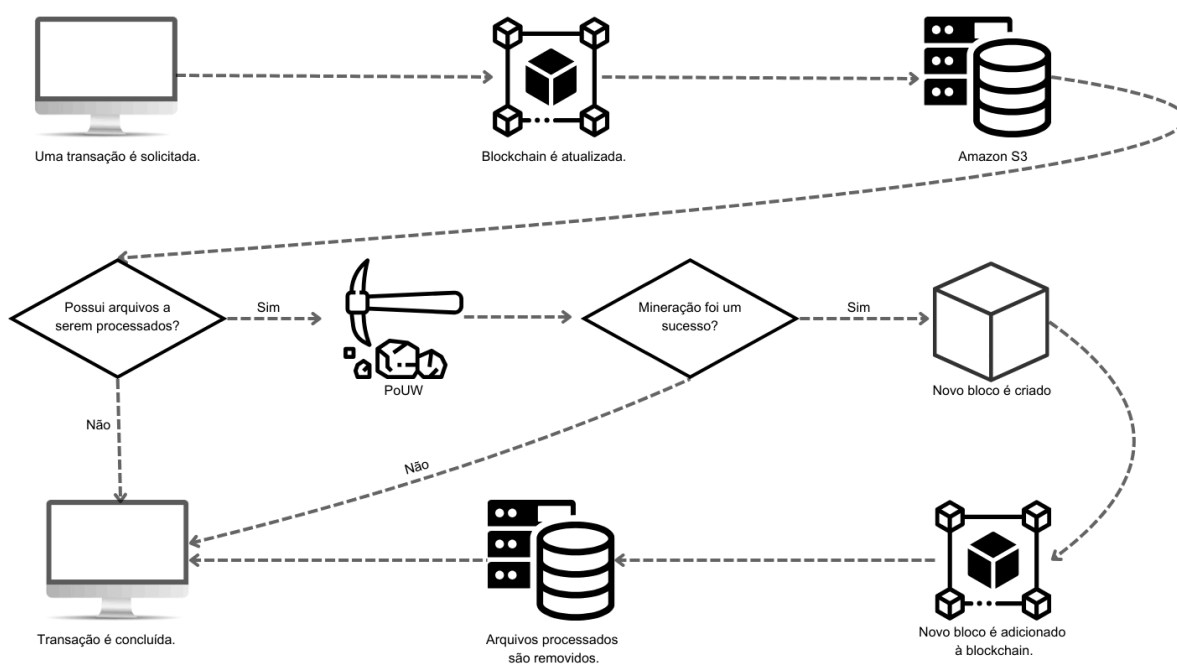


Figura 5.3 – Estrutura da blockchain construída

5.2.1 Adição de novos nós da Blockchain

Ao adicionar um novo nó à rede, é necessário verificar a presença de outros nós e adquirir seus endereços IP para estabelecer a conexão. Com os endereços dos outros nós, a conexão é realizada utilizando a rota *connect_node*. No corpo da requisição, o endereço IP do novo nó é enviado aos outros nós da rede, enquanto o endereço dos nós existentes é transmitido para o novo nó.

5.2.2 Mineração de blocos

A mineração de novos blocos é realizada através da rota *mine_block*. Cada nó recebe a requisição de mineração. Quando um nó recebe uma requisição de mineração, ele primeiro atualiza sua cópia da blockchain e acessa os arquivos

contendo as preferências dos hospitais e residentes armazenados no Amazon S3 utilizando a biblioteca boto3, que também é utilizada para toda interação com o S3. Esta etapa tem um funcionamento análogo a uma fila, uma vez que a mineração só ocorre quando há arquivos presentes no S3, seguindo uma ordem da edição mais antiga para a mais recente. O processo do PoUW é então executado com a biblioteca *matching*, que emparelha as preferências dos médicos e hospitais e verifica a validade e estabilidade dos emparelhamentos, utilizando o algoritmo de Gale e Shapley orientado aos residentes (RGS) conforme descrito na Seção 2.1.2.1. Se bem-sucedido, um novo bloco contendo informações como índice do bloco, *timestamp*, hash criptografado do bloco anterior (usando a função hash SHA-256), edição do NRMP e o resultado do emparelhamento é criado.

Após a mineração, os arquivos referentes à edição minerada são então removidos do S3, mas é importante ressaltar que essa exclusão não é permanente, já que os arquivos podem ser restaurados através do versionamento disponível no S3. Por fim, o bloco minerado é adicionado à resposta da requisição.

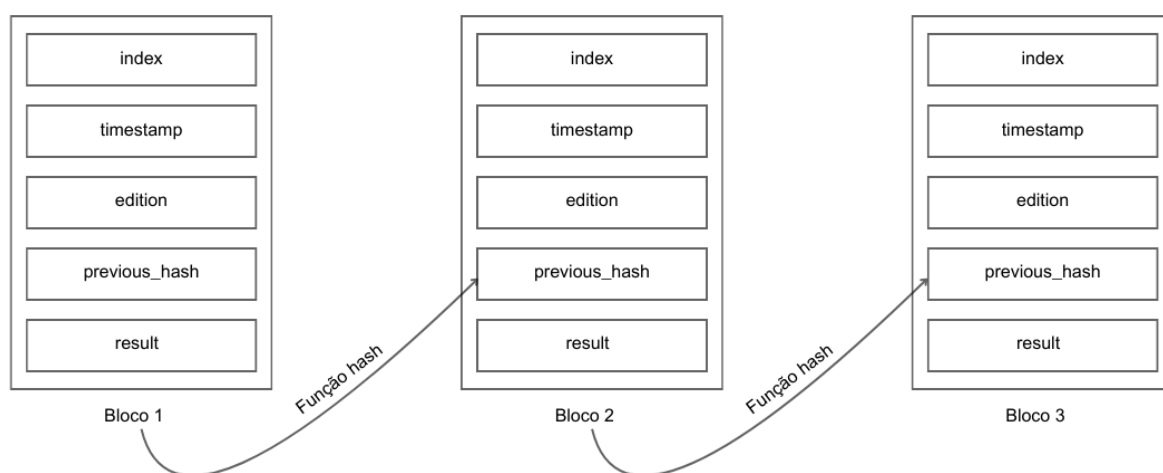


Figura 5.4 – Estrutura dos blocos

5.2.3 Propagação de blocos e consenso

Cada nó conectado à rede possui uma cópia da blockchain e, antes de iniciar a mineração, é essencial assegurar que esta cópia esteja atualizada. A

sincronização entre os nós da blockchain ocorre através da rota *update_chain*, que solicita as cadeias de blocos dos outros nós via a rota *get_chain*. A biblioteca *requests* gerencia toda a comunicação HTTP durante esse processo. As versões contidas em cada nó são então comparadas e validadas. Se um nó descobre que outro possui uma cadeia mais longa, essa cadeia é adotada.

O consenso é então realizado para determinar qual bloco será adicionado e se este é válido. Caso as cópias da blockchain possuam a mesma quantidade de blocos, é verificado se o último bloco possui *timestamps* diferentes – essa comparação é realizada utilizando a biblioteca Python-dateutil. Se for o caso, o bloco escolhido é o que contém o *timestamp* mais antigo, pois foi o primeiro a ser minerado.

A validação é realizada comparando o resultado contido no último bloco em comum entre os nós. Se os valores forem diferentes, o bloco não é aceito e os arquivos com as preferências dos hospitais e residentes são restaurados no S3 para serem reprocessados. Além disso, toda a cadeia de blocos é analisada pelo nó verificador, fazendo a criptografia do bloco anterior da mesma maneira que na criação do bloco e o hash gerado é comparado com o valor presente no bloco atual. Se forem diferentes, uma anomalia é detectada e a cópia da blockchain é descartada.

Todo o código da blockchain está disponível no seguinte repositório do *GitHub*: https://github.com/wallacevncs/Blockchain_PoUW.

TESTES

Neste capítulo, serão apresentados os procedimentos utilizados para testar a aplicação desenvolvida, juntamente com uma avaliação dos resultados obtidos.

6.1 Preparação dos dados

Inicialmente, foram criados arquivos contendo as preferências dos residentes e hospitais com base nos dados quantitativos das últimas cinco edições do NRMP (2024, 2023, 2022, 2021, 2020), conforme fornecidos em [34], utilizando a aplicação ResidentsHospitals-Matching-Generator. Em seguida, um *bucket* (contêiner para armazenamento de objetos) denominado 'nrmp-input' foi criado no S3 para armazenar esses arquivos, conforme apresentado na Figura 6.1.

Amazon S3 > Buckets > nrmp-input

nrmp-input Informações

Objetos Propriedades Permissões Métricas Gerenciamento Pontos de acesso

Objetos (10) Informações

Copiar URI do S3 Copiar URL Fazer download Abrir Excluir Ações Criar pasta Carregar

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon S3](#) para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisará conceder permissões explicitamente a eles. [Saiba mais](#)

Localizar objetos por prefixo

Mostrar versões











<input type="checkbox"/>	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	 hospitalsPreferences_2020.json	json	28 Jan 2024 09:30:00 PM -03	1.8 MB	Padrão
<input type="checkbox"/>	 hospitalsPreferences_2021.json	json	28 Jan 2024 09:30:02 PM -03	2.2 MB	Padrão
<input type="checkbox"/>	 hospitalsPreferences_2022.json	json	28 Jan 2024 09:30:03 PM -03	1.9 MB	Padrão
<input type="checkbox"/>	 hospitalsPreferences_2023.json	json	28 Jan 2024 09:30:04 PM -03	2.0 MB	Padrão
<input type="checkbox"/>	 hospitalsPreferences_2024.json	json	20 Apr 2024 04:50:33 PM -03	2.4 MB	Padrão
<input type="checkbox"/>	 residentsPreferences_2020.json	json	28 Jan 2024 09:30:07 PM -03	5.0 MB	Padrão
<input type="checkbox"/>	 residentsPreferences_2021.json	json	28 Jan 2024 09:30:09 PM -03	5.6 MB	Padrão
<input type="checkbox"/>	 residentsPreferences_2022.json	json	28 Jan 2024 09:30:11 PM -03	5.3 MB	Padrão
<input type="checkbox"/>	 residentsPreferences_2023.json	json	28 Jan 2024 09:30:13 PM -03	5.4 MB	Padrão
<input type="checkbox"/>	 residentsPreferences_2024.json	json	20 Apr 2024 04:50:36 PM -03	5.4 MB	Padrão

Figura 6.1 – Bucket no S3

6.2 Infraestrutura

Para executar a blockchain desenvolvida utilizando o ambiente da AWS, foi criado um repositório no ECR denominado 'blockchain_pouw'. Após isso, uma imagem docker da aplicação foi gerada e adicionada ao repositório criado, conforme exibido na Figura 6.2.

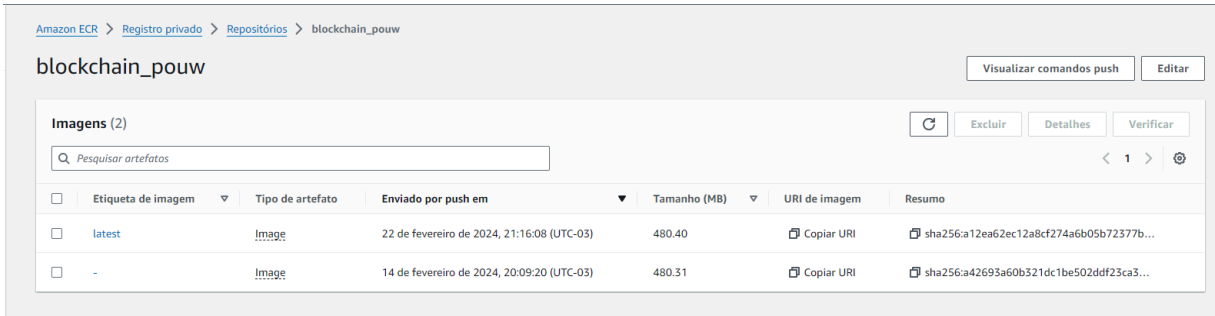


Figura 6.2 – Repositório no ECR

Posteriormente, um cluster denominado 'blockchain' foi criado no ECS, utilizando o tipo AWS Fargate. Após a criação do cluster, três nós foram criados dentro dele, como mostrado na Figura 6.3. Esses nós compõem a rede da nossa blockchain e executam a imagem docker mais recente armazenada no repositório criado no ECR.

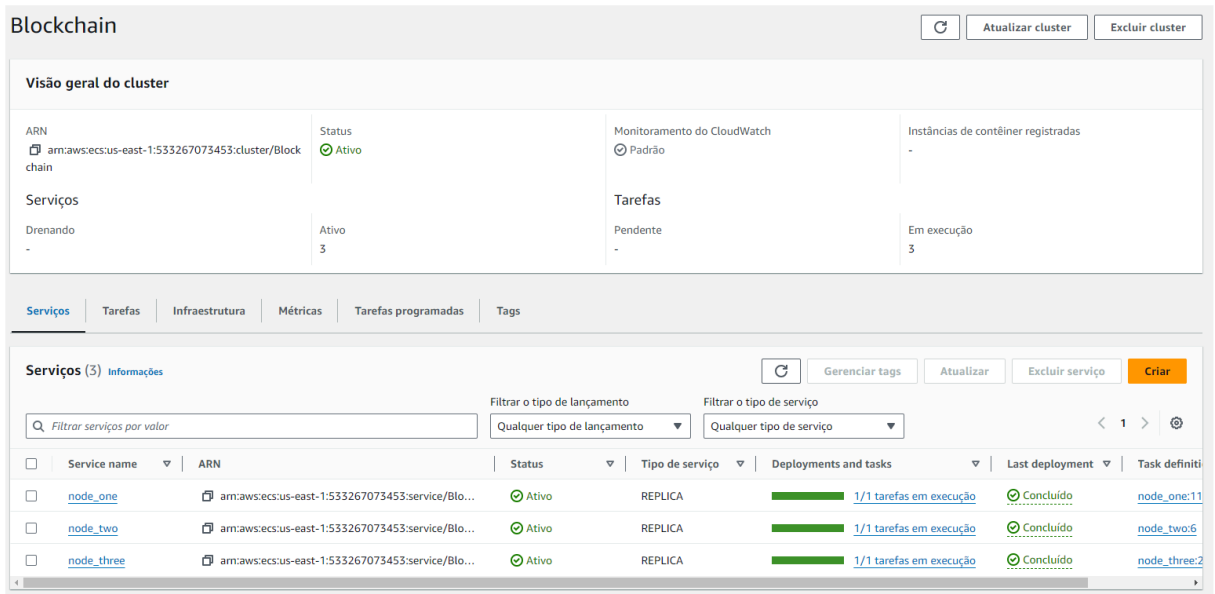


Figura 6.3 – Cluster com os nós da blockchain

Para realizar a validação da blockchain, conforme detalhado na Seção 4.4.1, os nós da rede foram configurados com diferentes capacidades de processamento.

A Tabela 6.1 apresenta o endereço IP (*Internet Protocol*) e os detalhes da capacidade de processamento de cada nó, tanto em CPU (*Central Process Unit*) quanto em memória. A CPU é expressa em vCPU (*Virtual Central Processing Unit*), na qual cada vCPU representa a capacidade de processamento de um núcleo de CPU virtual. E a memória é medida em *gigabytes* (GB).

Tabela 6.1 – Configuração de cada nó

Nó	Endereço IP	CPU (vCPU)	Memória (GB)
<i>node_one</i>	http://3.81.220.197:5001	0.25 vCPU	0.5 GB
<i>node_two</i>	http://54.224.23.169:5001	0.5 vCPU	1 GB
<i>node_three</i>	http://184.73.43.42:5001	1 vCPU	2 GB

6.3 Execução da blockchain

Com todas as configurações estabelecidas e a blockchain em funcionamento, podemos começar a testar a solução. Optamos por utilizar o Postman para efetuar as requisições aos nós da rede. Inicialmente, procedemos com a conexão dos nós e, em seguida, demos início à mineração. Dado que foram inseridos arquivos com as preferências de residentes e hospitais referentes a cinco edições do NRMP no S3, serão minerados e adicionados cinco blocos à blockchain. Os tempos de resposta relacionados à mineração de cada bloco pelos nós estão disponíveis na Tabela 6.2.

Conforme esperado, o nó com melhor desempenho, ou seja, o que produz um bloco no menor tempo, é o *node_three*, dado seu maior poder computacional em comparação com os outros nós da rede. Assim, os blocos adicionados à blockchain serão aqueles produzidos pelo *node_three*.

Tabela 6.2 – Tempo de resposta de cada bloco minerado

Bloco	Nó	Tempo de resposta (s)
1º Bloco	<i>node_one</i>	431s
	<i>node_two</i>	219s
	<i>node_three</i>	98s
2º Bloco	<i>node_one</i>	563s
	<i>node_two</i>	285s
	<i>node_three</i>	121s
3º Bloco	<i>node_one</i>	506s
	<i>node_two</i>	255s
	<i>node_three</i>	112s
4º Bloco	<i>node_one</i>	542s
	<i>node_two</i>	268s
	<i>node_three</i>	118s
5º Bloco	<i>node_one</i>	597s
	<i>node_two</i>	310s
	<i>node_three</i>	128s

É importante destacar que o *node_one* conseguiu produzir blocos em intervalos de tempo razoáveis, apesar de possuir um poder computacional reduzido. Isso se deve à utilização do algoritmo de Gale-Shapley puro, sem considerar as variações que aumentariam significativamente o tempo de execução da PoUW.

6.4 Análise dos resultados

A análise dos resultados gerados por cada nó foi conduzida utilizando o software KDiff3. Como explicado na Seção 4.4.2, espera-se que os nós gerem blocos com resultados idênticos, dado que a entrada é a mesma. A única diferença esperada é o *timestamp*, o qual indica a data de criação daquele bloco.

Nesse contexto, o KDiff3 foi empregado para identificar as diferenças entre os blocos gerados pelos nós para cada edição do NRMP. Os dados no formato JSON resultantes das interações com a API foram sempre inseridos no software na ordem *node_one*, *node_two* e *node_three* para detectar quaisquer discrepâncias nos resultados. Conforme demonstrado nas figuras 6.4 a 6.8, a única discrepância observada entre os blocos gerados é de fato o *timestamp*, atribuível ao fato de que cada nó teve um tempo de resposta distinto, reflexo das diferenças de poder computacional entre eles.

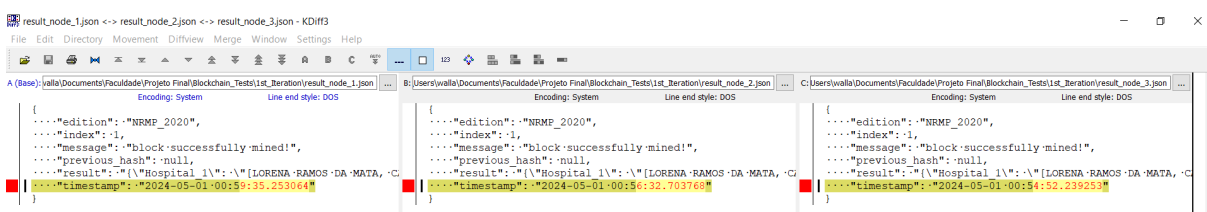


Figura 6.4 – Comparação entre os resultados do 1º bloco

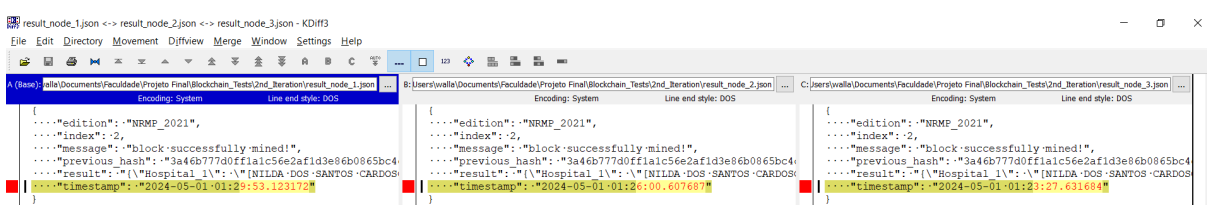


Figura 6.5 – Comparação entre os resultados do 2º bloco

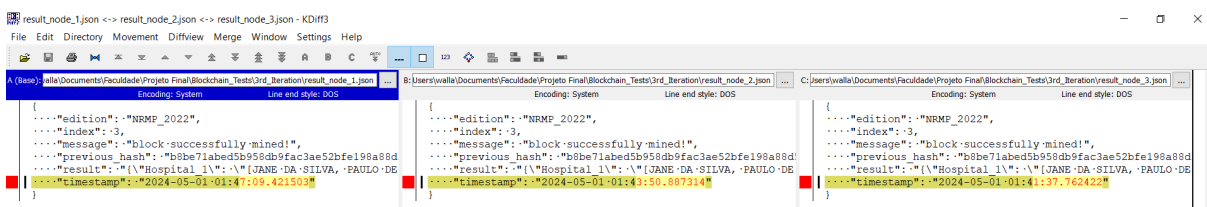


Figura 6.6 – Comparação entre os resultados do 3º bloco

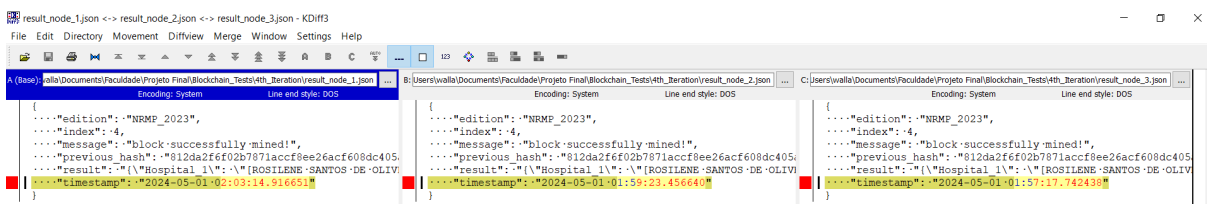


Figura 6.7 – Comparação entre os resultados do 4º bloco

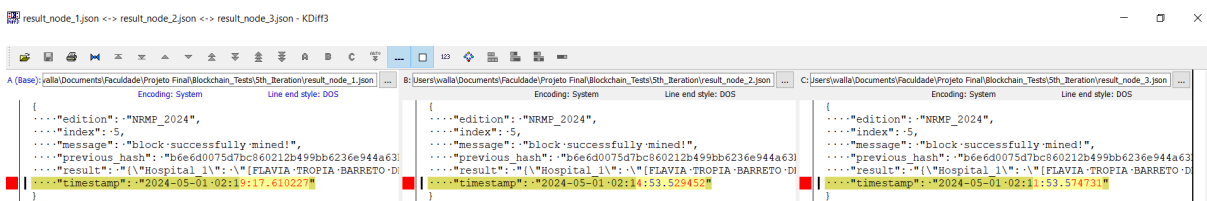


Figura 6.8 – Comparação entre os resultados do 5º bloco

No repositório do *GitHub* onde o código da aplicação está disponível, foi adicionada uma pasta denominada 'blockchain_tests'. Dentro desta pasta, existem duas subpastas: 'INPUT' e 'OUTPUT'. A primeira contém arquivos com as preferências dos residentes e hospitais, utilizados para testar a aplicação. Enquanto a segunda guarda os resultados após cada interação com a aplicação.

CONSIDERAÇÕES FINAIS

Neste último capítulo, será realizada uma avaliação conclusiva da solução apresentada neste trabalho, seguida da exploração de potenciais trabalhos futuros para aprimorar sua eficiência e eficácia. Por fim, serão abordadas as limitações identificadas ao longo do presente estudo.

7.1 Avaliação da solução

Ao longo deste estudo, foi destacada a importância da seleção de médicos por meio dos programas de residências hospitalares do NRMP, reconhecendo seu papel crucial no sistema de saúde, tanto nos Estados Unidos quanto globalmente, já que influencia outros programas de seleção. No entanto, as deficiências relacionadas à falta de transparência podem comprometer a relevância contínua do NRMP no processo de seleção de residentes pelos hospitais.

Nesse contexto, a integração da tecnologia blockchain, aliada à proposta de PoUW apresentada neste trabalho, surge como uma solução potencial para mitigar as críticas direcionadas ao NRMP atualmente. Apresentamos uma proposta teórica, seguida pela implementação e teste da solução, que demonstrou resultados promissores.

A blockchain desenvolvida neste estudo é um exemplo de como essa tecnologia pode ser aplicada para resolver problemas complexos, indo além do âmbito dos registros distribuídos de informações. Ao proporcionar uma solução concreta para um problema prático e relevante, esperamos contribuir para o avanço contínuo e aprimoramento tanto da PoUW quanto do processo de seleção de residentes.

7.2 Trabalhos futuros

O algoritmo atualmente adotado pelo NRMP apresenta uma complexidade superior à do algoritmo de Gale-Shapley utilizado como PoUW na blockchain desenvolvida neste estudo. Isso se deve ao fato de que o algoritmo do NRMP incorpora diversas variações, como a consideração de casais de residentes. Nesse contexto, uma melhoria potencial seria a implementação de um algoritmo que se assemelhe o mais possível ao utilizado pelo NRMP, abrangendo as suas principais variações conhecidas.

Para promover um escalonamento mais eficiente da blockchain, é fundamental automatizar certas operações realizadas pelos nós. Atualmente, o procedimento demanda o envio manual de uma requisição a cada nó da blockchain – geralmente realizado através de ferramentas externas como o Postman – a fim de iniciar operações como mineração ou estabelecer conexões entre os nós da rede. Uma abordagem mais eficaz seria a implementação de um sistema de orquestração capaz de identificar novas conexões entre nós e, automaticamente, solicitar a mineração assim que novos arquivos contendo as preferências dos residentes e hospitais no S3 forem identificados. Essa automação proporcionaria um funcionamento mais fluido e eficiente da rede, permitindo um processo mais ágil e dinâmico.

Além disso, a avaliação da robustez e eficiência da solução blockchain proposta requer testes adicionais que considerem diferentes níveis de carga de rede e latência. Esses testes incluiriam simulações de cenários com variações na quantidade de transações processadas, bem como a avaliação do desempenho da rede sob diferentes condições de latência. Identificar possíveis gargalos e otimizar o desempenho da blockchain em ambientes reais seriam resultados essenciais desses testes.

Outro aspecto crucial a ser avaliado é a segurança da blockchain contra ataques comuns. Implementar testes de penetração e simulações de ataques ajudaria a identificar vulnerabilidades e fortalecer a resiliência da rede contra tentativas de comprometimento. Isso é vital, pois uma blockchain segura é essencial para garantir a integridade e confiabilidade dos resultados do sistema proposto.

Por fim, comparar a blockchain desenvolvida neste estudo com outras blockchains em termos de custos operacionais e energéticos é importante para identificar oportunidades de melhoria. Isso envolveria uma análise detalhada do custo da execução da blockchain em serviços de nuvem, como a AWS, e uma

estimativa do consumo de energia de cada nó durante o processo de mineração. Comparar esses aspectos com outras alternativas permitiria uma avaliação abrangente da viabilidade econômica e energética da implementação, destacando áreas potenciais de otimização.

7.3 Limitações

No desenvolvimento deste trabalho, foram identificadas algumas limitações significativas. Primeiramente, foi constatada uma escassez de informações detalhadas sobre o funcionamento do algoritmo de emparelhamento utilizado pelo NRMP. A maioria dos trabalhos disponíveis que descrevem o algoritmo são antigos, e o próprio NRMP divulga apenas informações básicas sobre o algoritmo. Além disso, a busca por uma blockchain que efetivamente implementasse o algoritmo de consenso da PoUW revelou-se desafiadora, uma vez que as referências encontradas se limitavam a trabalhos teóricos, restringindo assim a capacidade de avaliar práticas existentes.

Outro ponto crítico foi a limitação de recursos, tanto em termos de tempo quanto financeiros. Esses recursos são essenciais para deixar a solução pronta para uso pelo mercado. O desenvolvimento e a implementação completa de uma blockchain robusta e escalável demandam um investimento significativo em infraestrutura, mão de obra especializada e testes extensivos. Ademais, a realização de testes em cenários equivalentes aos encontrados no mundo real requer um ambiente controlado e recursos financeiros que viabilizem a simulação de diversas condições de carga e latência. A ausência desses recursos limitou a capacidade de realizar uma validação mais abrangente e precisa da solução proposta.

REFERÊNCIAS

- [1] GALE, D.; SHAPLEY, L. S. "College Admissions and the Stability of Marriage". *The American Mathematical Monthly*, v. 69, n. 1, 1962, pgs. 9–15. DOI: 10.1080/00029890.1962.11989827 (citado nas pgs. 1, 6).
- [2] NRMP. "2024 Main Residency Match® By the Numbers". Disponível em: <https://www.nrmp.org/wp-content/uploads/2024/03/2024-Match-by-the-Numbers.pdf>. Acesso em: 4 abr. 2024.
- [3] NRMP. "The Matching Algorithm". Disponível em: <https://www.nrmp.org/imatch/matching-algorithm/>. Acesso em: 2 mar. 2024.
- [4] CaRMS. "Algorithm FAQs: How it works". Disponível em: <https://www.carms.ca/the-match/how-it-works/general/>. Acesso em: 2 mar. 2024.
- [5] MIT. "Stable Matching". Disponível em: https://openlearninglibrary.mit.edu/assets/courseware/v1/df88d46bd2d149efb373ff0acf785dd6/asset-v1:OCW+6.042J+2T2019+type@asset+block/MIT6_042JS15_stablematchg.pdf. Acesso em: 2 mar. 2024.
- [6] HABER, S.; STORNETTA, W. S. "How to time-stamp a digital document". *Journal of Cryptology*, v. 3, n. 2, p. 99–111, 1991. DOI: <https://doi.org/10.1007/BF00196791>.
- [7] BAYER, D.; HABER, S.; STORNETTA, W. S. "Improving the efficiency and reliability of digital time-stamping". In: *Sequences II*. New York, NY: Springer New York, 1993, p. 329–334. DOI: 10.1007/978-1-4613-9323-8_24.
- [8] MERKLE, R. C. "A Digital Signature Based on a Conventional Encryption Function". In: *Advances in Cryptology – CRYPTO '87*. Springer, Berlin, Heidelberg. 1988. p. 369–378. DOI: 10.1007/3-540-48184-2_32.

[9] NAKAMOTO, S. "Bitcoin: A Peer-to-Peer Electronic Cash System". 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 9 mar. 2024.

[10] Simply Explained. "How does a blockchain work". Disponível em: <https://simplyexplained.com/videos/how-does-a-blockchain-work/>. Acesso em: 10 mar. 2024.

[11] HUANG, J.; O'NEILL, C.; TABUCHI, H. "Bitcoin Uses More Electricity Than Many Countries. How Is That Possible?". New York Times, 3 set. 2021. Disponível em: <https://www.nytimes.com/interactive/2021/09/03/climate/bitcoin-carbon-footprint-electricity.html>. Acesso em: 10 mar. 2024.

[12] ELROM, E. (2019). "The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-Based Projects". Apress, p.17-30.

[13] AWS. O que é a tecnologia blockchain?. Disponível em: <https://aws.amazon.com/pt/what-is/blockchain/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc>. Acesso em: 12 Mar. 2024.

[14] BACK, A. "Hashcash - A Denial of Service Counter-Measure". 1997. Disponível em: <http://www.hashcash.org/papers/hashcash.pdf>. Acesso em: 14 Mar. 2024.

[15] ROTH, A. E.; PERANSON, E. "The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design." 1999. American Economic Review, 89 (4): 748-780. DOI: 10.1257/aer.89.4.748.

[16] KLEINBERG, J.; TARDOS, E. (2005). "Algorithm Design". Pearson, p.7-8.

[17] JRMP. "Japan Residency Matching Program". Disponível em: <https://www.jrmp.jp/>. Acesso em: 21 Mar. 2024.

- [18] OLIVER, C. G.; RICOTTONE, A.; PHILIPPOPOULOS, P. "Proposal for a fully decentralized blockchain and proof-of-work algorithm for solving NP-complete problems". 2017. DOI: 10.48550/arXiv.1708.09419.
- [19] BALDOMINOS, A.; SAEZ, Y. "Coin. AI: A proof-of-useful-work scheme for blockchain-based distributed deep learning". 2019. DOI: 10.3390/e21080723.
- [20] MA, R.; YAO, L.; SONG, L.; JIN, M. "A novel algorithm for peer-to-peer ridesharing match problem". *Neural Computing and Applications*, v. 31, 2019, p. 247-258. DOI: 10.1007/s00521-018-3733-5.
- [21] DAVIDOVIĆ, T.; TODOROVIĆ, M.; SHARMA, B.; RAMLJAK, D. "Exploring Arbitrary Real-Life Problems in Proof-of-Useful-Work: Myth Busting?". *Fifth International Conference on Blockchain Computing and Applications (BCCA)*, 2023, pp. 1-6. DOI: 10.1109/BCCA58897.2023.10338884.
- [22] HAOUARI, M.; MHIRI, M.; EL-MASRI, M.; AL-YAFI, K. "A novel proof of useful work for a blockchain storing transportation transactions". *Information Processing & Management*, v. 59, n. 1, p. 102749, 2022. DOI: 10.1016/j.ipm.2021.102749.
- [23] Pypi. "Matching". Disponível em: <https://pypi.org/project/matching/>. Acesso em: 7 abr. 2024.
- [24] Pypi. "Boto3". Disponível em: <https://pypi.org/project/boto3/>. Acesso em: 7 abr. 2024.
- [25] Pypi. "Python-dateutil". Disponível em: <https://pypi.org/project/python-dateutil/>. Acesso em: 7 abr. 2024.
- [26] Pypi. "Requests". Disponível em: <https://pypi.org/project/requests/>. Acesso em: 7 abr. 2024.
- [27] Flask. "Flask". Disponível em: <https://flask.palletsprojects.com/en/3.0.x/>. Acesso em: 7 abr. 2024.

[28] AWS. O que é o Amazon S3?. Disponível em: https://docs.aws.amazon.com/pt_br/AmazonS3/latest/userguide/Welcome.html.

Acesso em: 8 abr. 2024.

[29] AWS. “What is Amazon Elastic Container Registry?”. Disponível em: <https://docs.aws.amazon.com/AmazonECR/latest/userguide/what-is-ecr.html>. Acesso em: 8 abr. 2024.

[30] AWS. O que é o *Amazon Elastic Container Service*?. Disponível em: https://docs.aws.amazon.com/pt_br/AmazonECS/latest/developerguide/Welcome.html. Acesso em: 8 abr. 2024.

[31] Docker. “Docker overview”. Disponível em: <https://docs.docker.com/get-started/overview/>.

Acesso em: 8 abr. 2024.

[32] SAMBINELLI, M. Problemas de emparelhamentos estáveis. 2014. Disponível em: <https://repositorio.unicamp.br/Busca/Download?codigoArquivo=489633>.

Acesso em: 14 abr. 2024.

[33] Newtonsoft. Json.NET. Disponível em: <https://www.newtonsoft.com/json>. Acesso em: 23 abr. 2024.

[34] NRMP. “Match Data & Report Archives”. Disponível em: <https://www.nrmp.org/match-data-analytics/archives/>. Acesso em: 23 abr. 2024.

[35] AWS. AWS Fargate . Disponível em: <https://aws.amazon.com/pt/fargate/> . Acesso em: 23 abr. 2024.

[36] KDiff3. KDiff3 . Disponível em: <https://kdiff3.sourceforge.net/>. Acesso em: 02 maio 2024.

[37] POSTMAN. “What is Postman?”. Disponível em: <https://www.postman.com/product/what-is-postman/>. Acesso em: 02 maio 2024.

[38] SONI, A. "Crypto miners used the same amount of electricity as all of Australia last year". Yahoo Finance, 6 fev. 2024. Disponível em: <https://finance.yahoo.com/news/crypto-miners-used-same-amount-034552218.html>. Acesso em: 05 maio 2024.